The 6th International Semantic Web Conference and the 2nd Asian Semantic Web Conference

ISWC+ASWC 2007

Workshop 7

Ontology Matching

Workshop Organizers:

Pavel Shvaiko, Jérôme Euzenat, Fausto Giunchiglia, Bin He

11 Nov. 2007
BEXCO, Busan KOREA

ISWC 2007 Sponsor

Emerald Sponsor

Saltlux

Gold Sponsor

Google

u-MTec

KAIST

VULCAN
A Paul G. Allen Company

KoGb

Silver Sponsor

IBM Research

NeOn

EAST WEB

ASIA-LINK
EUROPEAID
CO-OPERATION OFFICE

ETRI

FRANZ INC.

ADUNA

IOS Press

We would like to express our special thanks to all sponsors

ISWC 2007 Organizing Committee

General Chairs

Riichiro Mizoguchi (Osaka University, Japan)
Guus Schreiber (Free University Amsterdam, Netherlands)

Local Chair

Sung-Kook Han (Wonkwang University, Korea)

Program Chairs

Karl Aberer (EPFL, Switzerland)
Key-Sun Choi (Korea Advanced Institute of Science and Technology)
Natasha Noy (Stanford University, USA)

Workshop Chairs

Harith Alani (University of Southampton, United Kingdom)
Geert-Jan Houben (Vrije Universiteit Brussel, Belgium)

Tutorial Chairs

John Domingue (Knowledge Media Institute, The Open University)
David Martin (SRI, USA)

Semantic Web in Use Chairs

Dean Allemang (TopQuadrant, USA)
Kyung-Il Lee (Saltlux Inc., Korea)
Lyndon Nixon (Free University Berlin, Germany)

Semantic Web Challenge Chairs

Jennifer Golbeck (University of Maryland, USA)
Peter Mika (Yahoo! Research Barcelona, Spain)

Poster & Demos Chairs

Young-Tack, Park (Sonngsil University, Korea)
Mike Dean (BBN, USA)

Doctoral Consortium Chair

Diana Maynard (University of Sheffield, United Kingdom)

Sponsor Chairs

Young-Sik Jeong (Wonkwang University, Korea)
York Sure (University of Karlsruhe, German)

Exhibition Chairs

Myung-Hwan Koo (Korea Telecom, Korea)
Noboru Shimizu (Keio Research Institute, Japan)

Publicity Chair: Masahiro Hori (Kansai University, Japan)

Proceedings Chair: Philippe Cudré-Mauroux (EPFL, Switzerland

Metadata Chairs

Tom Heath (KMi, OpenUniversity, UK)
Knud Möller (DERI, National University of Ireland, Galway)

Ontology Matching

OM-2007

Papers from the ISWC + ASWC Workshop

Introduction

Ontology matching takes ontologies as input and determines as output correspondences between the semantically related entities of those ontologies. These correspondences can be used for various tasks, such as ontology merging, query answering, data translation, or for navigation on the semantic web. Ontology matching is thus a key interoperability enabler for the semantic web as it allows knowledge and data expressed in the matched ontologies to interoperate.

The workshop had two goals:

- To bring together academic and industry leaders to assess how academic advances are addressing real world requirements. The workshop strives to improve academic awareness of industrial needs, and therefore, direct research towards those needs. Simultaneously, the workshop serves to inform industry representatives about existing research efforts that may meet their business needs. Moreover, it is central to the aims of the workshop to evaluate how technologies for ontology matching are going to evolve, which research topics are in the academic agenda and how these can fit emerging business issues.
- To conduct an extensive, rigorous and transparent evaluation of ontology matching approaches through the OAEI (Ontology Alignment Evaluation Initiative) 2007 campaign, `http://oaei.ontologymatching.org/2007`. The particular focus of this year's OAEI campaign is on real-world matching tasks from specific domains, such as medicine, food, library and environment. Therefore, the ontology matching evaluation initiative itself will provide a solid ground for discussion of how well the current approaches are meeting business needs.

We received 26 submissions for the technical track of the workshop. The program committee selected 8 submissions for oral presentation and 9 submissions for poster presentation. 18 matching systems participated in this year's OAEI campaign. Further information about the Ontology Matching workshop can be found at: `http://om2007.ontologymatching.org/`.

Acknowledgments. We thank all members of the program committee, authors and local organizers for their efforts. We appreciate support from the KnowledgeWeb European Network of Excellence (IST-2004-507482), the OpenKnowledge European STREP (FP6-027253) and the ASIA-Link EASTWEB project (Contract TH/Asia Link/010 (111084)).

Pavel Shvaiko
Jérôme Euzenat
Fausto Giunchiglia
Bin He

November 2007

Organization

Organizing Committee

Pavel Shvaiko, University of Trento, Italy
Jérôme Euzenat, INRIA Rhône-Alpes, France
Fausto Giunchiglia, University of Trento, Italy
Bin He, IBM Almaden Research Center, USA

Program Committee

Olivier Bodenreider, National Library of Medicine, USA
Francesco Guerra, University of Modena and Reggio Emilia, Italy
Wei Hu, Southeast University, China
Jingshan Huang, University of South Carolina, USA
Todd Hughes, DARPA, USA
Michael Huhns, University of South Carolina, USA
Ryutaro Ichise, National Institute of Informatics, Japan
Anthony Jameson, DFKI, Germany
Yannis Kalfoglou, University of Southampton, UK
Vipul Kashyap, Clinical Informatics R&D, Partners HealthCare System, USA
Christian Meilicke, University of Mannheim, Germany
Peter Mork, The MITRE Corporation, USA
Meenakshi Nagarajan, Wright State University, USA
Luigi Palopoli, University of Calabria, Italy
Luciano Serafini, Fondazione Bruno Kessler (ITC-IRST), Italy
Marco Schorlemmer, IIIA-CSIC, Spain
Steffen Staab, University of Koblenz, Germany
Umberto Straccia, ISTI-C.N.R., Italy
Heiner Stuckenschmidt, University of Mannheim, Germany
Petko Valtchev, Université du Québec à Montréal, Canada
Frank van Harmelen, Vrije Universiteit Amsterdam, Netherlands
Ludger van Elst, DFKI, Germany
Baoshi Yan, Bosch Research, USA
Mikalai Yatskevich, University of Trento, Italy
Songmao Zhang, Chinese Academy of Sciences, China

Table of Contents

PART 1 - Technical Papers

PART 2 - OAEI Papers

PART 3 - Posters

Solving Semantic Ambiguity to Improve Semantic Web based Ontology Matching

Jorge Gracia[1], Vanessa López[2], Mathieu d'Aquin[2],
Marta Sabou[2], Enrico Motta[2], and Eduardo Mena[1]

[1] IIS Department, University of Zaragoza, Spain
{jogracia,emena}@unizar.es
[2] Knowledge Media Institute (KMi), The Open University, United Kingdom
{v.lopez,m.daquin,r.m.sabou,e.motta}@open.ac.uk

Abstract. A new paradigm in Semantic Web research focuses on the development of a new generation of knowledge-based problem solvers, which can exploit the massive amounts of formally specified information available on the Web, to produce novel intelligent functionalities. An important example of this paradigm can be found in the area of Ontology Matching, where new algorithms, which derive mappings from an exploration of multiple and heterogeneous online ontologies, have been proposed. While these algorithms exhibit very good performance, they rely on merely syntactical techniques to anchor the terms to be matched to those found on the Semantic Web. As a result, their precision can be affected by ambiguous words. In this paper, we aim to solve these problems by introducing techniques from Word Sense Disambiguation, which validate the mappings by exploring the semantics of the ontological terms involved in the matching process. Specifically we discuss how two techniques, which exploit the ontological context of the matched and anchor terms, and the information provided by WordNet, can be used to filter out mappings resulting from the incorrect anchoring of ambiguous terms. Our experiments show that each of the proposed disambiguation techniques, and even more their combination, can lead to an important increase in precision, without having too negative an impact on recall.
Keywords: semantic web, ontology matching, semantic ambiguity.

1 Introduction

As result of the recent growing of the Semantic Web, a new generation of semantic applications are emerging, focused on exploiting the increasing amount of online semantic data available on the Web [5]. These applications need to handle the high semantic heterogeneity introduced by the increasing number of available online ontologies, that describe different domains from many different points of view and using different conceptualisations, thus leading to many ambiguity problems.

In this challenging context, a new paradigm, which uses the Semantic Web as background knowledge, has been proposed to perform automatic Ontology

Matching [8]. An initial evaluation of this method showed a 70% precision in obtaining mappings between ontologies [9]. These experiments have also shown that more than half of the invalid mappings are due to ambiguity problems in the *anchoring process* (see later Sections 2 and 3).

These ambiguity problems are shared by any other Ontology Matching system based on background knowledge. Indeed, they are shared by any other system which needs to find correspondences across heterogeneous sources. Nevertheless we focus on the above mentioned Semantic Web based matcher, because it deals with online ontologies, thus maximizing heterogeneity of sources (and ambiguity problems), and providing us a suitable scenario to develop our ideas.

In this paper we investigate the use of two different techniques from Word Sense Disambiguation. The objective is to improve the results of background knowledge based Ontology Matching, by detecting and solving the ambiguity problems inherent to the use of heterogeneous sources of knowledge. Our experiments, based on the system described in [8], confirm our prediction that precision can be improved by using the above mentioned semantic techniques, getting even better results by combining them.

The rest of this paper is as follows: Section 2 explains the paradigm of harvesting the Semantic Web to perform Ontology Matching. How semantic ambiguity hampers this method is explained in Section 3, whereas in Sections 4, 5, and 6 we show three different approaches to solve this problem. Our experimental results and some related work can be found in Sections 7 and 8, respectively. Finally conclusions and future work appear in Section 9.

2 Ontology Matching by Harvesting the Semantic Web

In [8] a *new paradigm to Ontology Matching* that builds on the Semantic Web vision is proposed: it derives semantic mappings by exploring multiple and heterogeneous online ontologies that are dynamically selected (using Swoogle[3] as semantic search engine), combined, and exploited. For example, when matching two concepts labelled *Researcher* and *AcademicStaff*, a matcher based on this paradigm would 1) identify, at run-time, online ontologies that can provide information about how these two concepts relate, and then 2) combine this information to infer the mapping. The mapping can be either provided by a single ontology (e.g., stating that $Researcher \sqsubseteq AcademicStaff$), or by reasoning over information spread among several ontologies (e.g., that $Researcher \sqsubseteq ResearchStaff$ in one ontology and that $ResearchStaff \sqsubseteq AcademicStaff$ in another). The novelty of the paradigm is that the knowledge sources are not manually provided prior to the matching stage but dynamically selected from online available ontologies during the matching process itself.

Figure 1 illustrates the basic idea of Ontology Matching by harvesting the Semantic Web. A and B are the concepts to relate, and the first step is to find online ontologies containing concepts A' and B' equivalent to A and B. This

---

[3] http://swoogle.umbc.edu/

process is called *anchoring* and A' and B' are called the *anchor terms*. Based on the relations that link A' and B' in the retrieved ontologies, a mapping is then derived between A and B.

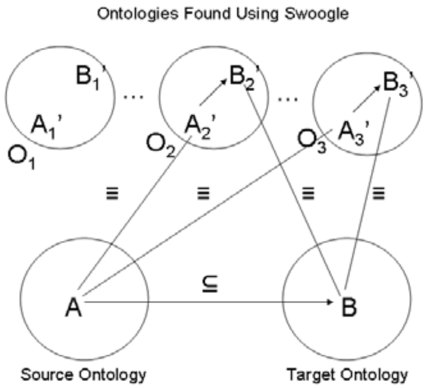


Fig. 1. Ontology Matching by harvesting the Semantic Web.

A baseline implementation of this technique has been evaluated [9] using two very large, real life thesauri that made up one of the test data sets in the 2006 Ontology Alignment Evaluation Initiative, AGROVOC and NALT[4]. A sample of 1000 mappings obtained thanks to this implementation has been manually validated, resulting in a promising 70% precision. However, a deeper analysis of the wrong mappings has shown that more than half of them (53%) were due to an incorrect anchoring: because of ambiguities, elements of the source ontology have been anchored to online ontologies using the considered terms with different senses. The employed naive anchoring mechanism is thus clearly insufficient, as it fails to distinguish words having several different senses and so, to handle ambiguity. Our hypothesis is that integrating techniques from Word Sense Disambiguation to complement the anchoring mechanism would lead to an important increase in precision.

3 Sense Disambiguation to Improve Anchoring

We have devised an improved way to perform Ontology Matching based on background knowledge, using techniques that take into account the *semantics* of the compared terms to validate the anchoring process.

For a better insight, let us see an example. The matcher described in Section 2 retrieved the following matching between two terms from the AGROVOC and NALT ontologies: $game \sqsupseteq sports$. "Game" is a "wild animal" in AGROVOC while "sports" appears in NALT as a "leisure, recreation and tourism" activity.

---

[4] http://www.few.vu.nl/ wrvhage/oaei2006/

The reason why this invalid mapping was derived is because "game" has been anchored in a background ontology[5], where it is defined as subclass of "Recreation or Exercise", and as superclass of "sport". This problem can be solved with an appropriate technique which deals with the *ambiguity* of the terms, being able to determine that "game" in the AGROVOC ontology (an animal) and "game" in the background ontology (a contest) are *different* concepts, thus avoiding their anchoring. Thus, our approach to handle semantic ambiguity is twofold:

First, we have considered the system proposed in [11]. Its goal is to disambiguate user keywords in order to translate them into semantic queries. In this context a *semantic similarity measure* has been defined to provide a *synonymy degree* between two terms from different ontologies, by exploring both their lexical and structural context. A configurable threshold allows the system to determine whether two ontological terms are considered or not the same (see Section 4 for more details).

Second, we have explored the use of a WordNet[6] based technique to perform a similar task. We reused parts of PoweMap [4], a hybrid knowledge-based matching algorithm, comprising terminological and structural techniques, and used in the context of multiontology question answering. Details of how PowerMap is used to filter semantically sound ontological mappings are given in Section 5.

In the following, we discuss the experiments we have conducted on the use of these two techniques, and on their combination, to improve Semantic Web based Ontology Matching.

4 Improving Anchoring by Exploring Ontological Context

In [2, 11] a system to discover the possible meanings of a set of user keywords by consulting a pool of online available ontologies is presented. First it proposes a set of possible ontological senses for each keyword, integrating the ones that are considered similar enough. Then these merged senses are used as input for a disambiguation process to find the most probable meaning of each keyword, to use them, finally, in the construction of semantic queries. These queries must represent the intended meaning of the initial user keywords.

Here we focus on the first step of the system, where an *ontological context* based similarity measure is applied to decide whether the semantics of two ontological terms represent the same sense or not.

4.1 Synonymy degree estimation

A detailed description of the above mentioned similarity measure is out of the scope of this paper, but we summarize here the key characteristics:

1. The algorithm receives two terms A, B from two different ontologies as input. Their ontological contexts are extracted (hypernyms, hyponyms, descriptions, properties,...).

---

[5] http://lists.w3.org/Archives/Public/www-rdf-logic/2003Apr/att-0009/SUMO.daml

[6] http://wordnet.princeton.edu/

2. An initial computation uses linguistic similarity between terms, considering labels as strings.
3. A subsequent recursive computation uses structural similarity, exploiting the ontological context of a term until a given *depth*. Vector Space Models are employed in the comparisons among *sampled* sets of terms extracted from the ontological contexts.
4. The different contributions (structural similarity, linguistic similarity, ...) are weighted, and a final *synonymy degree* between A, B is provided.

Therefore, this ontological context based similarity (let us call it $sim_{ont}(A, B)$) gets an estimated *synonymy degree* in $[0, 1]$ for a given *depth* of exploration (number of levels in the hierarchy that we explore).

4.2 Improved anchoring technique

Let us call, for the rest of the paper, A and B a particular pair of terms belonging respectively to the ontologies O_A and O_B to be aligned. We denote A' and B' their respective anchor terms in background ontologies, and O'_A and O'_B the respective background ontologies where they appear (sometimes $O'_A = O'_B$). Finally we denote as $\langle A, B, r, l\rangle$ a mapping between terms A and B, r representing the relation between them and l the level of confidence of the mapping.

Here is our first approach to take into account the semantics of the involved anchored terms in the matching process:

Scheme 1 ("filtering candidate mappings by exploring ontological context"). In this first approach, the validity of the anchoring is evaluated, a posteriori, on the mappings derived by the method explained in Section 2. The similarity between the ontological terms and their respective anchor terms is measured by analysing their ontological context up to a certain depth[7]: $sim_{ont}(A, A')$ and $sim_{ont}(B, B')$.

To qualify the mapping as a *valid* one, validity on each side of the mapping is required, hence both confidence degrees obtained must be above the required threshold. We compute the *confidence level* for the mapping $\langle A, B, r, l\rangle$ as:

$$l = min(sim_{ont}(A, A'), sim_{ont}(B, B')) \tag{1}$$

If $l > threshold$ then the mapping is accepted, otherwise is rejected.

The expected effect of this approach is an improvement in the precision, as many results erroneously mapped due to bad anchoring can be detected and filtered. Recalling the example discussed in Section 3: for the mapping $\langle game, sports, \sqsupseteq, l\rangle$ between AGROVOC and NALT ontologies, a value of $l = 0.269$ is computed. Then, if we have set up a threshold with a higher value, this erroneous mapping due to bad anchoring will be filtered out.

On the other hand, this approach is unable to improve the overall recall of the results (as it is unable to add new valid mappings). Indeed, we cannot discard a

---

[7] In this and subsequent experiments we compute sim_{ont} using $depth = 2$.

potential negative effect on recall, as some good mappings could also be filtered out if the computed similarities are not high enough (for example because of a poor description of the terms in ontologies).

5 Improving Anchoring by Exploring WordNet

As a complementary way, we have explored the use of a WordNet based algorithm implemented as part of PowerMap [4]. This makes possible to establish comparisons with the technique proposed in Section 4 and, eventually, to identify a combined use of both.

PowerMap is the solution adopted by PowerAqua, a multiontology-based Question Answering platform [4], to map user terminology into ontology-compliant terminology distributed across ontologies. The PowerMap algorithm first uses syntactic techniques to identify possible ontology matches, likely to provide the information requested by the user's query. WordNet based methods are then used to elicit the sense of candidate concepts by looking at the ontology hierarchy, and to check the semantic validity of those syntactic mappings, which originate from distinct ontologies, with respect the user's query terms.

5.1 PowerMap based method for the semantic relevance analysis

The PowerMap WordNet-based algorithm is adapted and used here to determine the validity of the mappings provided by the system described in Section 2. In this approach we do not perform similarity computation between terms and anchored terms, as we did in Schemes 1. Instead, similarity is computed directly between the matched ontology terms A and B.

Note that, here, *similarity* has a broader meaning than *synonymy*. We say that **two words are semantically similar** if they have a synset(s) in common (synonymy), or there exists an allowable IS-A path (in the hypernym/hyponym WordNet taxonomy) connecting a synset associated with each word. The rationale of this point is based on the two criteria of similarity between concepts established by Resnik in [7], where semantic similarity is determined as a function of the path distance between the terms, and the extent to which they share information in common. Formally, in the IS-A hierarchy of WordNet, similarity is given by the Wu and Palmer's formula described in [13].

5.2 Improved anchoring technique

In the following we explain how we apply this WordNet based method to determine the validity of mappings.

Scheme 2 ("filtering candidate mappings by exploring WordNet"). We compute the WordNet based confidence level $l = sim_{WN}(A, B)$ for the matching $\langle A, B, r, l\rangle$ as follows. Given the two ontological terms A and B, let $S_{B,A}$ be the set of those synsets of B for which there exists a semantically similar synset of A (according to Wu and Palmer's formula). If $S_{B,A}$ is empty, the mapping B is discarded because the intended meaning of A is not the same as that of the concept B. Finally, the true senses of B are determined by its place in the hierarchy of the ontology. That is, S_B^H consists only of those synsets of B that are similar to at least one synset of its ancestors in the ontology. We then obtain the valid senses as the intersection of the senses in S_B^H, with the senses obtained in our previous step, $\mathrm{S}_{B,A}$. Note that by intersection we mean the synsets that are semantically similar, even if they are not exactly the same synset. In case the intersection is empty it means that the sense of the concept in the hierarchy is different from the sense that we thought it might have in the previous step, and therefore that mapping pair should be discarded. The same process is repeated for the term A and its mapped term B.

The obtained confidence level l is in $\{0,1\}$. This is a binary filtering, which only estimates whether there is semantic similarity between the mapped terms or not. The ontology mapping pair will be selected ($l = 1$) only if there is similarity between at least one pair of synsets from the set of valid synsets for A-B and the set of valid synsets for B-A. Otherwise, the mapping is rejected ($l = 0$)

Note that this method is not appropriate to evaluate disjoint mappings, producing unpredictable results. Also it is affected if the terms has no representation in WordNet. Therefore if $r = \perp$ or one of the terms to be mapped is not found in WordNet (i.e "zebrafish"), we left the value l as *undetermined*. Otherwise we compute the WordNet based *confidence level* for the mapping $\langle A, B, r, l\rangle$ as:

$$l = sim_{WN}(A, B) \tag{2}$$

Different strategies can be applied in case $l = undetermined$. By default we will not apply the filtering in these cases, thus assigning $l = 1$.

6 Combined Approach to Improve Anchoring

Finally, we propose a last strategy to improve anchoring: the combined use of the filtering schemes presented in Sections 4.2 and 5.2. We argue that, due to the different nature of these approaches, some of the false positives not filtered by one method could be detected by the other as inappropriate mappings, and vice versa. As an example, let us remind that the WordNet based method cannot evaluate disjoint mappings, thus this type of relations could be assisted by the other method. On the contrary if the internal structure of background ontologies is not rich enough, the ontological context based method could not filter properly, while the WordNet based one can.

Scheme 3 ("filtering candidate mappings by combining WordNet and Ontological Context based techniques"). Let us call l_{ont} the confidence level based on ontological context, computed with Equation 1 and l_{WN} the WordNet based confidence level obtained from Equation 2. We have identified two ways of combining both measures in an unified one:

Scheme 3.1: Promoting precision. As reported in Section 5.2, l_{WN} cannot be always computed. In such cases ($l_{WN} = undetermined$) we assign $l = l_{ont}$. Otherwise we compute the *confidence level* for the mapping $\langle A, B, r, l \rangle$ as:

$$l = min(l_{ont}, l_{WN}) \tag{3}$$

Criterion of minimizing the confidence degree optimizes precision (but penalizes recall), because the resultant filtering criteria are much more exigent: only mappings that both methods estimate as valid can pass the filter.

Scheme 3.2: Promoting recall. If $l_{WN} = undetermined$ then $l = l_{ont}$, else:

$$l = max(l_{ont}, l_{WN}) \tag{4}$$

This alternative scheme, that maximizes the confidence degree, can be used if our primary target is to obtain as many potentially good mappings as possible (among the total of valid ones), thus promoting recall instead of precision.

7 Experimental Results

Our experiments have been conducted to verify the feasibility of the proposed methods to improve the Semantic Web based Ontology Matching method. We have tested a basic implementation of Schemes 1, 2 and 3. The results confirm our initial hypothesis (the precision is increased by solving ambiguity problems) thus proving the value of the approach.

We applied our different filtering mechanisms to a sample of 354 evaluated mappings, out of the total set of data provided by the initial matching experiment mentioned in Section 2 (which lead to a baseline precision of 70%).

We have measured *precision* as the number of retrieved valid mappings out of the total which pass the filtering. Nevertheless, the filtering also rejects a number of valid mappings. In order to assess this we would need a recall measure but, due to the nature of the experiment, we are not able to provide it (our starting point, the experiment mentioned in Section 2, did not consider recall). Nevertheless we can estimate the *effect* that the filtering of mappings causes on recall (even if we do not know it), by using this expression:

$$effect\ on\ recall = \frac{number\ of\ retrieved\ valid\ mappings}{number\ of\ initial\ valid\ mappings}$$

This is a value to be multiplied by the recall of the initial matching process, to obtain the final recall. We consider as *initial valid mappings* those out of the utilized sample that are valid according to human evaluation.

7.1 Experiment 1: filtering by using Ontological Context

We have run our first experiment by applying the filtering mechanism discussed in Section 4.2. In Figure 2 we show (Scheme 1), the precision achieved by the prototype in the experiment. The worst value coincides with the baseline (70%), with minimum threshold. As we increase it, we reject more invalid mappings than valid ones, as reflects the increase of precision, which reaches soon values above **80%**. At some point (thresholds between 0.33 and 0.38) the precision moderates its ascending trend, fluctuating around 87%. This value is the predicted precision one can reach due to the anchoring improvement according to [9].

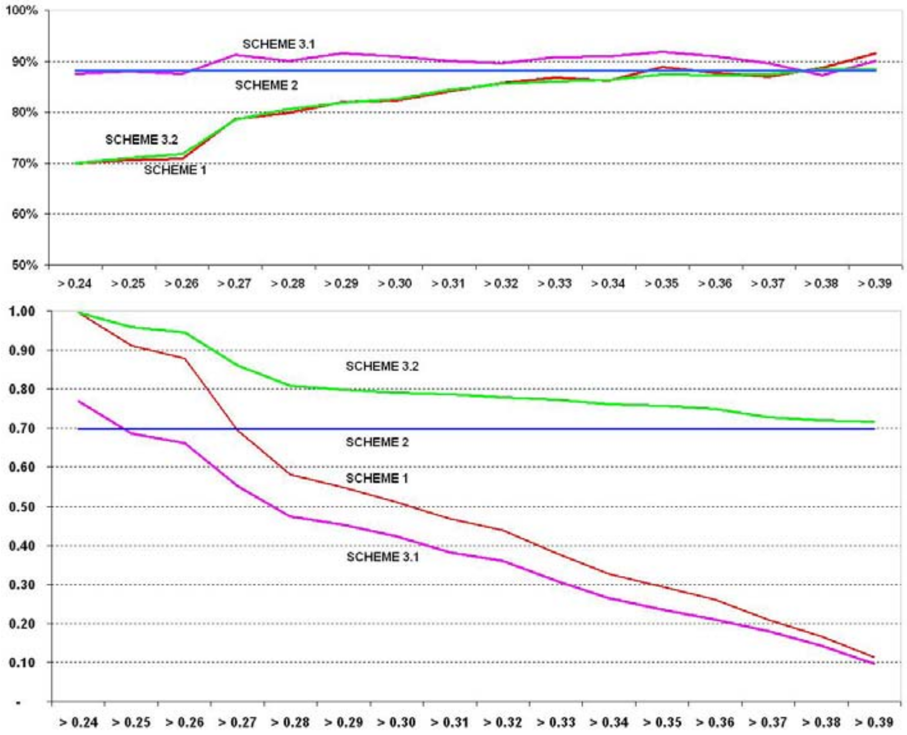


Fig. 2. *Precision* (upper) and *effect on recall* (lower). Baseline precision is 70%.

Figure 2 shows the effect on recall due to the filtering of mappings (Scheme 1). As we can expect, with the lowest threshold, no valid mappings are removed, so the recall is not influenced (effect on recall=1). As the threshold is raised the effect on recall decreases because more valid mappings are filtered out.

After first analysis of the mappings that hamper the method, we have discovered many ontological terms which are poorly described in background ontologies, and some other problems that we discuss later in Section 7.3.

Furthermore, we have run the experiment with a smaller set of randomly selected mappings (50), achieving almost identical effect on recall. This shows the feasibility of a training mechanism to obtain an optimal threshold with a small training set, to be reused later on the whole dataset.

7.2 Experiment 2: filtering by exploring WordNet

We analyse the results obtained from the same sample studied in Experiment 1. The WordNet based algorithm evaluated as correct 70% of valid mappings and 22% of invalid ones, leading to a *precision* of **88%** and an *effect on recall* of **0.70**. This sample help us to analyse the drawbacks of exclusively relying on sense information provided by WordNet to compute semantic similarity on ontology concepts. Those drawbacks are:

1. *Ontologies classes frequently use compound names without representation in WordNet.* Some compounds are not found in WordNet as such, i.e. "sugar substitutes" corresponds to two WordNet lemmas ("sugar", "substitutes"). Therefore, in many occasions the meaning can be misleading or incomplete.
2. *Synsets not related in the WordNet IS-A taxonomy.* Some terms considered similar from an ontology point of view, are not connected through a relevant IS-A path, i.e. for the term "sweeteners" and its ontological parent "food additive" (AGROVOC), therefore the taxonomical sense is unknown.
3. *The excessive fine-grainedness of WordNet sense distinctions.* For instance, the synsets of "crayfish" (AGROVOC) considering its parent "shellfish" are (1) "lobster-like crustacean..."; and (2) "warm-water lobsters without claws", but while considering its mapped term "animal" (NALT) the synset is (3) "small fresh water crustacean that resembles a lobster". This valid mapping is discarded as there is no relevant IS-A path connecting (3) with (1) or (2).
4. *Computing semantic similarity applying Resnik criteria to IS-A WordNet does not always produce good semantic mappings.* For instance the best synset obtained for, when computing the similarity between "Berries" and its parent "Plant" is "Chuck Berry – (United States rock singer)".

7.3 Experiment 3: combined approach

Here we have tested the behaviour of the improved anchoring schemes proposed in Section 6. In Figure 2 we can see the results (Schemes 3.1 and 3.2), and establish comparisons among all studied schemes.

As we predicted, Scheme 3.1 promotes precision. This combined approach slightly increases the precision achieved by Scheme 2, reaching a **92%** for a threshold of 0.285. Nevertheless we reduce recall almost to one half for this threshold. On the other hand Scheme 3.2 shows almost the same improvement in precision than Scheme 1, but with a very good behaviour in recall.

A precision of 92% obtained with Scheme 3.1 is the maximum we can reach combining both methods. At this point the system filters out most mappings considered invalid between AGROVOC and NALT, i.e. $\langle Fruit, Dessert, \perp, 0.25\rangle$ or $\langle Dehydration, drying, \sqsubseteq, 0\rangle$. Exploring the invalid mappings that pass our filters (particularly the ones that cause a slightly decrease in precision for high thresholds) we have found that the number of negative mappings due to bad anchoring is negligible, having found other types of errors that hamper our method, as bad modelling of relationships (using for example sumbsumption instead of part-of relation) i.e. $\langle East\_Asia, Asia, \sqsubseteq, 0.389\rangle$. Moreover, the meaning of an ontological concept must be precisely defined in the ontology: both similarity measures need to get the direct parents of the involved terms, but often the ancestor is Resource[8], and therefore the taxonomical meaning cannot be obtained, which introduces certain degree of uncertainty in the results.

8 Related Work

The anchoring process, where ambiguity problems can be present, is inherent to any Ontology Matching system based on background knowledge. Nevertheless most of them rely on merely syntactical techniques [1, 12]. Others, as S-Match [3], explore structural information of the term to anchor the right meaning, however it only accesses to WordNet as background knowledge source.

In some cases ambiguity in anchoring is a minor problem, because matched and background ontologies share the same domain [1], so it is expected that most polysemous terms have a well defined meaning. On the contrary, in our case the online ontologies constitute an open and heterogeneous scenario where, consequently, ambiguity becomes relevant.

Regarding the techniques we use from Word Sense Disambiguation, many others could be applied (see [10, 6] for example). Nevertheless we have selected the synonymy measure used in [11] to perform our disambiguation tasks because it has some convenient properties: it is not domain-dependent, it does not depend on a particular lexical resource, and it was conceived to deal with online ontologies. We also included the PowerMap based technique [4] to take advantage of the high quality description and coverage that WordNet provides, and because it combines in a clever way some well founded ideas from traditional Word Sense Disambiguation [7, 13].

9 Conclusions and Future Work

In this paper, we have presented different strategies to improve the precision of background knowledge based Ontology Matching systems, by considering the semantics of the terms to be anchored in order to deal with possible ambiguities during the anchoring process. We have explored the application of two similarity measures: one based on the ontological context of the terms, and another based on WordNet. A final strategy has been conceived by combining both measures.

In order to apply our ideas we have focused on a matcher that uses the Semantic Web as source of background knowledge. Our experimental results show that all filtering strategies we have designed improve the precision of the system (initially 70%). For example our Scheme 3.2 can reach a precision of 87%, affecting the overall recall in only a factor of 0.76.

Our experimental results encourage us to tackle further improvements and tests to our matching techniques. For example, a more advanced prototype will

---

[8] http://www.w3.org/2000/01/rdf-schema#Resource

be developed, which fully integrates the Semantic Web based Ontology Matcher with the filtering schemes that we have tested here. Also we will explore new ways to exploit semantics *during* the anchoring process, not only after it (as we currently do in our filtering schemes).

Acknowledgments. This work was founded by the OpenKnowledge IST-FF6-027253, NeOn IST-FF6-027595 and Spanish CICYT TIN2004-07999-C02-02 projects, and by the Government of Aragon-CAI grant IT11/06.

References

1. Z. Aleksovski, M. C. A. Klein, W. ten Kate, and F. van Harmelen. Matching unstructured vocabularies using a background ontology. In *EKAW*, volume 4248 of *LNSC*, pages 182–197, 2006.
2. M. Espinoza, R. Trillo, J. Gracia, and E. Mena. Discovering and merging keyword senses using ontology matching. In *1st International Workshop on Ontology Matching (OM-2006) at ISWC-2006, Athens, Georgia (USA)*, November 2006.
3. F. Giunchiglia, P. Shvaiko, and M. Yatskevich. Semantic schema matching. In *OTM Conferences (1)*, volume 3760 of *LNCS*, pages 347–365. Springer, 2005.
4. V. Lopez, M. Sabou, and E. Motta. Powermap: Mapping the real semantic web on the fly. In *Proc. of 5th International Semantic Web Conference, Athens, GA, USA, November 5-9, 2006, LNCS 4273*, 2006.
5. E. Motta and M. Sabou. Next generation semantic web applications. In *1st Asian Semantic Web Conference*, volume 4185 of *Lecture Notes in Computer Science*, pages 24–29. Springer, 2006.
6. R. Navigli and P. Velardi. Structural semantic interconnections: A knowledge-based approach to word sense disambiguation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(7):1075–1086, 2005.
7. P. Resnik. Disambiguating noun groupings with respect to Wordnet senses. In *Proceedings of the Third Workshop on Very Large Corpora*, pages 54–68. Association for Computational Linguistics, 1995.
8. M. Sabou, M. d'Aquin, and E. Motta. Using the semantic web as background knowledge for ontology mapping. In *1st International Workshop on Ontology Matching (OM-2006) at ISWC-2006, Athens, Georgia (USA)*, November 2006.
9. M. Sabou, M. d'Aquin, and E. Motta. Using the semantic web as background knowledge for ontology matching: A large-scale evaluation. *Submitted for peer review*, 2007.
10. S. P. Ted Pedersen, Satanjeev Banerjee. Maximizing semantic relatedness to perform word sense disambiguation, 2005. University of Minnesota Supercomputing Institute Research Report UMSI 2005/25.
11. R. Trillo, J. Gracia, M. Espinoza, and E. Mena. Discovering the semantics of user keywords. *Journal on Universal Computer Science. Special Issue: Ontologies and their Applications*, November 2007.
12. W. R. van Hage, S. Katrenko, and G. Schreiber. A method to combine linguistic ontology-mapping techniques. In *International Semantic Web Conference*, 2005.
13. Z. Wu and M. Palmer. Verb semantics and lexical selection. In *32nd. Annual Meeting of the Association for Computational Linguistics*, pages 133 –138, New Mexico State University, Las Cruces, New Mexico, 1994.

Structure preserving semantic matching

Fausto Giunchiglia[1], Mikalai Yatskevich[1] and Fiona McNeill[2]

[1] Dept. of Information and Communication Technology, University of Trento, 38050
Povo, Trento, Italy
{fausto,mikalai.yatskevich}@dit.unitn.it
[2] School of Informatics, University of Edinburgh, EH8 9LE, Scotland
f.j.mcneill@ed.ac.uk

Abstract The most common matching applications, e.g., ontology matching, focus on the computation of the correspondences holding between the nodes of graph structures (e.g., concepts in two ontologies). However there are applications, such as matching of web service descriptions, where matching may need to compute the correspondences holding between the full graph structures and to preserve certain structural properties of the graphs being considered. The goal of this paper is to provide a new matching operator, that we call *structure preserving match*. This operator takes two graph-like structures and produces a mapping between those nodes of the structures that correspond semantically to each other, (*i*) still preserving a set of structural properties of the graphs being matched, (*ii*) only in the case that the graphs *globally* correspond semantically to each other. We present an exact and an approximate structure matching algorithm. The latter is based on a formal theory of abstraction and builds upon the well known tree edit distance measures. We have implemented the algorithms and applied them to the web service matchmaking scenario. The evaluation results, though preliminary, show the efficiency and effectiveness of our approach.

1 Introduction

We are interested in the problem of location of web services on the basis of the capabilities that they provide. This problem is often referred to as the matchmaking problem; see [12,14,15] for some examples. Most previous solutions employ a single ontology approach, that is, the web services are assumed to be described by the concepts taken from a shared ontology. This allows the reduction of the matchmaking problem to the problem of reasoning within the shared ontology. In contrast to this work, as described in [6,19], we assume that the web services are described using terms from different ontologies and that their behaviour is described using complex terms, actually first order terms. This allows us to provide detailed descriptions of their input and output behaviour. The problem becomes therefore that of matching two web service descriptions (which can be seen as graph structures) and the mapping is considered as successful only if the two graphs are *globally* similar (e.g., $tree_1$ is 0.7 similar to $tree_2$, according to some metric). A further requirement of these applications is that the mapping must preserve certain structural properties of the graphs being considered. In particular, the syntactic types and sorts have to be preserved (e.g., a function symbol must be mapped to a function symbol and a variable must be mapped to a variable). At the same time we would like to enable the matchmaking of the web service descriptions that match only approximately (see [6] for a detailed description). For instance, $get\_wine(Region, Country, Colour, Price, Number\_of\_bottles)$ can be approximately mapped to $get\_wine(Region(Country, Area), Colour, Cost, Year, Quantity)$.

In this paper, we define an operator that we call *structure preserving match*. This operator takes two graph-like structures and produces a mapping between those nodes of the structures that correspond semantically to each other, (*i*) still preserving a set of structural properties of the graphs being matched, (*ii*) only in the case that the graphs *globally* correspond semantically to each other. Notice that this problem significantly differs from the ontology matching problem, as defined for instance in [8], where (*i*) is only partially satisfied and (*ii*) is an issue which is hardly ever dealt with (see [12,23] for some noticeable exceptions). We present an exact and an approximate structure matching algorithm. The former solves the exact structure matching problem. It is designed to succeed on equivalent terms and to fail otherwise. The latter solves an approximate structure matching problem. It is based on the fusion of the ideas derived from the theory of abstraction [7] and tree edit distance algorithms [3,28]. We have implemented the algorithms and evaluated them on the dataset constructed from different versions of the state-of-the-art first order ontologies. The evaluation results, though preliminary, show the efficiency and effectiveness of our approach.

Section 2 introduces a motivating example, Section 3 discusses the exact structure preserving semantic matching. Section 4 defines the abstraction operations and introduces the correspondence between them and tree edit operations. In Section 5 we show how existing tree edit distance algorithms can be exploited for the computation of the global similarity between two web service descriptions. Section 6 is devoted to the approximate structure matching algorithm. The evaluation results are presented in Section 7. Section 8 briefly reviews the related work and concludes the paper.

2 Motivating example

Figure 1 provides an example of exactly matched web service descriptions along with their tree representations (or term trees). Dashed lines stand for the correspondences holding among the nodes of the term trees. In particular, in Figure 1 we have an exact match, namely the first of the services requires the second to return *Cars* of a given *Brand*, *Year* and *Colour* while the other provides *Autos* of a given *Brand*, *Year* and *Colour*. Notice that there are no structural differences and that the only difference is in the function names. Where these names differ, their semantic content remains the same (*e.g.*, *Colour* is semantically identical to *Colour*) and therefore the two descriptions constitute an exact match.

Figure 2 provides an example of an approximate match. In this case a more sophisticated data translation is required. For example, the first web service description requires the fourth argument of *get_wine* function (*Colour*) to be mapped to the second argument (*Colour*) of *get_wine* function in the second description. On the other

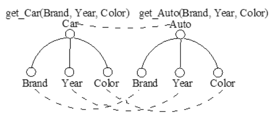


Figure 1: Exactly matched web service descriptions and their tree representations.

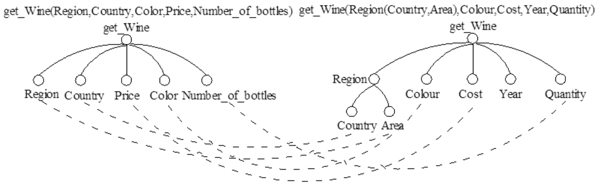


Figure 2: Approximately matched web service descriptions and their tree representations.

hand, *Region* on the right is defined as a function with two arguments (*Country* and *Area*) while on the left *Region* is an argument of *get_wine*. Thus, *Region* in the first web service description must be passed to the second web service as the value of the *Area* argument of the *Region* function. Moreover, *Year* on the right has no corresponding term on the left.

Therefore, in order to guarantee the successful data translation, we are interested in the correspondences holding among the nodes of the term trees of the given web service descriptions only in the case when the web service descriptions themselves are "similar enough". At the same time the correspondences have to preserve the certain structural properties of the descriptions being matched. In particular we require functions to be mapped to functions and variables to variables. We can see how the context is preserved through this mapping: for example, the two nodes *Colour* are mapped to one another, but this is done in the context that they are both children of nodes *get_wine* that are also mapped to one another. Thus we can tell that *Colour* is likely to mean the same thing in both cases.

3 Exact structure semantic matching

There are two stages in the matching process:

- *Node matching*: solves the semantic heterogeneity problem by considering only labels at nodes and domain specific contextual information of the trees. In our approach we use semantic matching, as extensively described in [8]. Notice that the result of this stage is the set of correspondences holding between the nodes of the trees.
- *Structural tree matching*: exploits the results of the node matching and the structure of the tree to find the correspondences holding between the trees themselves (e.g., $tree_1$ is 0.7 similar to $tree_2$).

The exact structure matching algorithm exploits the results of the node matching algorithm. It is designed to succeed for equivalent terms and to fail otherwise. It expects the trees to have the same depth and the same number of children. More precisely we say that two trees T_1 and T_2 match iff for any node n_{1i} (numbers in subscript refer to the tree and the node in this tree, respectively) in T_1 there is a node n_{2j} in T_2 such that:

- n_{1i} semantically matches n_{2j};
- n_{1i} and n_{2j} reside on the same depth in T_1 and T_2, respectively;
- all ancestors of n_{1i} are semantically matched to the ancestors of n_{2j}.

We do not discuss the exact structure preserving matching any further, since its implementation is straightforward, see [6] for details.

4 Approximate matching via abstraction/refinement operations

In [7], Giunchiglia and Walsh describe their theory of abstraction. We present here the key concepts in order to facilitate the presentation of our approach, which builds upon this work. Giunchiglia and Walsh categorise the various kinds of abstraction operations in a wide-ranging survey. They also introduce a new class of abstractions, called TI-abstractions (where TI means "Theorem Increasing"), which have the fundamental property of maintaining completeness, while losing correctness. In other words, any fact that is true of the original term is also true of the abstract term, but not vice versa. Similarly, if a ground formula is true, so is the abstract formula, but not vice versa. Dually, by taking the inverse of each abstraction operation, we can define a corresponding refinement operation which preserves correctness while losing completeness. The second fundamental property of the abstraction operations is that they provide all and only the possible ways in which two first order terms can be made to differ by manipulations of their signature, still preserving completeness. In other words, this set of abstraction/refinement operations defines all and only the possible ways in which correctness and completeness are maintained when operating on first order terms and atomic formulas. This is the fundamental property which allows us to study and consequently quantify the semantic similarity (distance) between two first order terms. To this extent it is sufficient to determine which abstraction/refinement operations are necessary to convert one term into the other and to assign to each of them a cost that models the "semantic distance" associated to the operation.

Giunchiglia and Walsh's categories are as follows:

Predicate: Two or more predicates are merged, typically to the least general generalisation in the predicate type hierarchy, e.g.,

– $Bottle(X) + Container(X) \mapsto Container(X)$.

We call $Container(X)$ a predicate abstraction of $Bottle(X)$ or $Container(X) \sqsupseteq_{Pd} Bottle(X)$. Conversely we call $Bottle(X)$ a predicate refinement of $Container(X)$ or $Bottle(X) \sqsubseteq_{Pd} Container(X)$.

Domain: Two or more terms are merged, typically by moving the functions (or constants) to the least general generalisation in the domain type hierarchy, e.g.,

– $Daughter(Me) + Child(Me) \mapsto Child(Me)$.

Similarly to the previous item we call $Child(Me)$ a domain abstractions of $Daughter(Me)$ or $Child(Me) \sqsupseteq_D Daughter(Me)$. Conversely we call $Daughter(Me)$ a domain refinements of $Child(Me)$ or $Daughter(Me) \sqsubseteq_D Child(Me)$.

Propositional: One or more arguments are dropped, e.g.,

– $Bottle(A) \mapsto Bottle$.

We call $Bottle$ a propositional abstraction of $Bottle(A)$ or $Bottle \sqsupseteq_P Bottle(A)$. Conversely $Bottle(A)$ is a propositional refinement of $Bottle$ or $Bottle(A) \sqsubseteq_P Bottle$.

Precondition: The precondition of a rule is dropped [1] , e.g.,

– $[Ticket(X) \rightarrow Travel(X)] \mapsto Travel(X)$.

Consider the pair of first order terms ($Bottle\ A$) and ($Container$). In this case there is no abstraction/refinement operation that make them equivalent. However consequent applications of propositional and predicate abstraction operations make the two terms

---

[1] We do not consider precondition abstraction and refinement in the rest of this paper as we do not want to drop preconditions, because this would endanger the successful matchmaking of web services.

equivalent:

$$(Bottle\ A) \mapsto^{\sqsubseteq_P} (Bottle) \mapsto^{\sqsupseteq_{Pd}} (Container) \tag{1}$$

In fact the relation holding among the terms is a composition of two refinement operations, namely $(Bottle\ A) \sqsubseteq_P (Bottle)$ and $(Bottle) \sqsubseteq_{Pd} (Container)$. We define an *abstraction mapping element (AME)* as a 5-tuple $\langle ID_{ij}, t_1, t_2, R, sim\rangle$, where ID_{ij} is a unique identifier of the given mapping element; t_1 and t_2 are first order terms; R specifies a relation for the given terms; and sim stands for a similarity coefficient in the range [0..1] quantifying the strength of the relation. In particular for the AMEs we allow the semantic relations $\{\equiv, \sqsupseteq, \sqsubseteq\}$, where $\equiv$ stands for equivalence, $\sqsupseteq$ represents an abstraction relation and connects the precondition and the result of a composition of arbitrary numbers of predicate, domain and propositional abstraction operations, and $\sqsubseteq$ represents a refinement relation and connects the precondition and the result of a composition of arbitrary numbers of predicate, domain and propositional refinement operations.

Therefore, the problem of AME computation becomes a problem of minimal cost composition of the abstraction/refinement operations allowed for the given relation R that are necessary to convert one term into the other. In order to solve this problem we propose to represent abstraction/refinement operations as tree edit distance operations applied to the term trees. Calculating the cost of moving between nodes therefore becomes the problem of determining whether these nodes are equivalent, an abstraction or refinement of one another, or none of these relations. Note that this calculation does not in general require specific background knowledge; the semantic matching techniques allow us to calculate this automatically. Naturally, the semantic matching techniques themselves require some kind of background knowledge but this is not specific: currently, our semantic matching techniques use WordNet; see [8] for more details. This allows us to redefine the problem of AME computation into a tree edit distance problem.

In its traditional formulation, the tree edit distance problem considers three operations: (i) vertex deletion, (ii) vertex insertion, and (iii) vertex replacement [25]. Often these operations are presented as rewriting rules:

$$(i)\ \ \upsilon \rightarrow \lambda \qquad (ii)\ \ \lambda \rightarrow \upsilon \qquad (iii)\ \ \upsilon \rightarrow \omega \tag{2}$$

where υ and ω correspond to the labels of nodes in the trees while λ stands for the special blank symbol. Figure 3 illustrates two applications of delete and replace tree edit operations.

Our proposal is to restrict the formulation of the tree edit distance problem in order to reflect the semantics of the first order terms. In particular we propose to redefine the tree edit distance operations in a way that will allow them to have one-to-one correspondence to the abstraction/refinement operations presented previously in this section. Table 1 illustrates the correspondence between abstraction/refinement and tree edit operations. The first column presents the abstraction/refinement operations. The second column lists corresponding tree edit operations. The third column describes the preconditions of the tree edit operation use. Consider, for example, the first line of Table 1. The predicate abstraction operation applied to first order term t_1 results with term t_2 ($t_1 \sqsupseteq_{Pd} t_2$). This abstraction operation corresponds to a tree edit replacement operation applied to the term tree of t_1 that replaces the node a with the node $b(a \rightarrow b)$. Moreover the operation can be applied only in the case that (i) label a is a generalisation of label b and (ii) both nodes a and b in the term trees correspond to predicates in the first order terms.

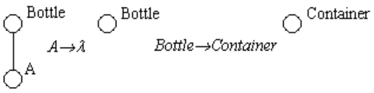


Figure 3: Delete and replace tree edit operations

Table 1: The correspondence between abstraction/refinement operations and tree edit operations.

Abstraction/refinement operation	Tree edit operation	Preconditions of operation
$t_1 \sqsupseteq_{Pd} t_2$	$a \rightarrow b$	$a \sqsupseteq b$; a and b correspond to predicates
$t_1 \sqsupseteq_{D} t_2$	$a \rightarrow b$	$a \sqsupseteq b$; a and b correspond to functions or constants
$t_1 \sqsupseteq_{P} t_2$	$a \rightarrow \lambda$	a corresponds to predicates, functions or constants
$t_1 \sqsubseteq_{Pd} t_2$	$a \rightarrow b$	$a \sqsubseteq b$; a and b correspond to predicates
$t_1 \sqsubseteq_{D} t_2$	$a \rightarrow b$	$a \sqsubseteq b$; a and b correspond to functions or constants
$t_1 \sqsubseteq_{P} t_2$	$a \rightarrow \lambda$	a corresponds to predicates, functions or constants

5 Computing the global similarity between two trees

Our goal now is to compute the similarity between two term trees. In order to perform this we need to compute the minimal cost composition of the abstraction/refinement operations that are necessary to convert one term tree (or first order term) into the other. The starting point is the traditional formulation of the tree edit distance problem.

$$Cost = \sum_{i \in S} n_i * Cost_i \tag{3}$$

The similarity between two trees is thus the minimal possible *Cost* as defined in Eq. 3; that is, the set of operations that transforms one tree into another at minimal cost. In Eq. 3, S stands for the set of the allowed tree edit operations; n_i stands for the number of *i-th* operations necessary to convert one tree into the other and $Cost_i$ defines the cost of the *i-th* operation. Our goal is to define the $Cost_i$ in a way that models the semantic distance between the two trees.

A possible uniform proposal is to assign the same unit cost to all tree edit operations that have counterparts in the theory of abstraction. These are defined in Table 1. Table 2 illustrates the costs of the abstraction/refinement (tree edit) operations, depending on the relation (equivalence, abstraction or refinement) being computed. These costs have to be adjusted depending on what relation is being considered: for example, the cost of applying an abstraction operation is different if we are considering abstraction relations than if we are considering refinement relations. In particular, the tree edit operations corresponding to abstraction/refinement operations that are not allowed by the definition of the given relation have to be prohibited by assigning to them an infinite cost. Notice also that we do not give any preference to a particular type of abstraction/refinement operations. Of course this strategy can be changed to satisfy certain domain specific requirements.

Consider, for example, the first line in Table 2. The cost of the tree edit distance operation that correspond to the propositional abstraction ($t_1 \sqsupseteq_{Pd} t_2$) is equal to 1 when used for the computation of equivalence ($Cost_{\equiv}$) and abstraction ($Cost_{\sqsupseteq}$) relations in AME. It is equal to ∞ when used for the computation of refinement ($Cost_{\sqsubseteq}$) relation.

Table 2: Costs of the abstraction/refinement (tree edit) operations, exploited for computation of equivalence ($Cost_{\equiv}$), abstraction ($Cost_{\sqsubseteq}$) and refinement ($Cost_{\sqsupseteq}$) relations holding among the terms.

Abstraction/refinement (tree edit) operation	$Cost_{\equiv}$	$Cost_{\sqsubseteq}$	$Cost_{\sqsupseteq}$
$t_1 \sqsupseteq_{Pd} t_2$	1	∞	1
$t_1 \sqsupseteq_{D} t_2$	1	∞	1
$t_1 \sqsupseteq_{P} t_2$	1	∞	1
$t_1 \sqsubseteq_{Pd} t_2$	1	1	∞
$t_1 \sqsubseteq_{D} t_2$	1	1	∞
$t_1 \sqsubseteq_{P} t_2$	1	1	∞

Eq. 3 can now be used for the computation of the tree edit distance score. However, when comparing two web service descriptions we are interested in similarity rather than in distance. We exploit the following equation to convert the distance produced by an edit distance algorithm into the similarity score:

$$sim = 1 - \frac{Cost}{max(number\_of\_nodes_1, number\_of\_nodes_2)} \tag{4}$$

where $number\_of\_nodes_1$ and $number\_of\_nodes_2$ stand for the number of nodes in the trees. Note that for the special case of $Cost$ equal to ∞ the similarity score is estimated as 0.

Many existing tree edit distance algorithms allow us to keep track of the nodes to which a replace operation is applied. Therefore, as a result they allow us to obtain not only the minimal tree edit cost but also a minimal cost mapping among the nodes of the trees. According to [25], this minimal cost mapping is (*i*) one-to-one; (*ii*) horizontal-order preserving between sibling nodes; and (*iii*) vertical-order preserving. These criteria are not always preserved in our approach. For example, the mapping depicted in Figure 1 complies to all these requirements while the mapping depicted in Figure 2 violates (*ii*). In particular the third sibling *Price* on the left tree is mapped to the third sibling *Cost* on the right tree while the fourth sibling *Colour* on the right tree is mapped to the second sibling *Colour* on the left tree.

For the tree edit distance operations depicted in Table 1, we propose to keep track of nodes to which the tree edit operations derived from the replace operation are applied. In particular we consider the operations that correspond to predicate and domain abstraction/refinement ($t_1 \sqsupseteq_{Pd}$, $t_1 \sqsubseteq_{Pd}$, $t_1 \sqsupseteq_{D}$, $t_1 \sqsubseteq_{D}$). This allows us to obtain a mapping among the nodes of the term trees with the desired properties (i.e., there is only one-to-one correspondences in the mapping). Moreover it complies to the structure preserving matching requirements that functions are mapped to functions and variables are mapped to variables. This is the case because (*i*) predicate and domain abstraction/refinement operations do not convert, for example, a function into a variable and (*ii*) the tree edit distance operations, as from Table 1, have a one-to-one correspondence with abstraction/refinement operations.

At the same time, a mapping returned by a tree edit distance algorithm preserves the horizontal order among the sibling nodes, but this is not desirable property for the data translation purposes. This is the case because the correspondences that do not comply to the horizontal order preservation requirements, like the one holding between *Colour* and *Colour* on Figure 2, are not included in the mapping. However, as from Table 1, the tree edit operations corresponding to predicate and domain abstraction/refinement ($t_1 \sqsupseteq_{Pd}$, $t_1 \sqsubseteq_{Pd}$, $t_1 \sqsupseteq_{D}$, $t_1 \sqsubseteq_{D}$) can be applied only to those nodes of the trees whose labels are either generalisations or specialisations of each other, as computed by the node matching algorithm. Therefore, given the mapping produced by the node matching algorithm we can always recognise the cases when the horizontal order between sibling nodes is not preserved and change the ordering of the sibling nodes to make the mapping horizontal order preserving. For example, swapping the nodes *Cost* and *Colour* in the right tree depicted on Figure 2 does not change the meaning of the corresponding term but allows the correspondence holding between *Colour* and *Colour* on Figure 2 to be included in the mapping produced by a tree edit distance algorithm.

We can see that this technique satisfies the two properties mentioned earlier: namely, that the operator finds a mapping (*i*) still preserving a set of structural properties of the graphs being matched, (*ii*) only in the case that the graphs *globally* correspond semantically to each other. If the graphs do not correspond semantically to one another, and the structural properties of the graphs do not match, the similarity score will be very low.

6 The approximate structure matching algorithm

As discussed above, our goal is to find good enough services [9] if perfect services are not available. We start by providing a definition of the approximate structure matching as the basis for the algorithm.

We say that two nodes n_1 and n_2 in trees T_1 and T_2 approximately match iff $c@n_1$ R $c@n_2$ holds given the available background knowledge, where $c@n_1$ and $c@n_2$ are the concepts at nodes of n_1 and n_2, and where $R \in \{\equiv, \sqsubseteq, \sqsupseteq\}$. We say that two trees T_1 and T_2 match iff there is at least one node n_{1i} in T_1 and a node n_{2j} in T_2 such that: (*i*) n_{1i} approximately matches n_{2j} and (*ii*) all ancestors of n_{1i} are approximately matched to the ancestors of n_{2j}.

First the approximate structure matching algorithm estimates the similarity of two terms by application of a tree edit distance algorithm with the tree edit operations and costs modified as described in Sections 4 and 5. The similarity scores are computed for equivalence, abstraction and refinement relations. For each of these cases the tree edit distance operation costs are modified as depicted in Table 2. The relation with the highest similarity score is assumed to hold among the terms. If the similarity score exceeds a given threshold, the mappings connecting the nodes of the term trees, as computed by the tree edit distance algorithm, are returned by the matching routine what allows for further data translation. Algorithm 1 provides pseudo code for the approximate structure matching algorithm.

approximateStructureMatch takes as input the *source* and *target* term trees and a *threshold* value. approximateTreeMatch fills the *result* array (line 3) which stores the mappings holding between the nodes of the trees. An AME *ame* is computed (line 4) by analyzeMismatches. If *ame* stands for equivalence, abstraction or refinement relations (line 5) and if an *approximationScore* exceeds *threshold* (line 6) the mappings calculated by approximateTreeMatch are returned (line 7). analyzeMismatches calculates the aggregate score of tree match quality by exploiting a tree edit distance algorithm as described in Section 5.

Algorithm 1 Pseudo code for approximate structure matching algorithm

```
AME struct of
    Tree of Nodes source;
    Tree of Nodes target;
    String relation;
    double approximationScore;

MappingElement[] approximateStructureMatch(Tree of Nodes source, target, double threshold)
 MappingElement[] result;
 approximateTreeMatch(source,target,result);
 AME ame=analyzeMismatches(source,target,result);
 if (getRelation(ame)=="=") or (getRelation(ame)=="<") or (getRelation(ame)==">")
     if (getApproximationScore(ame)>threshold)
     return result;
 return null;
```

7 Evaluation

We have implemented the algorithm described in the previous section. In the implementation we have exploited a modification of simple tree edit distance algorithm from Valiente's work [27]. We have evaluated the matching quality of the algorithms on 132 pairs of first order logic terms. Half of the pairs were composed of the equivalent terms (e.g., *journal(periodical-publication)* and *magazine (periodical-publication)*) while the other half were composed from similar but not equivalent terms (e.g., *web-reference(publication-reference)* and *thesis-reference (publication-reference)*). Te terms were extracted from different versions of the Standard Upper Merged Ontology (SUMO)[2] and the Advanced Knowledge Technology (AKT)[3] ontologies. We extracted all the differences between versions 1.50 and 1.51, and between versions 1.51 and 1.52 of the SUMO ontology and between versions 1, 2.1 and 2.2 of the AKT-portal and AKT-support ontologies[4]. These are both first-order ontologies, so many of these differences mapped well to the potential differences between terms that we are investigating. However, some of them were more complex, such as differences in inference rules, or consisted of ontological objects being added or removed rather than altered, and had no parallel in our work. These pairs of terms were discarded and our tests were run on all remaining differences between these ontologies. Therefore, we have simulated the situation when the service descriptions are defined exploiting the two versions of the same ontology.

In our evaluation we have exploited the commonly accepted measures of matching quality, namely precision, recall, and F-measure. Precision varies in the [0..1] range; the higher the value, the smaller the set of incorrect correspondences (false positives) which have been computed. Precision is a correctness measure. Recall varies in the [0..1] range; the higher the value, the smaller the set of correct correspondences (true positives) which have not found. Recall is a completeness measure. F-measure varies in the [0..1] range. The version computed here is the harmonic mean of precision and recall. It is a global measure of the matching quality, increasing as the matching quality improves. While computing precision and recall we have considered the correspondences holding among first order terms rather than the nodes of the term trees. Thus, for instance, *journal(periodical-publication*$_1$*)=magazine(periodical-publication*$_2$*)* was con-

---

[2] `http://ontology.teknowledge.com/`

[3] `http://www.aktors.org`

[4] See `http://dream.inf.ed.ac.uk/projects/dor/` for full versions of these ontologies and analysis of their differences.

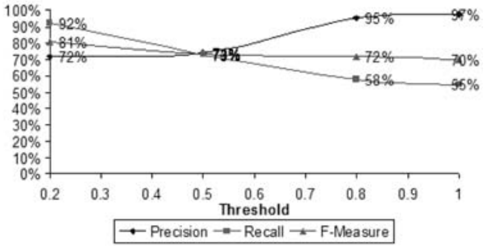


Figure 4: The matching quality measures depending on threshold value for approximate structure matching algorithm.

sidered as single correspondence rather than two correspondences, namely *journal*= *magazine* and *periodical-publication*$_1$=*periodical-publication*$_2$.

Interestingly enough, our exact structure matching algorithm was able to find 36 correct correspondences what stands for 54% of Recall with 100% Precision. All mismatches (or correct correspondences not found by the algorithm) corresponded to structural differences among first order terms which exact structure matching algorithm is unable to capture. The examples of correctly found correspondences are given below:

```
meeting-attendees(has-other-agents-involved) : meeting-attendee(has-other-agents-involved)

r&d-institute(Learning-centred-organization) : r-and-d-institute(Learning-centred-organization)

piece(Pure2,Mixture) : part(Pure2,Mixture)

has-affiliatied-people(Affiliated-person) : has-affililated-person(affiliated-person)
```

The first and second examples illustrate the minor syntactic differences among the terms, while the third and fourth examples illustrate the semantic heterogeneity in the various versions of the ontologies.

Figure 4 presents the matching quality measures depending on the cut-off threshold value for approximate structure preserving matching algorithm. As illustrated in Figure 4, the algorithm demonstrates high matching quality on the wide range of threshold values. In particular, F-Measure values exceed 70% for the given range. Table 3 summarizes the time performance of the matching algorithm. It presents the average time taken by the various steps of the algorithm on 132 term matching tasks. As illustrated in Table 3, Step 1 and 2 of the node matching algorithm significantly slow down the whole process. However these steps correspond to the linguistic preprocessing that can be performed once offline [8]. Given that the terms can be automatically annotated with the linguistic preprocessing results [8] once when changed, the overall runtime is reduced to 4.2 ms, which corresponds roughly to 240 term matching tasks per second.

Table 3: Time performance of approximate structure matching algorithm (average on 132 term matching tasks).

	Node matching: steps 1 and 2 [8]	Node matching: steps 3 and 4 [8]	Structure matching
Time, ms	134.1	3.3	0.9

8 Conclusions and Related work

We have presented an approximate structure matching algorithm that implements the *structure preserving match* operator. We have implemented the algorithm and applied it to the web service matchmaking scenario. The evaluation results, though preliminary, show the efficiency and effectiveness of our approach.

Future work includes further investigations on the cost assignment for the abstraction/refinement operations. In the version of the algorithm presented in the paper, no preference is given to the particular abstraction/refinement operation and all allowed operations are assigned a unit cost. One may argue, for example, that the semantic distance between *cat* and *mammal* is less then the semantic distance between *cat* and *animal*. Therefore, the operation abstracting *cat* to *mammal* should be less costly than the operation abstracting *cat* to *animal*.

The problem of location of web services on the basis of the capabilities that they provide (often referred as the matchmaking problem) has recently received a considerable attention. Most of the approaches to the matchmaking problem so far employed a single ontology approach (i.e., the web services are assumed to be described by the concepts taken from the shared ontology). See [14,15,21] for example. Probably the most similar to ours is the approach taken in METEOR-S [1] and in [20], where the services are assumed to be annotated with the concepts taken from various ontologies. Then the matchmaking problem is solved by the application of the matching algorithm. The algorithm combines the results of atomic matchers that roughly correspond to the element level matchers exploited as part of our algorithm. In contrast to this work, we exploit a more sophisticated matching technique that allows us to utilise the context provided by the first order term.

Many diverse solutions to the ontology matching problem have been proposed so far. See [23] for a comprehensive survey and [5,18,4,10,2,12,24] for individual solutions. However most efforts has been devoted to computation of the correspondences holding among the classes of description logic ontologies. Recently, several approaches allowed computation of correspondences holding among the object properties (or binary predicates) [26]. The approach taken in [11] facilitates the finding of correspondences holding among parts of description logic ontologies or subgraphs extracted from the ontology graphs. In contrast to these approaches, we allow the computation of correspondences holding among first order terms.

9 Acknowledgements

This work has been supported by the STReP OpenKnowledge (http://www.openk.org/). The authors are indebted to Pavel Shvaiko for his extensive feedback on earlier versions of this paper.

References

1. R. Aggarwal, K. Verma, J. A. Miller, and W. Milnor. Constraint driven web service composition in METEOR-S. In *Proceedings of IEEE SCC*, pages 23–30, 2004.
2. S. Bergamaschi, S. Castano, and M. Vincini. Semantic integration of semistructured and structured data sources. *SIGMOD Record*, 28(1):54–59, 1999.
3. W. Chen. New algorithm for ordered tree-to-tree correction problem. *Journal of Algorithms*, 40(2):135–158, 2001.
4. M. Ehrig, S. Staab, and Y. Sure. Bootstrapping ontology alignment methods with APFEL. In *Proceedings of ISWC*, pages 186–200, 2005.
5. J. Euzenat and P. Valtchev. Similarity-based ontology alignment in OWL-lite. In *Proceedings of ECAI*, pages 333–337, 2004.

6. F. Giunchiglia, F. McNeill, and M. Yatskevich. Web service composition via semantic matching of interaction specifications. Technical Report DIT-06-080, University of Trento, 2006.
7. F. Giunchiglia and T. Walsh. A theory of abstraction. *Artificial Intelligence*, 57(2-3):323–389, 1992.
8. F. Giunchiglia, M. Yatskevich, and P. Shvaiko. Semantic matching: Algorithms and implementation. *Journal on Data Semantics*, IX:1–38, 2007.
9. F. Giunchiglia and I. Zaihrayeu. Making peer databases interact – a vision for an architecture supporting data coordination. In *Proceedings of the CIA workshop*, pages 18–35, 2002.
10. R. Gligorov, Z. Aleksovski, W. ten Kate, and F. van Harmelen. Using google distance to weight approximate ontology matches. In *Proceedings of WWW*, 2007.
11. W. Hu and Y. Qu. Block matching for ontologies. In *Proceedings of ISWC*, pages 300–313, 2006.
12. Y. Kalfoglou and M. Schorlemmer. IF-Map: an ontology mapping method based on information flow theory. *Journal on Data Semantics*, I:98–127, 2003.
13. J. Kim, M. Jang, Y.-G. Ha, J.-C. Sohn, and S.-J. Lee. MoA: OWL ontology merging and alignment tool for the semantic web. In *Proceedings of IEA/AIE*, pages 722–731, 2005.
14. M. Klusch, B. Fries, and K. Sycara. Automated semantic web service discovery with OWLS-MX. In *Proceedings of AAMAS*, pages 915–922, 2006.
15. L. Li and I. Horrocks. A software framework for matchmaking based on semantic web technology. In *Proceedings of WWW*, pages 331–339, 2003.
16. V. Lopez, M. Sabou, and E. Motta. Powermap: Mapping the real semantic web on the fly. In *Proceedings of ISWC*, pages 414–427, 2006.
17. P. Mitra, N. Noy, and A. Jaiswal. Omen: A probabilistic ontology mapping tool. In *Proceedings of ISWC*, pages 537–547, 2005.
18. N. Noy and M. Musen. The PROMPT suite: interactive tools for ontology merging and mapping. *International Journal of Human-Computer Studies*, 59(6):983–1024, 2003.
19. OpenKnowledge Manifesto. http://www.openk.org/, 2006.
20. S. Oundhakar, K. Verma, K. Sivashanugam, A. Sheth, and J. Miller. Discovery of web services in a multi-ontology and federated registry environment. *International Journal of Web Services Research*, 2(3):1–32, 2005.
21. M. Paolucci, T. Kawamura, T. Payne, and K. Sycara. Semantic matching of web services capabilities. In *Proceedings of ISWC*, pages 333–347, 2002.
22. Y. Qu, W. Hu, and G. Cheng. Constructing virtual documents for ontology matching. In *Proceedings of WWW*, pages 23–31, 2006.
23. P. Shvaiko and J. Euzenat. A survey of schema-based matching approaches. *Journal on Data Semantics*, IV:146–171, 2005.
24. U. Straccia and R. Troncy. oMAP: Combining classifiers for aligning automatically OWL ontologies. In *Proceedings of WISE*, pages 133–147, 2005.
25. K.-C. Tai. The tree-to-tree correction problem. *Journal of the ACM*, 26(3):422–433, 1979.
26. J. Tang, J. Li, B. Liang, X. Huang, Y. Li, and K. Wang. Using Bayesian decision for ontology mapping. *Journal of Web Semantics*, 4(1):243–262, 2006.
27. G. Valiente. *Algorithms on Trees and Graphs*. Springer, 2002.
28. J. T.-L. Wang, B. Shapiro, D. Shasha, K. Zhang, and K. Currey. An algorithm for finding the largest approximately common substructures of two trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):889–895, 1998.

Analyzing Mapping Extraction Approaches

C. Meilicke, H. Stuckenschmidt

Computer Science Institute
University of Mannheim
A5, 6 68159 Mannheim, Germany
{christian, heiner} @informatik.uni-mannheim.de

Abstract. While lots of research in ontology matching is related to the issue of computing and refining similarity measures, only little attention has been paid to question how to extract the final alignment from a matrix of similarity values. In this paper we present a theoretical framework for describing extraction methods and argue that the quality of the final matching result is highly affected by the extraction method. Therefore, we discuss several extraction methods and apply them to some of the results submitted to the OAEI 2006. The results of our experimental study show that the proposed strategies differ with respect to precision and recall. In particular, theoretical considerations as well as emprirical results indicate that methods that additionally make use of information encoded in the ontologies result in better extractions compared to state of the art approaches.

1 Motivation

Automated matching systems have to be applied to detect semantic relations between ontologies representing overlapping domains. The majority of matching systems approach this problem by computing similarities between the entities (concepts, roles, ect.) to be matched. While similarities can be obtained by comparing the matchable entities one by one, structure and semantic based techniques use more sophisticated ways to compute, propagate and refine similarities by taking the context of these entities into account. As a final step of the matching process an alignment has to be extracted. Since the similarities computed in the first phases of the matching process suggest a diversity of matching hypotheses, the final result will be highly affected by the extraction process. Obviously, it makes sense to use the structural and semantic information in the process of similarity computation. We argue that the same holds for the extraction process. Therefore, we propose a theoretical framework to distinguish between several extraction methods and present as well as evaluate some extraction algorithms that make use of the additional information encoded in the ontologies to be matched.

1.1 Problem Statement

In accordance to Euzenat and Shvaiko [1] the problem of ontology matching can be defined as follows. For each ontology $\mathcal{T}$ there is a function $Q(\mathcal{T})$ that defines matchable elements of $\mathcal{T}$. Given ontologies $\mathcal{T}_1$ and $\mathcal{T}_2$, the task of matching is to determine correspondences between $Q(\mathcal{T}_1)$ and $Q(\mathcal{T}_2)$. Correspondences can be defined as 4-tuples

$\langle e, e', r, v\rangle$ where $e \in Q(\mathcal{T}_1)$ and $e' \in Q'(\mathcal{T}_2)$, r is a semantic relation, and $v \in [0, 1]$ is a confidence value. In this paper, we only consider the simple case where $Q(\mathcal{T})$ are the concepts of $\mathcal{T}$ and r is the equivalence relation. Given correspondence c we use the functions $source(c)$ and $target(c)$ to refer to the aligned concepts, $rel(c)$ to denote the semantic relation, and $conf(c)$ to refer to the confidence value.

The extraction problem we address is a subproblem of the matching problem. A process that solves this problem takes a mapping $\mathcal{M}$ (a set of correspondences) and two ontologies $\mathcal{T}_1$ and $\mathcal{T}_2$ as input and returns a mapping $\mathcal{M}' \subseteq \mathcal{M}$ as result. In many matching systems the final mapping is extracted from a similarity matrix S. Notice that for each similarity matrix S there exists a mapping $\mathcal{M}$, such that each correspondence in $\mathcal{M}$ corresponds to a cell in S. The methods we propose to solve this problem will thus be relevant for the large class of matching systems that first compute similarity matrices or comprehensive intermediary mappings and thereafter extract alignments as final outcome. An optimal solution to an extraction problem is a mapping $\mathcal{M}' \subseteq \mathcal{M}$ such that each $c \in \mathcal{M}'$ truly reflects the semantic relation $rel(c)$ between $source(c)$ and $target(c)$, and there exists no $\bar{c} \in \mathcal{M} \setminus \mathcal{M}'$ such that $\bar{c}$ truly reflects the semantic relation $rel(\bar{c})$ between $source(\bar{c})$ and $target(\bar{c})$.

1.2 Related Work

The extraction problem has only been of minor interest in research related to ontology matching. Euzenat and Shvaiko [1] spend five pages on the topic in their comprehensive book 'Ontology Matching'. They mainly discuss thresholds and a greedy startegy as well as several methods that optimize the results of a one-to-one extraction. Some of these approaches are taken up and extended in the following sections. In opposite to our approach Euzenat and Shvaiko model the extraction problem to be independent of the ontologies to be matched. To our suprise, the papers submitted to the OAEI 2006 contest describing the matching systems of the participants contain only minor information on extraction. We believe that the importance of the extraction problem is highly underestimated in research and that extraction methods have strong effects on characteristics and quality of the resulting mappings.

In prior work [7] we applied the notion of mapping consistency to eliminate potentially erroneous correspondences from automatically generated mappings. Not being aware of the applicability to the more general problem of mapping extraction we referred to this process as repairing of mappings. This approach has been extended and modified towards an extraction technique in [6] where it has been evaluated on synthetic data sets. The main contribution of these approaches compared to state-of-the-art extraction methods is based on the fact that knowledge endcoded in the ontologies cannot only be used for computing similarities but also plays a crucial role in the final extraction process.

1.3 Outline and Contribution

In section 2 we introduce the notion of an extraction function and describe several distinctive properties of extraction functions. In particular, we distinguish between ontology dependent functions that make use of the additional knowledge encoded in the ontologies and ontology independent functions. In section 3 we take up this distinction and present several algorithmic implementations of both ontology independent (section 3.1) and ontology dependent extraction functions (section 3.2). We implemented these algorithms and conducted several experiments on some of the results submitted to the OAEI 2006. The experimental results (section 4) indicate that certain extraction methods, in particular ontology dependent methods, yield better solutions compared to naive approaches. Nevertheless, our main contribution is to emphasise the importance of the extraction problem and to point out that different extraction techniques have strong effects on the characteristics and the quality of the final matching result.

2 Preliminaries

We understand the extraction process as the application of a sequence of extraction functions to an input mapping and the ontologies to be matched. Therefore, we first have to define the notion of an extraction function as well as some properties of extraction functions. For all of the following definitions let $\mathbb{T}$ denote the set of ontologies and let $\mathbb{M}$ denote the set of possible mappings between two ontologies $\mathcal{T}_1, \mathcal{T}_2 \in \mathbb{T}$.

Definition 1 (Extraction function). *A function* $f : \mathbb{T} \times \mathbb{T} \times \mathbb{M} \rightarrow \mathbb{M}$ *is an extraction function iff for all* $\mathcal{T}_1, \mathcal{T}_2 \in \mathbb{T}$ *and for all* $\mathcal{M} \in \mathbb{M}$ *we have* $f(\mathcal{T}_1, \mathcal{T}_2, \mathcal{M}) \subseteq \mathcal{M}$.

Some extraction functions solely rely on the input mapping $\mathcal{M}$, while some functions make use of the information encoded in $\mathcal{T}_1$ and $\mathcal{T}_2$. We take this distinction into account by refering to functions of the first type as ontology independent, while refering to functions of the second type as ontology dependent.

Definition 2 (Independence). *An extraction function* f *is ontology independent iff for all* $\mathcal{T}_1, \mathcal{T}_2 \in \mathbb{T}$ *and for all mappings* $\mathcal{M} \in \mathbb{M}$ *between* $\mathcal{T}_1$ *and* $\mathcal{T}_2$ *we have* $f(\mathcal{T}_1, \mathcal{T}_2, \mathcal{M}) = f(\emptyset, \emptyset, \mathcal{M})$ *where* $\emptyset$ *denotes an ontology with no axioms. Otherwise* f *is ontology dependent.*

In this work we focus on extraction functions that yield one-to-one mappings. A one-to-one mapping respectively a one-to-one extraction function can straight forward be defined as follows.

Definition 3 (One-to-one mapping and extraction function). *A mapping* $\mathcal{M}$ *is a one-to-one mapping iff for all correspondences* $c \in \mathcal{M}$ *there exists no* $c' \neq c \in \mathcal{M}$ *such that* $source(c) = source(c') \vee target(c) = target(c')$. *An extraction function* f *is a one-to-one extraction function iff the codomain of* f *is constrained to be the set of one-to-one mappings.*

As mentioned above, the extraction process can be understood as applying a sequence of extraction functions, in other words, applying the composition of several extraction functions. Since domain and codomain of an extraction function are not the same, we have to redefine composition with respect to extraction functions in the following self-evident way.

Definition 4 (Composition of extraction functions). *Given extraction functions f and g. The composition $f \circ g : \mathbb{T} \times \mathbb{T} \times \mathbb{M} \rightarrow \mathbb{M}$ is defined as $(f \circ g)(\mathcal{T}_1, \mathcal{T}_2, \mathcal{M}) = f(\mathcal{T}_1, \mathcal{T}_2, g(\mathcal{T}_1, \mathcal{T}_2, \mathcal{M}))$.*

Some matching systems are extracting the final mapping by applying a threshold on the similarity matrix. This way to extract is obviously not a one-to-one extraction. Nevertheless, it is an important component in the extraction process. Therefore, we define the application of a threshold as threshold extraction function.

Definition 5 (Threshold extraction function). *An extraction function f_t with $t \in [0, 1]$ is a threshold extraction function iff for all $\mathcal{T}_1, \mathcal{T}_2 \in \mathbb{T}$ and for all mappings $\mathcal{M} \in \mathbb{M}$ between $\mathcal{T}_1$ and $\mathcal{T}_2$ we have $f_t(\mathcal{T}_1, \mathcal{T}_2, \mathcal{M}) = \{c \in \mathcal{M} | conf(c) > t\}$.*

In most cases a one-to-one mapping will be obtained by the composition of a threshold function f_t and a one-to-one extraction function f. The order of applying these functions can have effects on the results depending on f. Therefore, we have to introduce the following property.

Definition 6 (Threshold commutative). *An extraction function f is threshold commutative iff $f \circ f_t = f_t \circ f$ for all $t \in [0, 1]$.*

Obviously, threshold extraction functions are threshold commutative. This follows from the fact that $f_{t_1} \circ f_{t_2} = f_{max\{t_1, t_2\}} = f_{t_2} \circ f_{t_1}$.

The most interesting class of extraction functions are, from a theoretical perspective, optimization functions. These functions are defined by an objective function and extract a final mapping that is optimal with respect to the chosen objective function.

Definition 7 (Optimization function). *An extraction function f_o is an optimization function iff for all $M^* \subseteq M$ we have $o(M^*) \leq o(f_o(\mathcal{T}_1, \mathcal{T}_2, M))$ with $o : \mathbb{M} \rightarrow \mathbb{R}$ being an objective function.*

We will see that the considerations of this section are not only of theoretical interest but constitute a useful framework to describe and distinguish between different extraction techniques.

3 One-to-one extraction algorithms

In this section we describe several algorithms that are implementations of different types of extraction functions. Some of these algorithms have to iterate over the elements of a mapping according to the ordering of confidence values. Therefore, we deal with mappings as sequences of correspondences in the following.

3.1 Ontology independent extraction algorithms

First, we focus on the algorithmic implementation of ontology independent extraction functions. These functions do not make any use of the information encoded in the ontologies to be aligned. At the end of this subsection we present a small example that compares the behavior of the introduced algorithms.

Algorithm 1

```
NAIVEDESCENDING($\mathcal{M}$)
$\mathcal{M}' \leftarrow \emptyset$
SORTDESCENDING($\mathcal{M}$)
while $\mathcal{M} \neq \emptyset$ do
    $c \leftarrow$ REMOVEFIRSTELEMENT($\mathcal{M}$)
    $\mathcal{M}' \leftarrow \mathcal{M}' \cup \{c\}$
    for all $c' \in$ GETALTERNATIVES($\mathcal{M}, c$) do
        REMOVEELEMENT($\mathcal{M}, c'$)
    end for
end while
return $\mathcal{M}'$
```

Naive descending extraction Algorithm 1 can be described as naive greedy strategy that transforms a many-to-many mapping into a one-to-one mapping by iterating over the elements of $\mathcal{M}$ in descending order. The algorithm consists of a sequence of local decisions. First, the correspondences in $\mathcal{M}$ are sorted descending due to their confidence value. Then the algorithm iterates over $\mathcal{M}$ removing step by step elements from $\mathcal{M}$. In each iteration correspondence c with the hightest confidence value is removed and added to the extraction result $\mathcal{M}'$. The reduced $\mathcal{M}$ is checked for alternative correspondences, where GETALTERNATIVES($\mathcal{M}, c$) is defined to return the set of all correspondences that have the same source or target concept in common with c. All alternatives are removed from $\mathcal{M}$ and will thus also not be contained in the extracted mapping $\mathcal{M}'$. Notice that this naive extraction function is threshold commutative and has been implemented by several participants of the OAEI 2006, for example by Falcon-AO [2] and RiMOM [4].

Naive ascending extraction Algorithm 2 is similar to algorithm 1 but more restrictive. While algorithm 1 accepts correspondences and removes their alternatives in descending order, algorithm 1 dismisses correspondences due to the existence of alternatives with a higher confidence value. The major difference is based on the principle that correspondences are dismissed if better alternatives exist even though these alternatives are also dismissed in one of the following iterations. Therefore, we have NAIVEDESCENDING($\mathcal{M}$) $\supseteq$ NAIVEASCENDING($\mathcal{M}$) for all $\mathcal{M} \in \mathbb{M}$.

Hungarian extraction Algorithms 1 and 2 are implementations of extraction functions that are based on sequences of local decisions. Contrary to these approaches, an optimization extraction function extracts a solution that is optimal from a global point of view, selecting the one-to-one subset $\mathcal{M}'$ of $\mathcal{M}$ that is optimal with respect to some objective function o. Choosing objective function $o(\mathcal{M}') = \sum_{c \in \mathcal{M}'} conf(c)$, the problem to find $\mathcal{M}'$ can be solved using the hungarian method. The hungarian method is a combinatorial optimization algorithm which solves assignment problems in polynomial time [3]. To use the algorithm in the context of mapping extraction a few modifications have to be applied. Due to the lack of space we cannot give a detailed description. Such

Algorithm 2

```
NaiveAscending($\mathcal{M}$)
$\mathcal{M}' \leftarrow \emptyset$
SortAscending($\mathcal{M}$)
for all $c \in \mathcal{M}$ do
    if GetAlternatives($\mathcal{M}, c$) $= \emptyset$ then
        $\mathcal{M}' \leftarrow \mathcal{M}' \cup \{c\}$
    end if
end for
return $\mathcal{M}'$
```

a description can be found in [6] in section 3.1. We refer to this algorithm as HUNGARIANEXTRACTION(). Notice that the function implemented in HUNGARIANEXTRACTION() is our first example for a function that is not threshold commutative. A proof for this claim can be given by counter example. Example 1 provides a counter example and illustrates the differences between the three proposed extraction functions.

Example 1. Given a mapping $\mathcal{M} = \{c_1, c_2, c_3, c_4\}$ between $\mathcal{T}_1$ and $\mathcal{T}_2$ based on a similarity matrix S. The following table describes S respectively $\mathcal{M}$ in detail. Applying the extraction functions presented above we obtain different results.

	$2:X$	$2:Y$
$1:A$	$c_1 = \langle 1\!: A, 2\!: X, =, 0.9\rangle$	$c_2 = \langle 1\!: A, 2\!: Y, =, 0.8\rangle$
$1:B$	$c_3 = \langle 1\!: B, 2\!: X, =, 0.7\rangle$	$c_4 = \langle 1\!: B, 2\!: Y, =, 0.5\rangle$

- NAIVEDESCENDING($\mathcal{M}$) $= \{c_1, c_4\}$: The algorithms first adds c_1 to the resulting mapping $\mathcal{M}'$. By making this choice c_2 and c_3 are removed from $\mathcal{M}$ since they are alternatives with a lower confidence value. Finally, c_4 is added to $\mathcal{M}'$.
- NAIVEASCENDING($\mathcal{M}$) $= \{c_1\}$: The algorithm starts with c_4. Since there are more probable alternatives available c_4 is discarded. The same holds for c_3 and c_2. Finally, only c_1 is left and accepted to be part of $\mathcal{M}'$.
- HUNGARIANEXTRACTION($\mathcal{M}$) $= \{c_2, c_3\}$: The hungarian method finds the best one-to-one mapping $\mathcal{M}'$ with respect to the objective function $o(\mathcal{M}) = \sum_{c \in \mathcal{M}'} c$. Notice that correspondence c_1 is not an element of $\mathcal{M}'$ contrary to the results of the greedy approaches.

What if we apply threshold function $f_{0.75}$ to this problem? Now we have HUNGARIANEXTRACTION($f_{0.75}(\mathcal{M})$) $= \{c_1\}$ and $f_{0.75}$(HUNGARIANEXTRACTION($\mathcal{M}$))$= \{c_2\}$. Thus, we can conclude that our implementation of an optimizing extraction function is not threshold commutative.

3.2 Ontology dependent extraction algorithms

We now introduce two straight forward extensions for each of the algorithms presented above. The resulting algorithms are implementations of ontology dependent extraction functions. First, we discuss how to use the additional information encoded in $\mathcal{T}_1$ and $\mathcal{T}_2$ by introducing the notion of a merged ontology (definition 8). Merging $\mathcal{T}_1$ and $\mathcal{T}_2$ results in the union of $\mathcal{T}_1$ and $\mathcal{T}_2$ using $\mathcal{M}$ as nexus.

Definition 8 (Merged ontology). *Given a mapping* $\mathcal{M}$ *between ontologies* $\mathcal{T}_1$ *and* $\mathcal{T}_2$*. The merged ontology* $\mathcal{T}_1 \cup_{\mathcal{M}} \mathcal{T}_2$ *of* $\mathcal{T}_1$ *and* $\mathcal{T}_2$ *connected via* $\mathcal{M}$ *is defined as* $\mathcal{T}_1 \cup_{\mathcal{M}} \mathcal{T}_2 = \mathcal{T}_1 \cup \mathcal{T}_2 \cup \{t(c) \mid c \in \mathcal{M}\}$ *with t being defined by* $t(\langle 1\colon C, 2\colon D, =, c\rangle) = 1\colon C \equiv 2\colon D$ *converting equivalence correspondences into equivalence axioms of* $\mathcal{T}_1 \cup_{\mathcal{M}} \mathcal{T}_2$*.*

Adding the correspondences of $\mathcal{M}$ as equivalence statements results in an ontology that is structured by subsumption relations connecting concepts of both ontologies. A concept of $\mathcal{T}_1$ is thus positioned in the taxonomy of $\mathcal{T}_2$ and vice versa. This can result in subsumptions between concepts of $\mathcal{T}_1$ or $\mathcal{T}_2$ that cannot be derived from $\mathcal{T}_1$ respectively $\mathcal{T}_2$ without taking $\mathcal{M}$ into account. We define the according property as instability of a mapping (also defined in [5] in the context of DDL).

Definition 9 (Stability of a mapping). *Given a mapping* $\mathcal{M}$ *between ontologies* $\mathcal{T}_1$ *and* $\mathcal{T}_2$*.* $\mathcal{M}$ *is stable iff there exists no pair of concepts* $\langle i\colon C, i\colon D\rangle$ *with* $i \in \{1,2\}$ *such that* $\mathcal{T}_i \not\models i\colon C \sqsubseteq i\colon D$ *and* $\mathcal{T}_1 \cup_{\mathcal{M}} \mathcal{T}_2 \models i\colon C \sqsubseteq i\colon D$*. Otherwise* $\mathcal{M}$ *is instable.*

The additional subsumption statements introduced by a mapping are major topic of distributed description logics (compare for example [9]). In this context, additional subsumption statements are equated with additional knowledge which has a positive connotation. Nevertheless, we think that - as far as we are concerned with automatically generated correspondences - additional subsumption statements introduced by $\mathcal{M}$ indicate that some of the correspondences in $\mathcal{M}$ are erroneous. Furthermore, we introduce the stronger property of mapping consistency (based on the corresponding definitions in [10], [5], and [7]).

Definition 10 (Consistency of a mapping). *Given a mapping* $\mathcal{M}$ *between ontologies* $\mathcal{T}_1$ *and* $\mathcal{T}_2$*.* $\mathcal{M}$ *is consistent iff there exists no concept* $i\colon C$ *with* $i \in \{1,2\}$ *such that* $\mathcal{T}_i \not\models i\colon C \sqsubseteq \bot$ *and* $\mathcal{T}_1 \cup_{\mathcal{M}} \mathcal{T}_2 \models i\colon C \sqsubseteq \bot$*. Otherwise* $\mathcal{M}$ *is inconsistent.*

Obviously, some of the correspondences of an inconsistent mapping $\mathcal{M}$ have to be incorrect, because we would not accept a mapping that imposes restrictions on $\mathcal{T}_1 \cup_{\mathcal{M}} \mathcal{T}_2$ making some of the concepts in $\mathcal{T}_1 \cup_{\mathcal{M}} \mathcal{T}_2$ unsatisfiable. By postulating that the extracted mapping $\mathcal{M}'$ has to be stable respectively consistent, we impose additional constraints on extraction functions. Notice that both properties are subject to criticism.

- Mapping stability: Extracting stable mappings makes only sense if all (or most of all) subsumption statements have been specified (directly or indirectly via entailment) in both $\mathcal{T}_1$ and $\mathcal{T}_2$. If one of the ontologies has been poorly structured by the ontology engineer the additional subsumption statements imposed by $\mathcal{M}$ will indeed introduce new acceptable knowledge.
- Mapping inconsistency: Inconstencies will only occur if $\mathcal{T}_1$ respectively $\mathcal{T}_2$ contain disjointness statements. Evaluation of ontologies show that these statements are often missing [13]. Thus, even a completely incorrect mapping will often be consistent.

Even though there are extraction problems where stability is too restrictive while consistency is not restrictive at all, we believe that applying these properties will in most cases have positive effects on the extraction results. Therefore, we introduce the notion of a minimal conflict set. A minimal conflict set is an instable respectively inconsistent subset of $\mathcal{M}$ which contains no real subset that is also instable respectively inconsistent.

Definition 11 (Minimal conflict sets). *Given a mapping $\mathcal{M}$ between ontologies $\mathcal{T}_1$ and $\mathcal{T}_2$. A subset $C \subseteq \mathcal{M}$ is a minimal conflict set with respect to stability (consistency), if C is instable (inconsistent) and each $C' \subset C$ is stable (consistent).*

From a more general point of view we can abstract from the conflicts introduced in this paper and define the class of conflict based extraction functions as subclass of ontology dependent extraction functions.

Definition 12 (Conflict based extraction function). *An extraction function f_g is conflict based iff for all $C \in g(\mathcal{M})$ there exists a correspondence $c \in C \wedge c \notin f_g(\mathcal{T}_1, \mathcal{T}_2, M)$ where $g : \mathbb{M} \rightarrow 2^{\mathbb{M}}$ is a conflict function defined by $g(\mathcal{M}) = \{C \subseteq \mathcal{M} | C$ is a minimal conflict set$\}$.*

Using a conflict based approach origins from its application in the context of diagnosis, as introduced by Reiter [8]. It has already been applied by the authors to the problem of automatically repairing mappings in [7] and partially to the problem of mapping extraction in [6]. Notice that related approaches can be found in [12] where Wang and Xu propose similar mapping properties. These properties are used supporting humans in semi-automatic repairing of mappings.

As decribed in [6] stability and consistency of $\mathcal{M}$ can be efficiently checked for each dual-element subset of $\mathcal{M}$ using a straight forward approach.[1] Therefore, we restrict ourselves to pairwise consistency and stability and have to define a conflict pair as a dual-element conflict set.

Definition 13 (Conflict pair). *Given a mapping $\mathcal{M}$ between ontologies $\mathcal{T}_1$ and $\mathcal{T}_2$. C is a stability (consistency) conflict pair, if $|C| = 2$ and C is a minimal conflict set with respect to stability (consistency). Correspondence $c_1 \in \mathcal{M}$ conflicts with correspondence $c_2 \in \mathcal{M}$ iff $\{c_1, c_2\}$ is a conflict pair.*

The extraction algorithms we propose as approximate conflict based extensions of algorithms 1 and 2 remove at least one correspondence from each conflict pair which results in an extracted mapping that will be pairwise stable respectively consistent. Both algorithms can be extended in a natural way by replacing GETALTERNATIVES($\mathcal{M}, c$) by calling a method that returns all alternatives for c as well as all correspondences that conflict with c. Obviously, the results of these extended algorithm are pairwise consistent respectively stable. For the experimental study we have implemented both variants. Notice that pairwise stability (consistency) is only an approximation of stability (consistency). Nevertheless, due to our experience most conflict sets turned out to be minimal conflict pairs.

The same approach cannot be applied to extend the hungarian method. Instead of that it is possible to search for an optimal solution that is free of pairwise conflicts. Due to the lack of space we have to omit a description of the algorithm which has already been described in detail in [6]. There we claimed that conflict based extraction not only increases precision but in some cases also increases recall. We have been critizised for this point of view, because it seems that solving a conflict is established

---

[1] In [6] this has only been described for consistency, but this approach can be modified in a self-evident way for stability.

by dismissing correspondences. That is only half the truth. Solving conflicts forces the hungarian method to rearranage parts of or even the whole assignment. Consider again example 1. Suppose now c_2 conflicts with c_3. Obviously, c_2 and c_3 cannot be both elements of the final extraction. But since alternatives are available, the final result of the extended hungarian extraction will be $\{c_1, c_4\}$ instead of $\{c_2, c_3\}$. Thus, it is possible that the conflict-based extraction increases both precision and recall. The same kind of argument can be applied to NAIVEDESCENDING() even if a rearrangement will only affect parts of the results due to the greediness of the algorithm. Contrary to this, increasing recall is not possible by extending algorithm 2. For this algorithm we have NAIVEASCENDING($\mathcal{M}$) $\supseteq$ NAIVEASCENDING$_{Consistency}$($\mathcal{M}$) $\supseteq$ NAIVEASCENDING$_{Stability}$($\mathcal{M}$).

4 Experiments

In the following we present some emprirical results by applying the one-to-one extraction algorithms presented above on real world matching problems. Before going into details it makes sense to bring together the theoretical considerations of the last sections and resume them as hypotheses.

- H_1: One-to-one extraction functions exceed threshold extraction functions with respect to the harmonic mean of precision and recall (f-measure).
- H_2: Ontology dependent extraction functions (implemented as conflict-based algorithms) will result in better extractions compared to alternative approaches.
- H_3: Optimization extraction functions will result in better extractions compared to alternative approaches.

4.1 Experimental Settings

We evaluated the presented algorithms using automatically created mappings between ontologies of the ontoFarm data set. The ontoFarm data set consists of a set of ontologies in the domain of conference organization that has been created by the Knowledge Engineering Group at the University of Economics Prague [11] and has been subject of the conference track at the Ontology Alignment Evaluation Initiative 2006.

Amongst the participants that submitted results for all pairs of ontologies only two matching systems generated many-to-many mappings that can be used as input to our one-to-one extraction procedures. We refer to these systems as system A and B. We were also able to refactor another matching system (system C) to generate many-to-many mappings while not modifying any other parameters that have been used to generate the submitted results. Thus, we could apply and evaluate our algorithms as final step of three matching systems. To evaluate the extraction results we had to manually construct reference mappings consisting of equivalence correspondences for all pairs of ontologies. Since this task is extremly time-consuming we decided to choose a subset of eight ontologies of the ontoFarm Dataset, thus creating 28 reference mappings. In our experiments we evaluated the extraction algorithms presented above for all matching systems and for all matching problems. Thus we had to evaluate $9 \times 3 \times 28 = 756$ different extraction results. Therefore, we cannot present the results in detail but have to focus on aggregated values.

4.2 Results

To examine hypothesis H_1 we computed precision and recall aggregated over all matching problems for both the many-to-many input mappings and the extraction results based on applying the simplest one-to-one extraction algorithm NAIVEDESCENDING(). The resulting differences are listed in table 1. The extraction method NAIVEDESCENDING() is already implemented as extraction method of system C. Thus, we compare against the intermediary many-to-many mapping of this system. System A has been applied with a setting that results in an extraction close to a one-to-one mapping but also adds correspondences that have nearly (specified by a certain range) the same confidence as the best match. Therefore, for system A the effects of a one-to-one extraction are only marginal. System B does not extract one-to-one mappings. The resulting mapping can be optimized to a significant degree. While we gain about 26% precision, we loose only 4% recall. Nevertheless, even for system A we get slightly better results. We conclude that hypothesis H_1 has been verified by our experiments.

Matcher	Δ Precision	Δ Recall	Δ F-Measure
System A	+ 2.2%	0.0%	+ 1.3%
System B	+ 25.8%	- 3.9%	+ 13.3%
System C	+ 27.4%	- 3.0%	+ 16.6%

Table 1. Many-to many mappings vs. one-to-one mappings.

Hypothesis H_2 is concerned with the issue wether the information encoded in the ontologies can be used to optimize the extraction result. We proposed two ways to extend extraction functions to be ontology dependent, in particular conflict based. Thus, we compare for each extraction function the ontology independent algorithm with the variants generating consistent respectively stable extraction results. Figure 1 presents the mean values aggregating over matching systems and matching problems. Each column describes in how far precision, recall, and f-measure has been increased or decreased by applying one of the extraction methods compared to the many-to-many input mapping. For all three extraction methods and their extensions we can observe a similar pattern. The ontology independent variant is less precise than the ontology dependent variant using consistency which is again less precise than the variant using stability. A similar pattern with a negative influence of ontology dependency can be observed for recall. But notice that the negative effects are smaller compared to the positive effect on precision. This can also be derived from the third group of columns describing the f-measure. Only for algorithm NAIVEDESCENDING() we have slightly worse results by extending it to an ontology dependent approach. We have already argued above that extending this algorithm will not result in an rearrangement of the extracted assignment. We can conclude that hypotheses H_2 has been verified by our experiments.

Finally, we have to consider hypotheses H_3. We have claimed that an extraction function that finds an optimal extraction with respect to an appropriate objective function yields better results than a greedy approach. This hypothesis cannot be verified

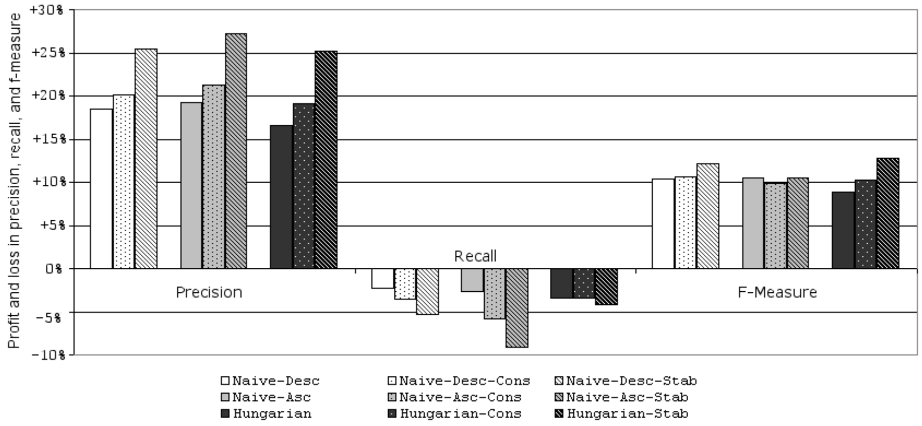


Fig. 1. Comparison of three basic extraction methods and their ontology dependent extensions.

by the experimental results. If we compare the f-measures of NaiveDescending() and HungarianExtraction() we observe that the greedy approach surprisingly extracts slightly better solutions in average. There is some evidence that choosing an additive objective function has not been the best choice. Understanding confidence values as probabilities, it seems to be more natural to use an objective function like $o(\mathcal{M}') = \prod_{c \in \mathcal{M}'} conf(c)$ which possibly will result in better extractions. But notice also that using additional conflict information has the strongest positive effect on the optimization approach. We have argued above that an optimization approach is more flexible with respect to rearrangements of assignments. Therefore, taking additional conflicts into account in the context of an optimization approach will strongly influcence the quality of the extraction results, while algorithm NaiveAscending() can use information on conflicts only to a very limited extent.

5 Summary and Outlook

We introduced a framework for describing and distinguishing between methods to extract a one-to-one mapping from a similarity matrix. In particular, we introduced the notion of an ontology dependent extraction function as well as the notion of an optimization extraction function. From a theoretical perspective we have argued that these types of extraction functions should result in better solutions to the extraction problem than naive approaches. We stated several algorithms as implementations of naive and more sophisticated extraction functions. To extend these algorithms towards ontology dependency, we introduced the concepts of mapping consistency and stability based on prior work. All in all, we stated nine different extraction methods. Thus, we had for each important type of extraction function at least one implementation.

In order to proove the hypothesis derived from theoretical considerations, we performed several experiments. In these experiments we could verify that ontology dependent functions yield better results than their ontology independent counterparts. We could not proove that our implementation of an optimization extraction function performs better than the naive approaches and argued that the choice of objective function is a crucial point. We also observed that extending optimization functions results in a significant improvement compared to the ontology independent counterparts. In future work the combination of optimization and conflict based extractions has to be examined. In particular, understanding confidences as probabilities seems to be an promising approach. Besides arguing that certain methods result in better solutions compared to naive approaches, the main contribution of this paper is to highlight the importance of the extraction problem as a self-contained subproblem of ontology matching.

Acknowledgement The work has been partially supported by the German Science Foundation (DFG) in the Emmy Noether Programme under contract STU 266/3-1.

References

1. Jerome Euzenat and Pavel Shvaiko. *Ontology Matching*. Springer Verlag, 2007.
2. Wei Hu, Gong Cheng, Dongdong Zheng, Xinyu Zhong, and Yuzhong Qu. The results of falcon-ao in the oaei 2006 campaign. In *Proceedings of the ISWC 2006 Workshop on Ontology Matching*, Athens, GA, USA, November 2006.
3. H.W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics*, pages 83–97, 1955.
4. Yi Li, Juanzi Li, Duo Zhang, and Jie Tang. Result of ontology alignment with rimom at oaei06. In *Proceedings of the ISWC 2006 Workshop on Ontology Matching*, Athens, GA, USA, November 2006.
5. Christian Meilicke. Reasoning about ontology mappings in distributed description logics. Bachelor Thesis, University of Mannheim, August 2006.
6. Christian Meilicke and Heiner Stuckenschmidt. Applying logical constraints to ontology matching. In *Proccedings of the 30th Annual German Conference on Artificial Intelligence (KI-07)*, Osnabrück, Germany, 2007.
7. Christian Meilicke, Heiner Stuckenschmidt, and Andrei Tamilin. Repairing ontology mappings. In *Proceedings of the Twenty-Second Conference on Artificial Intelligence (AAAI-07)*, Vancouver, Canada, 2007.
8. Ray Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32:57–95, 1987.
9. L. Serafini and A. Tamilin. DRAGO: Distributed reasoning architecture for the semantic web. In *Proceedings of the Second European Semantic Web Conference (ESWC'05)*, 2005.
10. Heiner Stuckenschmidt, Holger Wache, and Luciano Serafini. Reasoning about ontology mappings. In *Proceedings of the ECAI-06 Workshop on Contextual Representation and Reasoning*, August 2006.
11. Ondrej Svab, Svatek Vojtech, Petr Berka, Dusan Rak, and Petr Tomasek. Ontofarm: Towards an experimental collection of parallel ontologies. In *Poster Proceedings of the International Semantic Web Conference 2005*, 2005.
12. P. Wang and B. Xu. Debugging ontology mapping. In *International Workshop on Semantic e-Science (SeS2006) of the 1st Asian Semantic Web Conference (ASWC2006)*, 2006.
13. T. D. Wang. Gauging ontologies and schemas by numbers. In *Proceedings of the Workshop EON Evaluation of Ontologies for the Web*, 2006.

Towards Semantics-Based Ontology Similarity

Rudi Araújo and H. Sofia Pinto

INESC-ID
Rua Alves Redol 9, Apartado 13069, 1000-029 Lisboa, Portugal
`{rnna,sofia}@algos.inesc-id.pt`

Abstract. As the Semantic Web emerges the problem of semantic heterogeneity is becoming more acute. Ontology matching techniques aim at tackling this problem by establishing correspondences between elements of the ontologies. These techniques rely on distance metrics, often called (dis)similarity measures, to assess the similarity of elements within the ontologies. Most of these approaches are either terminological, structural and/or extensional. However, recently some proposals for semantics-based measures have been put forward. We reason that these latter should receive more attention, since semantics are one of the key advantages of ontologies. Therefore, we present a set of semantic ontology similarity measures.

1 Introduction and Motivation

Although the Semantic Web will be responsible for the proliferation of ontologies, it is unrealistic to expect that every Semantic Web agent shares the same set of ontologies. Therefore, the need for establishing a consensus among many cooperating agents arises. This is a key motivation for ontology matching.

Ontology matching techniques take as input a set of ontologies, and output a set of correspondences between elements of the ontologies. Distance metrics, often called (dis)similarity measures, are used to estimate the similarity of elements of the ontologies. There are four kinds of similarity measures: lexical, structural, extensional and semantics-based. The difference between the latter and the others lies on the fact that they are sensitive to the logical nature of the knowledge representation formalism in which the ontologies are formalized. Hence, they are enabled to resort to deduction services, such as consistency checking. Given that, on the one hand, modern representation languages have well-defined semantics, and on the other hand, one of the key advantages of ontologies lies on their semantics, we believe that this type of measures should deserve more attention from the ontology matching community.

This paper elaborates on our own work, presented in [2], where we define a purely semantics based similarity measure. Here, we propose a more refined approach and extend the proposed measure, which was only sensitive to concept constructs of $\mathcal{ALC}$ ($\mathcal{A}$ttributive $\mathcal{L}$anguage with $\mathcal{C}$omplement) – intersection $\sqcap$, conjunction $\sqcup$ and complement $\neg$, – to deal with role constructs – universal $\forall$ and existential quantifiers $\exists$. However, our measure does not yet cover the whole scope of $\mathcal{ALC}$. Though limited in expressivity, $\mathcal{ALC}$ is the foundation of more expressive Description Logics (DLs) and is already sufficiently expressive to formalize many practical ontologies.

Throughout the rest of the paper, we assume the standard DL notation [3], except for concept equivalence, $C \doteq D$, to distinguish from TBox equivalence, $\mathcal{T}_1 \equiv \mathcal{T}_2$ (meaning $\mathcal{T}_1 \models \mathcal{T}_2$ and $\mathcal{T}_2 \models \mathcal{T}_1$). We denote $\mathcal{C}^*$ as the closure of the set of concept names $\mathcal{C}$ under the DL constructs $\sqcap$, $\sqcup$ and $\neg$. We further assume that every set of concept names $\mathcal{C}$ and roles $\mathcal{R}$ are finite.

The paper is organized as follows: in section 2 we summarize related work. We introduce the theory underlying the proposed measures in section 3, followed by both the concept and role similarity measures, and a toy example. In section 4 we present the results for real-world examples. We discuss the measures based on our results in section 5 and finish with conclusions and future directions in section 6.

2 Related Work

The use of similarity measures is not limited to ontology matching. Other uses include unsupervised or semi-automatic ontology clustering [11], and automated ontology merging, as performed by PROMPT [10] and CHIMAERA [9].

Approaches to similarity measures in DLs can be found in [7, 5, 4, 8]. The measure presented in [7] is based on extracting and comparing *concept signatures*: the elements within the ontology that are related to one concept. The work presented in [5] is an extensional dissimilarity measure that estimates the difference between two concepts in different ontologies through the set of individuals they share. Borgida *et al.* [4] propose the adaptation of known similarity measures to DLs: feature-based models, the semantic-network approach and information-content models. Finally, the work presented in [8] defines a similarity measure for each DL construct and computes the similarity between two concepts by agreggating these values.

3 Theory

We start by presenting the general framework underlying the similarity measures. Though different from [2], the definitions presented here can be shown as equivalent.

3.1 Foundation

All measures presented here are based on the same idea: counting *characteristic concepts*. By characteristic concepts, we mean any concept, formed using standard concept constructs on a set of concept names, which cannot be any more specific. For example, given the concept names Man and Woman, the concepts $\neg\mathsf{Man} \sqcap \neg\mathsf{Woman}$ and $\neg\mathsf{Man} \sqcap \mathsf{Woman}$ are two of four possible characteristic concepts. Characteristic concepts can be seen as propositional models. The formal definition follows.

Definition 1 (Characteristic Concept). *Let $\mathcal{C}$ be a finite set of DL concept names. A concept $C \in \mathcal{C}^*$ is characteristic iff it is an intersection of n literals, with $n = |\mathcal{C}|$, and each concept name in $\mathcal{C}$ occurs exactly once in it. The set of all possible characteristic concepts w.r.t. $\mathcal{C}$ is $\zeta(\mathcal{C})$.*

Given a TBox $\mathcal{T}$ containing a set of concept names $\mathcal{C}$, we need to determine the *characteristic acceptance set*: the sub-set of $\zeta(\mathcal{C})$ with only *consistent* characteristic concepts w.r.t. $\mathcal{T}$.

Definition 2 (Characteristic Acceptance Set). *Let $\mathcal{T}$ be a TBox containing a finite set of DL concept names $\mathcal{C}$. Given a set of characteristic concepts $S \subseteq \zeta(\mathcal{C})$, S is* accepted *by $\mathcal{T}$ iff every $C \in S$ is consistent in $\mathcal{T}$ (i.e., there exists a model $\mathcal{I}$ of $\mathcal{T}$ such that $C^{\mathcal{I}} \neq \emptyset$). The* characteristic acceptance set*, $Z(\mathcal{T}) \subseteq \zeta(\mathcal{C})$, is the maximal set accepted by $\mathcal{T}$.*

3.2 The Measures

In this section we introduce the measures for comparing concepts and roles. The former determines the similarity of two TBoxes disregarding any information concerning role restrictions.

Comparing Concepts Given two TBoxes, $\mathcal{T}_1$ and $\mathcal{T}_2$, we need to determine to which extent their acceptance sets overlap. In particular, if the ontologies are equivalent, then their acceptance sets are the same [1] and, therefore, their similarity assessment should be the highest. Although this assumption is rather intuitive for the maximum value, the same does not apply to the minimum value. The question of when two ontologies are totally dissimilar does not have a straight answer. One could consider that total dissimilarity corresponds to the case where the acceptance sets are disjoint. However, that depends on the interpretation given to similarity, since when the acceptance sets are disjoint, it means there is no *agreement* on a set of *consistent* characteristic concepts, while there may be agreement on a large number of *inconsistent* characteristic concepts.

Let us consider the following TBoxes that represent several typical cases of overlap between acceptance sets:

$$\begin{aligned}&\mathcal{E}_1 = \{\mathsf{Man} \sqsubseteq \mathsf{Person} \sqcap \mathsf{Male}\}, \quad \mathcal{E}_2 = \{\mathsf{Man} \doteq \mathsf{Person} \sqcap \mathsf{Male}\},\\ &\mathcal{E}_3 = \{\mathsf{Man} \sqsubseteq \mathsf{Person}, \mathsf{Man} \doteq \neg\mathsf{Person} \sqcup \neg\mathsf{Male}\}\ .\end{aligned}$$

Their acceptance sets are shown in table 1 (with $Z_i = Z(\mathcal{E}_i)$). Each line of the table is a concept name, and each column is a characteristic concept, and a symbol at the intersection of a characteristic concept with a concept name indicates that the concept name occurs as a positive (+) or negative (−) literal.

Table 1. The characteristic acceptance sets for $\mathcal{E}_1$, $\mathcal{E}_2$ and $\mathcal{E}_3$.

<table>
<tr><td></td><td colspan="4">Z_2</td><td colspan="2">Z_3</td><td></td><td></td></tr>
<tr><td>Person</td><td>−</td><td>+</td><td>−</td><td>+</td><td>+</td><td>+</td><td>−</td><td>−</td></tr>
<tr><td>Man</td><td>−</td><td>−</td><td>−</td><td>+</td><td>−</td><td>+</td><td>+</td><td>+</td></tr>
<tr><td>Male</td><td>+</td><td>−</td><td>−</td><td>+</td><td>+</td><td>−</td><td>+</td><td>−</td></tr>
<tr><td></td><td colspan="5">Z_1</td><td></td><td></td><td></td></tr>
</table>

The *coverage* measure (γ) determines to which extent one of the ontologies covers the other, by measuring the percentage of overlap between the acceptance sets in proportion to the size of the acceptance set of one of the ontologies:

$$\gamma(\mathcal{T}_1, \mathcal{T}_2) = \frac{|Z(\mathcal{T}_1) \cap Z(\mathcal{T}_2)|}{|Z(\mathcal{T}_1)|}\ .$$

Applying coverage to the examples above yields $\gamma(\mathcal{E}_2, \mathcal{E}_1) = 100\%$, because $Z(\mathcal{E}_2)$ is a sub-set of $Z(\mathcal{E}_1)$. This measure is not symmetric, therefore inverting the order yields a different result, $\gamma(\mathcal{E}_1, \mathcal{E}_2) = 80\%$, because only 80% of $Z(\mathcal{E}_1)$ is contained within $Z(\mathcal{E}_2)$, since they only disagree on the consistency of one characteristic concept, $\neg\mathsf{Man} \sqcap \mathsf{Person} \sqcap \mathsf{Male}$ (which is consistent in $\mathcal{E}_1$ but not in $\mathcal{E}_2$). Given that $\mathcal{E}_2$ and $\mathcal{E}_3$ are disjoint, $\gamma(\mathcal{E}_2, \mathcal{E}_3) = \gamma(\mathcal{E}_3, \mathcal{E}_2) = 0\%$.

The *consistency agreement* measure (α_+) determines to which extent the ontologies agree on the set of consistent characteristic concepts. It measures the proportion of characteristic concepts consistent in both ontologies w.r.t. the set of characteristic concepts consistent in at least one of them:

$$\alpha_+(\mathcal{T}_1, \mathcal{T}_2) = \frac{|Z(\mathcal{T}_1) \cap Z(\mathcal{T}_2)|}{|Z(\mathcal{T}_1) \cup Z(\mathcal{T}_2)|} \ .$$

Contrary to γ, this measure is symmetric, thus $\alpha_+(\mathcal{E}_1, \mathcal{E}_2) = \alpha_+(\mathcal{E}_2, \mathcal{E}_1) = 80\%$. Again, since $\mathcal{E}_2$ and $\mathcal{E}_3$ are disjoint, their consistency agreement is 0%. We note that while coverage measures the overlap between $\mathcal{E}_1$ and $\mathcal{E}_3$ as 20% and 50%, the consistency agreement is lesser: $\alpha_+(\mathcal{E}_1, \mathcal{E}_3) = 16.67\%$. Therefore, consistency agreement measures the global overlap, while coverage only measures the overlap of one in relation to the other.

The previous measure only takes into account the set of consistent characteristic concepts. To measure the *inconsistency agreement*, the proportion of inconsistent characteristic concepts in both TBoxes w.r.t. the inconsistent characteristic concepts in at least one of them, we define the following measure:

$$\alpha_-(\mathcal{T}_1, \mathcal{T}_2) = \frac{|\zeta(\mathcal{C}) - (Z(\mathcal{T}_1) \cup Z(\mathcal{T}_2))|}{|\zeta(\mathcal{C}) - (Z(\mathcal{T}_1) \cap Z(\mathcal{T}_2))|} \ .$$

Even though all previous measures yield 0% for $\mathcal{E}_2$ and $\mathcal{E}_3$, this measure yields 25%, due to the fact that even though their acceptance sets are disjoint they agree that 2 out of 8 characteristic concepts are inconsistent.

The *agreement measure* (α) combines both consistency and inconsistency agreement, by measuring the proportion of characteristic concepts whose consistency is the same in both ontologies w.r.t. all possible characteristic concepts:

$$\alpha(\mathcal{T}_1, \mathcal{T}_2) = \frac{|\zeta(\mathcal{C}) - (Z(\mathcal{T}_1) \cup Z(\mathcal{T}_2))| + |Z(\mathcal{T}_1) \cap Z(\mathcal{T}_2)|}{2^{|\mathcal{C}|}} \ .$$

Agreement yields 87.50% for $\mathcal{E}_1$ and $\mathcal{E}_2$, since they agree on the consistency of 7 out of 8 characteristic concepts.

In [6] a set of similarity measures for the automatic evaluation of learnt ontologies against a gold-standard is proposed. In this context, if we apply the coverage measure, with, say, $\mathcal{T}_1$ as the gold-standard, then $\gamma(\mathcal{T}_1, \mathcal{T}_2)$ can be interpreted as *recall*, since it is the proportion of correct learnt characteristic concepts w.r.t. the total amount of correct characteristic concepts, while $\gamma(\mathcal{T}_2, \mathcal{T}_1)$ can be regarded as *precision*, since it measures the proportion of correct learnt characteristic concepts w.r.t. the set of all learnt characteristic concepts. We can thus define the F_1 measure:

$$F(\mathcal{T}_1, \mathcal{T}_2) = \frac{2 \times \gamma(\mathcal{T}_1, \mathcal{T}_2) \times \gamma(\mathcal{T}_2, \mathcal{T}_1)}{\gamma(\mathcal{T}_1, \mathcal{T}_2) + \gamma(\mathcal{T}_2, \mathcal{T}_1)} = \frac{2 \times |Z(\mathcal{T}_1) \cap Z(\mathcal{T}_2)|}{|Z(\mathcal{T}_1)| + |Z(\mathcal{T}_2)|} \ .$$

Comparing Roles Until this point we have only compared ontologies with concepts built using intersection ($\sqcap$), conjunction ($\sqcup$) and negation ($\neg$). However, this is very limited in expressiveness, given that standard DLs (at least as expressive as $\mathcal{ALC}$) allow building concepts with role constructs, namely universal quantification $\forall R.C$ and existential quantification $\exists R.C$. These constructs allow us to restrict the domain and range of a role:

$$\begin{aligned} \exists R.\top \sqsubseteq C & \qquad \text{(domain restriction)} \\ \top \sqsubseteq \forall R.C & \qquad \text{(range restriction)} \end{aligned} \tag{1}$$

The first axiom restricts instances of R to have as source an individual in C, while the second restricts instances of R to have as destination an individual contained in C. The power of these constructs is not limited, however, to general domain and range restriction. They can also be used to restrict the domain and range for a given concept D:

$$\begin{aligned} \exists R.D \sqsubseteq C & \qquad \text{(specific domain restriction)} \\ D \sqsubseteq \forall R.C & \qquad \text{(specific range restriction)} \end{aligned} \tag{2}$$

This means that the domain and range of R for individuals of D is C.

Given that $\mathcal{ALC}$ does not provide any other construct related to roles, it is reasonable to argue that the domain and range specifications characterize the semantics of a role in this language. Therefore, the task of computing role similarity can be reduced to comparing these domains and ranges. Since they are concepts, we can compare them using the measures introduced above.

Let us start by considering the case in (1) where the domain and range of a role consist of a single concept and they are not related to one another (i.e., it is not necessary to know the domain of a role to know its range). In this case, given a role R occuring in a TBox $\mathcal{T}$, we can define the concept δ_R and ρ_R as the domain and range of R respectively, i.e. such that $\mathcal{T} \models \exists R.\top \sqsubseteq \delta_R$ and $\mathcal{T} \models \top \sqsubseteq \forall R.\rho_R$. Given two TBoxes $\mathcal{T}_1$ and $\mathcal{T}_2$ in which the role R occurs, we can extract the domains of R, δ_R^1 and δ_R^2, and the ranges, ρ_R^1 and ρ_R^2. To compute the similarity of R, we can compare δ_R^1 with δ_R^2 and ρ_R^1 with ρ_R^2, which can be done by using the set of measures for concept similarity introduced above, but instead of using $Z(\mathcal{T}_1)$ and $Z(\mathcal{T}_2)$, we use only the sub-set of these that are subsumed by δ_R^1 and δ_R^2, to compare the domains, and by ρ_R^1 and ρ_R^2, to compare the ranges.

The procedure introduced above can be generalized to deal with specific domain and range restrictions (2). The problem is that the range of a role depends on the domain (and vice-versa), so there is no single δ_R and ρ_R concepts. The idea is to construct a different TBox describing the domain and range restrictions of a given role. Given a TBox $\mathcal{T}$ and a role R, our approach is to construct a TBox $\mathcal{T}^R$ from $\mathcal{T}$, such that, for each concept name A occuring in $\mathcal{T}$, we have two concept names in $\mathcal{T}^R$: dA and rA. dA should be seen as the set of *role instances* which have an individual in A as source, and rA the ones that have an individual in A as destination. The domain and range concepts are related through subsumption according to the restrictions in $\mathcal{T}$. This technique in many ways resembles the reification technique, in the sense that individuals of $\mathcal{T}^R$ are not instances of concepts of $\mathcal{T}$, but instances of the role R. A formal definition of $\mathcal{T}^R$ follows after the following auxiliary definition.

First we define the domain/range description, which is a TBox describing the relations between the domain and range of a role. For example, the axiom $\mathsf{A} \sqsubseteq \forall R.\mathsf{B}$

is described in the domain/range description as $\mathsf{dA} \sqsubseteq \mathsf{rB}$, and should be interpreted as "the set of role instances of R that have origin in A is a sub-set of the set of role instances of R that have destination in B."[1]

Definition 3 (Domain/Range Description). *Let R be a role occuring in a TBox $\mathcal{T}$. The* domain/range description *of R w.r.t. $\mathcal{T}$, $\mathrm{dr}_{\mathcal{T}}(R)$, is defined as:*

$$\mathrm{dr}_{\mathcal{T}}(R) = \{\delta(C) \sqsubseteq \rho(D) | \mathcal{T} \models C \sqsubseteq \forall R.D\} \ ,$$

where $\delta(C)$ (resp. $\rho(D)$) is the same as C (resp. D) with every concept name A occuring in C (resp. D) replaced by dA (resp. rA).

Next we define the domain/range TBox of $\mathcal{T}$, which is essentially the union of three sets of axioms: (1) the domain/range description of a role R w.r.t. $\mathcal{T}$, (2) a set of axioms that maintains the semantics of the concepts when acting as domain of R and (3) another set of axioms when acting as range of R. This TBox should be seen as describing the role in terms of the relations between its domain and range.

Definition 4 (Domain/Range TBox). *Let $\mathcal{T}$ be a TBox and R a role. The* domain/range TBox *of $\mathcal{T}$ w.r.t. R, written $\mathcal{T}^R$, is defined as:*

$$\mathcal{T}^R = \delta(\mathcal{T}) \cup \rho(\mathcal{T}) \cup \mathrm{dr}_{\mathcal{T}}(R) \ .$$

In definition 4, the terms $\delta(\mathcal{T})$ and $\rho(\mathcal{T})$ ensure that the concepts' semantics are maintained when they act as domain and range of the role R. Finally, we define the role similarity measure, which is simply the employment of any of the previously defined measures to the domain/range TBoxes.

Definition 5 (Role Similarity). *Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be ontologies, R a role and s a similarity measure, $s \in \{\gamma, \alpha_+, \alpha_-, \alpha, F\}$. The* role similarity measure *w.r.t. R under s, $\varrho_s^R : \mathcal{O} \times \mathcal{O} \rightarrow [0, 1]$, is defined as:*

$$\varrho_s^R(\mathcal{T}_1, \mathcal{T}_2) = s(\mathcal{T}_1^R, \mathcal{T}_2^R)$$

It is arguable that this solution is somewhat cumbersome, for three reasons:

1. the meaning of the prefixed concepts dA and rA is not clear;
2. it is necessary to define a domain/range TBox for each role, where the only variable is the domain/range description;
3. the computation of $\mathrm{dr}_{\mathcal{T}}(R)$ requires a full prover.

A more elegant solution would be to effectively reify the relations. This could be achieved by extending the formalism with a rule mechanism (e.g., $\mathcal{AL}$-log). In this case, we could simply add a rule base to the original TBoxes to relate the domain and range of any role. For example, the following rule could belong to such a rule base:

$$\mathrm{range}(\mathsf{marriedTo}, \mathsf{Woman}) \leftarrow \mathrm{domain}(\mathsf{marriedTo}, \mathsf{Man}) \ .$$

However, this solution does not fit as well with the previously defined measures. Moreover, it would be necessary to deal with a hybrid DL and rule-based formalism. In effect, we would be shifting from one problem to another.

Let us illustrate all measures with a more complex example. Consider the following ontologies, $\mathcal{T}_1$ and $\mathcal{T}_2$. Table 2 shows $Z(\mathcal{T}_1)$ and $Z(\mathcal{T}_2)$ (written Z_1 and Z_2).

---

[1] Note that $\exists R.D \sqsubseteq C$ is equivalent to $\neg C \sqsubseteq \forall R.\neg D$, so this definition captures domain as well as range restrictions.

$\mathcal{T}_1$	$\mathcal{T}_2$
$\neg\mathsf{Male} \sqsubseteq \mathsf{Female}$	$\mathsf{Female} \doteq \neg\mathsf{Male}$
$\mathsf{Man} \doteq \mathsf{Person} \sqcap \mathsf{Male}$	$\mathsf{Man} \doteq \mathsf{Person} \sqcap \mathsf{Male}$
$\mathsf{Woman} \doteq \mathsf{Person} \sqcap \mathsf{Female}$	$\mathsf{Woman} \doteq \mathsf{Person} \sqcap \neg\mathsf{Man}$
$\mathsf{MaleCat} \doteq \mathsf{Cat} \sqcap \mathsf{Male}$	$\mathsf{MaleCat} \sqsubseteq \mathsf{Cat} \sqcap \mathsf{Male}$
$\top \sqsubseteq \forall\mathsf{marriedTo}.\mathsf{Person}$	$\top \sqsubseteq \forall\mathsf{marriedTo}.\mathsf{Person}$
$\neg\mathsf{Person} \sqsubseteq \forall\mathsf{marriedTo}.\bot$	$\neg\mathsf{Person} \sqsubseteq \forall\mathsf{marriedTo}.\bot$
	$\mathsf{Man} \sqsubseteq \forall\mathsf{marriedTo}.\mathsf{Woman}$
	$\mathsf{Woman} \sqsubseteq \forall\mathsf{marriedTo}.\mathsf{Man}$

Table 2. The characteristic acceptance sets for $\mathcal{T}_1$ and $\mathcal{T}_2$.

	$Z_1 - Z_2$				$Z_1 \cap Z_2$								$Z_2 - Z_1$	
Person	+	−	−	+	+	+	+	+	−	−	−	−	+	−
Man	+	−	−	+	+	+	−	−	−	−	−	−	+	−
Male	+	+	+	+	+	+	−	−	−	−	+	+	+	+
Woman	+	−	−	+	−	−	+	+	−	−	−	−	−	−
Female	+	+	+	+	−	−	+	+	+	+	−	−	−	−
Cat	+	+	−	−	+	−	−	+	−	+	−	+	+	+
MaleCat	+	+	−	−	+	−	−	−	−	−	−	+	−	−

Z_1

Z_2

Table 3 shows the results of applying the similarity measures to the example.[2] We observe that one of the main differences between $\mathcal{T}_1$ and $\mathcal{T}_2$ is that in the former every individual is required to be either Male or Female or both, but in the latter they have to be either one or the other exclusively. If this were the only difference, then $Z(\mathcal{T}_2)$ would be totally contained within $Z(\mathcal{T}_1)$, but the difference in the definition of Cats and MaleCats accounts for the two characteristic concepts in $Z(\mathcal{T}_2)$ and not in $Z(\mathcal{T}_1)$. The fact that more of $Z(\mathcal{T}_2)$ is contained within $Z(\mathcal{T}_1)$ than vice-versa is responsible for the lower coverage result between $\mathcal{T}_1$ and $\mathcal{T}_2$ than the inverse. Indeed, we can observe that more of what can be modeled in $\mathcal{T}_2$ can also be modeled in $\mathcal{T}_1$ than vice-versa, so these results are rather intuitive. The consistency agreement result comes from the number of agreed consistent characteristic concepts, 8, in proportion to the number of characteristic concepts in the table, 14. This result is also intuitive, since the ontologies have a reasonable overlap, i.e. most concepts are similar in both ontologies. On the other hand, the inconsistency agreement is the number of agreed inconsistent characteristic concepts, i.e., the ones that do not appear in table 2, $128 - 14 = 114$, in proportion to the number of inconsistent characteristic concepts in at least one of the ontologies, $128 - 8 = 120$. This shows that, although there is a considerable difference on what can be modeled in the ontologies, there is a substantially higher similarity concerning what can *not* be modeled in none of the ontologies. The result obtained for the agreement measure is a consequence of the fact that $8+114 = 122$ characteristic concepts, out of 128, have the same consistency in both TBoxes. It is slightly higher than the inconsistency agreement, since it also takes into account the consistency agreement.

The results for role similarity show us that there is a low coverage between $\mathcal{T}_1$ and $\mathcal{T}_2$, since $\mathcal{T}_1$ is much more permissible regarding the marriage relationship. In

---

[2] For the sake of readability, we refer to marriedTo as simply m.

Table 3. Results of applying the similarity measures to $\mathcal{T}_1$ and $\mathcal{T}_2$.

$\gamma(\mathcal{T}_1,\mathcal{T}_2)$	$\gamma(\mathcal{T}_2,\mathcal{T}_1)$	$\alpha_+(\mathcal{T}_1,\mathcal{T}_2)$	$\alpha_-(\mathcal{T}_1,\mathcal{T}_2)$	$\alpha(\mathcal{T}_1,\mathcal{T}_2)$	$F(\mathcal{T}_1,\mathcal{T}_2)$
66.67%	80.00%	57.14%	95.00%	95.31%	72.73%

$\varrho^{\mathsf{m}}_{\gamma}(\mathcal{T}_1,\mathcal{T}_2)$	$\varrho^{\mathsf{m}}_{\gamma}(\mathcal{T}_2,\mathcal{T}_1)$	$\varrho^{\mathsf{m}}_{\alpha_+}(\mathcal{T}_1,\mathcal{T}_2)$	$\varrho^{\mathsf{m}}_{\alpha_-}(\mathcal{T}_1,\mathcal{T}_2)$	$\varrho^{\mathsf{m}}_{\alpha}(\mathcal{T}_1,\mathcal{T}_2)$	$\varrho^{\mathsf{m}}_{F}(\mathcal{T}_1,\mathcal{T}_2)$
22.22%	66.67%	20.00%	99.80%	99.80%	33.33%

this TBox, both men and women can marry any person of any gender. In $\mathcal{T}_2$, men can only marry women and vice-versa. On the other hand, many of the possible marriage relations in $\mathcal{T}_2$ are covered in $\mathcal{T}_1$, which accounts for the higher coverage between $\mathcal{T}_2$ and $\mathcal{T}_1$. What keeps it from being 100% is the fact that women and men are exclusively female and male in $\mathcal{T}_2$, respectively, while in $\mathcal{T}_1$, men and women can have both genders, so the characteristic concepts subsumed by $\mathsf{dMan} \sqcap \mathsf{dWoman}$ or $\mathsf{rMan} \sqcap \mathsf{rWoman}$ are inconsistent in $\mathcal{T}_2^{\mathsf{marriedTo}}$ but consistent in $\mathcal{T}_1^{\mathsf{marriedTo}}$ (the domain/range TBoxes w.r.t. $\mathsf{marriedTo}$).

4 Experimental Results

Evaluating similarity measures is a difficult task since it requires a standard definition of similarity, which is highly subjective. Therefore, it is hard to use common metrics such as precision and recall. In the following, we present and discuss two experiments conducted applying the similarity measures to two datasets: BibTeX and Directory.

BibTeX The first experiment used the BibTeX dataset extracted from OAEI.[3] We used a sub-set of three of those ontologies: the reference, $\mathcal{B}_{\mathsf{ref}}$, and the ones from Karlsruhe, $\mathcal{B}_{\mathsf{k}}$, and INRIA, $\mathcal{B}_{\mathsf{i}}$. Each ontology presents a different perspective on what bibliographic references are. All ontologies are formalized in OWL. Since the measures we propose are only applicable to ontologies with the same set of concepts, we need to align them. The ontologies are bundled with a set of alignments, between the reference ontology and the other two, which we use to merge them. These alignments contain not only equalities but also inclusion alignment relations. We discarded the latter since we focus on what is equal on both ontologies. Figure 1 shows a relevant section of the ontologies. Most concept names have the same (or similar) label in all ontologies, except $\mathsf{Reference}$, which is $\mathsf{Publication}$ and Entry in $\mathcal{B}_{\mathsf{k}}$ and $\mathcal{B}_{\mathsf{i}}$ respectively, and School, labeled as $\mathsf{University}$ in $\mathcal{B}_{\mathsf{k}}$.

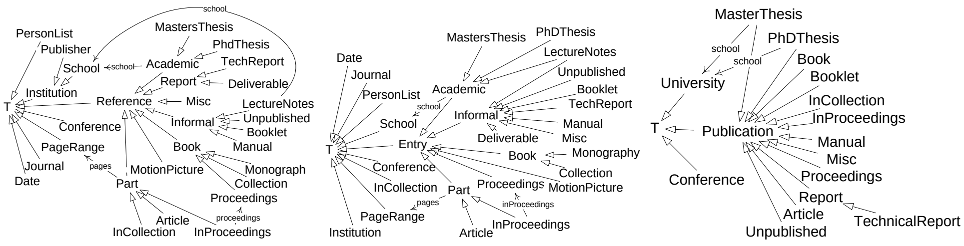
In this experiment we compared the reference ontology to the other two. For that, we focus only on the concept names that are present in both ontologies (the reference one and the other), by trimming out the ones which are not involved in the alignment. Table 4 shows the results of applying the similarity measures to the ontologies.[4]

Coverage results show that $\mathcal{B}_{\mathsf{ref}} \models \mathcal{B}_{\mathsf{k}}$. In fact, we can observe that, if we focus on the concept names that appear in both ontologies, every axiom in $\mathcal{B}_{\mathsf{k}}$ is either

---

[3] `http://oaei.ontologymatching.org/2006/`.

[4] We refer to the school role as s.



Reference INRIA Karlsruhe

Fig. 1. BibTeX ontologies.

Table 4. Results of applying the measures to the BibTeX ontologies.

$\gamma(\mathcal{B}_\mathsf{ref}, \mathcal{B}_\mathsf{k})$	$\gamma(\mathcal{B}_\mathsf{k}, \mathcal{B}_\mathsf{ref})$	$\alpha_+(\mathcal{B}_\mathsf{ref}, \mathcal{B}_\mathsf{k})$	$\alpha_-(\mathcal{B}_\mathsf{ref}, \mathcal{B}_\mathsf{k})$	$\alpha(\mathcal{B}_\mathsf{ref}, \mathcal{B}_\mathsf{k})$	$F(\mathcal{B}_\mathsf{ref}, \mathcal{B}_\mathsf{k})$
100%	75.00%	75.00%	86.96%	90.63%	85.71%
$\varrho^\mathsf{s}_\gamma(\mathcal{B}_\mathsf{ref}, \mathcal{B}_\mathsf{k})$	$\varrho^\mathsf{s}_\gamma(\mathcal{B}_\mathsf{k}, \mathcal{B}_\mathsf{ref})$	$\varrho^\mathsf{s}_{\alpha_+}(\mathcal{B}_\mathsf{ref}, \mathcal{B}_\mathsf{k})$	$\varrho^\mathsf{s}_{\alpha_-}(\mathcal{B}_\mathsf{ref}, \mathcal{B}_\mathsf{k})$	$\varrho^\mathsf{s}_\alpha(\mathcal{B}_\mathsf{ref}, \mathcal{B}_\mathsf{k})$	$\varrho^\mathsf{s}_F(\mathcal{B}_\mathsf{ref}, \mathcal{B}_\mathsf{k})$
100%	44.99%	44.99%	94.96%	95.16%	62.06%
$\gamma(\mathcal{B}_\mathsf{ref}, \mathcal{B}_\mathsf{i})$	$\gamma(\mathcal{B}_\mathsf{i}, \mathcal{B}_\mathsf{ref})$	$\alpha_+(\mathcal{B}_\mathsf{ref}, \mathcal{B}_\mathsf{i})$	$\alpha_-(\mathcal{B}_\mathsf{ref}, \mathcal{B}_\mathsf{i})$	$\alpha(\mathcal{B}_\mathsf{ref}, \mathcal{B}_\mathsf{i})$	$F(\mathcal{B}_\mathsf{ref}, \mathcal{B}_\mathsf{i})$
85.44%	50.62%	46.60%	96.69%	96.79%	63.58%
$\varrho^\mathsf{s}_\gamma(\mathcal{B}_\mathsf{ref}, \mathcal{B}_\mathsf{i})$	$\varrho^\mathsf{s}_\gamma(\mathcal{B}_\mathsf{i}, \mathcal{B}_\mathsf{ref})$	$\varrho^\mathsf{s}_{\alpha_+}(\mathcal{B}_\mathsf{ref}, \mathcal{B}_\mathsf{i})$	$\varrho^\mathsf{s}_{\alpha_-}(\mathcal{B}_\mathsf{ref}, \mathcal{B}_\mathsf{i})$	$\varrho^\mathsf{s}_\alpha(\mathcal{B}_\mathsf{ref}, \mathcal{B}_\mathsf{i})$	$\varrho^\mathsf{s}_F(\mathcal{B}_\mathsf{ref}, \mathcal{B}_\mathsf{i})$
72.51%	20.50%	19.02%	99.88%	99.88%	31.97%

also present in $\mathcal{B}_\mathsf{ref}$ or is a logical consequence of it. The 75% result in the coverage between $\mathcal{B}_\mathsf{k}$ and $\mathcal{B}_\mathsf{ref}$ can be explained by the fact that Proceedings is a subclass of Book in the latter, but not in the former. Therefore, the set of characteristic concepts consistent in $\mathcal{B}_\mathsf{k}$ but not in $\mathcal{B}_\mathsf{ref}$ are the ones that contain $\mathsf{Proceedings} \sqcap \neg \mathsf{Book}$. These are also the only characteristic concepts which are not agreed by the ontologies, which explains the very high consistency agreement.

Regarding the similarity of the role school, although it would seem that it has the same domain and range in $\mathcal{B}_\mathsf{ref}$ and $\mathcal{B}_\mathsf{k}$, and, therefore, it should be assessed as totally similar, it is not exactly that case. In $\mathcal{B}_\mathsf{ref}$, the domain and range of school are modeled as general domain and range restrictions:

$$\exists \mathsf{school}.\top \sqsubseteq \mathsf{Academic} \sqcup \mathsf{LectureNotes}\ ,$$
$$\top \sqsubseteq \forall \mathsf{school}.\mathsf{School}\ .$$

This means that there cannot be any instance of the school role that does not start in Academic or LectureNotes or end in School. In $\mathcal{B}_\mathsf{k}$, we have:

$$\mathsf{MasterThesis} \sqcup \mathsf{PhDThesis} \sqsubseteq \forall \mathsf{school}.\mathsf{University}\ .$$

This only restricts the range of school for instances in the domain of MasterThesis or PhDThesis. Since MastersThesis and PhDThesis are subclasses of Academic, the former two axioms entail the latter one, but the contrary does not hold. The fact that LectureNotes is in the domain of the role does not affect the similarity assessment since this concept is not present in $\mathcal{B}_\mathsf{k}$, and is, therefore, trimmed out. Although the INRIA ontology also uses the general domain and range restriction, we can observe that LectureNotes has different semantics.

We note that the reference ontology is assessed as more similar to the INRIA one than to the Karslruhe one regarding their inconsistency. This results from the fact that the more concept names an ontology has, the stricter it is likely to be, since every concept name is usually involved in some axiom, which restricts its characteristic acceptance set. The inconsistency agreement and the agreement measures suffer from this problem. However, for a set of ontologies sharing exactly the same concepts, which is not the case with this dataset, these measures would be helpful in determining which ones were more similar.

Directory We also took the Directory ontologies from the OAEI dataset. These ontologies were extracted from the Google, Yahoo and Looksmart web directories. The source ontology comprises over 2,000 concept names while the target ontology more than 6,000. Both ontologies contain only inclusion statements (taxonomies) and are biased towards the task of easily finding a set of web sites. Since there was no reference alignment available, we performed one using a semi-automatic algorithm.

Due to space constraints, we only present the basic idea of the algorithm. It is composed of (1) an automatic phase and (2) a manual one. In (1) we use a derivation of the terminological overlap measure using a notion similar to the semantic cotopy [6]. For each concept name in each ontology, its cotopy is extracted from the ontology and augmented with its WordNet cotopy. The algorithm chooses the pairs of concept names that maximize the terminological overlap of their cotopies. Phase (2) is composed of two steps: filtering out wrong mappings and adding correct ones. The final result is a set of 580 concept mappings. Since there is no reference mapping available, we estimate the precision of the automatic phase of the algorithm w.r.t. our notion of correct or incorrect mappings: 51.56%.

We compared the ontologies w.r.t. the set of concept names involved in the extracted mapping. Because of the previously identified problem concerning the relation between the number of concept names and the inconsistency agreement, this measure, along with the agreement one, assessed the ontologies as almost 100% similar. In contrast, the ontologies were given a very low similarity assessment, close to 0%, w.r.t. coverage and consistency agreement. A closer inspection revealed that out of the 585 axioms present in each of the ontologies, 150 of them differed, revealing that at least such amount of concept names assume a different position in the taxonomy. If these differences are in higher-level concept names, that can have a deep impact on the similarity assessment. This leads us to conclude that either (1) the mapping is inaccurate or (2) the ontologies are indeed very different. Since the manual filter was introduced to improve the precision of the mapping, the first hypothesis is not very plausible.

5 Discussion

The results for the BibTeX ontologies show that the proposed measures bear relevant information concerning the similarity of the ontologies. Moreover, the different measures are to some extent independent, which is an important feature of similarity measures identified in [6]. From the results we can conclude that the measures are more trustworthy when applied to ontologies with the same set of concept names (as opposed to the BibTeX dataset, where $\mathcal{B}_{\mathsf{ref}}$ shares less concept names with $\mathcal{B}_{\mathsf{k}}$ than with $\mathcal{B}_{\mathsf{i}}$). This comes from the fact that the more concept names the ontologies share, the more likely there are differences in them.

One of the problems we clearly identify is the growth of the inconsistency agreement in proportion to the number of concept names. Therefore, this measure, along with the agreement one, is only useful for ontologies with few concept names or when comparing a set of (three or more) ontologies with exactly the same set of concept names. In the BibTeX results presented, the set of concept names differed among the three ontologies, so the results of comparing $\mathcal{B}_{\mathsf{ref}}$ to $\mathcal{B}_{\mathsf{i}}$ are not directly comparable to the similarity results between $\mathcal{B}_{\mathsf{ref}}$ and $\mathcal{B}_{\mathsf{k}}$, because $\mathcal{B}_{\mathsf{i}}$ and $\mathcal{B}_{\mathsf{k}}$ don't share the same concept names. If all ontologies shared the same set of concept names, and even if the results of the inconsistency agreement are very high, they are in the same scale, and can thus be compared with each other.

Given that our measures require a previous mapping, instead of comparing characteristic concepts, an alternative approach would be to compare their set of consequences. Though countable, we should stress that this set is infinite. Nonetheless, the acceptance set can be seen as a finite set of consequences.

Contrary to related work, the concept similarity measures focus on the ontologies as wholes, which may hinder their usefulness. Local similarity measures are often used to support the matching process of individual concepts. Therefore, the concept similarity measures presented here are not directly useful in this task. However, they can be used to determine the *global a posteriori* quality of a given mapping.

We should also note that the concept similarity is not affected by roles. These measures assess the following TBoxes as equal:

$$\mathcal{T}_1 = \{\mathsf{A} \sqsubseteq \forall R.A\},\ \mathcal{T}_2 = \{\mathsf{A} \sqsubseteq \forall R.\neg A\}\ ,$$

although the role similarity measures do not. This is naturally undesirable and should be revised in the future. Since the role similarity measures focus on each role individually, they can be used to perform role matching given a previous mapping of concepts.

The question of which measure is the most useful is only sensible in view of the purpose it is used for. If we are comparing a learnt ontology against a gold-standard, as discussed earlier, the coverage operator corresponds to precision and recall, and can therefore be used as such. If we are only concerned with what can be represented with the ontologies, then the agreement operator is a good indication of similarity. Naturally, the different operators can be combined within a unique operator that suits a particular purpose, but this is outside the scope of this work.

Regarding efficiency, although $|\zeta(\mathcal{C})|$ is exponential to $|\mathcal{C}|$, it can be shown that #SAT solvers, used to compute the size of acceptance sets, perform well in practice [2]. Finally, although it is not shown here, the measures satisfy the set of criteria identified in [6].

6 Conclusions and Future Work

In this paper we proposed a set of similarity measures to assess the similarity of ontologies based on the notion of characteristic concept. We applied them in two experiments using real ontologies yielding intuitive and very promising results.

Extending the measures to more expressive DLs is one of the directions for future work. Further developments should tackle the problem regarding the weak relation between the concept and role similarities. Other future developments include integration of the measures in a system, for example to perform (semi-)automatic evaluation of learnt ontologies against a gold-standard. It should also be interesting to research into the usage of characteristic concepts for the assessment of the similarity of concept names instead of ontologies. This could then be used to create a novel matching algorithm.

References

1. R. Araújo and H. S. Pinto. SEMilarity theory. Technical report, INESC-ID, 2007. `http://algos.inesc-id.pt/~rnna/semilarity-theory.pdf`.
2. R. Araújo and H. S. Pinto. SEMilarity: Towards a model-driven approach to similarity. In *Proc. of DL2007*, volume 250 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2007.
3. F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.
4. A. Borgida, T. Walsh, and H. Hirsh. Towards measuring similarity in description logics. In *Proc. of DL2005*, volume 147 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2005.
5. C. d'Amato, N. Fanizzi, and F. Esposito. A dissimilarity measure for ALC concept descriptions. In *Proc. of ACM SAC 2006*, pages 1695–1699. ACM, 2006.
6. K. Dellschaft and S. Staab. On how to perform a gold standard based evaluation of ontology learning. In *Proc. of ISWC2006*, volume 4273 of *LNCS*, pages 228–241. Springer, 2006.
7. B. Hu, Y. Kalfoglou, H. Alani, D. Dupplaw, P. H. Lewis, and N. Shadbolt. Semantic metrics. In *Proc. of EKAW2006*, volume 4248 of *LNCS*, pages 166–181. Springer, 2006.
8. K. Janowicz. Sim-DL: Towards a semantic similarity measurement theory for the description logic ALCNR in geographic information retrieval. In *On the Move to Meaningful Internet Systems. Proc., Part II*, volume 4278 of *LNCS*, pages 1681–1692. Springer, 2006.
9. D. McGuinness, R. Fikes, J. Rice, and S. Wilder. An environment for merging and testing large ontologies. In *Proc. of KR2000*, pages 483–493. Morgan Kaufmann, 2000.
10. N. F. Noy and M. A. Musen. Prompt: Algorithm and tool for automated ontology merging and alignment. In *Proc. of AAAI/IAAI 2000*, pages 450–455. AAAI Press / The MIT Press, 2000.
11. V. A. M. Tamma, P. R. S. Visser, D. Malerba, and D. M. Jones. Computer assisted ontology clustering for knowledge sharing. In *Proc. of ECML'2000/ML net Workshop on Machine Learning in the New Information Age*, pages 75–83, 2000.

Ontology Mapping - A User Survey

Sean M. Falconer[1], Natalya F. Noy[2], and Margaret-Anne Storey[1]

[1] University of Victoria, Victoria BC V8W 2Y2, Canada
{seanf, mstorey}@uvic.ca
[2] Stanford University, Stanford, CA 94305, USA
noy@stanford.edu

Abstract. Ontology mapping is the key to data interoperability in the semantic web vision. Computing mappings is the first step to applications such as query rewriting, instance sharing, web-service integration, and ontology merging. This problem has received a lot of attention in recent years, but little is known about how users actually construct mappings. Several ontology-mapping tools have been developed, but which tools do users actually use? What processes are users following to discover, track, and compute mappings? How do teams coordinate when performing mappings? In this paper, we discuss the results from an online user survey where we gathered feedback from the community to help answer these important questions. We discuss the results from the survey and the implications they may have on the mapping research community.

1 Introduction

Ontology mapping is a complex and largely user-driven process that can benefit from tool support. In the past few years, researchers have developed many tools and techniques for creating ontology mappings. Tools include PROMPT[12], COMA++ [3], Clio [10], Chimaera [9] and OWL Lite Alignment (OLA) [1]. Much research has been spent on developing the algorithms used by these tools, and indeed the authors of [13, 15] cite more than 20 different algorithms that can be used to generate candidate mappings. However, in most cases, the mapping process cannot be fully automated [2, 4, 5] and user input is required to accept, reject, or create new mappings. Despite the necessary role users play in the mapping process, there has been little work done to understand how and why users perform mappings.

In order to design more effective tools and algorithms, we claim that a deeper understanding of the interplay between tool, user, and the process is needed. For example, who are these users that are going to use the tools? Why do they need to perform mappings and for which domains? Do they use the currently available tools and if so, how do they use them? And, do these tools meet their needs? To answer these questions, we designed a survey and gathered feedback from the ontology mapping community. To our knowledge, this survey is the first specifically designed with these goals. The information gained from this survey should be valuable to both tool and algorithm designers. For example, in part as a consequence of this survey, we believe that the biggest gains in mapping will not come from improvements in the precision and recall in matching algorithms, but rather from better tool support.

The paper is organized as follows. We first present related work in Section 2. Next, we describe our survey design in Section 3. Following this, we present the results from the survey (Section 4). In Section 5, we discuss the results and implications for tool designers. We then describe limitations of the survey in Section 6. Finally, in Section 7, we present our concluding remarks and directions for future work.

2 Related work

Specifying mappings between one or more ontologies is well recognized to be a challenging and complex process that can be made significantly easier through tool support. The typical mapping process is an iterative procedure whereby the tool presents to the user a set of candidate mappings and the user then decides to accept and reject some of those mappings. The process is repeated until the user is satisfied that the mapping is sufficiently complete.

Determining candidate mappings is a challenging algorithmic problem. Consequently much of the research to date has been expended on designing more efficient and effective algorithms for determining candidate mappings. But much of the mapping process involves a tight collaboration and coordination between the user and tool. For example, the user must decide which mappings to accept and reject, keep track of progress, and determine when enough mappings have been completed for the intended purpose. These tasks are cognitively challenging but can be made easier through an improved partnership with the tool during the mapping process. Despite the gains that can be made across the entire mapping process, little research has focused on improving the effectiveness of the user decision process. Notable exceptions include user studies with PROMPT and Chimaera [6, 11], mapping experience reports [7, 14] and our own observational user study [5].

The user study experiment conducted with PROMPT concentrated on evaluating mapping suggestions provided by the tool by having several users merge two ontologies [11]. The number of steps, suggestions followed, suggestions that were not followed, and what the resulting ontologies looked like were all recorded. Precision and recall was used to evaluate the quality of the suggestions. The experiment only involved four users, which was too small to draw generalizable conclusions. Independently, PROMPT was evaluated, along with Chimaera by Lambrix and Edberg [6] with the purpose of merging ontologies from bioinformatics. Eight users were involved in the study, four with computer science backgrounds and four with biology backgrounds. The participants were given a number of tasks to perform, a hard copy user manual , and the software's help system for support. They were also instructed to "think aloud" and an evaluator took notes during the experiment. Afterwards, the users were asked to complete a questionnaire about their experience. The tools were evaluated with the same precision and recall measurements as used in the previously described PROMPT experiment, while the user interfaces were evaluated using the REAL (Relevance, Efficiency, Attitude, and Learnability) [8] approach. Under both criteria, PROMPT outperformed Chimaera, however, the participants found learning how to merge ontologies in either tool was equally difficult. The participants found it particularly difficult to perform non-automated procedures in PROMPT, such as creating user-defined merges.

Reed and Lenat [14] reported on their experiences with manually mapping the CyC ontology to other ontologies over a period of 15 years. The process relied on trained ontologists collaborating with domain experts. Over time, interactive clarification-dialog-based tools were developed to help ease the mapping procedure. The authors believed that the major barrier to the adoption of ontology mapping is the heavy reliance on someone setting up the source schemas and access protocols. They also stated that better tools were needed in order to allow domain experts to perform mappings rather than relying on ontology experts.

Lomax and McCray [7] described their experiences with mapping the Gene Ontology (GO) to the National Library's Unified Medical Language System (UMLS). The authors used a combination of methods to perform the mapping, starting with a preliminary exploration of both ontologies looking for overlap and then using an automated system to map 25% of the GO terms to UMLS terms. Following this, one of the authors visited the UMLS team for a month to work with the team in an attempt to complete the mapping. While many problems surfaced during this time, these were eventually addressed, debated and resolved. Through a combination of automated techniques, analysis and collaboration from the UMLS and GO teams, and mapping verfication by humans, the GO ontology was fully mapped to UMLS.

In our previous work, we presented results from a user study where we observed teams participating in a "think-aloud" ontology mapping session with two different tools [5]. The goal of this study was to gain a deeper understanding of the user needs and how they could be met more effectively through tool support. Although the results of the study were informative, we were left with questions that required feedback from the ontology mapping community. For example, users had trouble remembering what mappings they had created or verified while working with the tools. Also, the participants in the study were frustrated by not knowing how much of the mapping task they had already completed and what was left to be completed. The participants also had difficulty learning and working with the tools, which reinforces the findings reported in the Lambix and Edberg study. The participants in our study were not typical ontology mapping tool users and indeed were trained to use the tools before the study. Hence, we are interested in discovering if the problems encountered by our test users are also experienced by ontology mapping tool users with pragmatic and pressing needs for ontology mappings.

Despite some preliminary work in this area of understanding mapping tool users, we believe there is a lack of knowledge about the tools currently used, the users themselves, and the problems faced during the mapping process. Hence, we designed the survey that is presented in the next section of this paper to gain more insight into these questions.

3 The survey

We conducted a user survey to gain a basic understanding of why people perform ontology mappings, the tools they use, problems they experience, and what processes, if any, they follow.

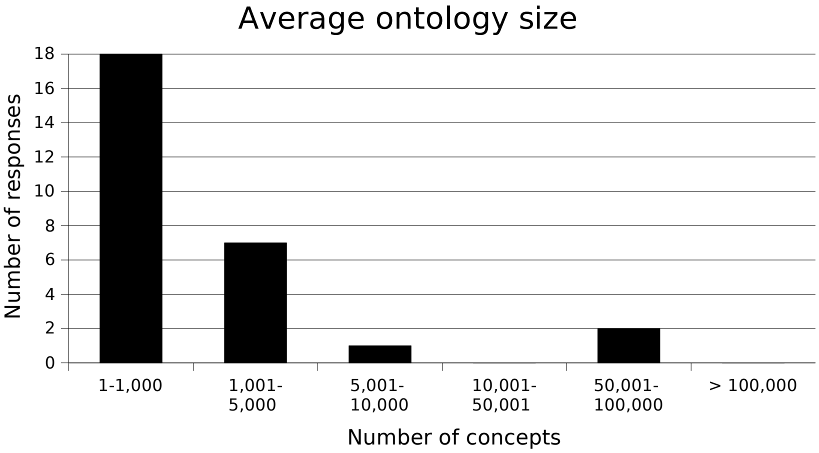


Fig. 1: Size of ontologies being used.

3.1 Participants

The survey was made available online for a two week period to both industry and academia via semantic web related mailing lists and news groups. Twenty-eight people responded to the survey.

3.2 Survey design

We designed the questions, organization, and presentation of the survey with the help of several researchers with human-computer interaction and ontology-mapping experience. The survey consisted of **multiple-choice** and **open-ended** questions (Table 1). Ten of the questions were completely open-ended, three were multiple-choice, and five were multiple choice answers that also allowed an open-ended response.

The questions can be classified into three categories. Questions 1 through 5 were **user context questions**, questions 6 through 8 and 17 were **tool questions**, and questions 9 through 16 were **process questions**. The user context questions were designed to gather data relating to the use cases for ontology mapping. The tool-related questions were designed to explore our questions regarding which tools people are using and whether the users find the tools useful. The process-related questions investigated the difficulties with performing mappings, whether users are working in team environments, and what processes people are following for coordinating their mapping efforts. Table 1 lists the specific questions that were asked.

4 Survey results

4.1 User context questions

The first question asked participants what domains of ontologies they work with. We had participants from a variety of domains, the most popular being biomedical, media, information-system design, business, and travel. Several respondents worked with

Table 1: Survey questions

No.	Question	Response
	User context questions	
1	What are the domains of ontologies that you work with?	open-ended
2	What is the average size of the ontologies you work with?	multiple-choice
3	What type of ontologies do you work with?	combined
4	How often do you create/edit ontology mappings?	open-ended
5	What are these mappings used for?	combined
	Tool questions	
6	What mapping tools have you used?	combined
7	Of the tools you've used, what do you find most useful and why?	open-ended
8	Of the tools you've used, what do you find to be deficient about these tools?	open-ended
17	If you were to design your perfect mapping tool, what features will it have?	open-ended
	Process questions	
9	Do you find it difficult to create an ontology mapping?	multiple-choice
10	If you answered "Yes" to question 9, please explain why.	open-ended
11	What process do you use when performing mappings?	combined
12	How do you remember which mappings you have created/verified versus mappings that are left to create/verify?	combined
13	When do you consider that the mapping is complete?	open-ended
14	Do you experience problems while performing ontology mappings? If so, please explain.	open-ended
15	How many people participate in the creation process of the mappings you perform?	multiple-choice
16	If the number of participants is greater than one, please briefly explain how the work is coordinated.	open-ended

different domains of ontologies, either for research purposes or based on their current project.

Figures 1 and 2 show the distribution of results for the questions on the size of the ontologies and the ontology languages used. Most participants worked with ontologies with less than 1000 concepts and the primary ontology languages are OWL and RDFS.

In the next user context question, we looked at how often participants create and edit ontology mappings. The responses varied, the most popular being that mappings were created either per-application or as often as their work dictated that a mapping was required. Two of the respondents indicated they performed this operation often, but for research and tool testing purposes.

The final user context question asked why the participants created mappings. Participants were allowed to select from a list of responses (as many that applied) and also provide a free-form response in an "Other" field (see Figure 3). The use cases listed were "Instance data sharing", "Ontology merging", "Query translation", "Web service integration", and "Other". Two of the "Other" responses stated that they create mappings purely for research purposes, while one was for automatic SQL generation.

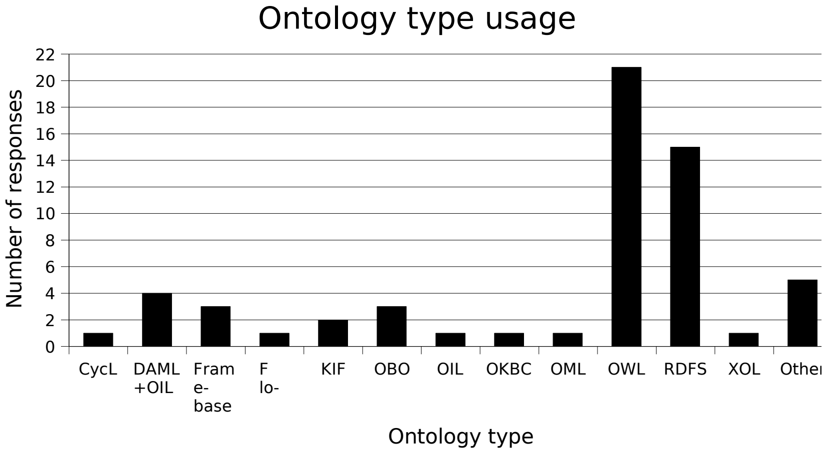


Fig. 2: Ontology language usage.

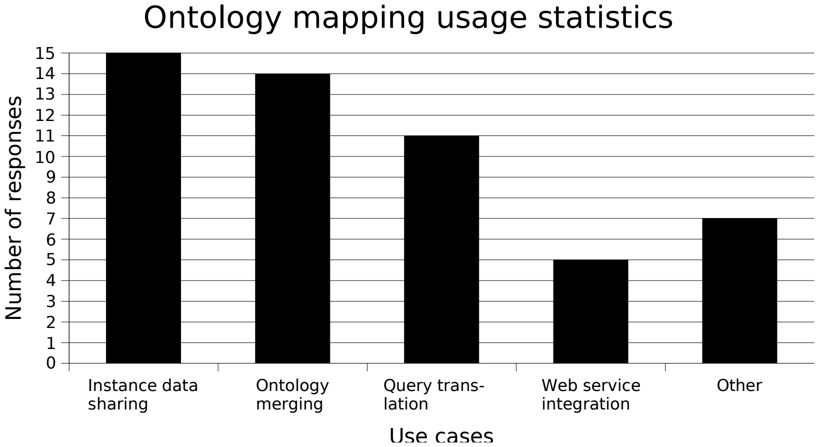


Fig. 3: Ontology mapping use cases.

4.2 Tool questions

The first tool-related question asked which tools users had tried. Respondents could choose from seven tools: Chimaera, COMA++, FOAM, MoA Shell, OLA, PROMPT, and QOM. They could also list any other tools in the “Other” field. Each of the listed tools was used by at most one to three participants with the exception of MoA Shell, which none of the respondents used. As shown in Figure 4, no tool was particularly dominant. The bulk of the feedback came in the “Other” category, which had 17 participant responses.

Other tools included Protégé, Internet Business Logic, AUTOMS, Crosi, WSMT with Ontostudio, X-SOM, OMAP, Falcon-AO, HMatch, and Snoggle. Each of these tools were used by only one participant, except X-SOM, which had been used by two.

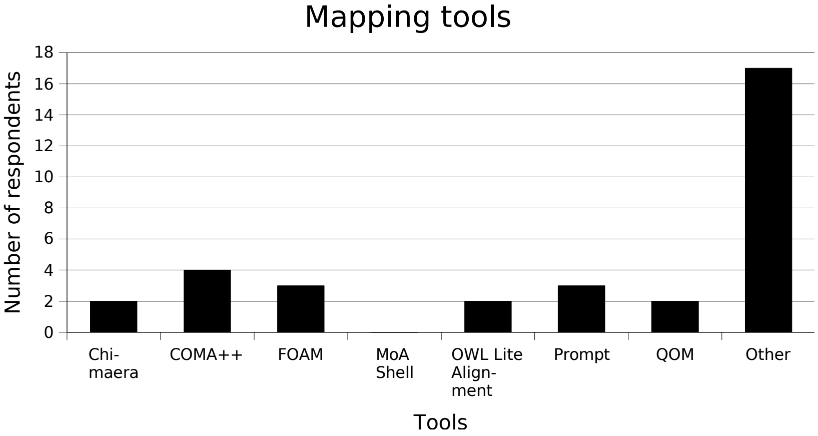


Fig. 4: Ontology mapping tools.

Two participants indicated that they use a custom built solution, while one indicated that they use a completely manual process.

We asked which tools and features participants found most useful and what deficiencies they found with the tools. Both Crosi and COMA++ were found to be useful because they integrate a large variety of similarity algorithms and are available online. One user indicated that they like tools to provide simple suggestions and automatic help, while another user had a contrasting view, stating that they like statistically-based tools because others require too much designer opinion. Protégé was highlighted as being good for manual creation of mappings as it makes it easy to create ontologies. Several participants pointed out that many tools are too general and are built without domain-specific mechanisms. One of the custom built solutions was indicated to be similar to PROMPT, but was built to take advantage of domain knowledge, specifically term normalization algorithms and synonyms for their domain of interest. The requirement for the tools to incorporate domain-specific analysis and features was a common theme in response to several questions in the survey. Another common theme was the lack of visual displays or easy to use tools. Specifically, one participant indicated that PROMPT's interface was too complicated to give to a domain expert to do the mapping. One user criticized specific tools for their lack of documentation, for being buggy, and for not working as described. Other interesting observations were the lack of precision and recall for mappings in real world ontologies and that the tools do not allow for expressive enough mappings (e.g. some tools only support 1-1 mappings).

In the final tool-related question, we asked the respondents to describe which features the perfect mapping tool would have. In the presentation of the survey, this question came at the end, but here we categorize it as a tool question. There were several interesting themes that came up in the responses.

The first theme was that six of the 19 responses related to the desire for *better and easier to use tools*. Specifically, participants stated that they needed better interfaces, graphical cognitive support, improved user interactivity, and facilities for explaining manual mappings. Users highlighted a large number of *desired features for* *the algorithms for generating candidate mappings*: powerful and pluggable matching algorithms, recognition of re-occurring patterns of representing information, identification of not only simple correspondences but also of complex ones, and extending beyond mere word-pair assocations and semantics. Four of the responses stated the requirement for perfect precision of recall for the mapping algorithms. Three participants also indicated that they want better facilities for *testing mappings and support for more expressive mappings*. The final interesting theme was *collaborative support*. Most of the respondents indicated that they work on their mappings in teams (see *process questions*). Most available tools do not support this type of team development scenario.

4.3 Process questions

The first process-related question asked whether the participants found the creation of an ontology mapping difficult. 30.8% replied “No” to this question, while 69.2% said “Yes”. The follow-up question to those participants that answered “Yes”, asked participants to explain why they found the process difficult. Ten of the 21 responses discussed semantic issues, such as the process being too subjective or ambiguous. One participant pointed out that the *“semantics of the underlying ontologies are not usually well defined. Without a solid understanding of the semantics, it is almost impossible to perform the mapping correctly.”* Respondents also discussed a lack of domain expertise for performing mappings, and that *“[y]ou have to get into the brains of the original developers of the ontologies being mapped.”* Participants also stated that tools are not flexible enough for application-specific needs, resulting in the manual creation of mappings, which is neither scalable nor efficient. One participant indicated that the OWL primitives for expressing mappings are poor and that users are faced with making difficult decisions when two related concepts *“almost but not exactly match.”* Three participants also indicated that problems with creating mappings resulted from poorly designed and documented tools.

We next asked participants what process they followed when performing mappings. Available responses were “Tackle the easy matches first” (37.0%), “Focus on a familiar area of the ontology” (51.9%), “Rely on the tool’s ordering of mappings” (14.8%), and “Other” (22.2%). Responses for the “Other” category included performing an automated matching up front and then a debugging step, while two of the responses indicated that they first applied lexical, then structural, and finally semantic methods.

In the next question, we asked how the participants remembered mappings they had created or verified. Most respondents chose from the provided answers, “The tool supports this” (37.0%) and “Paper and pencil” (55.6%), while 22.2% filled out the “Other” option. In the “Other” responses, one user indicated that the tool they use supports this, which works well when mappings are done in a single pass, but extra help is needed for multiple passes. Another respondent indicated that they use their own codes to report the mappings they create, which is similar to tracking the information by paper and pencil. Finally, one respondent indicated that they did not follow any good process. It is interesting that the majority resort to tracking this information manually by paper and pencil. Similar types of changes exist in software development and most IDEs and source control systems handle the tracking of this data.

We then asked when the participants considered the mapping to be complete. Ten of the 25 responses indicated that they used some form of testing (automated or manual) to verify that the mapping was completed to their satisfaction. For many respondents, this testing meant determining whether the mapping supports whatever application they were working on. Five responded that they knew the mapping to be complete when all concepts had been mapped. However, this implies either a perfect mapping, or that they knew when all reasonable concepts had been mapped. Interestingly, three participants responded that they never knew when the mapping was complete. Only one respondent indicated that they relied on tool support for determining whether the mapping was complete, although one participant stated that they must trust the system when mapping large ontologies because verification by hand is too slow.

We also asked participants about the types of problems they experienced while performing mappings. Similar issues outlined in previous questions came up again. Specifically, one respondent stated that *"most ontology tools are difficult for business users to understand."* Testing the mapping was also a popular theme along with issues related to the problem that people model conceptualizations differently.

The final two questions dealt with whether participants worked in teams and what sort of process they followed for coordinating their efforts. 53.9% indicated that only 1 or 2 people were involved in the mapping process, 42.3% worked in groups of 3 to 5 people, and finally, 3.9% worked with 6 to 10 people. Based on results from the team process question, we were able to determine that of the 53.9% working in teams of 1 to 2 people, 53.8% of these actually work in a team of 2, which means that only 24.0% work completely on their own. It is interesting that these users felt that there could be an automatic algorithm that they would trust completely.

The team-coordination processes ranged from weekly meetings to collaborating through wiki's to coordinating through CVS. 18 of the 20 respondents relied on non-software solutions for managing the team or a combination of *ad hoc* communication strategies like CVS, wiki's and e-mail along with meetings. Smaller teams typically had one team lead and one implementer, and coordinated with face to face meetings. Participants also indicated that they worked as a group or partitioned the ontologies and then performed a group check to validate the mapping. Some teams used domain expertise for reviewing the composed mappings or during the mapping process for input. One respondent indicated that they use a "brainstorming" team process for coordinating the mapping effort.

5 Discussion of results

We found it surprising how many tools had been tried by our respondents. There has clearly been a large effort from the research community to develop so many tools, yet there does not appear to be a dominant tool that is a benchmark for mapping tool design. This may be due in part to the variety of user needs. Some respondents highlighted that they had domain-specific needs or that existing tools do not support sophisticated enough mappings.

Most of the problems, deficiencies, and issues with ontology mapping uncovered by the survey can be classified into one of two categories: fundamental issues with language and semantics, and tool-specific issues. Fundamental issues, such as different model conceptualizations and language ambiguity, are difficult, if not impossible problems to solve. It is interesting that some of the responses to the "perfect mapping tool" question were that the tool would have 100% precision and recall or full natural language understanding. While a perfect, fully-automated solution would be ideal, it is probably not realistic for any but the most simple, straight-forward mappings. As the survey also highlighted, many of the problems that people face in ontology mapping, are difficult for even a team of human experts to resolve.

It is also interesting that these users felt that there could be an automatic algorithm that they would trust completely. Issues of trust also came up in our previously mentioned user study. Although our users stated that they liked PROMPT's explanation facility, they were also often confused when it made a suggestion that was obviously wrong. Incorrectly generated candidate mappings would sometimes lead to our users ignoring the suggestions and switching to a completely manual process.

Tool specific issues such as better user interfaces, graphical support, better testing facilities (data translation based on mappings), interactivity, algorithm explanation capabilities, and so forth are all problems that ontology-mapping tool developers can help with. As discussed in the results, one respondent indicated that PROMPT was too complex to give to their end-users. This sentiment was also echoed by a non-computer science participant in our user study. Mapping is a complex process, it is important that we do not further burden our users with learning a difficult to use software suite. Instead we must support them via the software. In agreement with Bernstein *et al.* [2], we believe that at this point the biggest productivity gains in mapping tasks will come from better cognitive support rather than from an improvement of precision and recall in matching algorithms.

The responses to the process-related questions brought up several interesting issues that tool designers and researchers may also need to address. First, it is noteworthy that many of the participants relied on paper and pencil to remember the mappings they perform. One individual even noted that some tools work for a first pass, but then they "forget" the steps previously performed during the second pass. Tool support should be able to address this issue. Second, it appears that most users work in small teams but tools currently lack support for team communication and collaboration, as well as for partitioning the mapping process into manageable chunks that can be tackled by individuals on the team. Many teams work together to validate the prepared mappings. Again, tool support could help with team work. Metadata annotations (perhaps visualized via color-codinng) could be used to record who composed the mapping and why they made certain decisions. In addition, the ontology mapping community could borrow ideas from the Web 2.0 social networking community [3]. E.g., tools could support voting on mappings, commenting on and annotating mappings, and associating instance data with a conceptualization. There has been some experimentation with community-driven ontology mapping [16], but tool support is currently limited.

---

[3] `http://del.icio.us`, `http://www.flickr.com`

6 Limitations of the survey

There are of course limitations to this study, the first and foremost being the sampling size and population. Although we feel that 28 respondents gave us a wide variety of interesting and useful responses, with such a small sample it is possible that our responses are biased. Also, as we solicited participants from mailing lists, most of which were academically oriented, our sample may be biased towards researchers in the area rather than a balance between those working in research and industry. Finding and recruiting users from our target population was also an issue, because it is difficult to know how to best reach ontololgy mapping tool users.

As with any on-line survey, the wording of some of our questions may have potentially been confusing to some participants. For example, we asked "If you were to design your perfect mapping tool, what features will it have?". We stated that some respondents indicated full natural language understanding and 100% precision and recall. Perhaps if the question had been worded differently to solicit feedback on a "realistic ideal mapping tool" the responses would have been different.

7 Conclusion and future work

In this paper, we discussed the design and results from our on-line user survey. The open-ended responses gave us valuable information about the types of problems users are experiencing, what features they'd like to see improved, some insight into their mapping and team process, and which tools are being used by the community.

Several issues raised by our participants indicate that their problems could be solved by improved tools and this would lead to better mapping results as well as greater adoption of the various mapping tools. This area of research has seen little activity to date. In the future, we plan to continue gathering feedback from the user community and carry out observational user studies. Our goal is to develop a comprehensive theoretical framework for cognitive support in ontology mapping. It is our hope that this will help guide the design and evaluation of future mapping tools such that the user's role in the mapping process is fully supported.

Acknowledgments

This work was supported by the National Center for Biomedical Ontology, under roadmap-initiative grant U54 HG004028 from the National Institutes of Health. We wish to thank Philip Bernstein for his helpful suggestions about where to post our survey requests. We also wish to thank Chris Bennett and Jody Ryall for their editing suggestions.

References

1. Ontology alignment with ola. In *Evaluation of Ontology-based Tools*.
2. P. A. Bernstein and S. Melnik. Model management 2.0: Manipulating richer mappings. In *SIGNMOD'07*, June 2007.

3. H.-H. Do. *Schema Matching and Mapping-based Data Integration*. PhD thesis, Department of Computer Science, Universität Leipzig, 2006.
4. S. M. Falconer, N. F. Noy, and M.-A. Storey. Towards understanding the needs of cognitive support for ontology mapping. In *Ontology Matching Workshop*, 2006.
5. S. M. Falconer and M.-A. Storey. Cognitive support for human-guided mapping systems. Technical Report DCS-318-IR, University of Victoria, Victoria, BC, Canada, 2007.
6. P. Lambrix and A. Edberg. Evaluation of ontology merging tools in bioinformatics. In *Proceedings Pacific Symposium on Biocomputing*, pages 589–600, Kauai, Hawaii, USA, 2003.
7. J. Lomax and A. T. McCray. Mapping the gene ontology into the unified medical language system: Research papers. *Comp. Funct. Genomics*, 5(4):354–361, 2004.
8. J. Löwgren. *Human-computer Interaction. What every system developer should know.* Studentlitteratur, Lund, 1993.
9. D. L. McGuinness, R. Fikes, J. Rice, and S. Wilder. The chimaera ontology environment. In *Proceedings of AAAI*, pages 1123–1124, 2000.
10. R. J. Miller, L. M. Haas, and M. A. Hernández. Schema mapping as query discovery. In *VLDB 2000*, pages 77–88, 2000.
11. N. F. Noy and M. A. Musen. Evaluating ontology-mapping tools: Requirements and experience. In *Proceedings of OntoWeb-SIG3 Workshop*, pages 1–14, 2002.
12. N. F. Noy and M. A. Musen. The PROMPT suite: Interactive tools for ontology merging and mapping. *International Journal of Human-Computer Studies*, 59(6):983–1024, 2003.
13. E. Rahm and P. A. Bernstein. A survey of approaches to automatic schema matching. *VLDB Journal: Very Large Data Bases*, 10(4):334–350, 2001.
14. S. L. Reed and D. B. Lenat. Mapping ontologies into cyc. In *Proceedings of AAAI*, 2002.
15. P. Shvaiko and J. Euzenat. A survey of schema-based matching approaches. *Journal on Data Semantics*, pages 146–171.
16. A. V. Zhdanova. Towards a community-driven ontology matching. In *K-CAP '05*, pages 221–222, 2005.

Multi-Concept Alignment and Evaluation

Shenghui Wang[1,2], Antoine Isaac[1,2,3], Lourens van der Meij[1,2], and Stefan Schlobach[1]

[1] Vrije Universiteit Amsterdam
[2] Koninklijke Bibliotheek, Den Haag
[3] Max Planck Institute for Psycholinguistics, Nijmegen
{swang,aisaac,lourens,schlobac}@few.vu.nl

Abstract. In this paper we discuss a book annotation translation application scenario that requires multi-concept alignment – where one set of concepts is aligned to another set. Using books annotated by concepts from two vocabularies which are to be aligned, we explore two statistically-grounded measures (Jaccard and LSA) to build conversion rules which aggregate similar concepts. Different ways of learning and deploying the multi-concept alignment are evaluated, which enables us to assess the usefulness of the approach for this scenario. This usefulness is low at the moment, but the experiment has given us the opportunity to learn some important lessons.

1 Introduction

The ontology alignment problem is crucial for many applications. Many methods and tools have been proposed to assist this task by proposing candidate correspondences between two ontologies. Yet the application field is largely unchartered, and some specific problems remain untouched by the community.

One such problem is multi-concept alignment.[4] In these, a given correspondence[5] can involve more than one ontological entity on either side. These entities are possibly combined in expressions – *e.g* using the boolean operators AND and OR – as in `o1:FruitsAndVegetables→(o2:Fruits OR o2:Vegetables)`.

However, only a few ontology alignment tools and methods address the production and exploitation of complex multi-concept alignment [3]. Similar to the difficulties in finding complex matches between database schemas [5], dealing with multi-concept alignment is difficult:

- Finding and using multi-concept alignments have to deal with much more complex search space. If two ontologies have u and v concepts, respectively, there are $u \times v$ potential 1:1 correspondences. However, there will be a combinatorial explosion of the number of possible $m : n$ alignments.

---

[4] This is sometimes referred to as multiple, complex, or a combination of one-to-many (1:n), many-to-one (m:1), many-to-many (m:n) alignment [3].

[5] Here, an alignment between two ontologies is understood as a set of correspondences – mapping links – between entities – classes, properties or expressions – coming from these ontologies.

- It is unclear how to compare the similarity between two sets of concepts, as the semantic relations often cannot be stated explicitly as in [2].
- Alignment frameworks and formats do not say much about the semantics of multi-concept alignment, delegating the problem – and, to an extent, rightly so – to specific ontology languages and/or alignment applications. A typical example is the semantic link between several simple correspondences involving the same entities and a single correspondence involving complex expressions.[6]

Also, in practice, most applications in the ontology alignment community currently focus on rather abstract cases, such as ontology merging or bridging. This requires finding the one most similar entity in the target ontology to the one considered in the source ontology. Accordingly, the evaluation of alignments has relied on purely intellectual assessment of pairs of concepts.

In this paper we will give the applicative motivation for multi-concept alignment — an annotation translation case in the National Library of the Netherlands.[7] As instances – books – are described or *annotated* by concepts from the two vocabularies to align, our focus is on using statistical methods exploiting co-occurrence of concepts in the annotation of the same book [6], such as the Jaccard similarity measure. Here, we elaborate on this approach from a multi-concept alignment perspective: we take two statistical similarity measures – Jaccard and Latent Semantic Analysis (LSA) – and we test different aggregation methods – clustering and ranking – to build multi-concept alignments with them.

Our goal is to assess the usefulness of these methods for our application scenario. We explore strategies with different levels of accuracy for applying the obtained alignment to our data set. We also train these methods on different sets of annotated objects. The questions we want to answer are the following:

- Do these combinations of statistically-grounded measures and aggregation techniques perform well enough for our scenario?
- Can we improve the performance by carefully selecting the training sets for some of the techniques used?
- Is there a benefit using a sophisticated similarity measure (LSA) for instance-based multi-concept alignment?

In section 2, we present our application scenario – book annotation translation – and explain why it requires multi-concept alignment. Section 3 explains the way we use the dually annotated books to compute similarity measures that are aggregated to obtain multi-concept alignments. In section 4 we detail our experimental setting, focusing on the different strategies that we apply to translate book annotations using the rules contained in the alignments, and on how we

---

[6] As in `(o2:Fruits→o1:FruitsAndVegetables)⊓(o1:Vegetables→o1:FruitsAndVegetables)` and `o1:FruitsAndVegetables→(o2:Fruits OR o2:Vegetables)`.

[7] Different from classifying documents in hierarchical categories, in our task, a set of concepts should be aligned to another set of concepts, while both sets represent the same or very similar semantics.

evaluate the results of this process. Section 5 presents the results for the different options tested: similarity measures, training sets, aggregation and rule firing strategies. In sections 6 and 7 we point out some lessons learned and conclude on the perspectives of this research.

2 Problem description

The library context The National Library of the Netherlands (KB)[8] maintains a large number of collections, among them the *Deposit Collection* containing all the Dutch printed publications (one million items), and the *Scientific Collection* with about 1.4 million books mainly about the history, language and culture of the Netherlands.

Each collection is described according to its own indexing system. On the one hand, the Scientific Collection is mainly described using the GTT, a huge vocabulary containing 35000 general terms ranging from *Wolkenkrabbers* (Skyscrapers) to *Verzorging* (Care). On the other hand, the books contained in the Deposit Collection are mainly indexed against the *Brinkman thesaurus*, containing a large set of headings (more than 5,000) that are expected to be global subject of a book. Both thesauri have similar coverage but differ in granularity, and provide the usual lexical and semantic information found in thesauri: broader and related concept, synonyms and notes.

The co-existence of these different annotation systems, even if historically and practically justified, is not satisfactory from the interoperability point of view. KB is therefore investigating ways to combine the two thesauri, trying to enhance integration while retaining compatibility with the legacy data of both systems. For this reason, mapping concepts between GTT and Brinkman are crucially needed.

Finally, it is important to mention that around 250,000 books are common in both depot and scientific collections, and have therefore been manually annotated with both GTT and Brinkman vocabularies. This allows us to investigate the co-occurrence information in the dually annotated dataset in order to find the semantic alignments between concepts from these two thesauri.

A annotation translation scenario The application scenario in this paper is that one thesaurus (*e.g.* GTT) would be dropped. In such a case, a huge volume of legacy data would have to be made compatible with the indexing system that remains (Brinkman). This requires converting the GTT annotations into equivalent Brinkman annotations.

A first approach would be to find, for each concept from GTT, the one in Brinkman which is the semantically closest. Each time this GTT concept would appear in a book description, the corresponding Brinkman concept could be added. Yet such a one-to-one conversion is not satisfactory. First, and most intuitively, one observes that the two vocabularies do not have the same granularity.

---

[8] http://www.kb.nl

Sometimes for a concept in a vocabulary it is impossible to find a corresponding concept, with the same meaning, in the other vocabulary. This is the case for instance for the Brinkman `gassen ; mechanica` (gas mechanics) which has no equivalent in GTT, while the latter includes both `Gassen` and `Mechanica`. This suggests the necessity to introduce alignment links involving several concepts from the same vocabulary. This is confirmed by a second aspect which is more guided by the vocabulary application itself. The GTT annotation process makes use of *post-coordination*: multiple concepts found in a same annotation shall not be considered independent but rather as facets of a more complex virtual subject. With GTT, if a book is annotated by `historische geografie` and `Nederland` you expect the book to be about a more complex "historical geography of the Netherlands" subject instead of being about these two individual subjects in a disconnected manner. Ideally, a conversion algorithm would therefore have to exploit alignments that involve concepts that are combined together. In this way, implicit complex subjects would be properly detected and converted when they occur in an annotation.

3 Multi-concept alignment generation

Closely related concepts form a virtual conceptual entity, and the alignment is generated in terms of these virtual entities. An intuitive way of generating such a virtual entity is to group similar concepts together.

3.1 Instance-based similarity measures

We base our concept aggregation on the similarity between concepts. In [6], some similarity measures we have investigated generate 1:1 mappings between concepts based on their co-occurrence in annotations of books. In this paper we further investigate two measures.

Jaccard similarity The Jaccard similarity coefficient is a simple measure for similarity of sets:

$$J(A,B) = \frac{|A^i \cap B^i|}{|A^i \cup B^i|}$$

where A^i is the set of instances of concept A, in our case, the books which are annotated by A. If there is a perfect correlation between two concepts A and B, the measure will have a value of 1, if there is no co-occurrence, the measure is 0.

Latent Semantic Analysis (LSA) [7] was used to analyse the concept-book co-occurrence matrix and calculate the similarity between concepts. LSA is a statistical technique developed for extracting and representing the similarity between words and between documents by analysis of large bodies of text. In our context we expect the method to provide a measure for the correlation between concepts in annotation (corrected for insufficient data), as well as a way to distinguish the relevant correspondences between such concepts.

The occurrence of each concept in the annotation of each book is first counted and stored in a concept-by-book matrix $X_{c\times b}$. By using the singular value decomposition (SVD), the matrix $X_{c\times b}$ is decomposed as

$$X_{c\times b} = C_{c\times r} S_{r\times r} B^T_{r\times b},$$

where c is the number of concepts, b is the number of books, $C_{c\times r}$ describes the original concepts as vectors in a space of r derived orthogonal factor values, $B^T_{r\times b}$ describes the original book annotations in the same way, and $S_{r\times r}$ is a diagonal matrix containing scaling values, which are all positive and ordered in decreasing magnitude. Using only the k largest eigenvalues, a reduced matrix is reconstructed as

$$X_{c\times b} \approx \hat{X}_{c\times b} = C_{c\times k} S_{k\times k} B^T_{k\times b},$$

which is closest in the least squares sense to $X_{c\times b}$. The aim of this *dimension reduction* is to capture the most important structure but reduce noise and variability in concept usage. We used the percentage of accumulated singular values to determine k. In our case, we kept 80% accumulation.

The product

$$D_{c\times c} = \hat{X}_{c\times k} \hat{X}^T_{c\times k} = (C_{c\times k} S_{k\times k})(C_{c\times k} S_{k\times k})^T$$

gives the paired similarity matrix between concepts [1].

Both methods provide measures of similarity between pairs of concepts. We will see that different similarity measures group concepts in different ways and their performance in our evaluation also vary.

3.2 Concept aggregation

Here we introduce two ways of aggregating concepts using the similarity measures calculated above.

Grouping concepts based on 1:1 mappings For a specific concept C_0, a list of concepts which are the top k ranked in similarity is easily generated, i.e.,

$$C_0 \rightarrow (C_1, C_2, \ldots, C_k).$$

This list is not limited to contain the concepts from the different thesauri only, instead, it contains concepts from both thesauri, ranked by their similarity to concept C_0.

Once a threshold k is chosen,[9] the top k concepts together with concept C_0 are expected to form a closely related conceptual entity. Dividing this set of $k + 1$ concepts into concepts from one thesaurus and the other, the two sets of

---

[9] In our experiments, we chose $k = 10$.

concepts from both thesauri are used to define an n to m mapping between both thesauri. That is, if

$$G_0 \rightarrow (B_1, G_1, \ldots, B_m, G_n),$$

where $n + m = k$, then a mapping rule

$$(G_0, G_1, \ldots, G_n) \rightarrow (B_1, B_2, \ldots, B_m)$$

is generated.

Partitioning concepts based on clustering One similarity-based clustering technique [4] was used to partition the concepts into clusters. If one cluster contains k concepts

$$(B_1, G_1, \ldots, B_m, G_n),$$

where $n + m = k$, then a mapping rule

$$(G_1, \ldots, G_n) \rightarrow (B_1, B_2, \ldots, B_m)$$

is generated.

The clustering technique takes the similarity between all concepts into account, therefore the generated clusters partition the concepts in a global manner. In this way, the generated $n : m$ mappings are expected to reproduce the general correlation between concepts from both thesauri.

4 Evaluation

We use the book annotation translation scenario to evaluate the generated $n : m$ alignment. The complete set of dually annotated books was divided into two parts: 2/3 of books was used for training, and the rest 1/3 of books as the testing data. Using different sampling methods, we have two training data sets:

$Train\ _{random}$: randomly selected books (5245 books and 7391 concepts)

$Train\ _{rich}$: books with at least a total of 8 annotations from both thesauri (5288 books and 10382 concepts)

4.1 Evaluation method

From the training data, we learn the alignment rules specifying which set of GTT concepts should be aligned to which set of Brinkman concepts, that is,

$$R : G_r \rightarrow B_r,$$

where G_r is a set of GTT concepts and B_r is the corresponding set of Brinkman concepts. In the testing dataset, each book has its GTT and Brinkman annotation, i.e. G_t and B_t. The GTT annotation was used to fire rules. This results in a generated set of Brinkman concepts $B_r^{'}$. By comparing B_t and $B_r^{'}$, we can evaluate the precision and recall of the learned alignment rules.

We now specify how to apply the rules to the GTT annotation and how to define precision and recall. Given a alignment rule $R : G_r \rightarrow B_r$ and a book out of the testing data with the annotations G_t and B_t, we define four different strategies – later denoted by *FireIf* – for firing rules:

1	$G_t = G_r$
2	$G_t \supseteq G_r$
3	$G_t \subseteq G_r$
ALL	Fire in all above three cases.

The different generated Brinkman concepts were distinguished by a subscript i: $B_{ri}^{'}$. We consider a book to be *matched* if its real Brinkman annotation and the generated set of Brinkman concepts overlap i.e. $B_t \cap B_{ri}^{'} \neq 0$, $i \in \{1, 2, 3, ALL\}$.

4.2 Precision and Recall

Precision and recall are calculated at two levels. At the book level, we measure the performance in terms of the fired books, which were fired by at least one rule. We define the precision as the fraction of the books in the testing data set that actually match their real Brinkman annotations, i.e.,

$$P_b = \frac{\#books\_matched}{\#books\_fired},$$

and the recall as how many books in the whole testing set are matched, i.e,

$$R_b = \frac{\#books\_matched}{\#books\_testing}.$$

where *#books_matched* is the number of books whose real Brinkman annotation overlap the generated set and *#books_testing* is the number of books in the testing data.

At the annotation level, we measure how well the generated set of Brinkman concepts match the real annotation, i.e.,

$$P_a = \frac{\sum \frac{\#good\_found}{|B_{ri}^{'}|}}{\#books\_fired}, \quad R_b = \frac{\sum \frac{\#good\_found}{|B_t|}}{\#books\_testing},$$

where $\#good\_found$ is the number of the real Brinkman concepts which are found in the generated set.

5 Results

Table 1 gives an overview of the relation between the alignment rules, the training sets and different methods. The number of GTT concepts in these rules is generally 2 or 3 times bigger than that of Brinkman concepts. This is consistent

Similarity	Methods	Training Set	#Rule	#GTT	#Brinkman
Jaccard	Ranking	random	5669	6.4	4.6
		rich	8334	8.6	4.2
	Clustering	random	246	15.1	5.8
		rich	242	84.2	14.4
LSA	Ranking	random	6916	6.2	4.2
		rich	7117	7.6	3.5
	Clustering	random	747	3.2	1.8
		rich	883	8.2	2.9

Table 1. Generated rules using different training sets and methods

Similarity	Methods	Training Set	P_b	R_b	P_a	R_a
Jaccard	Ranking	random	63.77%	12.26%	12.45%	10.26%
		rich	43.41%	12.43%	5.36%	9.83%
	Clustering	random	26.87%	3.59%	25.17%	2.42%
		rich	5.37%	0.80%	4.16%	0.53%
LSA	Ranking	random	67.55%	17.94%	6.60%	15.16%
		rich	62.51%	19.54%	8.10%	16.41%
	Clustering	random	39.68%	9.19%	22.06%	6.76%
		rich	33.03%	8.01%	10.65%	6.24%

Table 2. Performance overview of different methods

with the way of using GTT concepts for book annotation, i.e., *post-coordination*, which indicates several GTT concepts should be combined first and then aligned to a single Brinkman concept. One GTT concept in different combinations could also be aligned to different sets of Brinkman concepts.

Note that, when generating rules from the ranked lists, the top 10 most similar concepts were grouped with a target concept, therefore, the sum of GTT and Brinkman concepts is around 11. The ranked lists of different concepts may contain the same group of concepts, thus the total number of the rules is different and less than the total number of concepts. The clustering method creates partitions of concepts and each cluster generates one rule, so the generated rules are much less than those from the ranking method.

In Table 2, we compare the performance of different methods using different training sets, with the firing type ALL. One message from this table is that the arbitrary choice of using richly annotated books as training data, in general, does not bring much benefits. Instead, as it is biased towards the richly annotated books, it is more prone to overfitting the training data, and the performance on the testing data tends to be worse.

This is confirmed by the results: we notice that using the rich training data has a detrimental effect on the performance when Jaccard is used as a similarity measure, but this effect is not observed when LSA is used. This is to be expected, as LSA is much less sensitive to noise, and thus less prone to overfitting.

FireIF type	Method	C_r	P_b	R_b	P_a	R_a
1	Jaccard	47.56%	62.77%	0.65%	60.46%	0.55%
	LSA	21.95%	47.00%	0.57%	37.32%	0.44%
2	Jaccard	58.94%	50.43%	3.39%	47.76%	2.27%
	LSA	46.05%	36.95%	3.22%	29.98%	1.99%
3	Jaccard	88.21%	11.08%	0.85%	10.14%	0.69%
	LSA	82.20%	41.77%	6.53%	18.82%	5.21%
ALL	Jaccard	97.15%	26.87%	3.59%	25.17%	2.42%
	LSA	92.90%	39.68%	9.19%	22.06%	6.76%

Table 3. Comparison between Jaccard and LSA in clustering method, where C_r is the ratio of the fired rules over the total number of rules for the given technique.

5.1 Comparing Jaccard and LSA

By matching type *ALL*, the LSA similarity outperforms the simple Jaccard measure, in terms of the precision and recall of both ranking and clustering methods. Table 3 gives the detailed figures in terms of different rule firing types. On the one hand, using the simple Jaccard measure, more books were fired with the exact and subsume match (type 2). This indicates that Jaccard measure is good at finding explicit similarity from the co-occurrence information.[10] On the other hand, LSA is able to find some potentially similar concepts. Therefore, by slightly sacrificing in the precision of generated concepts, LSA significantly improves the precision and recall at the book level, and also gives a higher recall for annotations.

5.2 Comparing Ranking and Clustering

In Table 2, for type *ALL*, rules generated from the ranked lists give better precision and recall than those from clusters, except that the rules from the clusters have higher precision at the annotation level. Table 4 gives the detailed comparison between rules generated from ranking and clustering. Here, the similarity measure is LSA.

The rules generated from the clusters have better performance when fired by type 1 and 2. More than 20% rules were fired by the exact match. This indicates that those clusters matches the real annotations quite well, at least the GTT concepts within clusters are highly correlated and matches the real data. The high precision also guarantees the validity of the generated Brinkman concepts. The higher recall and lower precision of the rules generated from the ranked lists are because more concepts, including good candidates and noisy data, are introduced at the same time. Note that the rules generated from ranking were not fired at all by FireIf type 1, therefore the first line is empty.

---

[10] This is also confirmed in [6].

FireIF type	Method	C_r	P_b	R_b	P_a	R_a
1	Ranking					
	Clustering	21.95%	47.00%	0.57%	37.32%	0.44%
2	Ranking	0.01%	100.00%	0.01%	50.00%	0.01%
	Clustering	46.05%	36.95%	3.22%	29.98%	1.99%
3	Ranking	98.54%	67.55%	17.94%	6.60%	15.16%
	Clustering	82.20%	41.77%	6.53%	18.82%	5.21%
ALL	Ranking	98.54%	67.55%	17.94%	6.60%	15.16%
	Clustering	92.90%	39.68%	9.19%	22.06%	6.76%

Table 4. Comparison between rules generated from ranking and clustering.

6 Discussion

As shown in Table 1, using simple Jaccard similarity, the average size of clusters is much bigger compared to other methods. Actually, those clusters have a big variation in their size. Due to the existence of big clusters, many books are fired because their GTT annotations are subsumed by the rules (type 3). Because of the quality of the bigger clusters is somewhat low, the generated sets of Brinkman concepts are not really good, which decreases the precision and recall. However, if only counting smaller clusters (the size is equal to or less than 11), 80% of the clusters if using $Train\ _{random}$ (59% if using $Train\ _{rich}$) contain 1.4 GTT and 1.2 Brinkman concepts (1.9 : 1.2 if using $Train\ _{rich}$). These small clusters generally contain the concepts which are strongly related, therefore, 1:1 or 2:1 mappings generated from them are more reliable.

Rule size also matters when generating rules from the ranked lists. The choice of k directly affects the size of the rules. In our case, the choice of 10 involves too many concepts into the rules. Therefore, many books were fired because their GTT annotation was subsumed by the rules (type 3) while very few books are fired by exact (type 1) or subsume matches (type 2). It would be interesting to investigate the effect of the choice of k on the precision and recall performance.

Another concern is that one GTT concept could well be mapped to different sets of Brinkman concepts if it is combined with different GTT concepts. However, using the clustering methods, concepts are allocated into only one partition. This limits the number of rules as well as the final performance. We will investigate some probabilistic clustering techniques in order to allow more flexible rules.

7 Conclusion

In this paper we have introduced an application that needs multi-concept alignment. We have presented and evaluated strategies to generate and deploy such alignments, using statistical techniques that are expected to take into account the way concepts are combined in the instance-level annotation data.

At this point we still have to answer the question whether the measures and techniques we have used perform well for our annotation translation scenario.

First, the precision of the generated sets of Brinkman concepts (P_a) (see table 2) is not encouraging. Ranging from 4.16% to 25.17%, it prevents from trusting the candidate concepts as a unique source for annotation translation. A scenario where a user would be proposed these candidates and required to assess them to produce valid annotation would be more realistic: having to choose among, for example, 10 Brinkman concepts is already an improvement compared to selecting among all the Brinkman vocabulary. Additionally, table 4 shows that selecting the rule firing strategy can improve the precision. Yet, such a scenario seems to be ruled out by the annotation-level recall (R_a): in the most effective configuration, 16.41% of the correct annotations were found. The book-level recall (R_b) shows that at most 19.54% of the books in the testing set were given at least one good candidate. This means that an annotator should add his own concepts on top of the candidate ones. One single technique is therefore not enough to cope with the annotation translation for an entire collection.

Section 6 has shown that these weaknesses might be compensated by using aggregation techniques that approximate better the way concepts are used for book annotation. The objective there would be to increase the number of the more reliable rules to produce more valid results (type 1 and 2). Also, different strategies used here can be combined so as to obtain better performance. These two options have to be explored further.

Also, an immediate way to improve the results without even changing the methods could be to revisit the evaluation itself. Our evaluation technique gives a first and cheap assessment, which is likely to remain generally valid. Yet it is sensitive to *indexing variation*, the phenomenon that renders the fact that several annotators annotating a same book (or a same annotator annotating it at different time) will select slightly different concepts. Some manual application-specific evaluation shall be performed to assess the influence of this phenomenon and eventually compensate for its bias.

References

1. Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of The American Society for Information Science*, 41(6):391–407, 1990.
2. R. Dhamankar, Y. Lee, A. Doan, A. Halevy, and P. Domingos. imap: Discovering complex semantic matches between database schemas. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD)*, pages 383–394, 2004.
3. Jérôme Euzenat and Pavel Shvaiko. *Ontology Matching*. Springer-Verlag, 2007.
4. Brendan J. Frey and Delbert Dueck. Clustering by passing messages between data points. *Science*, February 16 2007.
5. B. He, K. C.-C. Chang, and J. Han. Discovering complex matchings across web query interfaces: A correlation mining approach. 2004.
6. Antoine Isaac, Lourens van der Meij, Stefan Schlobach, and Shenghui Wang. An empirical study of instance-based ontology matching. In *Proceedings of the 6th International Semantic Web Conference*, 2007. To appear.
7. T. K. Landauer, P. W. Foltz, and D. Laham. An introduction to latent semantic analysis. *Discourse Processes*, 25:259–284, 1998.

Emerging Consensus *in-situ*

Bo Hu, Srinandan Dasmahapatra, Paul Lewis

School of Electronic and Computer Science,
University of Southampton, SO17 1BJ, UK
{bh, sd, phl}@ecs.soton.ac.uk

Abstract Traditional ontology mapping techniques are not strictly applicable in a dynamic and distributed environment (e.g. P2P and pervasive computing) in which on-the-fly alignments are sought after. We propose an approach that collaborates the logic formalisms with collaboratively created web repositories. A logic conceptualisation based "signaturing" algorithm is to discover, from concept definitions, the "feature" vectors that uniquely identify concepts; web repositories are used to understand the implications of these features. Such a combination solidifies an on-demand and approximate mechanism that emerges a context-dependent and task-specific consensus among heterogeneous participants of an information exchange task.

1 Introduction

The prevalence of the Internet has made it possible to access a large amount of data. It has been commonly agreed that attaching machine-understandable semantics to web resources gives birth to "smart" applications and thus benefits ordinary web users by partially relieving them from routine tasks [3]. Thus far, the semantics is mainly depicted using ontologies. Due to a lack of universal standards and the diversity of human perspectives, it is inevitable that ontologies describing the same domain of discourse present semantic disagreement to some extent. Therefore, one of the primary tasks to facilitate the envisioned "smart" use of resources becomes establishing a mutual understanding among different ontology-driven, semantics-enhanced systems. This mutual understanding should faithfully reveal the intended meaning of different ontological entities.

Clearly, in order to have a mutual understanding, all the participants involved in an information exchanging task must agree upon a list of words as the semantics carriers and the meanings of these words must "pick out the same individuals in the same context" [14]. In other words, ontological entities pass meanings by not only values but also context-dependent referents. We refer to the structural and naming information attached to a concept the *value* of the concept while classified instances the *referents*. For example, we might use the sentence "*give me a French*" to ask for a French wine or a French movie depending on the conversational context. Current ontology mapping capability performs well in identifying task-independent semantic equivalences. While such techniques are good in generic ontology mapping scenarios, their applicability is suspect in certain circumstances. Let's take the peer-to-peer (p2p) environment as an example. In such a setting, when trying to establish a consensus, two major obstacles have prevented us from taking the conventional ontology mapping route. On the one hand, data owners would be more incline to well-targeted and task-specific solutions that allow them to share data within the context of a particular conversation instead of large scale and broad sense consensuses offered by some centralised authorities. On the other hand, the quality of a consensus is largely decided by the data that each individual holds. In many cases, such data might be so diverse, ambiguous and incomplete. Any consensuses stemmed therefrom demonstrate a certain degree of imperfectness, which is a function of the data possessed by a data provider. These two characteristics are, of course, not unique to p2p environments. Any applications aiming to provide on-the-fly semantic alignments present such characteristics.

In this paper, the first issue is accommodated by reinterpreting ontology mapping as a task situated in the background knowledge of a particular conversation: concepts are first decomposed into semantics-bearing signatures and are reenforced as feature vectors based on web encyclopedia repositories. When a particular conversation is to be conducted, we generate the corresponding feature vectors so as to reduce the network traffic and the subsequent computational burden (Section 3). During this process, Latent Semantic Analysis (LSA) is leveraged to alleviate the influence of modelling idiosyncrasy. In Section 3 and throughout the rest of this paper, the "Wine Ontology"[1] is used to detail our approach. LSA also leads to a feasible solution to the second topic of this paper. In a p2p or a similar environment, the consensus established w.r.t. a particular conversation should not be pealed off from the data that is held by each individual. The uncertainty of an answer is, therefore, defined as the degree of satisfaction, i.e. to what extent a request can be satisfied based on the local data that the query handling individuals possess. We propose a probabilistic model to quantify such satisfaction and regulate how an appropriate answer should be screened out from other candidates (Section 4). Finally, in Section 5, we conclude the paper with issues worth further discussion and investigation.

2 Preliminaries

Description Logic (DL) is a family of knowledge representation and reasoning formalisms. It has attracted substantial research interest recently, especially after the endorsement of DL-based ontology modelling languages (e.g. OWL [12]) by the Semantic Web initiative [3]. DLs are based on the notions of concepts (i.e. unary predicates) and properties (i.e. binary relations). Using defined constructs, complex concepts can be composed from primitive ones. In the context of DLs, an ontology is normally a 4-tuple $\langle CN, PN, \mathcal{C}, \mathcal{P} \rangle$ where CN is a set of concept names, PN a set of property names, $\mathcal{C}$ a set of concepts and $\mathcal{P}$ a set of properties. Let C and D be arbitrary concepts, P be a property, n be a non-negative integer,

---

[1] Available from http://www.schemaweb.info/schema/SchemaDetails.aspx?id=62

o_i ($1 \leq i \leq n$) be instances and $\top$, $\bot$ denote the top and the bottom. A $\mathcal{SHOIN}$ DL concept is: ($\mathcal{SHOIN}$ is the underlying logic of OWL-DL)

$$CN \mid \top \mid \bot \mid \mathsf{C} \sqcap \mathsf{D} \mid \mathsf{C} \sqcup \mathsf{D} \mid \neg \mathsf{C} \mid \exists \mathsf{R.C} \mid \forall \mathsf{R.C} \mid \geq_n \mathsf{R}.\top \mid \leq_n \mathsf{R}.\top \mid \{o_1, \ldots, o_n\}$$

An interpretation $\mathcal{I}$ is a couple ($\Delta^{\mathcal{I}}$, $\cdot^{\mathcal{I}}$) where the nonempty set $\Delta^{\mathcal{I}}$ is the domain of $\mathcal{I}$ and the $\cdot^{\mathcal{I}}$ function maps each concept to a subset of $\Delta^{\mathcal{I}}$ and each property to a subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$.

Latent Semantic Analysis (LSA) [6] is an approach to document indexing. For a large corpus of text documents, LSA assumes the existence of an underlying semantic model that can be captured using a term-document matrix with rows corresponding to terms and columns to documents. It then discovers such a model by projecting the original term-document matrix into a lower-dimensional vector space with effectively reduced noise. LSA has been found capable of simulating a variety of human cognitive phenomena and thus emulating the "meaning" discovering process. The advantage of LSA lies in the fact that the resulting correlation between an arbitrary pair of items (terms or documents) is not isolated from the rest of the representation system. The enabling technique of LSA is Singular Value Decomposition (SVD). SVD decomposes an $M \times N$ matrix $\mathfrak{M}$ and represents it as an approximation, $\hat{\mathfrak{M}}$, at a lower dimensionality k:

$$\mathfrak{M} \approx \hat{\mathfrak{M}} = USV^{\mathrm{T}} = (u_1 \cdots u_k) \begin{pmatrix} \delta_1 & & \\ & \ddots & \\ & & \delta_k \end{pmatrix} \begin{pmatrix} v^1 \\ \vdots \\ v^k \end{pmatrix} \tag{1}$$

where S is an $K \times K$ diagonal matrix of singular values, U is an $M \times K$ matrix of eigenvectors derived from the term-term correlation matrix given by $\mathfrak{M}\mathfrak{M}^{\mathrm{T}}$, and V is an $N \times K$ matrix of eigenvectors derived from the document-document correlation matrix given by $\mathfrak{M}^{\mathrm{T}}\mathfrak{M}$. Recently, methods based on LSA have been successfully applied to detect synonyms and acronyms [4].

3 Situating concept interpretation in WIKIPEDIA

Establishing consensus implies aligning different local ontologies. General ontology mapping has been extensively studied recently [2]. In this paper, we take an eclectic approach drawn from both the formal logics realm and Web2.0 applications. More specifically, we i) produce signatures that explicitly and quantitatively characterise the intensional restrictions of concepts, ii) retrieve web documents according to concept signatures as *virtual* instances, which are niched in the context of a particular "topic", and iii) generalise the signatures into individual-independent and task-specific *feature vectors* based on the landscape of their respective *virtual* instances. Similarity among concepts is then reduced to the similarity among their respective feature vectors.

3.1 Signaturing Concepts

In an ontology $\mathcal{O}$, semantics of concepts are concealed in the inter-concept relationships introduced through subsumptions and property references. The first step towards establishing consensus, therefore, becomes making explicit the semantics hidden behind the concept constructs. In order to reveal such “hidden” semantics, we recursively unfold concepts against their constructs till no further actions can be taken. In this paper, we focus on ontologies that can be represented with OWL-DL. More specifically, we restrict ourselves to $\mathcal{SHOIN}$(D) DL [10]. This restriction is due to both theoretical and practical considerations. On the one hand, $\mathcal{SHOIN}$(D) is Beth-definable. Although reasoning w.r.t. $\mathcal{SHOIN}$(D) is NExpTime-complete, it has been demonstrated [10] that deterministic complexity can be achieved by restricting the concept constructs to a carefully selected subset of $\mathcal{SHOIN}$(D) and/or translating $\mathcal{SHOIN}$(D) into the less expensive $\mathcal{SHIN}$(D) whose satisfiability reasoning is ExpTime [1]. On the other hand, after examining the available ones from the Internet, we observe that many ontologies can be or have already been rewritten in RDF(S) or OWL-DL, both of which are recommended by W3C. Methods developed for $\mathcal{SHOIN}$(D) is, therefore, applicable to those ontologies based on less expressive languages.

If cyclic definitions are not allowed—no primitive concept (property) appears on both sides of an introduction axiom, and all definitions are in their Negation-Normal Form—the negations are applied only to concept names, it is possible to fully unfold the righthand side of all concept introduction axioms and guarantee the termination of such an unfolding process.

$$
\begin{aligned}
\mathsf{WhiteBordeaux} &\doteq \mathsf{Bordeaux} \sqcap \mathsf{WhiteWine}\\
\mathsf{Bordeaux} &\doteq \mathsf{Wine} \sqcap \exists\, \mathsf{locatedIn}.\{\mathsf{BordeauxRegion}\}\\
\mathsf{Wine} &\sqsubseteq\ =_1 \mathsf{hasBody} \sqcap =_1 \mathsf{hasColor}\\
&\quad \sqcap =_1 \mathsf{hasFlavor} \sqcap =_1 \mathsf{hasMaker} \sqcap =_1 \mathsf{hasSugar}\\
&\quad \sqcap \forall\, \mathsf{hasMaker}.\mathsf{Winery} \sqcap \exists\, \mathsf{locatedIn}.\mathsf{Region}\\
&\quad \sqcap \geq_1 \mathsf{madeFromGrape}\\
\mathsf{WhiteWine} &\sqsubseteq \mathsf{Wine} \sqcap \exists \mathsf{hasColor}.\{\mathsf{White}\}\\
\mathsf{Region} &\sqsubseteq \top \qquad \mathsf{Winery} \sqsubseteq \top
\end{aligned}
$$

Figure 1. The WhiteBordeaux example

We adopt the construct transformation rules [1] to facilitate the concept unfolding. In Fig. 2, we demonstrate how concept WhiteBordeaux (defined in Fig. 1) is unfolded by repetitively applying the transformation rules. There are cases that concepts are only partially defined with necessary conditions (inclusions) instead of fully defined with both necessary and sufficient conditions (equalities). Before unfolding, inclusions (i.e. axioms in the form $\mathsf{C} \sqsubseteq \mathsf{D}$) are rewritten in equalities (i.e. axioms in the form $\mathsf{C} \doteq \mathsf{D}$). This is achieved by introducing a new primitive concept to represent the difference between C and D. For instance, we introduce $\mathsf{C\text{-}spec} \sqsubseteq \top$ and rewrite $\mathsf{C} \sqsubseteq \mathsf{D}$ into $\mathsf{C} \doteq \mathsf{D} \sqcap \mathsf{C\text{-}spec}$. In this paper, we assume that the set of newly introduced primitive concept names is disjoint with $CN \cup PN$ and bears clues to the original partially defined concepts. The unfolding process stops when no transformation rules are applicable. It has been demonstrated that by carefully selecting a set of admissible constructs, a termination of unfolding is guaranteed w.r.t. acyclic ontologies.

$$\mathit{\Pi}_1^{\mathsf{WB}} = \{\, x : \mathsf{Bordeaux} \sqcap \mathsf{WhiteWine} \,\}$$

$$\ldots$$

$$\mathit{\Pi}_1^{\mathsf{WB}} = \left\{ \begin{array}{l} x : (=_1 \mathsf{hasBody} \sqcap =_1 \mathsf{hasColor} \sqcap =_1 \mathsf{hasFlavor} \\ \quad \sqcap =_1 \mathsf{hasMaker} \sqcap =_1 \mathsf{hasSugar} \\ \quad \sqcap \forall \mathsf{hasMaker.Winery} \sqcap \exists \mathsf{locatedIn.Region} \\ \quad \sqcap \geq_1 \mathsf{madeFromGrape}) \\ \quad \sqcap \exists \mathsf{locatedIn}.\{\mathsf{BordeauxRegion}\} \\ \quad \sqcap \exists \mathsf{hasColor}.\{\mathsf{White}\} \sqcap \mathsf{WhiteWine\text{-}spec} \\ \quad \sqcap \mathsf{Wine\text{-}spec} \end{array} \right\}$$

$$\ldots$$

$$\mathit{\Pi}_1^{\mathsf{WB}} = \left\{ \begin{array}{l} \langle x, y_0 \rangle : \mathsf{hasBody},\ \langle x, y_1 \rangle : \mathsf{hasColor}, \\ \langle x, y_2 \rangle : \mathsf{hasFlavor},\ \langle x, y_3 \rangle : \mathsf{hasMaker}, \\ \langle x, y_4 \rangle : \mathsf{hasSugar},\ y_3 : \mathsf{Winery}, \\ \langle x, y_5 \rangle : \mathsf{locatedIn},\ y_5 : \mathsf{BordeauxRegion\text{-}spec}, \\ \langle x, y_6 \rangle : \mathsf{madeFromGrape},\ y_1 : \mathsf{White\text{-}spec} \end{array} \right\}$$

Figure 2. Unfolding concept WhiteBordeaux

As illustrated in Fig. 2, WhiteBordeaux is completely unfolded into its semantics-bearing signature, $\mathit{\Pi}_1^{\mathsf{WB}}$. In order to reduce the computational complexity, when unfolding we introduce primitive concepts to substitute nominal individuals. For instance, the fragment "$\ldots \sqcap \exists \mathsf{locatedIn}.\{\mathsf{BordeauxRegion}\} \sqcap \ldots$" of Bordeaux in Fig. 1 refers to instance BordeauxRegion of concept Region. We introduce primitive concept $\mathsf{BordeauxRegion} \sqsubseteq \mathsf{Region}$ accordingly and modify the above fragment into a set of equations as:

$$\begin{array}{rl} \mathsf{WhiteBordeaux} \doteq & \ldots \sqcap \exists\, \mathsf{locatedIn.BordeauxRegion} \ldots \\ \mathsf{BordeauxRegion} \doteq & \mathsf{Region} \sqcap \mathsf{BordeauxRegion\text{-}spec} \end{array}$$

Effectively, the resulting signature is composed by DL ABox assertions generated formally according to the conceptualisation. We regard them as semantics-preserving breakdowns of the constraints that are satisfied by any instances belonging to a concept. Fully breaking down into primitive concepts and properties, in some cases, is difficult to achieve. For instance, universal property value restrictions (UPVRs) can only be further expanded when in the same signature, there are elements defined over the quantified property. In Fig. 3(b), "$x : \forall \mathsf{hasSex.Male}$" is not unfolded due to the absence of "$\langle x, y \rangle : \mathsf{hasSex}$". It is different from "$\forall \mathsf{hasMaker.Winery}$" in Fig. 2 because of the presence of "$=_1 \mathsf{hasMaker}$" in the latter case. "$x : \forall \mathsf{hasMaker.Winery}$" would have been left unexpanded if otherwise.

It is possible that a concept has more than one signature, if it is defined as the disjunction of other concepts. Applying the non-deterministic unfolding rule of disjunction construct ($\sqcup$) results in alternative signatures, each of which captures a part of the intended meaning of the original concept. For instance, in Fig. 3, Human is unfolded into two different signatures.

$$\begin{array}{rl} \mathsf{Human} \doteq & \mathsf{Man} \sqcup \mathsf{Woman} \\ \mathsf{Man} \doteq & \ldots \sqcap \forall\,\mathsf{hasSex.Male} \sqcap \ldots \\ \mathsf{Woman} \doteq & \ldots \sqcap \forall\mathsf{hasSex.Female} \sqcap \ldots \\ & \text{(a)} \end{array} \qquad \begin{array}{rl} \Pi_1^{\mathsf{Man}} & = \{\ldots, x : \forall\mathsf{hasSex.Male}, \ldots\} \\ \Pi_1^{\mathsf{Human}} & = \{\ldots,\ x : \forall\mathsf{hasSex.Male},\ \ldots\} \\ \Pi_2^{\mathsf{Human}} & = \{\ldots,\ x : \forall\mathsf{hasSex.Female},\ \ldots\} \\ & \text{(b)} \end{array}$$

Figure 3. Rewriting BordeauxRegion

3.2 Weighting signature elements

Signaturing concepts can be seen as a process that gradually makes the semantic restrictions (expressed via concept constructs) explicit. As a result, each concept is associated with finite sets of formulae, being the primitive concepts, primitive properties and unexpanded universal property value restrictions. The initial feature vector is extracted from these formulae.

Π_i^{x} is subject to two "tuning" actions. Firstly, suffixes of X-spec concepts are removed. For instance, "y_1 : White-spec" in our example is rewritten as "y_1 : White" and is considered the same as those featured by "White". Secondly, residual UPVRs are simplified. The unexpanded UPVRs are dissected into properties and concepts. For instance, "x : ∀hasSex.Male" in Π_1^{Human} is expanded into "$\langle x, y\rangle$: *hasSex" and "y : *Male". The new signature elements generated therefrom are marked as optional to be differentiated from the others.

Obviously, different signature elements contribute differently to shaping the final semantics of concepts. In order to evaluate the significance of individual signature elements, a weighting schema is conceived. Basically, we consider concepts that contribute directly to the construction of others more important than those that impinge on others indirectly through properties or chains of properties. This is to emphasis on those elements that are semantically more significant than others. For instance, Bordeaux and WhiteWine are equally important in shaping the meaning of WhiteBordeaux (see Fig. 1) while BordeauxRegion is less significant than Wine w.r.t. Bordeaux as Bordeaux should be narratively interpreted as a special Wine first before narrowing it down to those that are produced in a particular geographic region. Our interests of Bordeaux, therefore, are arguably more in the former than the latter. In order to reflect such a difference in different restrictions and thus different signature elements of a concept, we introduce the weight adjusting coefficient β. β_e of signature element e is estimated as follows: we split an element as the *head* (e.g. "x" and "$\langle x, y_0\rangle$") and the *tail* (e.g. Region and hasBody) separated by a colon.

- if e is a first-class signature element headed by "x", $\beta_e = \omega^c$;
- if e is a first-class signature element headed by "$\langle x, y_i\rangle$", $\beta_e = \omega^p$;
- if e is a non-first-class signature element introduced in Π_i^{C} through a property P or a property chain $\mathsf{P}_1 \cdots \mathsf{P}_n$:

$$\beta_e = \beta_{\mathsf{P}} * \omega^c \qquad \text{or} \qquad \beta_e = \prod_i \beta_{\mathsf{P}_i} * \omega^c;$$

- if e is restricted by Negation, $\beta_e = -\beta_e$;

- if e is an optional element introduced through unfolding residual universal property value restrictions, $\beta_e = 0.5\,\beta_e$.

The weight of an arbitrary e then relies on two initial values ω^c and ω^p corresponding to the first-class element featured by a primitive concept and a primitive property respectively. The exact values of ω^c and ω^p are obtained by either i) assigning manually based on one's domain knowledge and expectation or ii) adopting the *tf-idf* weighting schema used in Information Retrieval (IR) with the assumption that an element appearing in every concept is less significant than those appearing only in a handful of concepts. Weight adjusting coefficients are memorised for each signature element. We then simplify the signatures to a set of terms/phrases. We extract the *bodies* of signature elements and apply Natural Language Processing (NLP) methods to tokenise and stem the *bodies* [11]. Those resulting terms or short phrases that appear more than once in a signature are collapsed into one with an aggregative weight as the sum of those corresponding to every occurrence, i.e. $\beta_e^{\text{total}} = \sum_i \beta_e^i$. When merging multiple appearance, we observe the disjointness between primitive concepts and primitive properties. For convenience, we denote the set of weighted terms/phrases obtained at this stage as γ.

3.3 Generalising Signatures

Terms or short phrases in γ are individual-specific, presenting interindividual variation. Therefore, in order to emerge a consensus among individuals each holding a different local ontology—possibly in different natural languages, it is necessary to situate the interpretation of those terms/phrases stemmed from concept signatures of different ontologies into the same background knowledge. A straightforward approach to drawing such information is treating web repositories as the source of common background knowledge. For instance in order to exploit the feature vectors of WhiteBordeaux, one has to understand all the words (i.e. Color, Flavor, etc.) appearing therein. Such an understanding should not rely on a particular ontology or vocabulary and should reflect the general human cognition of the words. In this paper, inspired by existing studies (e.g. [9]), we juxtapose terms/phrases against the titles of WIKIPEDIA articles and represent each concept as a vector of weighted WIKIPEDIA article titles, referred to as a *wiki-enhanced feature vector*. Note that hereafter we use bold font to denote the wiki-enhanced feature vectors.

WIKIPEDIA is a very appealing and probably the largest source of encyclopaedic knowledge. As a collaboratively edited document repository, it seems reasonable to conjecture that WIKIPEDIA presents most of the modelling (e.g. naming) variation that one will expect in independently developed ontologies. Meanwhile, the great diversity of wikipedians' background ensures that the contents published on WIKIPEDIA generally has better quality than other non-peer-reviewed web resources.

We assume that every conversation or an information exchange task focuses on a particular topic. For instance, when one asks others about "the taste of white Bordeaux", the topic of this incident is "Wine". If we denote the main WIKIPEDIA article as `t`, we compute the enhanced feature vector as in Fig. 4. In Step 1), when harvesting "virtual instances" from WIKIPEDIA, we utilise three different types of articles to pool a well targeted text corpus: i) the main article (Λ_{Main}) together with other articles that are the m neighbours of `t` (Λ_{Neigbr}), ii) the *List_of_**xxx*** page π and WIKIPEDIA articles directly linked to π and their m neighbours (Λ_{List}), and iii) the corresponding articles in other languages (Λ_{Lang}). WIKIPEDIA articles are retrieved from the following URL patterns:

URL for τ = *http://*⟨`ln`⟩*.wikipedia.org/wiki/*⟨`t`⟩
URL for π = *http://*⟨`ln`⟩*.wikipedia.org/wiki/List_of_*⟨***xxx***⟩

where ⟨`ln`⟩ is the language code (e.g. "`en`" for English and "`fr`" for French). When a particular topic does not have corresponding WIKIPEDIA entry, one has to manually specify the correct keywords to find the appropriate articles. Outbound links from Λ_{Main} are followed to retrieve articles that are closely related to Λ_{Main}. In many topics, WIKIPEDIA maintains the so-called "*List of*" pages, e.g. the "list of wine producing countries". Articles referred to from such collective pages are normally well-situated. For instance, "France" in "List of wine producing countries" leads to the WIKIPEDIA article titled "French Wine". Links to collective pages might also be available from within the main article. In our approach we gather both types of collective pages and their m-neighbours in Λ_{List}. Pooling all the WIKIPEDIA articles together, we have a well-targeted corpus of text documents, $\Lambda = \Lambda_{\text{Main}} \cup \Lambda_{\text{Neigbr}} \cup \Lambda_{\text{List}}$. Harvested WIKIPEDIA articles are parsed to remove WIKIPEDIA specific tags and commands. In Step 4), the SVD operation helps to reduce modelling variation and discover the latent semantics—unrevealed correlations between terms and articles. Similarly, SVD is performed again in Step 6) to optimise the weights of WIKIPEDIA articles w.r.t. concepts in $\mathcal{O}$.

If multiple natural languages are involved when establishing the consensus, **c** (see Step 8 in Fig. 4) needs to be translated into other languages. Although the cross-lingual capability of LSA has been investigated [8], we would rather avoid experimenting with such an approach and opt for a simple solution: when constructing $\mathfrak{M}_{\text{ac}}$ in Step 5), instead of the articles in the same language as the local ontology $\mathcal{O}$, we retrieve those corresponding WIKIPEDIA articles in the target languages as $\Lambda^{x}_{\text{Lang}}$, links to which normally present in the English articles. x will be decided by the context in which the consensus is to be established. For instance, if an individual is expected to communicate with French-speaking groups, $\Lambda^{\text{Fr}}_{\text{Lang}}$ is populated. One example of such articles is the French correspondence of "Wine" available at "*http://fr.wikipedia.org/wiki/Vin*". Subsequently, Step 6) and 7) are carried out based on the new matrix.

After signaturing and generalisation, an arbitrary concept $\mathsf{C} \in \mathcal{O}$ is associated with a WIKIPEDIA-enhanced and semantics-enriched feature vector, **c**, that represents the context within which C is to be interpreted. Note that the elements in the final enhanced feature vector is not specific to the naming habit of an individual ontology engineer. That is to say that **c** contains those terms that

1) harvest relevant WikipediA articles against $\mathtt{t}$ and pool them into Λ;
2) index every article $a \in \Lambda$ with a list of terms, l_a, and weight $t \in l_a$ based on *tf-idf* schema as $w_{\langle \mathrm{t}_i, \mathrm{a} \rangle}$;
3) construct the term-article matrix $\mathfrak{M}_{\mathrm{ta}}$ with WikipediA articles as columns and assign cell entries as

$$c_{ij} = \begin{cases} \beta_{kw} \cdot w_{\langle \mathrm{t}_i, \mathrm{a} \rangle}, & \text{if} \quad t_i = t_{\mathrm{kw}} \in \gamma; \\ 0, & \text{if} \quad t_i \notin \gamma. \end{cases}$$

4) perform SVD on $\mathfrak{M}_{\mathrm{ta}}$ and compute the correlation, $\sigma_{\langle \gamma, a \rangle}$, between γ and every indexed WikipediA article as the cosine of the angle between γ and article term vectors;
5) construct an article-concept matrix $\mathfrak{M}_{\mathrm{ac}}$ with WikipediA articles as rows, concepts from $\mathcal{O}$ as columns, and cell $c_{ij} = \sigma_{\langle \gamma, a \rangle}$;
6) condense the dimensionality of $\mathfrak{M}_{\mathrm{ac}}$ into $\hat{\mathfrak{M}}_{\mathrm{ac}}$ with SVD;
7) associate every $\mathsf{C} \in \mathcal{O}$ with a vector of WikipediA articles, denoted as $\mathbf{c}$.
8) (Optional) go to Step 5) and translate $\mathbf{c}$ into other languages based on $\Lambda^{\mathtt{x}}_{\mathrm{Lang}}$.

Figure 4. Algorithm for generating wiki-topic vectors

might not appear in $\mathcal{O}$ but are frequently correlated with parts of the restrictions of concept C. Such extras are pulled out from the referenced web repository, in our case WikipediA. This is consistent with the observation on inter-individual modelling variability: people tent to use different terms (e.g. synonyms and/or hypernyms) to refer to the same object [6].

4 Answering queries approximately

When trying to establish consensus, a key task is to find the local substitutes of those foreign concepts. In this occasion, we expect that all the individuals have already done their "homework" off-line and trained their article-concept matrix ($\hat{\mathfrak{M}}_{\mathrm{ac}}$) against the same web repository, namely WikipediA. Acquiring the similarity between a foreign concept C and the local ones requires the query handling peer to incorporate the foreign feature vector $\mathbf{c}$ of C into its local $\hat{\mathfrak{M}}_{\mathrm{ac}}$. Constructing $\hat{\mathfrak{M}}_{\mathrm{ac}}$ from scratch w.r.t. every received foreign feature vector is undesirable due to the high cost in recomputing SVD. A cheap solution is the so-called *folding-in* [6] operation. Let $\hat{\mathfrak{M}}_{\mathrm{ac}} = USV^{\mathrm{T}}$ be the reduced matrix of the query handler, p_{qh}. Every input $\mathbf{c}$ is projected onto the span of U as $(\mathbf{c})^{\mathrm{T}}US^{-1}$. Once feature vectors of the foreign concepts are put juxtaposed with those local ones, similarity measures such as the *cosine* angle can be employed to compare the foreign concepts against the local ones. Since both the query submiter p_0 and the query handler p_{qh} train and obtain their local *interpreation vectors* based on the same web repository, the folding-in approach is applicable. Alternatively, a more sophisticated and slightly more expensive approach, the SVD-updating method, can be used. Thus far, a few fast updating algorithms have been investigated. Our experience suggested that the one proposed in [15] outperformed many others.

4.1 LSA-based probability

There is a very little opportunity for individuals with independently developed ontologies to find perfect equivalences among themselves. In a majority of cases, we might need to rewrite Q_0 into Q_{qh}: "[...]*the list of* <u>*French*</u> <u>*WhiteWine*</u> *from* <u>*Boreaux*</u>" and attach to it similarity values. When more than one foreign concept is involved, there is not a plausible model to combine the multiple similarities. In order to tackle this daunting prospect of query rewriting with multiple foreign terms, we adopt the probability model of LSA [7] and a similarity approximation of probability [5]. If C is a concept appearing in Q_0 with $\mathbf{c}$, answers to Q_{qh} can be regarded as a faithful answer to Q_0 with a probability of $p\left(\mathsf{C} \mid \mathsf{C}'\right) = p\left(\mathbf{c} \mid \mathbf{c}',\, \mathbf{u}'_i\right)$ where C' is the local translation of C in $\mathcal{O}_{\mathrm{qh}}$ and $\mathbf{u}'_i$ are the left singular vectors computed based on $\mathcal{O}_{\mathrm{qh}}$ (see Equation 1).

$p\left(\mathbf{c} \mid \mathbf{c}',\, \mathbf{u}'_i\right)$ can be derived from two probabilities, $p\left(\mathbf{c}' \mid \mathbf{u}'_i\right)$ and $p\left(\mathbf{c} \mid \mathbf{u}'_i\right)$ [5]. $p\left(\mathbf{c} \mid \mathbf{u}'_i\right)$ is given by a probability variation of LSA model [7]. In this model, the author demonstrated that with some relaxations, an LSA-based similarity can be presented as the probability of a particular document (a column of a term-document matrix $\mathfrak{M}$) in the term-space (rows of $\mathfrak{M}$) based on document-document similarities [7]. This probability is computed as

$$p\left(\mathbf{c}' \mid \mathbf{u}'_i\right) = e^{(\mathbf{c}'\cdot\mathbf{u}'_1)^2+\cdots+(\mathbf{c}'\cdot\mathbf{u}'_k)^2} \,/\, Z_k \tag{2}$$

where Z_k is the normalisation constant[2]. Similarly, $p\left(\mathbf{c} \mid \mathbf{u}'_i\right)$ is computed. Deriving $p\left(\mathbf{c} \mid \mathbf{c}', \mathbf{u}'_i\right)$ from $p\left(\mathbf{c}' \mid \mathbf{u}'_i\right)$ and $p\left(\mathbf{c} \mid \mathbf{u}'_i\right)$, on the other hand, is not straightforward. In this paper we use a similarity emulation of probability [5],

$$p\left(\,\mathbf{c} \mid \mathbf{c}', \mathbf{u}'_i\right) = p\left(\,\mathbf{c}' \mid \mathbf{u}'_i\right)^{\epsilon}, \text{ where } \epsilon = \left(\frac{1-\mathbf{c}\cdot\mathbf{c}'}{1+\mathbf{c}\cdot\mathbf{c}'}\right)^{1-p\,(\mathbf{c}|\mathbf{u}'_i)} \tag{3}$$

where $\mathbf{c}\cdot\mathbf{c}'$ gives the similarity value of concepts C and C' computed from their respective feature vectors.

4.2 Fidelity of query rewriting

Since $\mathbf{u}'_i$ can be seen as a representation of p_{qh}'s local data, we define to what extent p_{qh} can understand p_0 as *fidelity* of the query-specific consensus between p_0 and p_{qh} as: $\mathit{fid}\left(\,Q_0 \mid Q^{\mathrm{Tr}}\right) = p\left(\,\mathbf{c}_1, \ldots, \mathbf{c}_n \mid \mathbf{c}'_1, \ldots, \mathbf{c}'_n, \mathbf{u}'_i\right)$, where $\mathbf{c}_i$ are the respective feature vectors of concepts in Q_0 while $\mathbf{c}'_i$ are feature vectors of the corresponding concepts in the translation Q^{Tr}. One possible way of continuing the derivation of the fidelity function would be approximating $p\left(\,\mathbf{c}_1, \ldots, \mathbf{c}_n \mid \mathbf{c}'_1, \ldots, \mathbf{c}'_n, \mathbf{u}'_i\right)$ with Equation 3. We collapse vectors $\mathbf{c}_1, \ldots, \mathbf{c}_n$ to their centroid $\mathbf{c}_c$ while $\mathbf{c}'_1, \ldots, \mathbf{c}'_n$ to $\mathbf{c}'_c$ and apply Equation 3. Hence, we have $\mathit{fid}\left(\,Q_0 \mid Q_i^{\mathrm{Tr}}\right) \approx p\left(\,\mathbf{c}_c \mid \mathbf{c}'_c, \mathbf{u}'_i\right)$. The computation of the fidelity requires only limited information from query submitting individuals: the actual queries and those feature vectors

---

[2] Please refer to [7] for details of how Z_k is defined.

associated with the queries. Meanwhile, computation is localised to each query handler and characterised by its local ontology.

The fidelity, $\mathit{fid}(Q_0 \mid Q^{\mathrm{Tr}})$, can be regarded as a criteria for judging the capability of query handlers. Let Q_i^{Tr} be the translation of Q_0 in the local ontology $\mathcal{O}_i$ of p_i, the query rewriting fidelity reflects how good an answer to Q_0 drawn from Q_i^{Tr} is and thus how well p_i can handle Q_0 based on its local knowledge. The higher the fidelity is, the greater chance the original query is satisfactorily answered by the query handler. Such a probabilistic model paves the way for a ranking mechanism of candidate query handlers. A feasible scenario could be that every p_i in a group G estimates its own fidelity based on the local translation of Q_i^{Tr}, the original Q_0 and the incoming foreign feature vectors. A group coordinator can then allocate the query handling task to the one with the highest estimated fidelity value.

5 Concluding Remarks

In a loosely regulated environment, global consensus may not be enforced. It is more likely that each individual or a small group of peers maintains a local ontology. For an outsider to query an established group and retrieve useful information, conventional ontology mapping techniques (e.g. discussed in [13]) are not sufficient. Dynamic and on-the-fly methods for establishing on demand consensuses becomes more desirable. In the meantime, exact equivalences have given way to those less perfect ones. In this paper, we propose a mechanism to emerge and exploit imperfect consensuses among heterogeneous data holders. We situate an alignment task in the background knowledge drawn from public web repositories, e.g. WIKIPEDIA. We combine the strength of both representation formalisms and collaboratively created web repositories when discovering semantics. Our semantic alignment approach also gives birth to a probabilistic model to evaluate approximate query answering by identifying the most appropriate individual to handle the query and ranking candidate answers returned from the chosen one. There are two sources of complexity w.r.t. the proposed approach: i) signaturing concepts using DL transformation rules (NEXPTIME for expressive DLs) and ii) SVD dimensionality reduction. We, however, would like to make the following argument. On the one hand, DL-based reasoning is very expensive in itself. Any methods aiming to manipulate semantics embedded therein, therefore, have to pay the price of high complexity. Meanwhile, high complexity is normally associated with a set of “culprit” DL constructs. Introducing substituting constructs and/or alternative modelling techniques might help us to find a route around the complexity issue. Meanwhile, construct transformation rules are implemented and optimised in many DL systems and the empirical evaluation has confirmed the performance of such systems in tackling real-life ontologies [1]. On the other hand, LSA has been extensively studied in IR. We share the optimistic expectation with those dedicated researchers regarding the practical value of LSA/SVD in latent semantics discovery.

Finalising the implementation and evaluating it against real-life scenarios are our immediate future work. The evaluation will focus on the following aspects: i) the applicability of Wikipedia w.r.t. topics of different popularity, ii) the scalability of the signaturing and weighting algorithms and their applicability in the current landscape of ontologies, and iii) the performance of the proposed probability mechanism for query handling.

Acknowledgements

This work is supported under the OpenKnowledge STREP projects funded by EU Framework 6 under Grant numbers IST-FP6-027253.

References

1. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications.* Cambridge University Press, 2003.
2. R. Benjamins, J. Euzenat, N. Noy, P. Shvaiko, and M. Stuckenschmidt, H. adn Uschold, editors. *International Workshop on Ontology Matching*, 2006.
3. T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, pages 28–37, 2001.
4. V. Bhat, T. Oates, V. Shanbhag, and C. Nicholas. Finding aliases on the web using latent semantic analysis. *Data & Knowledge Engineering*, 49(2):129–143, 2004.
5. S. Blok, D. Medin, and D. Osherson. Probability from similarity. In P. Doherty, J. McCarthy, and M. Williams, editors, *AAAI Spring Symposium on Logical Formalization of Commonsense Reasoning*, pages 36–42. AAAI Press, 2003.
6. S. Deerwester, S.T. Dumais, T.K. Landauer, G.W. Furnas, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.
7. C. Ding. A probabilistic model for latent semantic indexing. *JASIST*, 56(6):597–608, 2005.
8. S. Dumais, T. Landauer, and M. Littman. Automatic cross-linguistic information retrieval using latent semantic indexing. In *AAAI Spring Symposium on Cross Language Information Retrieval*, 1997.
9. E. Gabrilovich and S. Markovitch. Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis. In *Proceedings of the 20th IJCAI*, pages 1606–1611, 2007.
10. I. Horrocks and U. Sattler. A tableaux decision procedure for $\mathcal{SHOIQ}$. In *Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI 2005)*, 2005.
11. R. Korfhage. *Information storage and retrieval.* Wiley Computer Publishing, 1997.
12. D. L. McGuinness and F. van Harmelen. *OWL Web Ontology Language Overview.* W3C, 2003.
13. E. Rahm and P.A. Bernstein. A survey of approaches to automatic schema matching. *The VLDB Journal*, 10:334–350, 2001.
14. L. Steels and F. Kaplan. Bootstrapping grounded word semantics. In T. Briscoe, editor, *Linguistic Evolution through Language Acquisition: Formal and Computational Models*, chapter 3. Cambridge University Press, 1999.
15. H. Zha and H. Simon. On updating problems in latent semantic indexing. *SIAM J. Sci. Comput.*, 21(2):782–791, 1999.

In Vitro Study of Mapping Method Interactions in a Name Pattern Landscape

Ondřej Šváb, Vojtěch Svátek

University of Economics, Prague, Dep. Information and Knowledge Engineering,
Winston Churchill Sq. 4, 130 67 Praha 3, Prague, Czech Republic
`svabo@vse.cz, svatek@vse.cz`

Abstract. Ontology mapping tools typically employ combinations of methods, the mutual effects of which deserve study. We propose an approach to analysis of such combinations using synthetic ontologies. Initial experiments have been carried out for two string-based and one graph-based method. Most important target of the study was the impact of name patterns over taxonomy paths on the mapping results.

1 Introduction

Ideal taxonomy of ontology mapping methods clearly distinguishes different classes of methods such as string-based, graph-based or semantics-based, however most state-of-the-art systems nowadays exploit multiple approaches and combine them into (usually ad-hoc) workflows. Their results are then, e.g. within comparative tests such as OAEI[1], presented in their final form only; the contribution and mutual interaction of different constituent methods is not analysed.

In our current work we focused on the two classes of methods that are at the same time generally usable (unlike semantic methods, which depend on the presence of logical constraints in ontologies) and self-contained (unlike e.g. thesauri-based method, which depend on external thesauri): *string-based* and *graph-based.* In the first, proof-of-concept, series of experiments, we restricted our analysis to a single representative of graph-based methods plus two common string-based methods that can be used for its initialisation.

At the same time, we wanted to investigate the impact of *name patterns* that frequently follow the specialisation paths in ontologies[2]. The most typical patterns amount to the more specific (child) entity being right-, left- or internal extension of the more general (parent) entity. Clearly, the presence of such patterns could have influence on the effect of individual string matchers and, consequently, on the effect of the whole string-matching + graph-matching pipeline.

However, as we wanted to 'play' with different proportions and distributions of name patterns in the ontologies, as well as with different structural properties

---

[1] `http://oaei.ontologymatching.org/`

[2] We verified this assumption through informal analysis of about 50 ontologies 'randomly' picked from three popular ontology repositories.

of ontologies themselves, it was not convenient for us to stick to existing (real) ontologies. Instead, we made recourse to an 'in vitro' approach. We developed a *generator* that enabled us to automatically build 'ontologies' (syntactic OWL artefact) with customised structure and presence of name patterns as previously observed with real ontologies. Similarly to the 'calibration' collection of OAEI[3] we then derived the second model to be matched using pre-defined 'distortions'.

As the generated models lack any meaning and the 'distortions' are not necessarily identity-preserving, there cannot be any *reference mapping*. Our experimental results thus only consist in numerous observations about the strengths of mappings for different constellations of generated models and mapping methods, which we then attempt to interpret and generalise to some degree.

The structure of the paper is as follows. Section 2 very briefly characterises the mapping methods we took as testing material for our initial study. Section 3 describes the process of generating the synthetic 'ontologies' (as preliminary phase of our experiments). Section 4 is devoted to the actual mapping experiments and to case-by-case interpretation of their results. Section 5 contains some generalising discussion. Finally, section 6 reviews related work and section 7 outlines some future work.

2 Underlying Mapping Methods

The graph-based matching method, *similarity flooding* (SF), has been successfully tested on several mapping problems, especially those related to database schemata. The SF algorithm takes two models (database schemata, ontologies or the like) as graphs and transform them into a so-called *connectivity graph*, where each node (map pair) consists of two nodes from original models (one from each) that are connected with the same edges in their original models. Connectivity graph is then transformed into *propagation graph* where *propagation coefficients* are induced, expressing how well the similarity of a given map pair propagates to its neighbours and back [3].

As representatives of string-based methods we took two methods that merely differ in their granularity: the *Jaccard* method is token-based[4], while the *char-Jaccard* method is character-based (hence 'less semantic'). Both methods compute the similarity as the ratio of the intersection and union of the token/character sets of both entities.

---

[3] `http://oaei.ontologymatching.org/2007/benchmarks/`

[4] We also, originally, wanted to use the default (also token-based) string matcher contained in the SF package itself. It was quite interesting for our purposes, as it is (allegedly) based on common prefixes/suffixes of concept names—a feature relevant wrt. the name patterns we considered—but it produced apparently wrong results.

3 Generation of Synthetic Ontologies with Name Patterns

In order to swiftly obtain artifacts (in OWL language) syntactically similar to real ontologies, we developed a 'generator of ontologies'. The process of creating such an 'ontology' is guided by several *parameters*, such as the total number of classes, the length of the longest path, the maximum number of subclasses of one class etc. Parts of the process are 'random'[5], which concern the structure, labels (as 'reasonably long' sequences of random characters avoiding 'unpronounceable' sequences of more than two consonants), token delimiters, and, finally, name patterns. Currently we consider three patterns, all describing the relationship between the label of a parent and child in the taxonomy:

- (the most obvious) pattern 1 consists in left-hand side expansion of label, e.g. 'article' → 'scientificArticle'
- pattern 2 represents right-hand side expansion of label, e.g. 'IPR' → 'IPRTransfer' (example from the `ipronto.owl` from OntoSelect collection (`http://olp.dfki.de/ontoselect/`)),
- pattern 3 represents inward expansion of label, e.g. 'airMission' → 'airMovementMission' (example from the `ATO_Mission_Models.owl` ontology from DAML collection (`http://www.daml.org/ontologies/`).

In order to obtain pairs of ontologies for the mapping task, the generation process is followed with a *distortion* phase[6] where some classes are removed, some classes are added and/or there are changes in name patterns. In the section 4, there is list of all distortions that we tried.

Our generator enables us to prepare different ontologies with different features. For our current testing purposes we chose one synthetic ontology from a couple of diverse generated ontologies, see Figure 1. The chosen ontology has 24 classes, six of them being root classes. Only two root classes have children: `zutu` (for the sake of brevity we will shortcut it with Z) and `jonopizasfa` (J). J has two children and does not belong to any pattern. Z has three children and belongs to pattern 1. The occurence of this pattern among the children of Z is absolute (frequency is 100%), while in the case of its subclass `cexilipluZutu` the frequency of pattern 1 is 40%. J has subclass `dinicipi` (D) that has pattern 1 with 100% occurences in its children. In the experiments below, we took the classes J, Z and D as 'spy points' for observing the mapping results.

Although our ontology is synthetic, its structures are analogous to those of real ontologies. In Fig. 2, there is snippet of real ontology (`Communications.owl` from DAML collection) corresponding to J and in Fig. 3, there is snippet of the same real ontology corresponding to Z. Obviously, an ontology of this size could have been created purely by human effort, e.g. by manually tweaking

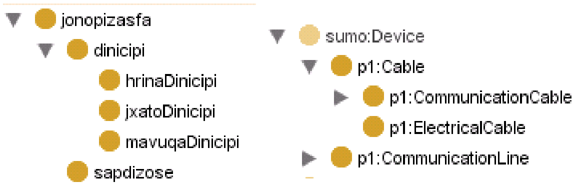
---

[5] I.e. governed by a sequence of pseudorandom numbers.

[6] In our first experiments, this phase was partly manual. Namely, for the current, rather small, testing model, it was much easier to do the small tweaks manually than to redesign the automated generation tool.

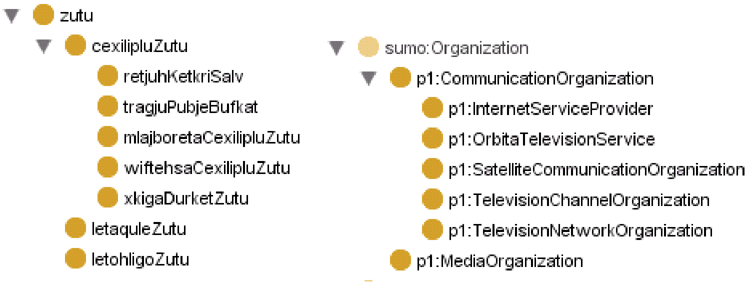
Fig. 1. Base synthetic ontology used in our experiments

real ontologies (which we actually did in prior research). We however anticipate further experiments where automatic generation would pay off, and also did not want to be distracted, during the result interpretation phase, by the human semantics of concrete labels.



(a) synthetic part (b) corresponding real part

Fig. 2. Part of real ontology corresponding to part of synthetic ontology



(a) synthetic part (b) corresponding real part

Fig. 3. Part of real ontology corresponding to part of synthetic ontology

4 Mapping Experiments

In our experiments we always tried to map the base synthetic ontology on some of its distortions. We observed the similarity values for three mappings: J with J (from the base ontology vs. from its distortion, in turn), Z with Z, and D with D. Their similarity values were computed with the similarity flooding algorithm; for initial mapping once the Jaccard method was used and once the charJaccard method.

We can roughly divide the distortions employed to the four groups below (particular distortions can belong to more than one group). They do not of course exhaustively represent all possibilities, nor are they very systematic; they rather reflect ad hoc 'polls' effected during a feedback-driven analysis process.

- (1) changes in structure, ie. adding/removing intermediate classes:
    - deletion of three root classes, except J and Z,
    - addition of some subclasses to root classes, except J and Z,
    - direct subclass of J was deleted,
    - direct subclass of J was deleted and all subclasses of D were also deleted,
    - deletion of direct subclasses of class D or Z,
    - all subclasses of D were relocated to root level,
- (2) changes in labels of classes:
    - two of three subclasses of D were relabelled in order to be different than original labels,
- (3) changes in delimiters in whole ontology (ceteris paribus); they are not considered below as they have no impact on similarity flooding,
- (4) changes in pattern, ie. altering the frequency of occurence of the pattern:
    - deletion of subclasses, that (does not) belong to pattern at class Z,
    - change pattern from pattern 1 to pattern 2,
    - relabeling all subclasses of Z so as to remove pattern.

Now we will present and discuss several 'interesting' situations (cases) with regard to the strengths of mappings for different constellations. Each case consists of textual description of distortions with regard to the original synthetic ontology (see Fig. 1), table with similarity values (charJaccard method was used for value in the first column, while Jaccard method was used for computation of value in the second column). All the cases are compared with the situation (initial case—Case #0) where the original synthetic ontology is mapped to itself: the absolute difference wrt. the initial case is always in parentheses. Some cases are additionally illustrated by figures. Finally, some tentative interpretations of observed results are formulated.

Case #1, see Fig. 4
Distortion: three root classes were deleted except J and Z.
Interpretation: Deletion of root classes leads to smaller connectivity graph, hence the effect of SF is smaller (higher similarity).

Case #2, see Fig. 5
Distortion: some classes were added to root classes, except to J and Z.
Interpretation: Addition of subclasses to root classes leads to larger connectivity graph, hence the effect of SF is greater (lower similarity).

Case #3, see Fig. 6 and 7 In this case we compare two situations - deletion of occurences of pattern or not.
Distortion: (a) three subclasses of `cexilipluZutu` were deleted, which had string Z in their labels (in Fig. 6 on the left-hand side); (b) two subclasses

mapping	charJaccard	Jaccard
J with J	0.28(+0.02)	0.44(+0.01)
Z with Z	0.52(+0.03)	0.82(+0.03)
D with D	0.49(+0.03)	0.77(+0.02)

(a) similarity table

(b) distortion

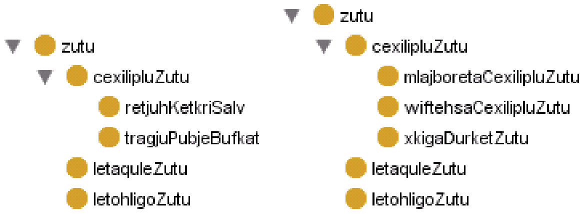
Fig. 4. Case #1

mapping	charJaccard	Jaccard
J with J	0.26(+0.00)	0.42(-0.01)
Z with Z	0.48(-0.01)	0.79(+0.00)
D with D	0.45(-0.01)	0.75(+0.00)

(a) similarity table

(b) distortion

Fig. 5. Case #2



(a) without Z (b) with Z

Fig. 6. Distortions for case #3

mapping	charJaccard	Jaccard
J with J	0.26(+0.00)	0.43(+0.00)
Z with Z	0.46(-0.03)	0.74(-0.05)
D with D	0.46(+0.00)	0.75(+0.00)

(a) similarity table without Z

mapping	charJaccard	Jaccard
J with J	0.26(+0.00)	0.43(+0.00)
Z with Z	0.47(-0.02)	0.77(-0.02)
D with D	0.46(+0.00)	0.75(+0.00)

(b) similarity table with Z

Fig. 7. Similarity tables for case #3

of `cexilipluZutu` were deleted, which had not string Z in their labels (in Fig. 6 on the right-hand side).
Interpretation: In both cases similarity values for mapping of classes Z decrease, but in the case on the right-hand side the value is higher. Higher values for the Z on the right-hand side situation could be strange because of greater connectivity graph. In this case (right-hand side) higher values can be explained with higher initial similarity values going from both string methods, because a name pattern is present.

Case #4, see Fig. 8
Distortion: Pattern 1 was changed to pattern 2 and one new subclass of D was added.
Interpretation: This change leads to greater connectivity graph, however higher similarity value is the effect of pattern at D.

Case #5, see Fig. 9
Distortion: All subclasses of D were relocated to root level.
Interpretation: It is similar example as in case #4 but now there is no pattern (by relocating classes pattern at D was aborted), thus greater connectivity graph leads to lower similarities at J and Z (similar to cases #1, and #2).

Case #6, see Fig. 10
Distortion: Z does not belong to pattern.

mapping	charJaccard	Jaccard
J with J	0.27(+0.01)	0.44(+0.01)
Z with Z	0.49(+0.00)	0.79(+0.00)
D with D	0.55(+0.09)	0.86(+0.11)

(a) similarity table

- owl:Thing
- jonopizasfa
  - dinicipi
    - dinicipiHrina
    - dinicipiJxato
    - dinicipiMavuqa
    - dinicipiRifnobiga
  - sapdizose

(b) distortion

Fig. 8. case #4

mapping	charJaccard	Jaccard
J with J	0.20(-0.06)	0.36(-0.07)
Z with Z	0.40(-0.09)	0.77(-0.02)
D with D	0.08(-0.38)	0.16(-0.59)

(a) similarity table

- owl:Thing
- hrinaDinicipi
- jonopizasfa
  - dinicipi
  - sapdizose
- jxatoDinicipi
- mavuqaDinicipi
- mfibaleja
- mjeriso
- xqafmopaweza
- xusquhzajenzo
- zutu

(b) distortion

Fig. 9. case #5

mapping	charJaccard	Jaccard
J with J	0.27(+0.01)	0.45(+0.02)
Z with Z	0.24(-0.25)	0.17(-0.62)
D with D	0.47(+0.01)	0.79(+0.04)

(a) similarity table

(b) distortion

Fig. 10. case #6

Interpretation: Lowering effect on Z because now there is no pattern. Higher effect (greater difference) in the case of Jaccard is expectable, because violation of prefixes and suffixes is more probable.

5 Discussion of Results

Our mapping experiments are in general consistent with the expectation that the SF algorithm computes lower similarity values for greater connectivity graphs. Greater connectivity graph means more candidate mapping pairs overall, and therefore there is provision for a high number of lower similarity values in neighbourhood of the map pair in question (eg. J with J), see cases #1, #2, and #5. In such a situation, the initially high similarity of the pair in question is largely redistributed to its neighbours in the connectivity graph.

Our experiments are also consistent with our hypothesis that the presence of name pattern leads to higher similarity values because of higher initial values going from both string based methods despite the size of connectivity graph (case #3, #4, and #6). This is produced by a relatively high degree of match among the neighbouring concepts pertaining to the same name pattern.

Cases #5, and #6 are also consistent with our hypothesis that the impact of name patterns on similarity values based on token-based string method Jaccard are higher than similarity values based on character-based string method charJaccard (differences between values in two column at appropriate tables). This naturally follows from the fact that patterns are represented as occurrences of same token within the labels of different neighbouring concepts.

Admittedly, the interpretations of current experiments suffer from our so far unsystematic, spontaneous approach to designing the experimental cases. However, we assume that the initial set of experiments will help creating such a systematic set of mapping experiments in the future.

6 Related Work

According to four (intertwined) main aspects of our work, we can distinguish among four areas of related research.

The effect of *combining string-based and graph-based mapping methods* was examined in several projects, e.g. [3, 1], however, essentially, in the sense of measuring the numerical improvement of quality (for a graph method applied on initial string-based mapping) using some kind of reference alignment. In contrast, we focus on detailed phenomena arising when applying the methods together, for concrete entities within their structural context, and without reference alignment. Additionally, in our own previous work we studied the interdependencies of several string-based methods alone, using probabilistic dependency models, namely Bayesian networks [4].

The analysis of *name patterns* along the ontology paths was carried out in numerous projects aiming at transformation of shallow ontologies such as thesauri or web directory headings into deeper ones (a recent representative is e.g. [2]). Our subject of study is however not transformation of the ontology but its mapping to another one; as this goal is less ambitious, our analysis is also much less thorough to date than the approaches mentioned.

The generation of *artificial ontologies* based on the features of real ones was envisaged by Tempich&Volz [6], for the purpose of creating benchmarks for semantic web reasoners. The difference of our generator lays in the focus on name patterns and local structural contexts rather than in large-scale statistical properties of ontologies. In addition, paper [6] does not contain any details about the actual generation process. As mentioned above, the distortion of original ontology in view of obtaining the two sides for matching experiments was carried out by J. Euzenat for the purpose of OAEI[7]. Our notion of distortion is at the same time finer grained and more restricted in its types. We also do not a priori assume that matching the original entity with its counterpart from the distorted model is 'correct' (even if its context and/or label are changed).

Finally, the problem of evaluating (or, rather, getting useful information from) ontology mapping results *without reference model* has also been investigated in our prior work [5]. We are not aware of any other major effort in this direction.

7 Conclusions and Future Work

We carried out a set of experiments with synthetic OWL artifacts ('ontologies' in syntactical sense) that should allow us get deeper insight into the behaviour of certain chosen mapping methods, applied in combination (in this case, sequence). An important feature we enforced to the synthetic models was the presence (to varying degrees) of name patterns that mimic those present in real ontologies. Rather than the concrete experimental results by themselves, the primary contribution of this initial study is meant to be the proof of concept of the whole

---

[7] `http://oaei.ontologymatching.org/2007/benchmarks/`

experimental setting, consisting in the 'ontology' generator (incl. the distortion component), interface to the scrutinised mapping methods, plus some (not yet too systematic) principles of human interpretation and generalisation of results.

Most imminent future work will consist in

- Doing a similar excercise for some other pairs of concepts in the structure (and possibly in differently generated 'ontologies'), including pairs that don't initially have the same label. Future experiments of this sort will be more systematic, possibly adhering to a clean formal model.
- Improvements of the generator, reflecting more sophisticated 'pattern' aspects (not confined to 'name' ones) of real OWL ontologies, and replacing the manual parts of the distortion phase with fully automatic ones (thus allowing for building large models if needed).
- Tests, possibly on the same data, of other string-based and graph-based methods. The latter could be e.g. probabilistic methods [1] or methods based on neighbour feature vector [7], provided we get access to their implementations.

In long term, we assume that the accumulated experience from testing current mapping approaches wrt. patterns in ontologies will help us design new mapping algorithms explicitly accounting for such patterns.

The research was partially supported by the IGA VSE grants no.12/06 "Integration of approaches to ontological engineering: design patterns, mapping and mining", no.20/07 "Combination and comparison of ontology mapping methods and systems", and by the Knowledge Web Network of Excellence (IST FP6-507482).

References

1. Doshi P., Thomas C.: Inexact Matching of Ontology Graphs Using Expectation-Maximization. In: Proc. AAAI 2006.
2. Hepp M., de Bruijn J.: GenTax: A Generic Methodology for Deriving OWL and RDF-S Ontologies from Hierarchical Classifications, Thesauri, and Inconsistent Taxonomies. In: Proc. ESWC 2007.
3. Melnik S., Garcia-Molina H., Rahm E.: Similarity Flooding: A Versatile Graph Matching Algorithm and its Application to Schema Matching. In: Proc. 18th ICDE, 2002.
4. Šváb O., Svátek V.: Combining Ontology Mapping Methods Using Bayesian Networks. In: Proc. OM workshop at ISWC 2006.
5. Šváb O., Svátek V., Stuckenschmidt H.: A Study in Empirical and 'Casuistic' Analysis of Ontology Mapping Results. In: Proc. ESWC 2007.
6. Tempich C., Volz R.: Towards a benchmark for Semantic Web reasoners - an analysis of the DAML ontology library. In: EON Workshop at ISWC 2003.
7. Udrea O., Getoor L., Miller R. J.: Leveraging Data and Structure in Ontology Integration. In: Proceedings of ACM-SIGMOD 2007.

Results of the Ontology Alignment Evaluation Initiative 2007 ⋆

Jérôme Euzenat[1], Antoine Isaac[2], Christian Meilicke[3], Pavel Shvaiko[4], Heiner Stuckenschmidt[3], Ondřej Šváb[5], Vojtěch Svátek[5], Willem Robert van Hage[2], and Mikalai Yatskevich[4]

[1] INRIA Rhône-Alpes & LIG, Montbonnot, France
`jerome.euzenat@inrialpes.fr`
[2] Vrije Universiteit Amsterdam, The Netherlands
`{wrvhage,aisaac}@few.vu.nl`
[3] University of Mannheim, Mannheim, Germany
`{heiner,christian}@informatik.uni-mannheim.de`
[4] University of Trento, Povo, Trento, Italy
`{pavel,yatskevi}@dit.unitn.it`
[5] University of Economics, Prague, Czech Republic
`{svabo,svatek}@vse.cz`

Abstract. Ontology matching consists of finding correspondences between ontology entities. OAEI campaigns aim at comparing ontology matching systems on precisely defined test sets. Test sets can use ontologies of different nature (from expressive OWL ontologies to simple directories) and use different modalities (e.g., blind evaluation, open evaluation, consensus). OAEI-2007 builds over previous campaigns by having 4 tracks with 7 test sets followed by 18 participants. This is a major increase in the number of participants compared to the previous years. Moreover, the evaluation results demonstrate that more participants are at the forefront. The final and official results of the campaign are those published on the OAEI web site.

1 Introduction

The Ontology Alignment Evaluation Initiative[1] (OAEI) is a coordinated international initiative that organizes the evaluation of the increasing number of ontology matching systems. The main goal of the Ontology Alignment Evaluation Initiative is to be able to compare systems and algorithms on the same basis and to allow anyone for drawing conclusions about the best matching strategies. Our ambition is that from such evaluations, tool developers can learn and improve their systems. The OAEI campaign provides the evaluation of matching systems on consensus test cases.

Two first events were organized in 2004: (i) the Information Interpretation and Integration Conference (I3CON) held at the NIST Performance Metrics for Intelligent

---

⋆ This paper improves on the "First results" initially published in the on-site proceedings of the ISWC+ASWC workshop on Ontology Matching (OM-2007). The only official results of the campaign, however, are on the OAEI web site.

[1] `http://oaei.ontologymatching.org`

Systems (PerMIS) workshop and (ii) the Ontology Alignment Contest held at the Evaluation of Ontology-based Tools (EON) workshop of the annual International Semantic Web Conference (ISWC) [13]. Then, unique OAEI campaigns occurred in 2005 at the workshop on Integrating Ontologies held in conjunction with the International Conference on Knowledge Capture (K-Cap) [2] and in 2006 at the first Ontology Matching workshop collocated with ISWC [7]. Finally, in 2007, OAEI results are presented at the second Ontology Matching workshop collocated with ISWC+ASWC, in Busan, South Korea.

We have continued last year's trend by having a large variety of test cases that emphasize different aspects of ontology matching. We have kept particular modalities of evaluation for some of these test cases, such as a consensus building workshop.

This paper serves as an introduction to the evaluation campaign of 2007 and to the results provided in the following papers. The remainder of the paper is organized as follows. In Section 2 we present the overall testing methodology that has been used. Sections 3-9 discuss in turn the settings and the results of each of the test cases. Section 10 overviews lessons learned based on the campaign. Finally, Section 11 outlines future plans and Section 12 concludes.

2 General methodology

We present the general methodology for the 2007 campaign as it was defined and report on its execution.

2.1 Tracks and test cases

This year's campaign has consisted of four tracks gathering seven data sets and different evaluation modalities.

The benchmark track (§3): Like in previous campaigns, systematic benchmark series have been produced. The goal of this benchmark series is to identify the areas in which each matching algorithm is strong or weak. The test is based on one particular ontology dedicated to the very narrow domain of bibliography and a number of alternative ontologies of the same domain for which alignments are provided.

The expressive ontologies track. Anatomy (§4): The anatomy real world case deals with matching the Adult Mouse Anatomy (2.744 classes) and the NCI Thesaurus (3.304 classes) describing the human anatomy.

The directories and thesauri track:

Directory (§5): The directory real world case consists of matching web site directories (like the Open directory or Yahoo's). It has more than four thousand elementary tests.

Food (§6): Two SKOS thesauri about food have to be matched using relations from the SKOS Mapping vocabulary. Samples of the results are evaluated by domain experts.

Environment (§7): Three SKOS thesauri about the environment have to be matched (A-B, B-C, C-A) using relations from the SKOS Mapping vocabulary. Samples of the results are evaluated by domain experts.

Library (§8): Two SKOS thesauri about books have to be matched using relations from the SKOS Mapping vocabulary. Samples of the results are evaluated by domain experts. In addition, we run application dependent evaluation.

The conference track and consensus workshop (§9): Participants were asked to freely explore a collection of conference organization ontologies (the domain being well understandable for every researcher). This effort was expected to materialize in usual alignments as well as in interesting individual correspondences ("nuggets"), aggregated statistical observations and/or implicit design patterns. There was no a priori reference alignment. Organizers of this track offered manual a posteriori evaluation of results. For a selected sample of correspondences, consensus was sought at the workshop and the process of its reaching was tracked and recorded.

Table 1 summarizes the variation in the results expected from these tests.

test	language	relations	confidence	modalities
benchmark	OWL	=	[0 1]	open
anatomy	OWL	=	1	blind
directory	OWL	=	1	blind
food	SKOS, OWL	narrow-, exact-, broadMatch	1	blind+external
environment	SKOS, OWL	narrow-, exact-, broadMatch	1	blind+external
library	SKOS, OWL	narrow-, exact-, broad-, relatedMatch	1	blind+external
conference	OWL-DL	=, $\leq$	1	blind+consensual

Table 1. Characteristics of test cases (open evaluation is made with already published reference alignments, blind evaluation is made by organizers from reference alignments unknown to the participants, consensual evaluation is obtained by reaching consensus over the found results).

2.2 Preparatory phase

The ontologies and (where applicable) the alignments of the evaluation have been provided in advance during the period between May 15th and June 15th, 2007. This gave potential participants the occasion to send observations, bug corrections, remarks and other test cases to the organizers. The goal of this preparatory period is to ensure that the delivered tests make sense to the participants. The final test base was released on July 2nd. The tests did not evolve after this period.

2.3 Execution phase

During the execution phase the participants used their systems to automatically match the ontologies from the test cases. Participants have been asked to use one algorithm and the same set of parameters for all tests in all tracks. It is fair to select the set of parameters that provide the best results (for the tests where results are known). Beside parameters, the input of the algorithms must be the two ontologies to be matched and any general purpose resource available to everyone, i.e., no resource especially designed for the test. In particular, the participants should not use the data (ontologies and reference alignments) from other test sets to help their algorithms.

In most cases, ontologies are described in OWL-DL and serialized in the RDF/XML format. The expected alignments are provided in the Alignment format expressed in RDF/XML [6]. Participants also provided the papers that are published hereafter and a link to their systems and their configuration parameters.

2.4 Evaluation phase

The organizers have evaluated the results of the algorithms used by the participants and provided comparisons on the basis of the provided alignments.

In order to ensure that it is possible to process automatically the provided results, the participants have been requested to provide (preliminary) results by September 3rd. In the case of blind tests only the organizers did the evaluation with regard to the withheld reference alignments.

The standard evaluation measures are precision and recall computed against the reference alignments. For the matter of aggregation of the measures we use weighted harmonic means (weights being the size of the true positives). This clearly helps in the case of empty alignments. Another technique that has been used is the computation of precision/recall graphs so it was advised that participants provide their results with a weight to each correspondence they found. New measures addressing some limitations of precision and recall have also been used for testing purposes as well as measures compensating for the lack of complete reference alignments.

In addition, the Library test case featured an application-specific evaluation and a consensus workshop has been held for evaluating particular correspondences.

2.5 Comments on the execution

This year again, we had more participants than in previous years: 4 in 2004, 7 in 2005, 10 in 2006, and 18 in 2007. We can also observe a common trend: participants who keep on developing their systems improve the evaluation results over time.

We have had not enough time to validate the results which had been provided by the participants, but we scrutinized some of the results leading to improvements for some participants and retraction from others. Validating these results has proved feasible in the previous years so we plan to do it again in future.

We summarize the list of participants in Table 2. Similar to last year not all participants provided results for all tests. They usually did those which are easier to run, such as benchmark, directory and conference. The variety of tests and the short time given to provide results have certainly prevented participants from considering more tests.

There are two groups of systems: those which can deal with large taxonomies (food, environment, library) and those which cannot. The two new test cases (environment and library) are those with the least number of participants. This can be explained by the size of ontologies or their novelty: there are no past results to compare with.

This year we have been able to devote more time to performing these tests and evaluation (three full months). This is certainly still too little especially during the summer

Software	confidence	benchmark	anatomy	directory	food	environment	library	conference
AgreementMaker	√		√					
AOAS	√		√					
ASMOV	√	√	√	√				√
DSSim	√	√	√	√	√	√	√	
Falcon-AO v0.7	√	√	√	√	√	√	√	√
Lily		√	√	√				√
OLA2	√	√		√				√
OntoDNA		√		√				√
OWL-CM		√						
Prior+	√	√	√	√	√			
RiMOM	√	√	√	√	√			
SAMBO		√	√					
SCARLET					√			
SEMA		√						√
Silas							√	
SODA	√	√						
TaxoMap	√	√	√					
X-SOM	√	√	√	√	√			
Total=18	10	14	11	9	6	2	3	6

Table 2. Participants and the state of their submissions. Confidence stands for the type of result returned by a system: it is ticked when the confidence has been measured as non boolean value.

period allocated for that. However, it seems that we have avoided the rush of previous years.

The summary of the results track by track is provided in the following six sections.

3 Benchmark

The goal of the benchmark tests is to provide a stable and detailed picture of each algorithm. For that purpose, the algorithms are run on systematically generated test cases.

3.1 Test set

The domain of this first test is Bibliographic references. It is, of course, based on a subjective view of what must be a bibliographic ontology. There can be many different classifications of publications, for example, based on area and quality. The one chosen here is common among scholars and is based on publication categories; as many ontologies (tests #301-304), it is reminiscent to BibTeX.

The systematic benchmark test set is built around one reference ontology and many variations of it. The ontologies are described in OWL-DL and serialized in the RDF/XML format. The reference ontology is that of test #101. It contains 33 named classes, 24 object properties, 40 data properties, 56 named individuals and 20 anonymous individuals. Participants have to match this reference ontology with the variations. Variations are focused on the characterization of the behavior of the tools rather than having them compete on real-life problems. They are organized in three groups:

Simple tests (1xx) such as comparing the reference ontology with itself, with another irrelevant ontology (the wine ontology used in the OWL primer) or the same ontology in its restriction to OWL-Lite;

Systematic tests (2xx) obtained by discarding features from some reference ontology. It aims at evaluating how an algorithm behaves when a particular type of information is lacking. The considered features were:

- *Name of entities* that can be replaced by random strings, synonyms, name with different conventions, strings in another language than English;
- *Comments* that can be suppressed or translated in another language;
- *Specialization hierarchy* that can be suppressed, expanded or flattened;
- *Instances* that can be suppressed;
- *Properties* that can be suppressed or having the restrictions on classes discarded;
- *Classes* that can be expanded, i.e., replaced by several classes or flattened.

Four real-life ontologies of bibliographic references (3xx) found on the web and left mostly untouched (there were added xmlns and xml:base attributes).

Since the goal of these tests is to offer some kind of permanent benchmarks to be used by many, the test is an extension of the 2004 EON Ontology Alignment Contest, whose test numbering it (almost) fully preserves. This year, no modification has been made since the last year benchmark suite.

The kind of expected alignments is still limited: they only match named classes and properties, they mostly use the "=" relation with confidence of 1.

After evaluation we have noted two mistakes in our test generation software, so that tests #249 and 253 still have instances in them. This problem already existed in 2005 and 2006. So the yearly comparison still holds. Full description of these tests can be found on the OAEI web site.

3.2 Results

13 systems participated in the benchmark track of this year's campaign. Table 3 provides the consolidated results, by groups of tests. We display the results of participants as well as those given by some simple edit distance algorithm on labels (edna). The computed values are real precision and recall and not an average of precision and recall. The full results are on the OAEI web site.

algo	edna		ASMOV		DSSim		Falcon		Lily		OLA2		OntoDNA	
test	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
1xx	0.96	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.94	1.00
2xx	0.40	0.55	0.95	0.90	0.99	0.60	0.92	0.85	0.97	0.89	0.91	0.86	0.80	0.43
3xx	0.46	0.79	0.85	0.82	0.89	0.67	0.89	0.79	0.81	0.80	0.63	0.76	0.90	0.71
Total	0.44	0.60	0.95	0.90	0.98	0.64	0.92	0.86	0.96	0.89	0.89	0.87	0.83	0.49
Ext	0.59	0.80	0.97	0.92	0.99	0.64	0.96	0.89	0.97	0.90	0.93	0.90	Error	

algo	OWL-CM		Prior+		RiMOM		SAMBO		SEMA		SODA		TaxoMap		X-SOM	
test	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
1xx	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.98	1.00	1.00	1.00	0.67	1.00	0.34	0.99	0.99
2xx	0.82	0.51	0.92	0.79	0.97	0.86	0.98	0.51	0.92	0.72	0.96	0.50	0.91	0.19	0.73	0.67
3xx	0.95	0.37	0.87	0.83	0.69	0.80	0.94	0.67	0.67	0.79	0.51	0.41	0.92	0.26	0.94	0.68
Total	0.85	0.54	0.93	0.81	0.95	0.87	0.98	0.56	0.90	0.74	0.92	0.51	0.92	0.21	0.76	0.70
Ext	Error		0.96	0.84	0.96	0.87	Error		0.93	0.77	Error		Error		Error	

Table 3. Means of results obtained by participants on the benchmark test case (corresponding to harmonic means). The Ext line corresponds to the three extended precision and recall measures (see [5] and further explanations next).

These results show already that three systems are relatively ahead (ASMOV, Lily and RiMOM) with three close followers (Falcon, Prior+ and OLA2). No system had strictly lower performance than edna. Each algorithm has its best score with the 1xx test series. There is no particular order between the two other series.

The results have also been compared with the three measures proposed in [5] (symmetric, effort-based and oriented). These are generalisation of precision and recall in order to better discriminate systems that slightly miss the target from those which are grossly wrong. The three measures provide the same results, so they have been displayed only once in Table 3 under the label "Ext". This is not really surprising given the proximity of these measures. As expected, they only improve over traditional precision and recall. Again, the new measures do not dramatically change the evaluation of the participating systems (all scores are improved and the six leading systems are closer to each others). This indicates that the close followers of the best systems (Falcon, OLA2) could certainly easily be corrected to reach the level of the best ones (RiMOM in particular). Since last year, the implementation of the precision and recall evaluator has changed. As a consequence, a number of results which would have been rejected last year, and then corrected by the participants, were accepted this year. As a consequence, now, the extended precision and recall reject them: this concerns the systems marked with "Error".

This year the apparently best algorithms provided their results with confidence measures. It is thus possible to draw precision/recall graphs in order to compare them. We provide in Figure 1 the precision and recall graphs of this year. They are only relevant for the results of participants who provided confidence measures different from 1 or 0 (see Table 2). They also feature the results for edit distance on class names (edna) and the results of previous years (Falcon-2005 and RiMOM-2006). This graph has been drawn with only technical adaptation of the technique used in TREC. Moreover, due to lack of time, these graphs have been computed by averaging the graphs of each of the tests (instead to pure precision and recall).

These results and those displayed in Figure 2 single out a group of systems, ASMOV, Lily, Falcon 0.7, OLA2, Prior+ and RiMOM which seem to perform these tests at the highest level of quality. Of these, ASMOV, Lily and RiMOM seem to have slightly better results than the three others. Like the two previous years, there is a gap between these systems and their followers. In addition, one system (OLA2) has achieved to fill this gap without significantly changing its strategy[2].

We have compared the results of this year's systems with the results of the previous years on the basis of 2004 tests, see Table 4. The results of three best systems (ASMOV, Lily and RiMOM) are comparable but never identical to the results provided in the previous years by RiMOM (2006) and Falcon (2005). Like Falcon last year, RiMOM provided this year lower results than last year. Figure 1 shows that RiMOM has increased its precision and decreased its overall performance. There seems to be a limit that systems are not able to overcome. At the moment, it seems that these systems are at a level at which making more progress is very hard: we now have strong arguments that having a 100% recall and precision on all these tests is not a reachable goal.

---

[2] Disclosure: the author of these lines is a member of the OLA2 team.

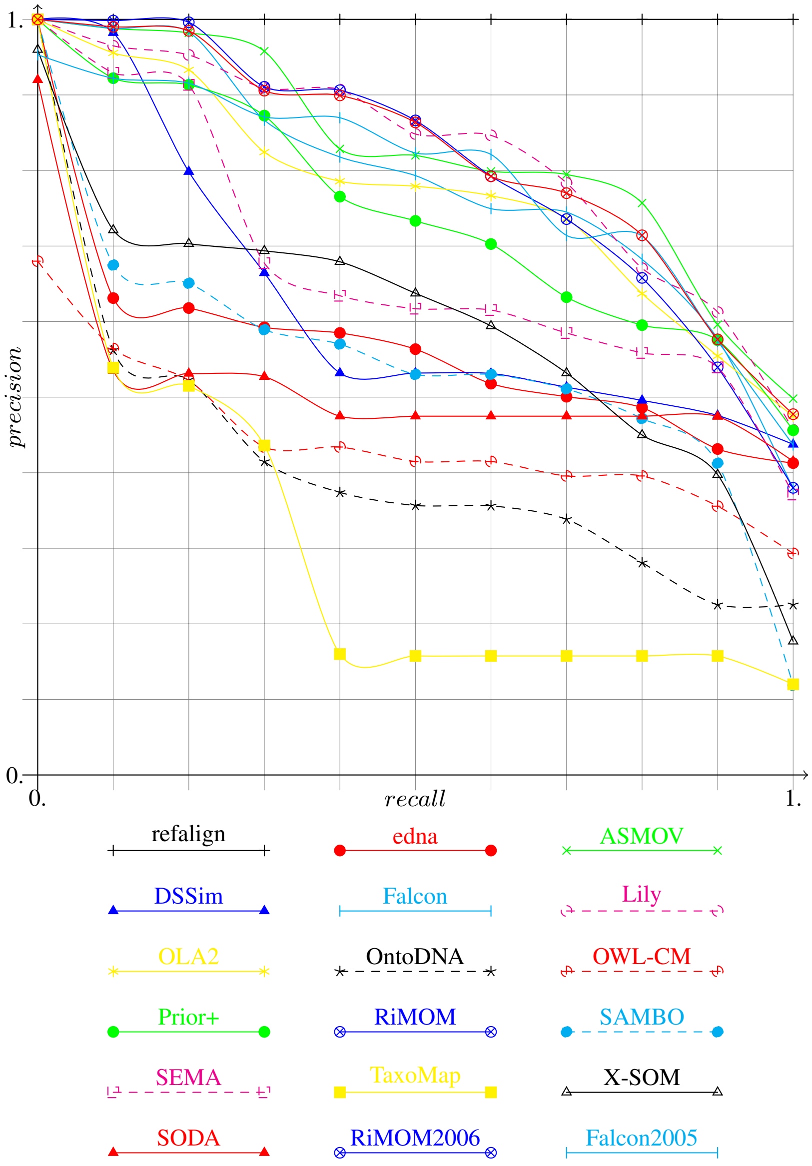


Fig. 1. Precision/recall graphs. They cut the results given by the participants under a threshold necessary for achieving $n\%$ recall and compute the corresponding precision. Systems for which these graphs are not meaningful (because they did not provide graded confidence values) are drawn in dashed lines. We remind the graphs for the best systems of the previous years, namely of Falcon in 2005 and RiMOM in 2006.

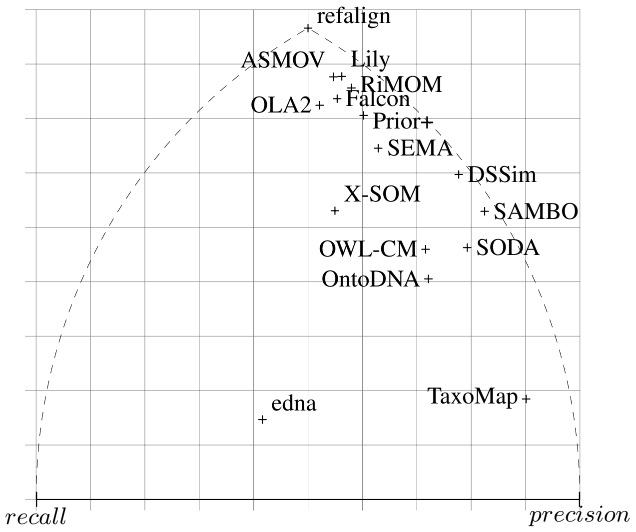


Fig. 2. Each point expresses the position of a system with regard to precision and recall.

Year	2004				2005		2006		2007					
System	Fujitsu		PromptDiff		Falcon		RiMOM		ASMOV		Lily		RiMOM	
test	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
1xx	0.99	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
2xx	0.93	0.84	0.98	0.72	0.98	0.97	1.00	0.98	0.99	0.99	1.00	0.98	1.00	0.97
3xx	0.60	0.72	0.93	0.74	0.93	0.83	0.83	0.82	0.85	0.82	0.81	0.80	0.69	0.80
H-means	0.88	0.85	0.98	0.77	0.97	0.96	0.97	0.96	0.97	0.97	0.97	0.96	0.95	0.95

Table 4. Evolution of the best scores over the years on the basis of 2004 tests.

4 Anatomy

The focus of the anatomy track is to confront existing matching technologies with real world ontologies. Currently, we find such real world cases primarily in the biomedical domain, where a significant number of ontologies have been built covering different aspects of medical research. Manually generating alignments between these ontologies requires an enormous effort by highly specialized domain experts. Supporting these experts by automatically providing correspondence proposals is both challenging, due to the complexity and the specialized vocabulary of the domain, and relevant, due to the increasing number of ontologies used in clinical research.

4.1 Test data and experimental setting

The ontologies of the anatomy track are the NCI Thesaurus describing the human anatomy, published by the National Cancer Institute (NCI)[3], and the Adult Mouse Anatomical Dictionary[4], which has been developed as part of the Mouse Gene Expression Database project. Both resources are part of the Open Biomedical Ontologies (OBO). The complex and laborious task of generating the reference alignment has been conducted by a combination of computational methods and extensive manual evaluation. In addition, the ontologies were extended and harmonized to increase the number of correspondences between both ontologies. An elaborate description of creating the reference alignment can be found in [4] and in work to be published by Hayamizu et al.

The task is placed in a domain where we find large, carefully designed ontologies that are described in technical terms. Besides their large size and a conceptualization that is only to a limited degree based on the use of natural language, they also differ from other ontologies with respect to the use of specific annotations and roles, e.g., the extensive use of the $partOf$ relation. The manual harmonization of the ontologies leads to a situation, where we have a high number of rather trivial correspondences that can be found by simple string comparison techniques. At the same time, we have a good share of non-trivial correspondences that require a careful analysis and sometimes also medical background knowledge. To better understand the occurrence of non-trivial correspondences in alignments, we implemented a straightforward matching tool that compares normalized concept labels. This trivial matcher generates for all pairs of concepts $\langle C, D\rangle$ a correspondence if and only if the normalized label of C is identical to the normalized label of D. In general we expect an alignment generated by this approach to be highly precise while recall will be relatively low. With respect to our matching task we measured approximately 99% precision and 60% recall. Notice that the value for recall is relatively high, which is partially caused by the harmonization process mentioned above.

Because we assumed that all matchers would easily find the trivial correspondences, we introduce an additional measure for recall, called $recall+$. $Recall+$ measures how many non-trivial correct correspondences can be found in an alignment M. Given a reference alignment R and an alignment S generated by the naive string equality matching,

---

[3] `http://www.cancer.gov/cancerinfo/terminologyresources/`

[4] `http://www.informatics.jax.org/searches/AMA_form.shtml`

$recall+$ is defined as follows:

$$Recall+ = \frac{|(R \cap M) - S|}{|R - S|}$$

We divided the task of automatically generating an alignment between these ontologies into three subtasks. Task #1 was obligatory for participants of the anatomy track, while task #2 and #3 were optional. For task #1 a matching system has to be applied with standard settings to obtain a result that is as good as possible with respect to the expected F-measure. For task #2 an alignment with increased precision has to be found. This seems to be an adequate requirement in a scenario where the automatically generated alignment will be directly used without subsequent manual evaluation. Contrary to this approach, in task #3 an alignment with increased recall has to be generated. Such an alignment could be seen as basis for subsequent expert evaluation. We believe that systems configurable with respect to these requirements will be much more useful in concrete application scenarios.

4.2 Results

In total, 11 systems participated in the anatomy task. These systems can be roughly divided in three groups. Systems of type A are highly specialized on matching biomedical ontologies and make extensive use of medical background knowledge. These systems are AOAS and SAMBO. Systems of type B can solve matching problems of different domains, but include a component exploiting biomedical background knowledge (e.g., using UMLS as lexical reference system). ASMOV and RiMOM fall into this category. Systems of type C, finally can be seen as general matching systems that do not distinguish between medical ontologies and ontologies of different domains. Most systems in the experiment fall into this category. Table 5 gives an overview of participating systems.

Runtime. The runtime of the systems differs significantly[5]. In average type-C systems outperformed systems that use medical knowledge. Falcon-AO, a system that solves large matching problems by applying a partition-based block matching strategy, solves the matching task in about 12 minutes without loss of quality with respect to the resulting alignment compared to other systems of type C. It has to be considered if similar approaches can also be applied to systems like ASMOV or Lily to solve their problems with runtime.

Type-C systems. The most astounding result is based on the suprisingly good performance of the naive label comparison approach compared to the alignments generated by systems of type C. The results of the naive approach are better with respect to recall as well as precision for task #1 compared to almost all matching systems of type C. Only TaxoMap and AgreementMaker generate alignments with higher recall but a significant

---

[5] Runtime information has been provided by the participants. All alignments have been generated on similarly equipped standard PCs. Advantages based on hardware differences could be neglected due to the significant differences in runtime.

System	Type	Task #1				Task #2		Task #3		Recall+	
		Runtime	Prec	Rec	F-meas	Prec	Rec	Prec	Rec	#1	#3
AOAS	A	2h	0.928	0.804	0.861	-	-	-	-	0.505	-
SAMBO	A	6 h	0.845	0.786	0.815	-	-	-	-	0.580	-
ASMOV	B	15 h	0.803	0.701	0.749	0.870	0.696	0.739	0.705	0.270	0.284
RiMOM	B	4 h	0.377	0.659	0.480	-	-	-	-	0.390	-
- Label Eq. -	-	*3 min*	*0.987*	*0.605*	*0.750*	-	-	-	-	0.0	-
Falcon-AO	C	12 min	0.964	0.591	0.733	0.986	0.540	0.814	0.655	0.123	0.280
TaxoMap	C	5 h	0.596	0.732	0.657	0.985	0.642	-	-	0.230	-
AgreementM.	C	30 min	0.558	0.635	0.594	0.930	0.286	0.424	0.651	0.262	0.302
Prior+	C	23 min	0.594	0.590	0.592	0.663	0.497	0.371	0.657	0.338	0.426
Lily	C	4 days	0.481	0.559	0.517	0.672	0.380	0.401	0.588	0.374	0.410
X-SOM	C	10 h	0.916	0.248	0.390	0.942	0.104	0.783	0.565	0.008	0.079
DSSim	C	75 min	0.208	0.187	0.197	-	-	-	-	0.067	-

Table 5. Participants and results with respect to runtime, precision, recall and F-measure. Results are listed in descending order with respect to the type of the system and the F-measure of task #1. The values for recall+ are presented in the rightmost columns for task #1 and #3.

loss in precision. We would have expected the participating systems to find more correct correspondences than applying straightforward label comparisons. It seems that many matching systems do not accept a correspondence even if the normalized labels of the concepts are equal. On the one hand, this might be caused by not detecting this equality at all (e.g., due to a partition based approach). On the other hand, a detected label equality can be rejected as correspondence due to the fact that additional information related to the concepts suggests that these concepts have a different meaning.

Type-A/B systems. Systems that use additional background knowledge related to the biomedical domain clearly generate better alignments compared to type-C systems. This result conforms with our expectations. The only exception is the low precision of the RiMOM system. The values for $recall+$ points to the advantage of using domain related background knowledge. Both AOAS and SAMBO detect about 50% of the non-trivial correspondences, while only Lily and Prior+ (systems of type C) achieve about 42% for task #3 with a significant loss in precision. Amongst all systems the AOAS approach generates the best alignment closely followed by SAMBO. Notice that AOAS is not available as a standalone system, but consists of a set of coupled programs which eventually require user configuration.

4.3 Discussion and conclusions

Obviously, the use of domain related background knowledge is a crucial point in matching biomedical ontologies and the additional effort of exploiting this knowledge pays off. This observation supports the claims for the benefits of using background knowledge made by other researchers [8; 1; 11]. Amongst all systems AOAS and SAMBO generate the best alignments, especially the relatively high number of detected non-trivial correspondences has to be mentioned positively. Nevertheless, for type C systems it is possible to detect non-trivial correspondences, too. In particular, the results of Lily and Prior+ on task #3 demonstrate this. Thus, there also seems to be a significant potential of exploiting knowledge encoded in the ontologies. Even if no medical background knowledge is used, it seems to make sense to provide a configuration that is specific to this type of domain. This is clearly demonstrated by the fact that most of the general matching systems fail to find a significant number of trivial correspondences. While in general it makes sense for a matcher not to accept all trivial correspondences to avoid the problem of homonymy, there are domains like the present one, however, where homonymy is not a problem, for example, because the terminology has been widely harmonized.

One major problem of matching medical ontologies is related to their large size. Though type C systems achieve relatively low values for recall, matching large ontologies seems to be less problematic. On the other hand the extensive use of domain related background knowledge has positive effects on recall, but does not seem to scale well. Thus, a trade-off between runtime and recall has to be found.

In further research we have to distinguish between different types of non-trivial correspondences. While for detecting some of these correspondences domain specific knowledge seems to be indispensable, the results indicate that there is also a large subset that can be detected by the use of alternative methods that solely rely on knowledge encoded in the ontologies. The distinction between different classes of non-trivial correspondences will be an important step for combining the strengths of both domain specific and domain independent matching systems. In summary, we can conclude that the data set used in the anatomy track is well suited to measure the characteristics of different matching systems with respect to the problem of matching biomedical ontologies.

5 Directory

The directory test case aims at providing a challenging task for ontology matchers in the domain of large directories.

5.1 Test set

The data set exploited in the directory matching task was constructed from Google, Yahoo and Looksmart web directories following the methodology described in [3; 9]. The data set is presented as taxonomies where the nodes of the web directories are modeled as classes and classification relation connecting the nodes is modeled as rdfs:subClassOf relation.

The key idea of the data set construction methodology is to significantly reduce the search space for human annotators. Instead of considering the full matching task which is very large (Google and Yahoo directories have up to $3 * 10^5$ nodes each: this means that the human annotators need to consider up to $(3{*}10^5)^2 = 9{*}10^{10}$ correspondences), it uses semi automatic pruning techniques in order to significantly reduce the search space. For example, for the data set described in [3], human annotators consider only 2265 correspondences instead of the full matching problem.

The specific characteristics of the data set are:

- More than 4.500 node matching tasks, where each node matching task is composed from the paths to root of the nodes in the web directories.
- Reference correspondences for all the matching tasks.
- Simple relationships, in particular, web directories contain only one type of relationships, which is the so-called classification relation.
- Vague terminology and modeling principles, thus, the matching tasks incorporate the typical real world modeling and terminological errors.

5.2 Results

In OAEI-2007, 9 out of 18 matching systems participated on the web directories data set, while in OAEI-2006, 7 out of 10, and in OAEI-2005, 7 out of 7 did it. Only the Falcon system participated in all three evaluations of the web directories data set. In 2007, participating systems demonstrated substantially higher quality results than in previous two years.

Precision, recall and F-measure of the systems on the web directories test case are shown in Figure 3. These indicators have been computed following the TaxMe and TaxMe2 methodologies [3; 9] and with the help of Alignment API [6].

Let us make several observations concerning quality of the results of the participated systems. In particular, the average F-measure of the systems increased from approximately 29% in 2006 to 49% in 2007. The highest F-measure of 71% was demonstrated by the OLA2 system in 2007. The average precision of the systems increased from approximately 35% in 2006 to 57% in 2007. The highest precision of 62% was demonstrated by both the OLA2 system and X-SOM in 2007. The average recall of the systems increased from approximately 22% in 2005 to 26% in 2006 and to 50% in 2007. The highest recall of 84% was demonstrated by the OLA2 system in 2007. Notice that in 2005 this data set allowed for estimating only recall, therefore in the above observations there are no values of precision and F-measure for 2005.

A comparison of the results in 2006 and 2007 for the top-3 systems of each of the years based on the highest values of the F-measure indicator is shown in Figure 4. The key observation is that quality of the best F-measure result of 2006 demonstrated by Falcon is almost doubled (increased by ~1.7 times) in 2007 by OLA2. The best precision result of 2006 demonstrated by Falcon was increased by ~1.5 times in 2007 by both OLA2 and X-SOM. Finally, for what concerns recall, the best result of 2005 demonstrated by OLA was increased by ~1.4 times in 2006 by Falcon and further increased by ~1.8 times in 2007 by OLA2. Thus, the OLA team managed to improve by ~2.6 times its recall result of 2005 in 2007.

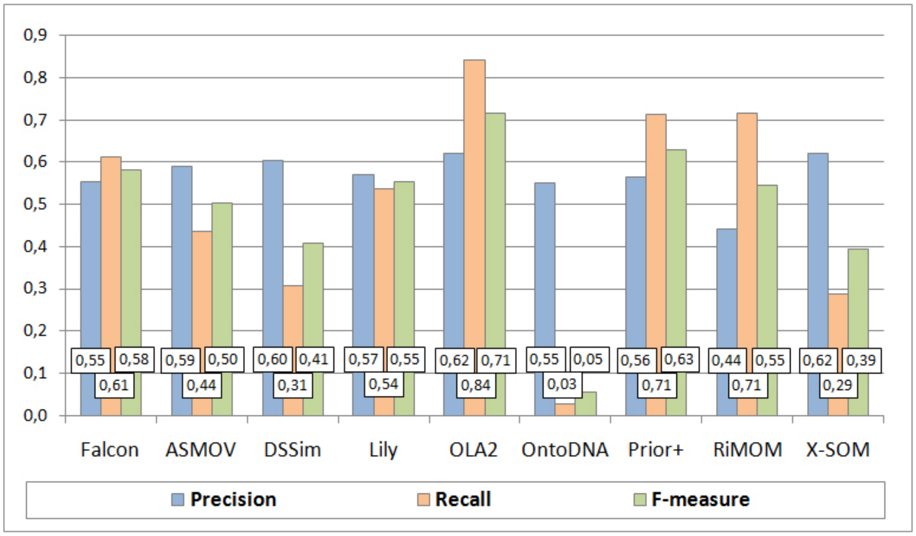


Fig. 3. Matching quality results.

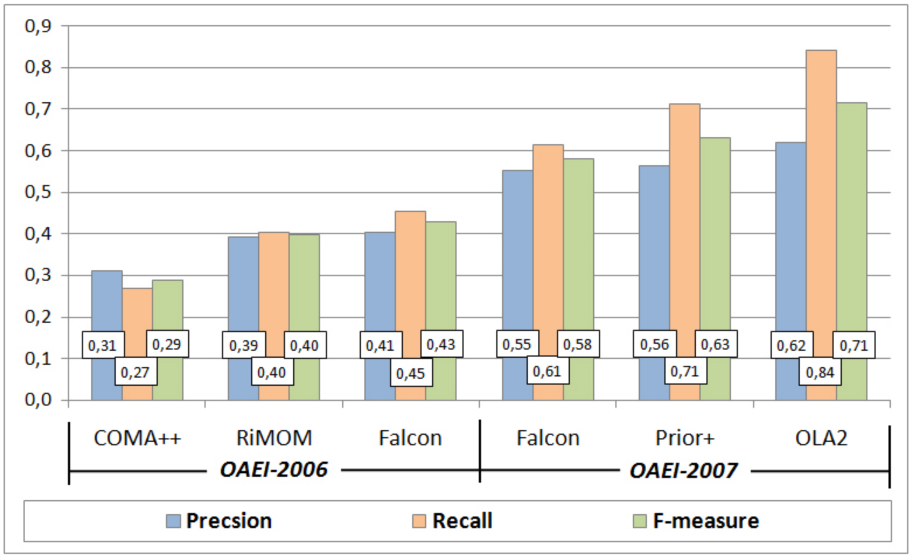


Fig. 4. Comparison of matching quality results in 2006 and 2007.

Partitions of positive and negative correspondences according to the system results are presented in Figure 5 and Figure 6, respectively.

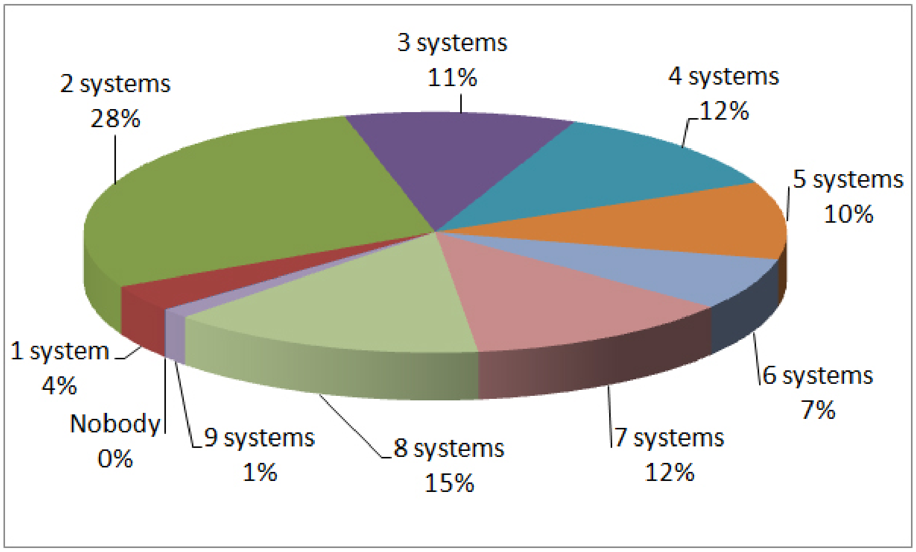


Fig. 5. Partition of the system results on positive correspondences.

Figure 5 shows that the systems managed to discover all the positive correspondences (Nobody - 0%). Only 15% of positive correspondences were found by almost all (8) matching systems. Figure 6 shows that almost all (8) systems found 11% of negative correspondences, i.e., mistakenly returned them as positive. The last two observations suggest that the discrimination ability of the data set is still high.

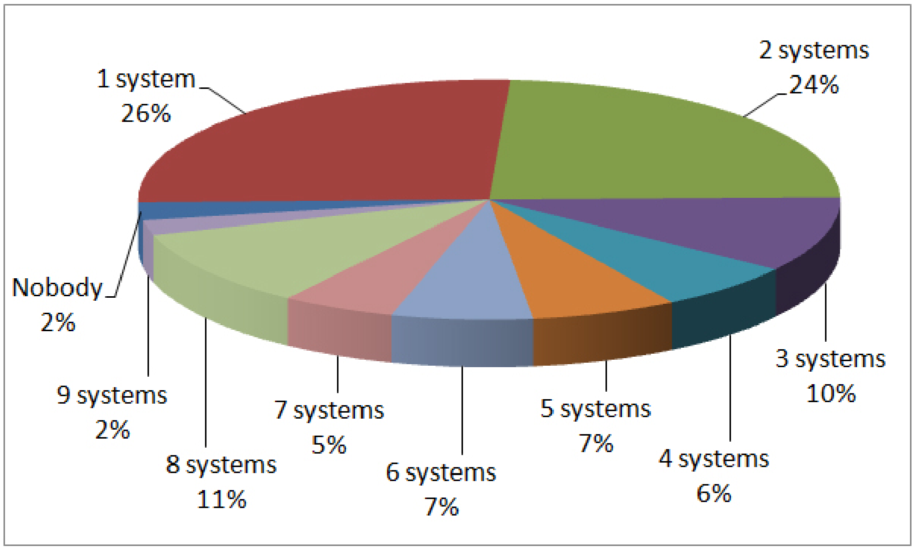


Fig. 6. Partition of the system results on negative correspondences.

Let us now compare partitions of the system results in 2006 and 2007 on positive and negative correspondences, see Figure 7 and Figure 8, respectively.

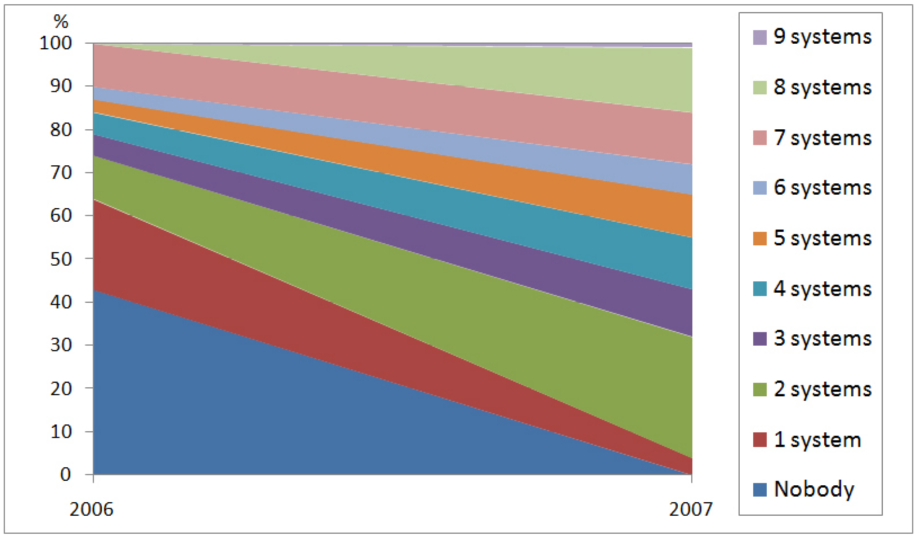


Fig. 7. Comparison of partitions of the system results on positive correspondences in 2006 and 2007.

Figure 7 shows that 43% of the positive correspondences have not been found by any of the matching systems in 2006, while in 2007 all the positive correspondences have been collectively found; see also how the selected regions (e.g., for 2 systems) consequently enlarge from 2006 to 2007.

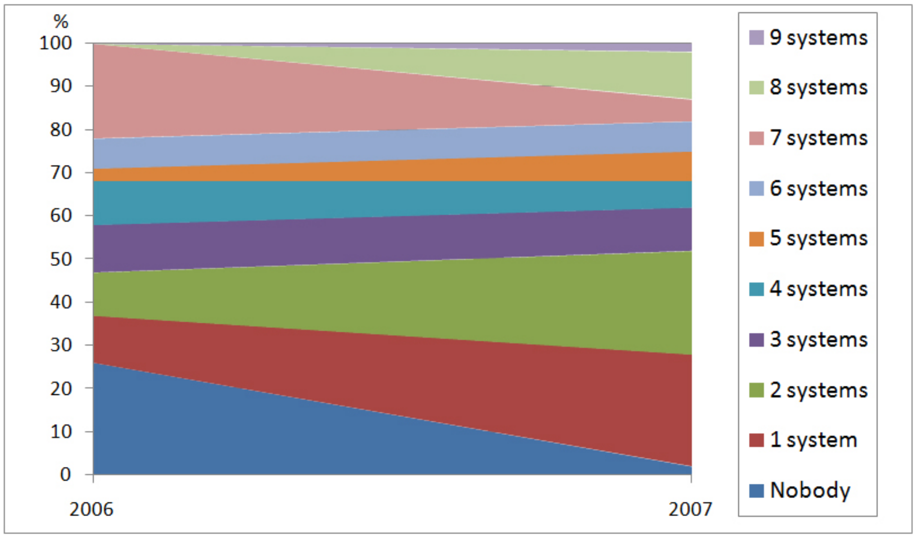


Fig. 8. Comparison of partitions of the system results on positive correspondences in 2006 and 2007.

Figure 8 shows that in 2006 in overall the systems have correctly not returned 26% of negative correspondences, while in 2007, this indicator decreased to 2%. In turn in 2006, 22% of negative correspondences were mistakenly found by all (7) the matching systems, while in 2007, this indicator decreased to 5%. An interpretation of these observations could be that systems keep trying various combinations of both "brave" and "cautious" strategies in discovering correspondences with a convergence towards better quality, since average precision increased from 2006 to 2007.

5.3 Comments

The key observation out of this evaluation is that the ontology matching community has made a substantial progress on the web directories task this year. In fact, as Figure 4 indicates, quality of the results is almost doubled from 2006 to 2007. This suggests that the systems experience fewer difficulties on the test case, although there still exists room for further improvements. Finally, as partitions of positive and negative correspondences indicate (see Figure 5 and Figure 6), the data set retains good discrimination ability, i.e., different sets of correspondences are still hard for the different systems.

6 Food

The food test case is another task in which the hierarchies come from thesauri, i.e., they have a lot of text involved compared to the previous test case, and they are expressed in SKOS. Success in this task greatly depends on linguistic term disambiguation and recognition of naming conventions.

6.1 Test set

The task of this case consists of matching two thesauri formulated in SKOS:

AGROVOC The United Nations Food and Agriculture Organization (FAO) AGROVOC thesaurus, version February 2007. This thesaurus consists of 28.445 descriptor terms, i.e., prefered terms, and 12.531 non-descriptor terms, i.e., alternative terms. AGROVOC is multilingual in eleven languages (en, fr, de, es, ar, zh, pt, cs, ja, th, sk).

NALT The United States National Agricultural Library (NAL) Agricultural thesaurus, version 2007. This thesaurus consists of 42.326 descriptor terms and 25.985 non-descriptor terms. NALT is monolingual, English.

Participants had to match these SKOS versions of AGROVOC and NAL using the exactMatch, narrowMatch, and broadMatch relations from the SKOS Mapping Vocabulary.

6.2 Evaluation procedure

Precision. In order to give dependable precision results within the time span of the campaign given a limited number of assessors we performed a combination of semi-automatic evaluation for alignments between taxonomical concepts and sample evaluation on roughly 5% of the other alignments. This sample was chosen to be representative of the type of topics covered by the thesauri and to be impartial to each participant and impartial to how much consensus amongst the participants there was about each alignment, i.e., the "hardness" or "complexity" of the alignment.

We distinguished four categories of topics in the thesauri that each required a different level of domain knowledge of the assessors: (i) taxonomical concepts (plants, animals, bacteria, etc.), (ii) biological and chemical terms (structure formulas, terms from genetics, etc.), (iii) geographical terms (countries, regions, etc.), and (iv) the remaining concepts (agricultural processes, natural resources, etc.).

Under the authority of taxonomists at the US Department of Agriculture the taxonomical category of correspondences was assessed using the strict rules that apply to the naming scheme of taxonomy. These are that if the preferred term of one concept is exactly the same as either the preferred or the alternative term of another concept then the concepts are considered to be exact matches. This rule works, because the taxonomical parts of the thesauri are based on the same sources. Samples from the other three categories were assessed by five groups of domain experts from the following institutions and companies: USDA NAL, UN FAO, Wageningen Agricultural University (WUR), Unilever, and the Netherlands organisation for applied scientific research (TNO). The sizes of the categories and the part that was assessed are shown in Table 6.

topic	# alignments	# assessed alignments (sample size)
taxonomical	22.542	22.542
biological / chemical	3.816	200
geographical	1.284	86
miscellaneous	9.678	476

Table 6. Categories of alignments that were separately assessed for the estimation of precision.

Recall. To give dependable recall results within the time span of the campaign we estimated recall on a set of sample sub-hierarchies of the thesauri. Specifically, everything under the NALT concept animal health and all AGROVOC concepts that have alignments to these concepts and their sub-concepts, all oak trees (everything under the concept representing the Quercus genus), all rodents (everything under Rodentia), countries of Europe, and part of the geographical concepts below country level (cities, provinces, etc.). These sample reference alignments consisted of exactMatch, narrowMatch, and broadMatch alignments. The sizes of the samples are shown in Table 7, along with the percentage of exactMatch alignments in each sample.

topic	# alignments	% exactMatch
animal health	34	57%
oak trees (taxonomical)	41	84%
rodents (vernacular)	42	32%
Europe (country level)	74	93%
geography (below country level)	164	35%

Table 7. Reference alignments that were used for the estimation of recall.

Significance. As a significance test on the percentile scores of the systems we used the Bernoulli distribution. The performance (precision or recall) of system A, P_A, can be considered to be significantly greater than that of system B for a sample set of size N when the following formula holds:

$$|P_A - P_B| > 2\sqrt{\frac{P_A(1-P_A)}{N} + \frac{P_B(1-P_B)}{N}}$$

6.3 Results

Five participants took part in the OAEI-2007 food test case: South East University (Falcon-AO 0.7), Tsinghua University (RiMOM), Politecnico di Milano (X-SOM), and the Knowledge Media Institute with two systems (DSSim and SCARLET). Each team provided between 18.420 (RiMOM) and 6.583 (X-SOM) alignments. This amounted to 37.384 unique alignments in total. Table 8 shows the total number of alignments that were submitted by each of the systems.

system	# alignments	alignment type
Falcon-AO	15.300	exactMatch
RiMOM	18.420	exactMatch
X-SOM	6.583	exactMatch
DSSim	14.962	exactMatch
SCARLET	81	exactMatch
	6.038	broadMatch & narrowMatch

Table 8. Number and type of alignments that were returned by the participating systems.

Best precision. The taxonomical parts of the thesauri accounted for by far the largest part of the alignments. The more difficult correspondences that required lexical normalization, such as structure formulas, and relations that required background knowledge, such as many of the relations in the miscellaneous domain, accounted for a smaller part of the alignments. This caused systems that did well at the taxonomical part to have a great advantage over the other systems. The Falcon-AO system performed consistently best at the largest two strata, taxonomical and miscellaneous, and thus achieved high precision. An overview of all the precision results is shown in Table 9. The results of the SCARLET system have been evaluated separately for each alignment type and hence are shown separately in Table 9.

Best recall. All systems except SCARLET only returned exactMatch alignments. This significantly limits recall. In Table 10 the first number represents recall of all types of alignment relations. Systems that only find exactMatch alignments are unable to achieve 1.00 here. The second number (between parentheses) shows recall of only exactMatch alignments. That means all systems can achieve 1.00 here. The RiMOM system had the highest recall for the OAEI 2006. This year, however, the Falcon-AO system has a higher recall than the RiMOM system. For some categories the RiMOM result equals to that of Falcon-AO, but on average the difference is significant.

					SCARLET	
Precision	Falcon-AO	RiMOM	X-SOM	DSSim	e.M.	b.M. & n.M.
taxonomical	0.81	0.54	0.26	0.37	0.60	0.13
bio/chem	0.91	0.84	0.92	0.86	1.00	0.17
geographical	0.95	0.97	1.00	0.94	0.00	1.00
miscellaneous	0.86	0.69	0.62	0.57	0.75	0.44
overall	**0.84**	**0.62**	**0.45**	**0.49**	**0.66**	**0.25**

Table 9. Precision results based on sample evaluation.

Recall	Falcon-AO	RiMOM	X-SOM	DSSim	SCARLET
animal health	0.21 (0.64)	0.21 (0.64)	0.00 (0.00)	0.06 (0.18)	0.00 (0.00)
oak trees	0.93 (1.00)	0.93 (1.00)	0.10 (0.12)	0.22 (0.24)	0.00 (0.00)
rodents	0.40 (0.71)	0.24 (0.42)	0.07 (0.10)	0.17 (0.29)	0.00 (0.00)
Europe	0.81 (0.97)	0.70 (0.84)	0.08 (0.10)	0.34 (0.40)	0.00 (0.00)
geography	0.32 (0.90)	0.26 (0.74)	0.05 (0.14)	0.18 (0.50)	0.01 (0.02)
overall	**0.49 (0.90)**	**0.42 (0.78)**	**0.06 (0.11)**	**0.20 (0.37)**	**0.00 (0.00)**

Table 10. Recall results based on sample evaluation. The numbers between parentheses show recall when only the exactMatch alignments of the reference alignments are considered.

7 Environment

The environment test set comprises three matching task between three thesauri: the two thesauri of the food task (AGROVOC and NALT), and the European Environment Agency thesaurus, GEMET. The participants were allowed to the third thesaurus as background knowledge to match the other two for the construction of any of the three alignments.

7.1 Test set

The task of this case consists of matching three thesauri formulated in SKOS:

GEMET The European Environment Agency (EEA) GEneral Multilingual Environmental Thesaurus, version July 2007. This thesaurus consists of 5.298 concepts, each with descriptor terms in all of its 22 languages (bg, cs, da, de, el, en, en-us, es, et, eu, fi, fr, hu, it, nl, no, pl, pt, ru, sk, sl, sv).

AGROVOC The United Nations Food and Agriculture Organization (FAO) AGROVOC thesaurus, version February 2007. This thesaurus consists of 28.445 descriptor terms, i.e., prefered terms, and 12.531 non-descriptor terms, i.e., alternative terms. AGROVOC is multilingual in eleven languages (en, fr, de, es, ar, zh, pt, cs, ja, th, sk).

NALT The United States National Agricultural Library (NAL) Agricultural thesaurus, version 2007. This thesaurus consists of 42.326 descriptor terms and 25.985 non-descriptor terms. NALT is monolingual, English.

Participants had to match these SKOS versions of GEMET, AGROVOC and NAL using the exactMatch, narrowMatch, and broadMatch relations from the SKOS Mapping Vocabulary.

7.2 Evaluation procedure

The evaluation procedure used is the same as for the food task with the exception that we used slightly different categories of sample topics.

Precision. For the evaluation of precision for the GEMET-AGROVOC and GEMET-NALT alignments we distinguished six categories of topics in the thesauri that each required a different level of domain knowledge of the assessors: (i) taxonomical concepts (plants, animals, bacteria, etc.), (ii) biological and chemical terms (structure formulas, terms from generics, etc.), (iii) geographical terms (countries, regions, etc.), (iv) natural resources (fishery, forestry, agriculture, mining, etc.), (v) health risk management (pollution, food, air, water, disasters, etc.), and (vi) the remaining concepts (administration, materials, military aspects, etc.). The results for the NALT-AGROVOC are shown in the section about the food task. The sizes of the categories and the part that was assessed are shown in Table 11.

	GEMET-AGROVOC		GEMET-NALT	
topic	# alignments	# assessed	# alignments	# assessed
taxonomical	500	39	802	33
biological / chemical	541	43	841	51
geographical	167	40	164	39
natural resources	412	51	450	39
health risk management	602	38	738	52
miscellaneous	1.884	48	1.988	51

Table 11. Categories of alignments that were separately assessed for the estimation of precision.

Recall. For the evaluation of recall we used a set of sub-hierarchies of the thesauri. Specifically, concepts from agriculture in the broad sense of the word, including: fishery (fishing equipment, aquaculture methods, etc.) and animal husbandry (animal diseases, animal housing, etc.), and geological concepts like countries and place types (the Baltic states, alluvial plains, etc.). The sizes of the samples are shown in Table 12, along with the percentage of exactMatch alignments in each sample.

	GEMET-AGROVOC		GEMET-NALT	
topic	# alignments	% exactMatch	# alignments	% exactMatch
agriculture	89	69%	92	66%
geology	136	64%	138	56%

Table 12. Reference alignments that were used for the estimation of recall.

7.3 Results

Two systems took part in the OAEI 2007 environment task: Falcon-AO 0.7 (South East University) and DSSim (Knowledge Media Institute). Both systems returned only exactMatch alignments. Table 13 shows the number of correspondences the two systems returned for each of the three tasks.

	# correspondences		
system	NALT-AGROVOC	GEMET-AGROVOC	GEMET-NALT
Falcon-AO	15.300	1.384	1.374
DSSim	14.962	3.030	4.278

Table 13. Number of correspondences that were returned by the participating systems.

Best precision. The GEMET thesaurus is very shallow compared to the AGROVOC and NALT thesauri, but it does offer definitions and labels in many languages. In consequence, lexical comparison is usually the only source of information that the matching system can exploit. This means that there is very little information for the matching systems to reason with. The Falcon-AO system performed best at both tasks, achieving a similar precision as with the easier NALT-AGROVOC task. An overview of all the precision results is shown in Table 14.

	GEMET-AGROVOC		GEMET-NALT	
Precision for	Falcon-AO	DSSim	Falcon-AO	DSSim
taxonomical	0.95	0.27	0.87	0.16
bio/chem	0.54	0.00	0.88	0.53
geographical	1.00	0.30	0.77	0.29
natural resources	1.00	0.53	0.95	0.32
health risk man.	0.95	0.38	0.88	0.50
miscellaneous	0.90	0.39	0.82	0.53
overall	**0.88**	**0.33**	**0.86**	**0.44**

Table 14. Precision results based on sample evaluation.

Best recall. The Falcon-AO system performs significantly better than the DSSim system on the GEMET-AGROVOC and GEMET-NALT tasks. However, it does not achieve similar recall scores as for the NALT-AGROVOC task.

	GEMET-AGROVOC		GEMET-NALT	
Recall for	Falcon-AO	DSSim	Falcon-AO	DSSim
agriculture	0.43 (0.62)	0.11 (0.16)	0.36 (0.54)	0.16 (0.25)
geology	0.37 (0.59)	0.18 (0.29)	0.26 (0.47)	0.17 (0.30)
overall	**0.39 (0.60)**	**0.15 (0.24)**	**0.30 (0.50)**	**0.16 (0.27)**

Table 15. Recall results based on sample evaluation. The numbers between parentheses show recall when only the exactMatch alignments of the reference alignments are considered.

8 Library

This is the last test case from the directory and thesauri track. It deals with two large Dutch thesauri.

8.1 Data set

The National Library of the Netherlands (KB) maintains two large collections of books: the Deposit Collection, containing all the Dutch printed publications (one million items), and the Scientific Collection, with about 1.4 million books.

Each collection is annotated – *indexed* – using its own controlled vocabulary. The Scientific Collection is described using the GTT thesaurus, a huge vocabulary containing 35.194 general concepts, ranging from Wolkenkrabbers (Sky-scrapers) to Verzorging (Care). The books in the Deposit Collection are mainly described against the Brinkman thesaurus, which contains a large set of headings (5.221) for describing the overall subjects of books. Both thesauri have similar coverage (2.895 concepts actually have exactly the same label) but differ in granularity.

Each concept has (exactly) one preferred label, synonyms (961 for Brinkman, 14.607 for GTT), extra hidden labels (134 for Brinkman, a couple of thousands for GTT) or scope notes (6.236 for GTT, 192 for Brinkman). The language of both thesauri is Dutch, albeit around 60% of GTT concepts also have English labels, which makes this track ideal for testing alignment in a non-English situation.

Concepts are also provided with structural information, in the form of *broader* and *related* links. However, GTT (resp. Brinkman) contains only 15.746 (resp 4.572) hierarchical *broader* links and 6.980 (resp. 1.855) associative *related* links. On average, one can expect at most one parent per concept, for an average depth of 1 and 2, respectively (in particular, the GTT thesaurus has 19.752 root concepts). The thesauri's structural information is thus very poor.

For the purpose of the OAEI campaign, the two thesauri were made available in the SKOS format. OWL versions were also provided, according to the – lossy – conversion rules detailed on the track page[6].

8.2 Evaluation and results

Three teams handed in final results: Falcon (3.697 `exactMatch` mappings), DSSim (9.467 `exactMatch` mappings), Silas (3.476 `exactMatch` mappings and 10.391 `relatedMatch` mappings). Two evaluation procedures were chosen, each of them motivated by a potential case of mapping usage.

Evaluation in a thesaurus merging scenario. The first scenario is *thesaurus merging*, where an alignment is used to build a new, unified thesaurus from GTT and Brinkman thesauri. Evaluation in such a context requires assessing the validity of each individual mapping, as in "standard" alignment evaluation.

---

[6] `http://oaei.ontologymatching.org/2007/library/`

Here, there was no reference alignment available. Given the size of the vocabularies, it was impossible to build one. Inspired by the anatomy and food tracks of OAEI 2006, we opted for evaluating precision using a reference alignment based on a lexical procedure. This makes use of direct comparison between labels, but also exploits a Dutch morphology database that allows to recognize variants of a word, e.g., singular and plural. 3.659 reliable equivalence links are obtained this way. We also measured coverage, which we define as the proportion of all good correspondences found by an alignment divided by the total number of good correspondences produced by all participants and those in the reference.

For manual evaluation, the set of all *equivalence* correspondences[7] was partitioned into parts unique to each combination of participant alignments plus reference set (15 parts in all). For each of those parts which were not in the lexical reference alignment, a sample of correspondences was selected, and evaluated manually. A total of 330 correspondences were assessed by two Dutch native experts.

From these assessments, precision and coverage were calculated with their 95% confidence intervals, taking into account sampling size and evaluator variability. The results are shown in Table 16, which identifies clearly Falcon as performing better than both other participants.

Alignment	Precision	Coverage
DSSim	0.134 ± 0.019	0.31 ± 0.19
Silas	0.786 ± 0.044	0.661 ± 0.094
Falcon	0.9725 ± 0.0033	0.870 ± 0.065

Table 16. Precision and coverage for the thesaurus merging scenario.

A detailed analysis reveals that Falcon results are very close to the lexical reference, which explains their observed quality. 3.493 links are common to Falcon and the reference, while Falcon has 204 correspondences not in the reference – of which 100 are good – and the lexical reference has 166 correspondences not identified by Falcon. DSSim also uses lexical comparisons, but its edit-distance-like approach is more prone to error: between 20 and 200 out its 8.399 correspondences not in the reference are correct. Silas is the one that succeeds most in adding to the reference: 234 of its 976 "non-lexical" correspondences are correct. But it fails to reproduce one third of the reference correspondences, therefore its coverage is relatively low.

Evaluation in an annotation translation scenario. The second usage scenario, aimed at indexers with an intricate expertise of Brinkman or GTT, consists in an *annotation translation* process supporting the re-indexing of GTT-indexed books with Brinkman concepts. This is particularly useful if GTT is dropped: a huge volume of legacy data has to be converted to the remaining annotation system.

This evaluation scenario requires building a tool that can interpret the correspondences provided by the different participants so as to translate existing GTT book annotations into equivalent Brinkman annotations. Based on the quality of the results for

---

[7] We did not proceed with manual evaluation of the *related* links, as only one contestant provided with such links, and their manual assessment is much more error-prone.

books we know the correct annotations of, we can assess the quality of the initial correspondences. This approach, based on evaluation of user's information needs (here, book annotations) is more in line with the application-specific, end-to-end approach described in [14].

Evaluation settings and measures. The simple concept-to-concept correspondences sent by participants were transformed into more complex mapping rules that associate one GTT concept and a set of Brinkman concepts – some GTT concepts are indeed involved in several mapping statements. Considering `exactMatch` only, this gives 3.618 rules for Falcon, 3.208 rules for Silas and 9.467 rules for DSSim.

The set of GTT concepts attached to each book is then used to decide whether these rules are *fired* for this book. If the GTT concept of one rule is contained by the GTT annotation of a book, then the rule is fired. As several rules can be fired for a same book, the union of the consequents of these rules forms the translated Brinkman annotation of the book.

On a set of books selected for evaluation, the generated concepts for a book are then compared to the ones that are deemed as correct for this book. At the book level, we measure how many books have a rule fired on them, and how many of them are actually *matched* books, i.e., books for which the generated Brinkman annotation contains at least one correct concept. These two figures give a precision (P_b) and a recall (R_b) for this book level.

At the annotation level, we measure (i) how many translated concepts are correct over the annotation produced for the books on which rules were fired (P_a), (ii) how many correct Brinkman annotation concepts are found for all books in the evaluation set (R_a), and (iii) a combination of these two, namely a Jaccard overlap measure between the produced annotation (possibly empty) and the correct one (J_a).

The ultimate measure for alignment quality here is at the annotation level. Measures at the book level are used as a raw indicator of users' (dis)satisfaction with the built system. A R_b of 60% means that the alignment does not produce any useful candidate concept for 40% of the books. We would like to mention that, in these formulas, results are counted on a book and annotation basis, and not on a rule basis. This reflects the importance of different thesaurus concepts: a translation rule for a frequently used concept is more important than a rule for a rarely used concept. This option suits the application context better.

Automatic evaluation and results. Here, the reference set consists of 243.887 books belonging both to KB Scientific and Deposit collections, and therefore already indexed against both GTT and Brinkman. The existing Brinkman indices from these books are taken as a reference to which the results of annotation translation are automatically compared.

Table 17 gives an overview of the evaluation results when we only use the `exactMatch` mappings. Falcon and Silas perform similarly, and much ahead of DSSim. Nearly half of the books were given at least one correct Brinkman concept in the Falcon case, which corresponds to 65% of the books a rule was fired on. At the annotation level, half of the translated concepts are not validated, and more than 60% of the real Brinkman annotation is not found. We already pointed out that the correspondences from Falcon are mostly generated by lexical similarity. This indicates that lexically equivalent correspondences alone do not solve the annotation translation problem. It also confirms the sensitivity of mapping evaluation methods to certain application scenarios.

Participant	P_b	R_b	P_a	R_a	J_a
Falcon	65.32%	49.21%	52.63%	36.69%	30.76%
Silas	66.05%	47.48%	53.00%	35.12%	29.22%
DSSim	18.59%	14.34%	13.41%	9.43%	7.54%
Silas+related	69.23%	59.48%	34.20%	46.11%	24.24%

Table 17. Performance of annotation translations generated from correspondences.

Among the three participants, only Silas generated `relatedMatch` mappings. To evaluate their usefulness for annotation translation, we combined them with the `exactMatch` ones so as to generate a new set of 8,410 rules. As shown in the *Silas+related* line in Table 17, the use of `relatedMatch` mappings increases the chances of having a book given a correct annotation. However, unsurprisingly, precision of annotations decreases, because of the introduction of noisy results.

Manual evaluation and results. Automatic evaluation against existing annotations gives a first large and relatively cheap assessment of participants' results. Yet it is sensitive to *indexing variation*: several indexers annotating a same book, or a same annotator annotating it at different times, will select different concepts. We decided to perform an additional *manual* evaluation to assess the influence of this phenomenon, as well as to validate or invalidate the results of the automatic evaluation.

For this evaluation, we have partly followed the approach presented in [10]. First, a sample of 96 books was randomly selected among the dually annotated books annotated in 2006. On these books we applied the translation rules derived from each participants' results – using only the `exactMatch` links. For each book, the results of these different procedures are merged in a single list of candidate concept annotations. As we wanted some insight on the automatic evaluation based on existing Brinkman annotations, we also included these original annotations in the candidate lists.

To collect assessments of the candidate annotations, a paper form was created for each book in the sample. Each form constitutes an evaluation task where the evaluator validates the proposed annotations: for each of the candidates, she is asked whether it is *acceptable*[8] for an index. Afterwards, she is asked to select, among the candidates, the ones she would have *chosen as indices*. She also has the possibility to add to the list of chosen indices some concepts which are not in the proposed annotation. This form was validated by running a pilot evaluation.

The judges involved in the evaluation are four professional book indexers – native Dutch speakers – from the Depot department at the KB. Each of the evaluators assessed the candidates for every book in the test set.

---

[8] This precision is made to avoid too narrow choices, e.g., when the subject of the book is unclear, the thesaurus contains several concepts equally valid for the book, or when the evaluator feels other indexers could have selected indices different from hers.

Table 18 presents the acceptability assessments, averaged over the four evaluators. These are significantly and regularly higher than the figures obtained for automatic evaluation. This confirms the dependence of the scenario on the way indexing variability is taken into account in the evaluation setting.

Participant	P_a	R_a	J_a	P_a	R_a	J_a
Falcon	74.95%	46.40%	42.16%	52.63%	36.69%	30.76%
Silas	70.35%	39.85%	35.46%	53.00%	35.12%	29.22%
DSSim	21.04%	12.31%	10.10%	13.41%	9.43%	7.54%

Table 18. Comparison of correspondences as assessed by manual evaluation (left), and automatic evaluation results (right, from Table 17).

To assess *evaluation variability*, we computed the (Jaccard) overlap between the evaluators' assessments. On average, two evaluators agree on 60% of their assessments. We also measured the agreement between evaluators using Krippendorff's *alpha* coefficient – a common measure for computational linguistics tasks. The overall *alpha* coefficient is 0.62, which, according to standards, indicates a great variability. This is however to be put into perspective: the tasks usually analyzed with this coefficient, e.g., part-of-speech tagging, are less "variable" than subject indexing.

Indexing variability was first measured by assessing the original Brinkman indices for the books, which we had added in the candidate concepts to be evaluated. These concepts are the results of a careful selection, and do not render all the acceptable concepts for a book. It is therefore no surprise that the recall is relatively low (R_a =66.69%). However, it is very surprising to see that almost one original index concept out of five is not acceptable (P_a =81.60%). This result shows indeed that indexing variability matters a lot, even when the annotation selection criteria are made less selective.

To measure agreement between the indexers involved in our evaluation, we have computed the average Jaccard overlap between their chosen indices, as well as their Kripendorff's alpha. Again, we have quite a low overall agreement value – 57% for Jaccard, 0.59 for Krippendorff – which confirms the high intrinsic variability of the indexing task.

8.3 Discussion

The first comment on this track concerns the *form* of the alignment returned by the participants, especially *wrt.* the type and cardinality of alignments. All three participants proposed alignments using the SKOS links we asked for. However, only symmetric links (`exactMatch` and `relatedMatch`) were used: no participant proposed hierarchical `broader` and `narrower` links. Yet these links are useful for the application scenarios at hand. The `broader` links are useful to attach concepts which cannot be mapped to an equivalent corresponding concept but a more general or specific one. This is likely to happen, since the two thesauri have different granularity but a same general scope.

Second, there is no precise handling of one-to-many or many-to-many alignments. Sometimes a concept from one thesaurus is mapped to several concepts from the other. This proves to be very useful, especially in the annotation translation scenario where concepts attached to a book should ideally be translated as a whole. As a result, we have to post-process alignment results, building multi-concept correspondences from alignments which initially do not contain such links. This processing makes the evaluation of the relative quality of the alignments more difficult for the annotation scenario.

Of course these problems can be anticipated by making participants more aware of the different scenarios that will guide the evaluation. The campaign's timing made it impossible this year, but this is an option we would like to propose for next campaigns.

The results we have obtained also show that the performance of matching systems vary from one scenario to the other, highlighting the strengths of different approaches. For the merging scenario, Falcon outperforms the two other participants. While in the translation scenario, Silas, which detects links based on extensional information of concepts[9], performs similarly to Falcon.

Finally, we would like to discuss the overall quality of the results. The annotation translation scenario showed a maximum precision of 50%, and around 35% for recall. This is not much, but we have to consider that this scenario involves a high degree of variability: different annotators may choose different concepts for a same book. The manual evaluation by KB expert illustrate this phenomenon, and show that under specific but realistic application conditions the quality of participant's result is more satisfactory.

This still leaves the low coverage of alignments with respect to the thesauri, especially GTT: in the best case, only 9.500 of its 35.000 concepts were linked to some Brinkman concept. This track, arguably because of its Dutch language context, seems to be difficult. Silas' results, which are partly based on real book annotations, demonstrate that the task can benefit from the release of such extensional information. We will investigate this option for future campaigns.

9 Conference

The conference test set deals with matching several ontologies on the same topic. It also features a consensus workshop aimed at studying the elaboration of consensus when establishing the reference alignments.

9.1 Test set

The Conference collection consists of fourteen ontologies (developed within the OntoFarm project[10]) in the domain of organizing conferences. In contrast to the last year's conference track, there are four new ontologies. The main features of this test set are:

- *Generally understandable domain.* Most ontology engineers are familiar with organizing conferences. Therefore, they can create their own ontologies as well as evaluate the alignments among their entities with enough erudition.
- *Independence of ontologies.* Ontologies were developed independently and based on different resources, they thus capture the issues in organizing conferences from different points of view and with different terminology.

---

[9] Silas was trained on a set of books which is different from the evaluation set we used.

[10] `http://nb.vse.cz/~svatek/ontofarm.html`

- *Relative richness in axioms.* Most ontologies were equipped with DL axioms of various kinds, which opens a way to use semantic matchers.

Ontologies differ in number of classes, properties, their DL expressivity, but also in underlying resources. Nine ontologies are based on *tools* supporting the task of organizing conferences, two are based on experience of people with *personal participation* in a conference organization, and three are based on *web pages* of concrete conferences.

Participants provided either complete alignments or interesting correspondences (nuggets), for all or some pairs of ontologies. There was no reference alignment. Instead, organizers of this track offered manual a posteriori evaluation of results. Organizers also plan to offer a posteriori evaluation of results by data-mining techniques. Manual evaluation produced statistics such as precision and will serve as input into data-mining based evaluation. During manual evaluation some interesting correspondences were chosen as a background material for the consensus building discussion.

9.2 Results

During the evaluation phase, we manually labelled correspondences by several tags in order to enable further processing of the results. In part we used those tags for computing traditional precision and so-called *relative-recall* (see Figure 9), which is the ratio of the number of correct correspondences found by a system over the number of the correct correspondences found by any of the systems. Next we also counted two quite soft metrics: *ratioSubs* and *ratioTriv*, where ratioSubs shows ratio of the number of subsumption errors and the number of incorrect correspondences, and ratioTriv shows ratio of the number of the so-called trivial correspondences and the number of correct correspondences (see Figure 10). Trivial correspondences are correct correspondences where aligned concepts have the very same label, thus exact string matching can fully work. All results from this phase are available on the result report page[11]. Those global statistics more or less reflect the quality of results of participants. Moreover, these tags were suitable for choosing controversial correspondences as input to the Consensus Building Workshop where additional fine grain results were obtained (see next).

Participants differ in the number of alignments submitted for evaluation:

- The ASMOV team and the Falcon team delivered totally 91 alignments. All ontologies were matched to each other. The Lily team also matched all ontologies to each other, moreover they also matched ontologies to themselves. The OLA2 team and the OntoDNA team matched all ontologies to each other.
- In order to make evaluation process more balanced, we transformed all results of participants into 91 alignments, except results of the SEMA tool. They delivered 13 alignments by matching all ontologies to the EKAW ontology.

Consensus Building Workshop (CBW). As a part of the Ontology Matching workshop we organized for the second year the so-called Consensus Building Workshop. Motivation for this event is to thoroughly discuss controversial correspondences and collaboratively trying to achieve consensus about (in)correctness of correspondences.

---

[11] `http://oaei.ontologymatching.org/2007/result/conference/`

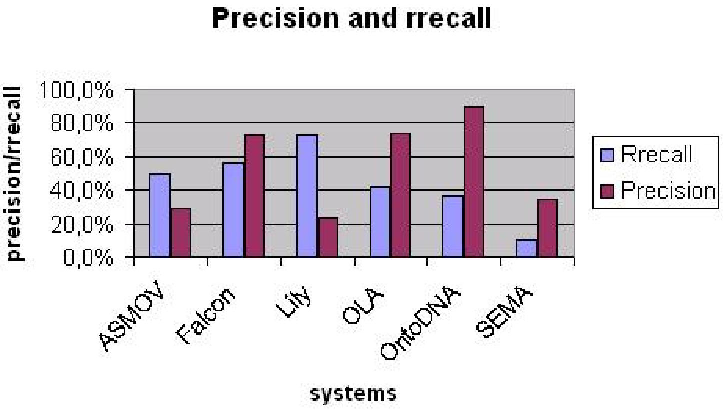


Fig. 9. Precision and relative-recall.

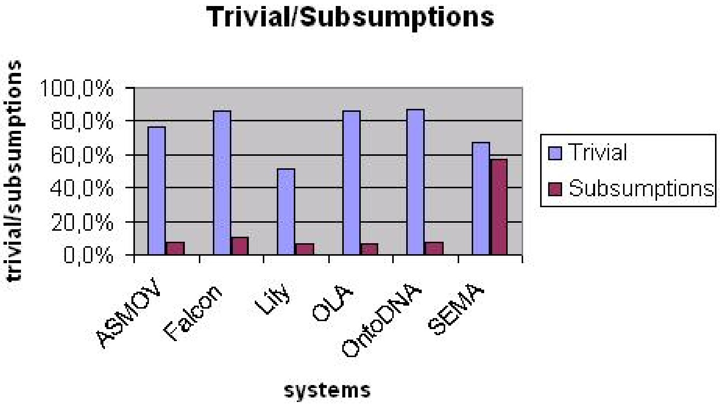


Fig. 10. ratioTriv and ratioSubs.

The controversial correspondences are those that were quite uncertain, unclear, interesting or obscure for different reasons during the process of evaluation. Intended results of this event are two-fold: (i) provide *feedback* for authors of the involved systems and (ii) examine the *argumentation process*. In comparison with 2006, participants of the discussion could argue for or against correspondences generally, and newly, we also considered application usage of correspondences. Organizers selected the application usage as a transformation application, i.e., final correspondences are to be used for conference data transformation from one software tool for organizing conference to another one. Altogether 15 correspondences were discussed. The board of experts finally achieved consensus for each of the correspondences. For some correspondences consensus was built easier, while for other argumentation had to be more sophisticated. The web site provides information about discussed controversial correspondences[12]. Additionally, for reaching agreement, visualization in the Protégé editor was used.

Some examples of the discussed correspondences include:

- `Individual_Presentation` and `Presentation`. Inspection of context, i.e., all subclasses, shows that `Individual_Presentation` is more general because `Presentation` contains just types of presentation format.
- `Name_of_conference` and `Topic_of_conference`. These concepts cannot be equivalent given common sense (or with regard to application-usage). On the other side, ontologies do not provide enough specification for distinguishing these concepts. In this case, it is needed to use some background knowledge.
- `has_author` and `submitted_by`. We concluded that these properties are equivalent according to their equivalent domains and ranges, i.e., how they are specified in the ontologies. However, regarding application-usage, they cannot be equivalent, because it can happen that sometimes the person who submitted the paper is not the author at the same time.
- `Presenter` and `Speaker`. In this case final consensus about equality of concepts was based on additional axiom: Paper presentedBy only Speaker.

Like in last CBW, the arguments are used in the same order: besides *lexical* reasons, the *context* of elements in question is considered primarily. Then, subclasses and superclasses which can unveil different extensions of classes are used. Finally, related properties and axioms are also considered.

CBW demonstrated that a board of experts can achieve agreement regarding (in)correctness of correspondences. CBW also shows that the problem can be with the meaning of concepts in terms of intensions of designers; some ontologies have not fully specified concepts. Application-usage of correspondences generally speeds up the discussion and makes the correspondence clearer.

10 Lesson learned and suggestions

The most important applied lesson learned from last year is that we have been able to revise the schedule so we had more time for evaluation. But there remain lessons not

---

[12] `http://oaei.ontologymatching.org/2007/result/conference/cbw07.ppt`

really taken into account that we identify with an asterisk (*). We reiterate those lessons that still apply with new ones:

A) This is a trend that there are now more matching systems and more systems are able to enter such an evaluation. This is very encouraging for the progress of the field.

B*) We also see systems that enter the campaign for several times. This means that we are not dealing with a continuous flow of prototypes, but with systems on which there is a persistent development. These systems tend to improve over years.

C*) The benchmark test case is not discriminant enough between systems. It is still useful for evaluating the strength and weakness of algorithms but does not seem to be sufficient anymore for comparing algorithms. We will have to improve these tests while preserving the comparability over years.

D) We have had more proposals for test cases this year (we had actively looked for them). However, the difficult lesson is that proposing a test case is not enough, there is a lot of remaining work in preparing the evaluation. Fortunately, with tool improvements, it will be easier to perform the evaluation. We would also like to have more test cases for expressive ontologies.

E*) It would be interesting and certainly more realistic, to provide some random gradual degradation of the benchmark tests (5% 10% 20% 40% 60% 100% random change) instead of a general discarding of features one by one. This has still not been done this year but we are considering it seriously for the next year.

F) This year, we have detected (through random verifications) some submissions which were not strictly complying to the evaluation rules. This suggests to be more strict about control in future campaigns.

G) Contrary to what has been noted in 2006, a significant number of systems were unable to output syntactically correct results (i.e., automatically usable by another program). Since fixing these mistakes by hand is becoming too much work, we plan to go towards automatic evaluation in which participants have to input correct results.

H) There seems to be partitions of the systems, between systems able to deal with large test sets and systems unable to do it, between system robust on all tracks and those which are specialized (see Table 2). These observations remain to be further analyzed.

11 Future plans

Future plans for the Ontology Alignment Evaluation Initiative are certainly to go ahead and to improve the functioning of the evaluation campaign. This involves:

- Finding new real world test cases, especially expressive ontologies;
- Improving the tests along the lesson learned;
- Accepting continuous submissions (through validation of the results);
- Improving the measures to go beyond precision and recall (we have done this for generalized precision and recall as well as for using precision/recall graphs, and will continue with other measures);
- Developing a definition of test hardness.

Of course, these are only suggestions that will be refined during the coming year.

12 Conclusion

This year we had more systems that entered the evaluation campaign as well as more systems managed to produce better quality results compared to the previous years. Each individual test case had more participants than ever. This shows that, as expected, the field of ontology matching is getting stronger (and we hope that evaluation has been contributing to this progress).

On the side of participants, it seems that there is clearly a problem of size of input that should be addressed in a general way. We would like to see more participation on the large test cases. On the side of organizers, each year the evaluation of matching systems becomes more complex.

Most of the participants have provided description of their systems and their experience in the evaluation[13]. These OAEI papers, like the present one, have not been peer reviewed. Reading the papers of the participants should help people involved in ontology matching to find what makes these algorithms work and what could be improved.

The Ontology Alignment Evaluation Initiative will continue these tests by improving both test cases and testing methodology for being more accurate. Further information can be found at:

`http://oaei.ontologymatching.org`.

---

[13] The SCARLET system is described in [12].

Acknowledgments

We warmly thank each participant of this campaign. We know that they have worked hard for having their results ready and they provided insightful papers presenting their experience. The best way to learn about the results remains to read the following papers.

We are grateful to Henk Matthezing, Lourens van der Meij and Shenghui Wang who have made crucial contributions to implementation and reporting for the Library track. The evaluation at KB could not have been possible without the commitment of Yvonne van der Steen, Irene Wolters, Maarten van Schie, and Erik Oltmans.

The following persons were involved in the Food and Environment tasks: Lori Finch (National Agricultural Library, US department of agriculture); Johannes Keizer, Margherita Sini, Gudrun Johannsen, Patricia Merrikin (Food and Agriculture Organization of the United Nations); Jan Top, Nicole Koenderink, Lars Hulzebos, Hajo Rijgersberg, Keen-Mun de Deugd (Wageningen UR); Fred van de Brug (TNO Quality of Life) and Evangelos Alexopoulos (Unilever). We would like to thank the teams of Agricultural Organization of the United Nations (FAO) for allowing us to use their ontologies.

We are grateful to Martin Ringwald and Terry Hayamizu for providing the reference alignment for the anatomy ontologies.

We also thank the other members of the Ontology Alignment Evaluation Initiative Steering committee: Wayne Bethea (John Hopkins University, USA), Alfio Ferrara (Università degli Studi di Milano, Italy), Lewis Hart (AT&T, USA), Tadashi Hoshiai (Fujitsu, Japan), Todd Hughes (DARPA, USA), Yannis Kalfoglou (University of Southampton, UK), John Li (Teknowledge, USA), Miklos Nagy (The Open University (UK), Natasha Noy (Stanford University, USA), Yuzhong Qu (Southeast University (China), York Sure (University of Karlsruhe, Germany), Jie Tang (Tsinghua University (China), Raphaël Troncy (CWI, Amsterdam, The Netherlands), Petko Valtchev (Université du Québec à Montréal, Canada), and George Vouros (University of the Aegean, Greece).

This work has been partially supported by the Knowledge Web European Network of Excellence (IST-2004-507482). In addition, Ondřej Šváb and Vojtěch Svátek were partially supported by IGA VSE grants no.12/06 “Integration of approaches to ontological engineering: design patterns, mapping and mining” and no.20/07 “Combination and comparison of ontology mapping methods and systems”.

References

1. Zharko Aleksovski, Warner ten Kate, and Frank van Harmelen. Exploiting the structure of background knowledge used in ontology matching. In *Proceedings of the ISWC workshop on Ontology Matching*, pages 13–24, Athens (GA US), 2006.
2. Ben Ashpole, Marc Ehrig, Jérôme Euzenat, and Heiner Stuckenschmidt, editors. *Proceedings of the K-Cap 2005 workshop on Integrating ontologies*, Banff (CA), 2005.
3. Paolo Avesani, Fausto Giunchiglia, and Mikalai Yatskevich. A large scale taxonomy mapping evaluation. In *Proceedings of the 4th International Semantic Web Conference (ISWC)*, pages 67–81, Galway (IE), 2005.
4. Oliver Bodenreider, Terry F. Hayamizu, Martin Ringwald, Sherri De Coronado, and Songmao Zhang. Of mice and men: Aligning mouse and human anatomies. In *Proceedings of the American Medical Informatics Association (AIMA) Annual Symposium*, pages 61–65, 2005.

5. Marc Ehrig and Jérôme Euzenat. Relaxed precision and recall for ontology matching. In *Proceedings of the K-Cap 2005 workshop on Integrating Ontologies*, pages 25–32, Banff (CA), 2005.
6. Jérôme Euzenat. An API for ontology alignment. In *Proceedings of the 3rd International Semantic Web Conference (ISWC)*, pages 698–712, Hiroshima (JP), 2004.
7. Jérôme Euzenat, Malgorzata Mochol, Pavel Shvaiko, Heiner Stuckenschmidt, Ondrej Svab, Vojtech Svatek, Willem Robert van Hage, and Mikalai Yatskevich. Results of the ontology alignment evaluation initiative 2006. In Pavel Shvaiko, Jérôme Euzenat, Natalya Noy, Heiner Stuckenschmidt, Richard Benjamins, and Michael Uschold, editors, *Proceedings of the ISWC workshop on Ontology Matching, Athens (GA US)*, pages 73–95, 2006.
8. Fausto Giunchiglia, Pavel Shvaiko, and Mikalai Yatskevich. Discovering missing background knowledge in ontology matching. In *Proceedings of the 17th European Conference on Artificial Intelligence (ECAI)*, pages 382–386, Riva del Garda (IT), 2006.
9. Fausto Giunchiglia, Mikalai Yatskevich, and Paolo Avesani. A large scale dataset for the evaluation of matching systems. In *Posters of the 4th European Semantic Web Conference (ESWC)*, Innsbruck (AU), 2007.
10. Antoine Isaac, Claus Zinn, Henk Matthezing, Lourens van der Meij, Stefan Schlobach, and Shenghui Wang. The value of usage scenarios for thesaurus alignment in cultural heritage context. In *Proceedings of the ISWC+ASWC International workshop on Cultural Heritage on the Semantic Web*, Busan (KR), 2007.
11. Marta Sabou, Mathieu d'Aquin, and Enrico Motta. Using the semantic web as background knowledge for ontology mapping. In *Proceedings of the ISWC workshop on Ontology Matching*, pages 1–12, Athens (GA US), 2006.
12. Marta Sabou, Jorge Gracia, Sophia Angeletou, Matthieu d'Aquin, and Enrico Motta. Evaluating the semantic web: A task-based approach. In *Proceedings of the 6th International Semantic Web Conference (ISWC)*, Busan (KR), 2007.
13. York Sure, Oscar Corcho, Jérôme Euzenat, and Todd Hughes, editors. *Proceedings of the ISWC workshop on Evaluation of Ontology-based tools (EON)*, Hiroshima (JP), 2004.
14. Willem Robert van Hage, Antoine Isaac, and Zharko Aleksovski. Sample evaluation of ontology-matching systems. In *Proceedings of the ISWC+ASWC International workshop on Evaluation of Ontologies and Ontology-based Tools*, Busan (KR), 2007.

Grenoble, Amsterdam, Trento, Mannheim, and Prague, December 10th, 2007

Using the AgreementMaker to Align Ontologies for the OAEI Campaign 2007[⋆]

William Sunna and Isabel Cruz

University of Illinois at Chicago, USA
`wsunna@cs.uic.edu, ifc@cs.uic.edu`

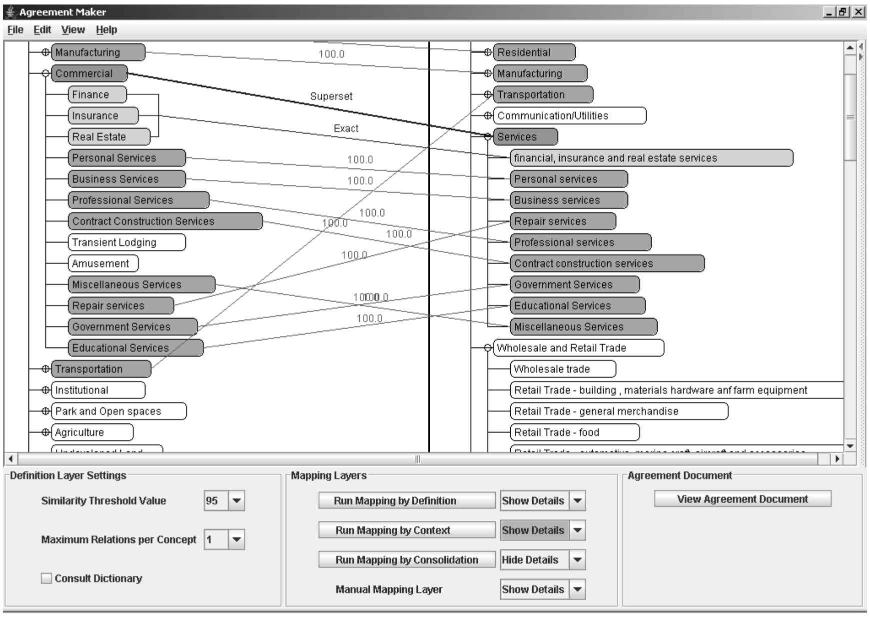
Abstract. In this paper, we present the AgreementMaker, an ontology alignment tool that incorporates the *Descendants Similarity Inheritance (DSI)* method. This method uses the structure of the ontology graphs for contextual information, thus providing the matching process with more semantics. We have tested our method on the ontologies included in the anatomy track of the OAEI 2007 campaign.

1 Presentation of the System

In distributed database applications with heterogeneous classification schemes that describe related domains, an ontology-driven approach to data sharing and interoperability relies on the alignment of concepts across different ontologies. Once the alignment is established, *agreements* that encode a variety of mappings between the concepts of the aligned ontologies are derived. In this way, users can potentially query the concepts of a given ontology in terms of other ontologies. To enable scalability both in the size and the number of the ontologies involved, the alignment method should be automatic. In order to achieve this, we have been working on a framework that supports the alignment of two ontologies. In our framework, we introduce an alignment approach that uses different matching techniques between the concepts of the aligned ontologies. Each matching technique is embedded in a what we refer to a mapping layer [2]. We have currently four layers in our framework with the possibility of adding more mapping layers in the future. The motivation behind our framework is to allow for the addition of as many mapping layers as possible in order to capture a wide range of relationships between concepts.

We have developed a tool, the AgreementMaker, which implements our approach. The user interface of our tool displays the two ontologies side by side as shown in Figure 1. We refer to the first ontology which is displayed on the left as the source ontology, and to the second ontology which is displayed on the right as the target ontology. After loading the ontologies, the domain expert can start the alignment process by mapping corresponding concepts manually or invoking procedures that map them automatically (or semi-automatically). The mapping information is displayed in the form of annotated lines connecting the matched nodes. Many choices were considered in the process of displaying the ontologies and their relationships [2].

[⋆] This research was supported in part by the National Science Foundation under Awards ITR IIS-0326284 and IIS-0513553.



Similarity Threshold Value 95

Maximum Relations per Concept 1

☐ Consult Dictionary

Fig. 1. Results of running three of the mapping layers.

1.1 Specific Techniques Used

In order to achieve a high level of confidence in performing the automatic alignment of two ontologies, a thorough understanding of the concepts in the ontologies is highly desired. To this end, we propose methods that investigate the ontology concepts prior to making a decision on how they should be mapped. We consider both the labels and the definitions of the ontology concepts and the relative positions of the concepts in the ontology tree. Our alignment method enables the user to select one of the following two matching methods: (1) applying the base similarity calculations only or (2) applying the base similarity calculations followed by the Descendant's Similarity Inheritance *(DSI)* method. The *DSI* method has been introduced to enhance the alignment results that were obtained from using the base similarity method previously proposed [2]. In what follows, we will present both our base similarity and our *DSI* methods.

Base similarity calculations The very first step in our approach is to establish initial mappings between the concepts of the source ontology and the concepts of the target ontology. These initial mappings will be a starting point for the *DSI* method. We try to find matching concepts in the target ontology for each concept in the source ontology. This is achieved by defining a similarity function that takes a concept in the source ontology and a concept in the target ontology and returns a similarity measure between them. If the similarity measure is equal or above a certain threshold decided by the domain expert, then the two concepts match each other. In order to find the base similarity measure between two concepts, we utilize the concepts' labels and in some cases their definitions as provided by a dictionary [2]. In what follows, we present the details of finding the base similarity between a concept in the source ontology and a concept in the local ontology:

- Let *S* be the source ontology and *T* be the target ontology.
- Let *C* be a concept in *S* and C' be a concept in *T*.
- We use function *base_sim*(C, C') that yields a similarity measure *M*, such that $0 \leq M \leq 1$.
- Parameter *TH* is a threshold value such that C' is matched with C when *base_sim*$(C, C') \geq TH$.
- For every concept *C* in *S*, we define the mapping set of C, denoted *MS*(C), as the set of concepts C' in T that are matched with C (i.e., *base_sim*$(C, C') \geq TH$).

Establishing base similarities between concepts of the source ontology and concepts of the target ontology may not be sufficient to achieve a high degree of precision in relating concepts in the two ontologies. To exemplify this point, we give an example in the geospatial domain, in particular, we align two ontologies describing wetlands. The first ontology describes the "Cowardin" wetland classification system [1] which is adopted in the United States. The second ontology describes the South African wetland classification system [3]. Figure 2 shows part of the "Cowardin" classification on the left, which is the source ontology, and part of the South African classification on the right, which is the target ontology. When calculating the base similarities between concepts of the two ontologies, the concept *Reef* that belongs to the *Intertidal* wetland subsystem in the source ontology will yield a base similarity measure of 100% with the concept *Reef* that belongs to the *Intertidal* wetland subsystem in the target ontology. Furthermore, it will also yield a base similarity measure of 100% with the concept *Reef* that belongs to the *Subtidal* wetland subsystem in the target ontology. This example shows that the base similarity measure is misleading because it does not correctly express the true meaning of the relationship between the two concepts, which should not be related because they belong to different wetland subsystems.

In order to eliminate such situations, we propose the Descendant's Similarity Inheritance *(DSI)* method, which reconfigures the base similarity between the concepts based on the similarity of their parent concepts.

Descendant's Similarity Inheritance (DSI) method We define the *DSI* reconfigured similarity between a concept *C* in *S* and a concept C' in *T* as *DSI_sim*(C, C'). In what follows, we present the details on how to determine *DSI_sim*(C, C'):

- Let *path_len_root*(C) be the number of edges between the concept *C* in *S* and the root of the ontology tree *S*. For example, in Figure 3, *path_len_root*$(C) = 2$. Similarly, we define *path_len_root*(C') with respect to T. For example, in Figure 3, *path_len_root*$(C') = 2$.
- Let *parent*$_i(C)$ be the *ith* concept from the concept *C* to the root of the source ontology *S*, where $0 \leq i \leq$ *path_len_root*(C). Similarly define *parent*$_i(C')$ with respect to T. For example, in Figure 3, *parent*$_1(C) = B$ and *parent*$_1(C') = B'$.

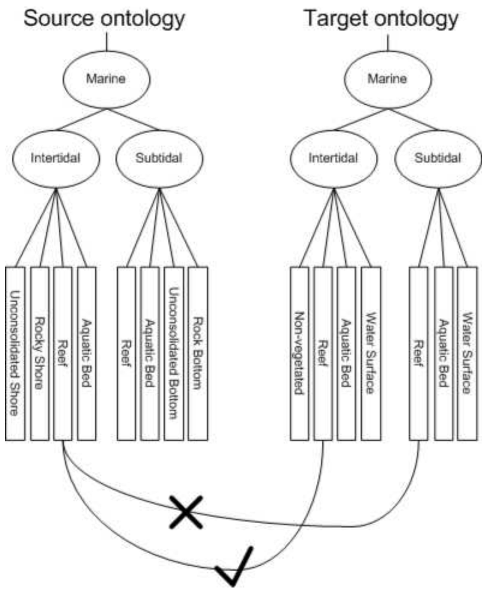


Fig. 2. An example of a case where misleading mappings may occur when two concepts have the same label.

- Define *MCP* as the *main contribution percentage*, which is the fraction of the similarity measure between *C* and C' that will be used in determining the overall *DSI_sim(C,C')*.
- We compute *DSI_sim*(C, C') as follows:

$$\mathit{MCP}\cdot \mathit{base\_sim}(C,C')+\frac{2(1-\mathit{MCP})}{n(n+1)}\sum_{i=1}^{n}(n+1-i)\mathit{base\_sim}(\mathit{parent}_i(C),\mathit{parent}_i(C')))$$

where $n = \min(\mathit{path\_len\_root}(C), \mathit{path\_len\_root}(C'))$

The main characteristic of the *DSI* method is that it allows for the parent and in general for any ancestor of a concept to play a role in the identification of the concept. Intuitively, the parent of a concept should contribute more to the identity of the concept than its grandparent. This is achieved by setting a relatively high value to *MCP*. The grandparent concept contributes more than the great grandparent, and so on, until the root is reached. This can be demonstrated by considering the example in Figure 3. In the figure, we show how the *DSI* similarity is determined between the concept *C* in the source ontology *S* (shown left) and the concept C' in the target ontology *T* (shown right) when applying the *DSI* method using an *MCP* value of 75%. The *DSI* similarity is determined by adding 75% of the base similarity between *C* and C' to 17% of the base similarity of their immediate parents (*B* and B') and finally to 8% of the base similarity of their grandparents (*A* and A'). Experiments have shown that 75% for the value of the

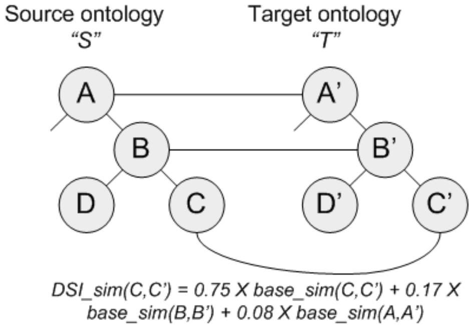


Fig. 3. Applying the DSI method to calculate the similarity between C and C'

.

MCP factor works well (in fact, any values in that neighborhood performed similarly). The following example illustrates just one such case.

Considering the case of Figure 2, the base similarity between the concepts *Intertidal* in the source ontology and the concept *Subtidal* in the target ontology is 37%. The base similarity between the concepts *Marine* in the source ontology and the concept *Marine* in the target ontology is 100%. When applying the *DSI* method with an *MCP* value of 75%, the *DSI* similarity between the concept *Reef* that belongs to the *Intertidal* wetland subsystem in the source ontology and the concept *Reef* that belongs to the *Subtidal* wetland subsystem in the target ontology will be 88%. Applying the *DSI* method again between the concept *Reef* that belongs to the *Intertidal* wetland subsystem in the source ontology and the concept *Reef* that belongs to the *Intertidal* wetland subsystem in the target ontology will yield a similarity of 100%. Therefore, we conclude that the last match is the best one (in fact the optimal one). This is just one example that shows how the *DSI* method can be useful in determining more accurate similarity measures between concepts.

1.2 Link to the system and parameters file

http://www.cs.uic.edu/˜advis/OAEI2007/align-code.zip

1.3 Link to the set of provided alignments (in align format)

The results of the three tasks for the anatomy track can be found at:
http://www.cs.uic.edu/˜advis/OAEI2007/sunna-cruz.zip

2 Anatomy Track Results

We have focused on the "Anatomy" track of the 2007 campaign. The purpose of this track is to find alignments between the ontology of the Adult Mouse Anatomy and the

NCI Thesaurus, which describes the human anatomy. The ontology of the Adult Mouse Anatomy has 2744 classes and the NCI Thesaurus has 3304 classes. Since the class IDs of all the classes of the ontologies do not describe what they refer to, a lookup file which contains the IDs of the classes and their labels has been produced. The lookup file has been used in the alignment process of the ontologies using the *DSI* method. The alignment process of the anatomy ontologies took around 9 minutes on an 1.6 GHz Intel Centrino Duo CPU with 1GB of RAM, running Windows XP.

3 Conclusions

We have presented the *Descendant's Similarity Inheritance (DSI)* method, that enhances our Base similarity method. The *DSI* method uses the structure of the ontology graph by utilizing the information associated with the descendants of each concept for contextual information thus providing the matching process with more semantics. We have applied our *DSI* method on the ontologies in the anatomy track of the OAEI 2007 campaign.

In addition to the *DSI* method, we have proposed the *Sibling's Similarity Contribution (SSC)* method [4], which uses the relationships between sibling concepts to further enhance the process of the alignment. For the purposes of this campaign, we decided to only apply the *DSI* method which performs better than the *SSC* in most of the alignment test cases we considered.

References

1. L. M. Cowardin, V. Carter, F. C. Golet, and E. T. LaRoe. Classification of Wetlands and Deepwater Habitats of the United States. U.S. Department of the Interior, Fish and Wildlife Service, Washington, D.C. Jamestown, ND: Northern Prairie Wildlife Research Center Online. `http://www.npwrc.usgs.gov/resource/wetlands/classwet/index.htm` (Version 04DEC1998), 1979.
2. I. F. Cruz, W. Sunna, N. Makar, and S. Bathala. A Visual Tool for Ontology Alignment to Enable Geospatial Interoperability. *Journal of Visual Languages and Computing*, 18(3):230–254, 2007.
3. J. Dini, G. Gowan, and P. Goodman. South African National Wetland Inventory, Proposed Wetland Classification System for South Africa. `http://www.ngo.grida.no/soesa/nsoer/resource/wetland/inventory`, 1998.
4. W. Sunna and I. F. Cruz. Structure-based Methods to Enhance Geospatial Ontology Alignment. In *Second International Conference on Geospatial Semantics (GeoS)*, volume 4853 of *Lecture Notes in Computer Science*. Springer, 2007. To appear.

Hybrid Alignment Strategy for Anatomical Ontologies Results of the 2007 Ontology Alignment Contest

Songmao Zhang[1], Olivier Bodenreider[2]

[1] Institute of Mathematics, Academy of Mathematics and Systems Science
Chinese Academy of Sciences, Beijing, P. R. China
[2] U.S. National Library of Medicine, NIH, Bethesda, Maryland, USA
[1] `smzhang@math.ac.cn`, [2] `olivier@nlm.nih.gov`

Abstract. An ontology is a formal representation of a domain modeling the entities in the domain and their relations. When a domain is represented by multiple ontologies, there is a need for creating mappings among these ontologies in order to facilitate the integration of data annotated with and reasoning across these ontologies. The objective of this paper is to present our experience in aligning two medium-size anatomical ontologies and to reflect on some of the issues and challenges encountered along the way. The anatomical ontologies under investigation are the Adult Mouse Anatomy (MA) and the anatomy subset of the NCI Thesaurus (NCI). We also use the Foundational Model of Anatomy as a reference ontology. We present a hybrid alignment strategy for anatomical entities, combining direct and indirect alignment techniques, both supported by the NLM Anatomy Ontology Alignment System (AOAS). Overall, the hybrid strategy combining direct and indirect alignment techniques identified 1,338 matches between MA and NCI, accounting for about 49% of all MA concepts and 41% of all NCI concepts. 1,007 matches are shared by both alignments, leaving 277 matches specific to the direct alignment and 54 specific to the indirect alignment.

1 Presentation of the system

An ontology is a formal representation of a domain modeling the entities in the domain and their relations. Many domains, including anatomy, are represented by multiple ontologies, with variable overlap among them. There is a need for creating mappings among these ontologies in order to facilitate the integration of data annotated with and reasoning across these ontologies [1].

1.1 Introduction

Over the past five years, as part of the Medical Ontology Research project at the U.S. National Library of Medicine, we have developed domain knowledge-based techniques for aligning large anatomical ontologies, with the objective of exploring approaches to aligning representations of anatomy differing in formalism, structure, and domain coverage [2]. In this effort, we aligned the two anatomical ontologies under investigation in the 2007 OAEI campaign, namely, the Adult Mouse Anatomy (MA) [3] and the anatomy subset of the NCI Thesaurus (NCI) [4]. More precisely, we explored two distinct approaches to aligning these two ontologies. The first approach is a direct alignment realized with the NLM Anatomy Ontology Alignment System (AOAS) [5]. The second approach is an indirect alignment through a reference ontology of anatomy: the Foundational Model of Anatomy (FMA) [6]. In the current study, we then combine these two alignment techniques into a hybrid strategy. To our knowledge, the combination of direct and indirect approaches represents a novel strategy for aligning ontologies.

1.2 Hybrid alignment strategy

The hybrid strategy for aligning anatomical entities combines direct and indirect alignment techniques, both supported by the NLM Anatomy Ontology Alignment System (AOAS).

Direct alignment

The direct alignment technique consists in the identification of one-to-one concept mappings between the MA and NCI using lexical resemblance between concept names, followed by the validation of the mappings through shared hierarchical paths among concepts across ontologies. A brief presentation of the method is given below. The interested reader is referred to [2] for further details.

Identifying matches lexically. The lexical alignment compares two ontologies at the term level, by exact match and after normalization. This process makes the source and target terms potentially compatible by eliminating such inessential differences as inflection, case, hyphen, and word-order variation. Both preferred terms and synonyms in the two ontologies are used in the alignment process. Moreover, synonymy in the Unified Medical Language System (UMLS) Metathesaurus is used to identify additional matches. For example, *Profunda femoris artery* in MA and *Deep femoral artery* in NCI, although lexically different, are considered as a match because they name the same anatomical concept in the UMLS. Our method does not address partial lexical matches.

Validating matches structurally. In order to facilitate the comparison of relations across ontologies, the structural alignment first normalizes the hierarchical relations among concepts, including *IS-A* and *PART-OF*. Missing inverse relations are complemented as necessary. Inference rules are used to generate a partitive relation between a specialized part and the whole or between a part and a more generic whole. Once all relations are represented consistently, the structural alignment is applied on the matches resulting from the lexical alignment in order to identify similar relations to other matches across ontologies (i.e., shared hierarchical paths). For example, the matching concepts *Forelimb* in MA and *Upper extremity* in NCI exhibit similar relations to other matches in the two ontologies, including *Limb* (through *IS-A*), *Arm* and *Hand* (through *HAS-PART*) across ontologies. Such structural similarity is used as positive evidence for the alignment. Instead of similar relations, one match may exhibit relations to other matches in opposite directions in the two ontologies. Such relations suggest a structural conflict across ontologies. For example, in MA *Pericardial cavity* is in HAS-PART relationship to *Pericardium*, while in NCI *Pericardial cavity* is defined as part of *Pericardium*. These conflicts are used as negative evidence for the alignment, indicating the semantic incompatibility between concepts across ontologies despite their lexical resemblance. In some cases, no evidence (positive or negative) is found to support or reject the match.

Indirect alignment

An alternative to aligning MA and NCI concepts directly consists in the identification of mappings through a reference ontology, here, the Foundational Model of Anatomy (FMA). In practice, the following method was used for automatically deriving a mapping between MA and NCI from the two direct alignments MA-FMA and NCI-FMA. When a FMA concept C_F is aligned with both a MA concept ({MA: C_M, FMA: C_F}) and a NCI concept ({NCI: C_N, FMA: C_F}), the concepts C_M and C_N are automatically aligned ({MA: C_M, NCI: C_N}).

For example, as shown in **Figure 1**, the direct alignment MA-FMA identifies the match {MA: *Forelimb*, FMA: *Upper limb* (synonym: *Forelimb*)}, which is supported by positive evidence. The direct alignment NCI-FMA identifies the match {NCI: *Upper extremity*, FMA: *Upper limb* (synonym: *Upper extremity*)}, also supported by positive evidence. Therefore, the match {MA: *Forelimb*, NCI: *Upper extremity*} is derived automatically, through the FMA concept *Upper limb*, supported by positive structural evidence in both direct alignments.

The direct alignment method between MA and FMA (and between NCI and FMA) follows the same steps described above for the direct alignment between MA and NCI. Additional knowledge augmentation techniques are used to acquire and normalize relations from the FMA, including the extraction of partitive reified relations from terms (e.g., <*Heel*, PART-OF, *Foot*> was derived from <*Heel*, IS-A, *Subdivision of foot*>).

In the indirect alignment through a reference ontology, where an indirect match (e.g., between C_M and C_N) results from two direct matches to the reference (e.g., between C_M and C_F and between C_N and C_F), the evidence for the indirect match is the combination of the evidence for the two direct matches to the reference. Intuitively, the strongest mappings correspond to those cases supported by positive evidence in both direct alignments. Weaker mappings are identified when positive evidence is found in only one of the two direct alignments and no evidence is found in the other. The weakest mappings correspond to cases where no evidence is found in either direct alignment. Finally, mappings exhibiting negative evidence in either direct alignment are rejected.

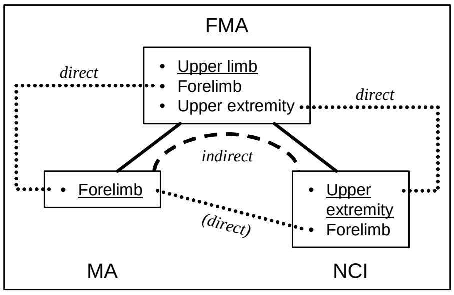


Figure 1. Indirect MA-NCI alignment through FMA

Combining direct and indirect alignment techniques

Although most mappings are identified by both direct and indirect alignments, we showed that each technique also produced specific mappings. Therefore, we propose to combine the mappings yielded by the two alignment techniques. In practice, a mapping can be identified by either technique or by both techniques. Moreover, the degrees of confidence attached to each mapping in the direct and indirect alignments can also be combined, resulting in a composite score reflecting the overall degree of confidence in the mapping.

In the direct alignment, direct matches can be supported by positive structural evidence, negative structural evidence, or no evidence. In the indirect alignment, as mentioned earlier, the degree of confidence in the matches goes from strong positive evidence (in both direct alignments), weak positive evidence (in only one direct alignment), no evidence (in either alignment). In all cases, the presence of negative evidence in either alignment causes the mapping to be rejected.

Intuitively, for the combined alignment strategy, the strongest mappings correspond to matches supported by positive evidence in both direct and indirect alignments. The presence of negative evidence in either alignment causes the mapping to be rejected. Otherwise, an additive model is used to combine degrees of evidence. The credit given to those matches specific to either direct or indirect alignment corresponds to one half of what they would receive if confirmed in the other alignment. The numeric scores reported in the results for the various combinations are summarized in **Table 1**.

Table 1. Degree of confidence for various combinations of support in direct and indirect alignments

		Indirect alignment				
		Strong positive evidence	Weak positive evidence	No evidence	Negative evidence	No match
Direct alignment	Positive evidence	1.0	.875	.75	0	.50
	No evidence	.75	.625	.50	0	.25
	Negative evidence	0	0	0	0	0
	No match	.50	.375	.25	0	

1.3 Adaptations made for the evaluation

In previous alignment experiments, we purposely stayed away form any particular formalism in order to avoid overfitting tools and techniques to a given formalism. Since our alignment approach is not designed to specifically take advantage of OWL, we simply extracted concept names and <concept, relationship, concept> triples from the class definitions in OWL. In addition to the files provided for MA and NCI (in OWL DL), we used the file provided last year in the OAEI 2006 campaign for the FMA (in OWL Full) as our reference ontology.

In practice, we used ***rdf:ID*** to identify concepts, ***rdf:label*** to acquire concept names, ***oboInOwl:hasRelatedSynonym*** to acquire synonyms, and ***rdfs:subClassOf*** to acquire taxonomic relations. The various partitive relations represented in the FMA (e.g., *part_of*, *constitutional_part_of*, *regional_part_of*) were acquired using the corresponding properties and merged – for alignment purposes – with *UNDEFINED_part_of* in MA and NCI. The other properties (e.g., ***oboInOwl:Definition***) were not used in the alignment.

1.5 Link to the set of provided alignments (in align format)

The result of our alignment for the ‘anatomy’ data set is available at: http://mor.nlm.nih.gov/pubs/supp/2007-oaei-sz/Zhang&Bodenreider.rdf in the format specified by the OAEI organizers at: http://oaei.ontologymatching.org/2007/.

2 Results

The only data set for which we report results is ‘anatomy’. Overall, the hybrid strategy combining direct and indirect alignment techniques identified 1,338 matches between MA and NCI, accounting for about 49% of all MA concepts and 41% of all NCI concepts.

Acquiring concept names and relations. The main characteristics of the data sets under investigation are listed in **Table 2**, including the number of classes, concept names, and types of partitive relationships. The number of *IS-A* and partitive relations extracted from the OWL file and generated by complementation, augmentation and inference is shown in **Table 3**. Not surprisingly, in the three ontologies, a majority of relations come from inference, which performs similarly to a transitive closure of the hierarchical relations. Also listed in **Table 3** is the small number of relations removed from the FMA because they create cycles.

Table 2. Main characteristics of the three ontologies

#	MA	NCI	FMA
Concepts	2,738	3,298	72,560
Normalization of preferred terms	3,024	3,586	98,942
Synonyms	324	1,796	44,597
Normalization of synonyms	370	1,947	60,051
Part-of relationships	1	1	7

Table 3. Number of relations in the three ontologies

Types of relations	MA	NCI	FMA
Explicitly represented is-a	2,857	3,761	72,560
Explicitly represented partitive relations	1,631	1,662	101,161
Explicitly represented associative relations	0	0	48,804
Complemented inverse-isa	2,857	3,761	72,560
Complemented partitive relations	1,631	1,662	3,561
Complemented associative relations	0	0	11,697
Removed because of cycles	0	0	- 40
Augmented	0	0	169,378
Inferred	23,504	35,092	5,169,034
Total	32,480	45,938	5,648,715

Three direct alignments. Results for three direct alignments are summarized in **Table 4**. The alignment NCI-FMA yielded the largest number of matches (2,314) and MA-NCI the smallest (1,284). A very small number of conflicts (matches exhibiting negative evidence) was identified in each direct alignment. In the three direct alignments, a vast majority of the matches (> 96%) was supported by positive structural evidence. No evidence (positive or negative) was found for 2-4% of the matches in three direct alignments. For example, although *Elbow joint* has relations to other matches in both MA (e.g., PART-OF *Forelimb*) and NCI (e.g., PART-OF *Skeletal system*), none of these relations are shared.

Table 4. Three direct alignments

	MA - NCI **1,284 matches**	**MA - FMA** **1,562 matches**	**NCI - FMA** **2,314 matches**
No evidence	25 (1.9%)	49 (3.1%)	85 (3.7%)
Positive evidence	1,254 (97.7%)	1,507 (96.5%)	2,215 (95.7%)
Negative evidence	5 (0.4%)	6 (0.4%)	14 (0.6%)

Indirect alignment MA-NCI through FMA. 1,061 matches between MA and NCI were automatically derived from the 1,562 matches in the direct alignment MA-FMA and the 2,314 matches in NCI-FMA. 1,008 of them (95%) received positive structural evidence in both direct alignments MA-FMA and NCI-FMA, 15 (1.4%) received negative evidence in one of the two direct alignments, and 38 (3.6%) received no evidence in at least one of the two direct alignments.

Combining direct and indirect alignments. Of the 1,284 matches obtained by direct alignment and 1,061 matches derived from the indirect alignment through the FMA, 1,007 matches are shared by both alignments, leaving 277 matches specific to the direct alignment and 54 specific to the indirect alignment. The repartition of the matches with respect to the degree of confidence is presented in **Figure 2**.

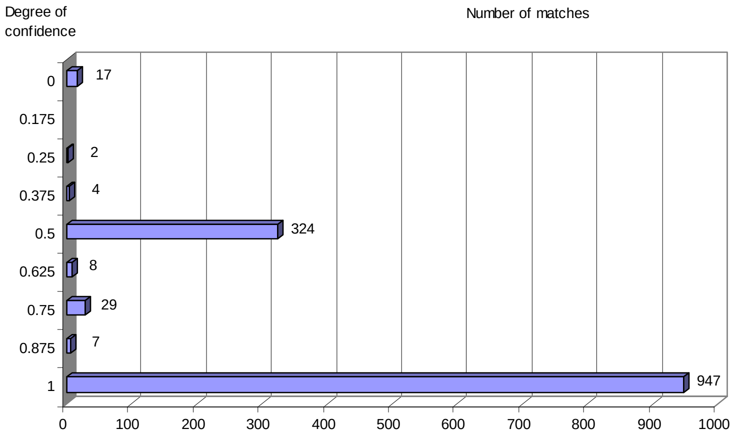


Figure 2. Repartition of the matches with respect to the degree of confidence

Matches in common. Of the 1,007 matches common to both alignments, 947 (94%) received positive evidence in the direct alignment and strong positive evidence in the indirect alignment. 15 matches (1.5%) received negative evidence in at least one of the alignments and were therefore eliminated. All but one of the remaining matches received positive evidence in one alignment and no evidence in the other. One match did not receive any evidence in either alignment. For example, the match {MA: *Forelimb*, NCI: *Upper Extremity*}is common to both alignments and receives positive evidence in the direct alignment and strong positive evidence in the indirect alignment.

Matches specific to the direct alignment. The direct alignment yielded 277 matches that were not identified by the indirect alignment. All but one received positive evidence and one received no evidence. For example, the match {MA: *tibial artery*, NCI: *Tibial Artery*} is specific to the direct alignment and receives positive evidence.

Matches specific to the indirect alignment. The indirect alignment yielded 54 matches that were not identified by the direct alignment. A majority of them (51 or 94%) received positive evidence (weak in 4 cases) and one received no evidence. Two matches received negative evidence and were eliminated. For example, the match {MA: *ovary follicle*, NCI: *Ovarian Follicle*} is specific to the indirect alignment (through the FMA concept *Follicle of ovary*) and receives strong positive evidence.

3 General comments

3.1 Comments on the results

The objective of combining several approaches is to increase recall and precision over each technique used in isolation. We note a significant increase in recall compared to the indirect alignment (1,338/1,061 or +26%) and a more modest increase compared to the direct alignment (1,338/1,284 or +4%). From a qualitative perspective, although most matches are common to both approaches, each technique contributes a number of specific matches.

In terms of precision, the use of two independent techniques represents a form of cross-validation of the matches. While insufficient for a completely automated mapping strategy, this validation method can help establish a degree of confidence in the matches, based on corroborated findings across alignment techniques. This degree of confidence can be used to guide the effort of manually curating the matches, by targeting those matches identified with a lesser confidence.

The scoring strategy presented in this paper (**Table 1**) also offers a more precise framework for estimating precision. In fact, while most alignment systems use some kind of arbitrary threshold over a continuous score between 0 and 1, the scheme we use for scoring the quality of the mappings is totally transparent and mathematically sound. Unlike most systems, the scoring system based on our hybrid approach attaches an explanation to each score (e.g., .75 consistently means that the lexical match is supported by positive evidence in one alignment, but uncorroborated in the other). The scoring scheme is also mathematically sound as it simply averages the scores from both direct and indirect alignments, except in the case where negative evidence is found, in which case the matches are excluded.

3.2 Discussion on the way to improve the proposed system

The strengths and weaknesses of our system have been analyzed in previous papers [7]. The major difference with other systems is that we take advantage of domain knowledge throughout the mapping process. For example, we use specific tools and resources, including normalization techniques developed for biomedical terms and synonyms from the Unified Medical Language System. We also developed techniques specific to the anatomical ontologies under investigation in order to represent explicitly relations implicitly present in these ontologies. These additional synonyms and relations increase the chances of identifying matches both at the lexical and structural level.

In a recent analysis of the matches produced by several systems for the alignment between FMA and GALEN in the 2006 OAEI campaign [8], we noted a small number of false negatives in our system. Alignment systems such as FALCON [9] and PRIOR [10] are based on an information retrieval paradigm and accommodate partial matches. Thus, they tend to handle gracefully the kinds of spelling variation (and sometimes misspelling) encountered in some anatomical ontologies. In contrast, with a stricter, domain-specific model of lexical similarity, our system has a better precision, but is more sensitive to missing synonyms and misspellings.

As mentioned before, our alignment strategy does not take advantage of the textual definitions available for most concepts in NCI. Textual definitions are not systematically present in anatomical ontologies and require natural language processing to be meaningfully interpreted. For these two reasons, we did not use them, nor do we have any plans to do so in the future.

3.3 Comments on the OAEI 2007 procedure

Unlike the FMA and GALEN in the 2006 OAEI campaign, provided in OWL Full after conversion from their native environments (Protégé-frames for the FMA and GRAIL for GALEN), MA and NCI are provided this year in OWL DL, which is the native format for NCI and an easy conversion from the Open Biological Ontology (OBO) format for MA. This simpler formalism, along with the smaller size and limited complexity of MA and NCI compared to FMA and GALEN, made the 2007 OAEI campaign for anatomy more interesting as it made it possible for the participants to focus on alignment issues.

Moreover, there exists a gold standard mapping between MA and NCI developed by biologists at the Jackson Laboratory. This will be the first time in the young history of the OAEI campaign that such a reference mapping is provided for anatomical ontologies. Although limited to equivalent concepts, this mapping will make the evaluation much more meaningful as we demonstrated that mere cross-validation among alignment systems was not adequate for evaluation purposes [8].

3.4 Comments on the OAEI 2007 test cases

Every year, the anatomy test case is one of the most challenging test cases in the OAEI campaign, generally because of the sheer size of the ontologies to be aligned, as well as the level of domain expertise required. Nonetheless, the anatomy task has attracted a growing number of participants over the years. However, we have shown that the performance of most participating systems, usually general-purpose, domain-independent alignment systems applied to the anatomy task, is generally mediocre [8]. Beside absolute performance, it is interesting to see the progress made by some systems (e.g., FALCON), due in part to the stimulation generated by the OAEI campaign and enabled by the availability of large test datasets.

4 Conclusion

Anatomy is central to biomedicine and a key element to translational medicine, i.e., the effective exchange of information between the “bench” (basic research) and the “bedside” (clinical practice). While aligning large anatomical ontologies remains challenging, because of the sheer size of these resources and the need for domain knowledge, the progress made in the past years is encouraging. By providing a forum for comparing methods and datasets for evaluation purposes, the OAEI campaign has contributed significantly to this progress.

Acknowledgments. This research was supported in part by the Intramural Research Program of the National Institutes of Health (NIH), National Library of Medicine (NLM), and by the Natural Science Foundation of China (No.60496324), the National Key Research and Development Program of China (Grant No. 2002CB312004), the Knowledge Innovation Program of the Chinese Academy of Sciences, MADIS of the Chinese Academy of Sciences, and Key Laboratory of Multimedia and Intelligent Software at Beijing University of Technology.

References

1. Euzenat, J., Shvaiko, P.: Ontology matching. Springer, New York (2007)
2. Zhang, S., Bodenreider, O.: Experience in aligning anatomical ontologies. International Journal on Semantic Web and Information Systems **3** (2007) 1-26
3. Hayamizu, T.F., Mangan, M., Corradi, J.P., Kadin, J.A., Ringwald, M.: The Adult Mouse Anatomical Dictionary: a tool for annotating and integrating data. Genome Biology **6** (2005) R29
4. Sioutos, N., Coronado, S.d., Haber, M.W., Hartel, F.W., Shaiu, W.L., Wright, L.W.: NCI Thesaurus: A semantic model integrating cancer-related clinical and molecular information. Journal of Biomedical Informatics **40** (2007) 30-43

5. Bodenreider, O., Hayamizu, T.F., Ringwald, M., de Coronado, S., Zhang, S.: Of mice and men: Aligning mouse and human anatomies. Proc AMIA Symp (2005) 61-65
6. Zhang, S., Bodenreider, O.: Alignment of multiple ontologies of anatomy: Deriving indirect mappings from direct mappings to a reference. Proc AMIA Symp (2005) 864-868
7. Zhang, S., Mork, P., Bodenreider, O.: Lessons learned from aligning two representations of anatomy. In: Hahn, U., Schulz, S., Cornet, R. (eds.): Proceedings of the First International Workshop on Formal Biomedical Knowledge Representation (KR-MED 2004) (2004) 102-108
8. Zhang, S., Bodenreider, O.: Lessons learned from cross-validating alignments between large anatomical ontologies. Medinfo (2007) 822-826
9. Hu, W., Cheng, G., Zheng, D., Zhong, X., Qu, Y.: The results of Falcon-AO in the OAEI 2006 Campaign. Proceedings of the Ontology Alignment Evaluation Initiative 2006 Campaign (OAEI 2006) (2006) 118-125
10. Mao, M., Peng, Y.: PRIOR system: Results for OAEI 2006. Proceedings of the Ontology Alignment Evaluation Initiative 2006 Campaign (OAEI 2006) (2006) 165-172

ASMOV Results for OAEI 2007

Yves R. Jean-Mary[1], Mansur R. Kabuka [1,2]

[1] INFOTECH Soft, 9200 South Dadeland Blvd, Suite 620, Miami, Florida, USA 33156
[2] University of Miami, Coral Gables, Florida, USA 33124
{reggie, kabuka}@infotechsoft.com

Abstract. Numerous ontology alignment algorithms have appeared in the literature in recent years, but only a few make use of the semantics enclosed within the ontologies in order to improve the accuracy. In this paper, we present the Automated Semantic Mapping of Ontologies with Validation (ASMOV) algorithm for ontology alignment. We first provide a brief overview of the algorithm followed by an analysis of its results on the 2007 Ontology Alignment Evaluation Initiative tests. We conclude the paper by identifying the specific strengths and weaknesses of ASMOV, while pointing out the necessary improvements that need to be made.

1 Presentation of the System

In recent years, ontology alignment (or ontology mapping) has become popular in solving interoperability issues across heterogonous systems in the semantic web. Though many techniques have emerged from the literature [1] [6], the distinction between them is accentuated by the manner in which they exploit the features within an ontology.

ASMOV, an algorithm that automates the ontology alignment process while optionally accepting feedback from a user, uses automatically-adjusting weights based on four features of the ontologies; a more complete description of ASMOV has been presented in [4]. ASMOV computes similarity measures by analyzing the entities in the manner in which they are modeled in the ontology, and the iterative alignments produced by ASMOV are validated by a number of rules and a mapping validation process.

1.1 State, Purpose, General Statement

ASMOV is an automatic ontology matching tool which has been designed in order to facilitate the integration of heterogeneous systems, using their data source ontologies. The OAEI tests help us validate that ASMOV produces ontology alignments with high accuracy and that little user interaction is needed to correct these results. The current ASMOV prototype produces both class-to-class and property-to-property mappings, including mappings from object properties to datatype properties and vice versa.

1.2 Specific Techniques Used

The ASMOV algorithm iteratively calculates the similarity between concepts for a pair of ontologies by analyzing four features: textual description (id, label, and comment), external structure (parents and children), internal structure (property restrictions for classes; types, domains, and ranges for properties), and individual similarity. The measures obtained by comparing these four features are combined into a single confidence value using a weighted sum in a similar manner to [2]. In the case of ASMOV, the initial weights were chosen arbitrarily, and have been optimized based on the benchmark test results. During an automated pre-processing phase, ASMOV contains a mechanism that automatically adjusts the weights based on the information contained in the ontologies. For example, when analyzing the textual information in the pre-processing phase, if ASMOV cannot find meaningful words, it decreases the textual similarity weight based on predetermined rules. These rules are static and have not been adjusted for any of the OAEI 2007 tests.

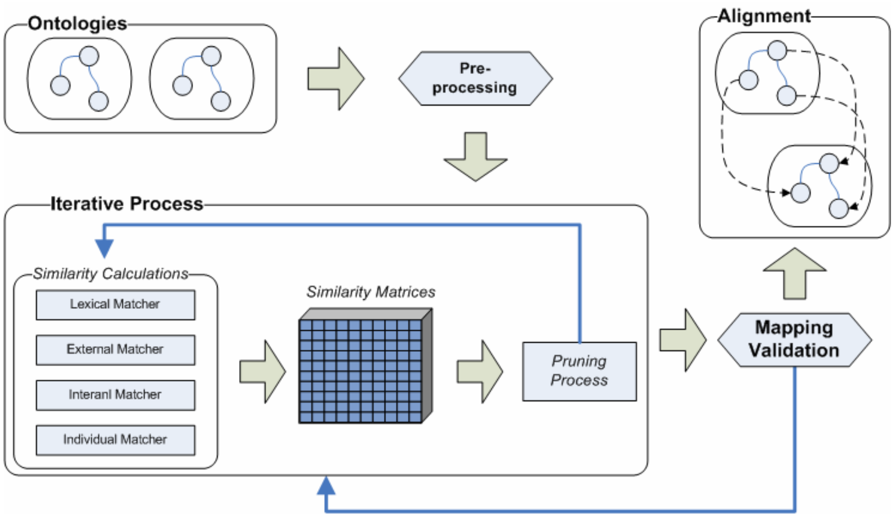


Fig. 1. The Mapping process of ASMOV

Fig. 1 illustrates the Mapping process of ASMOV. The whole process is fully automated.

In the pre-processing phase, the ontologies are loaded into memory using Jena [5]. Each class and property is wrapped and tagged with the meaning of its id and label(s). The meaning of these texts is retrieved using UMLS Metathesaurus [7] for the anatomy test and WordNet [8] for the other tests including the benchmark tests. Through a configuration parameter, a user can force the ASMOV system to use either one, neither, or both of the lexical systems. During the pre-processing phase, a quick analysis of the ontologies being mapped is performed. This analysis entails checking for the presence of properties and meaningful words within the textual description of the classes and properties; the weights are adjusted depending on the result of this analysis.

The second phase of the algorithm is the iterative process. During this phase, pairs of entities (classes and properties) are compared using the four features described earlier, with the resulting overall similarity measure (or confidence value) being stored in a 2-dimensional matrix. At the end of each iteration, a pruning process eliminates the invalid mappings by analyzing two semantic inconsistencies: crisscross mappings and many-to-one mappings. A crisscross mapping occurs whenever a source entity ($\mathcal{SE}p$) and its child ($\mathcal{SE}c$) are mapped to a target entity ($\mathcal{TE}c$) and its parent ($\mathcal{TE}p$) respectively. Many-to-one mappings are inconsistent if it cannot be asserted through the ontology that all the classes in a many-to-one mapping are either equivalent or if each of the classes is subsumed into another. The iterative process stops when the difference in confidence values for two subsequent interations is below a given threshold and no inconsistencies are found by the pruning process, or until a cyclic situation is detected.

After the iterative process is completed, a mapping validation starts. This validation process performs a structural analysis using graphs built from the alignment and information from the ontologies. The validation is performed in three phases: class validation, property validation, and concept-property validation. If any inconsistency is found by this process, the iterative process restarts at the end of the validation process. The inconsistent mappings discovered by the mapping validation process and the pruning process are retained so that ASMOV does not try to align those same entities in subsequent mappings.

1.3 Adaptations Made for the Evaluation

No special adaptations have been made to the ASMOV system in order to run the 2007 OAEI tests; however, four Java classes have been added in order to respectively run the benchmark series of tests, the anatomy tests, the director test and the conference tests, and output the results in the OAEI alignment format. Although the rules stated clearly that all alignments should be run from the same set of parameters, ASMOV was unable to run the anatomy tests under the same conditions due to its iterative nature and the large size of the anatomy ontologies. Two changes had to be made: UMLS was used instead of WordNet as the lexical reference system and the iterative process was stopped when 90% (instead of 100%) of the mappings in the similarity matrix were unchanged in two subsequent iterations. These changes were supplied to the system via properties of a parameters file and required no changes in the coding implementation of ASMOV.

1.4 Link to the ASMOV System

The ASMOV system (including the parameters file) can be downloaded from http://support.infotechsoft.com/integration/ASMOV. A document detailing our approach can also be found there.

1.5 Link to the Set of Alignments Produced by ASMOV

The results of the 2007 OAEI campaign for the ASMOV system can be found at http://support.infotechsoft.com/integration/ASMOV.

2 Results

The 2007 Ontology Alignment Evaluation Initiative campaign consists of four tracks which include: (a) a comparison track, (b) expressive ontologies, (c) directories and thesauri, and (d) a consensus workshop track. Although ASMOV was able to participate in all four tracks, only one out of four tests of the directories and thesauri track was able to be performed due to the large size of the ontologies in the other tests. ASMOV is a Java implementation which uses Jena to parse the RDF and OWL files. All tests were carried out on a PC running Windows XP Professional with a dual-core Intel Pentium processor (2.8 GHz) and 3 gigabytes of memory.

2.1 Benchmark

Because ASMOV's overall similarity calculation (or confidence value) is based on a weighted sum and the weights are automatically adjusted based on the structure of the ontologies being aligned, all tests were ran under the same conditions, the system's default configuration. For the analysis of the results, the benchmark tests are divided into three groups: tests 101-247, tests 248-266, and tests 301-304. The precision, the recall, and the time cost for the individual tests are listed in the Appendix.

2.1.1 Test 101-247

ASMOV performs very well in this set of tests, producing an overall precision and recall of 99%. The less accurate results were produced by the tests 202, 209 and 210. In test 202, although the identifiers of the entities were replaced by random strings and their labels and comments suppressed, ASMOV was still able to leverage other semantic information of the ontologies (namely the hierarchical information, the internal structure of the entities, and the similarity between individuals) in order to generate an alignment of 88% accuracy in both precision and recall. By our analysis, nine out of the eleven incorrectly mapped properties are as accurate as the ones provided within the gold standard, since these properties can only be differentiated by their lexical information (id, label, and comment) and the target ontology (202) has this information suppressed or replaced by random strings for its entities. In test 209, the identifiers and labels were replaced by synonyms and the comments suppressed; the obtained precision and recall were respectively 92% and 90%. In this test, ASMOV suffers mostly because of the measure produced by the similarity calculation. For example, the property '`abstract`' has been mapped to the property '`rights`' instead of the property '`summary`'; the latter was due to the fact that the lexical similarity measure between '`abstract`' and '`rights`' is 0.94 whereas the measure between ‘`abstract`’ and ‘`summary`’ is 0.92. In test 210, ASMOV found four incorrect mappings, producing a 97% precision and 95% recall. These errors were due to the fact that the lexical information was in French, which is not supported by WordNet.

2.1.2 Test 248-266

ASMOV’s accuracy decreased in these tests. Both the lexical information and the structure of the target ontologies have been heavily changed. As stated in [3], these tests are the most challenging ones, and it was extremely difficult to recognize the correct alignments. For these tests, the precision ranges from 0.77 to 0.91 and the recall was between 0.24 and 0.89.

2.1.3 Test 301-304

These tests represent four real-world ontologies of bibliographic references. Although there is a high lexical and structure similarity between these tests and the reference ontology, ASMOV encountered some difficulties in the mapping of datatype properties to object properties and vice versa. The overall precision and recall were respectively 85% and 82%.

The following table shows the average performance of ASMOV in terms of the groups of tests described above. The total time cost is also included.

Table 1. Overall Performance on the Benchmark Tests

	101-247	***248-266***	***301-304***	***H-mean***	***Time (sec)***
Precision	0.99	0.85	0.85	0.95	2654.001
Recall	0.99	0.68	0.82	0.90	

2.2 Anatomy

ASMOV’s implementation relies on Jena [5] in order to parse the ontologies to be aligned. We have encountered a few memory issues and found out that Jena does not scale well with large ontologies due to its reasoner. We have thus implemented solutions in ASMOV so that queries that involve the Jena reasoner are bypassed. For example, in order to answer a query for sub-classes or sub-properties, Jena needs to run its reasoner, which is not efficient when dealing with ontologies with large hierarchical structures; the solution to this issue was to maintain a map of parent-child relationships and query this map in order to retrieve the sub-classes and sub-properties. Also, in order to improve the accuracy of the alignment, we have implemented an adapter interface to the UMLS Metathesaurus [9]. With this, the semantic distance between the lexical information within entities (classes and properties) was calculated more accurately, ultimately improving the alignment produced by the system. An alignment was created for the anatomy ontologies using three different configurations: standard configuration, optimal precision configuration, and optimal recall configuration.

- Due to the large size of the anatomy ontologies and the iterative nature of the ASMOV algorithm, three parameters of the s*standard configuration* had to be changed in order to generate an alignment in an acceptable time frame. The iterative threshold has been changed form 1.0 to 0.9, which means that the iterative process of ASMOV converges once 90% of the mappings do not change in two subsequent iterations. Also the '`ignoreIdInLexicalSim`' parameter was set to false; this parameter setting indicates that the lexical matcher will ignore the local name of the entities. Since the anatomy ontologies deal with the biomedical domain, the UMLS Metathesaurus is more suitable than WordNet. Moreover, since querying the UMLS Metathesaurus for each of the thousands of labels is time-consuming, we have pre-processed the ontologies and stored the semantic information retrieved from the UMLS Metathesaurus into two separate database tables: the first storing the indexed words retrieved (An indexed word is an object that represents a word tied to its semantic meaning or UMLS concept), the second containing the hierarchy of the hypernym relationships from the indexed word in question to a fixed root chosen a priori. Only a subset of the UMLS was used, containing concepts from the NCI Thesaurus and the required dependencies.
- In order to obtain an *optimal overall precision*, the threshold of valid mappings was adjusted, from 0.5 in the system standard configuration, to 0.7. This threshold indicates the acceptable confidences for valid mappings. A value of 70% means that a mapping is deemed acceptable if its confidence value ranges from the best confidence value to 70 % of that value.
- A similar approach was used to obtain the *optimal overall recall*. However, in this case, the threshold of valid mappings was set to 0.0.

Due to the lack of a gold standard in this case, our evaluation was performed by textual analysis of the mappings within the resulting alignments. In this analysis, a correct mapping is a mapping where the entities are equivalent or synonyms, according to their labels and the UMLS Metathesuarus. The results of this analysis are illustrated by the table below in terms of precision and time cost.

Table 2. Performance of ASMOV in the Anatomy test

	Precision	***Time (sec)***
Standard configuration	0.82	54943.656
Optimal Precision	0.89	145382.953
Optimal Recall	0.75	87339.437

2.3 Directory

The standard configuration of ASMOV was used in order to run the directory tests. It took 44 minutes and 27 seconds to run and produced alignments that seem accurate for the most part. As a gold standard for these tests is not available, we are not yet able to report accuracy measures such as precision, recall, and F_1-measure.

2.4 Food

The food ontologies were too large to be run using the current prototype implementation of ASMOV. It took over an hour for Jena to parse the ontologies, and since ASMOV calculates similarity calculations for every pair of entities (class-to-class and property-to-property), the time cost for the alignment is prohibitive. We therefore opt-out of this track; we are currently working on mechanisms to improve the performance of ASMOV.

2.5 Conference

This collection of tests dealing with conference organization contains 14 ontologies. ASMOV was able to create 91 alignments from the ontologies. These ontologies were not analyzed in terms of precision and recall since no gold standard alignments were available.

3 General Comments

3.1 Comments on the Results

ASMOV performed well in the 2007 OAEI tests: the precision and recall of the benchmark tests are higher than those obtained by all entrants in OAEI 2006. This has been achieved by the use of multiple different ontology features and the ability of ASMOV to auto-adjust its weights to the characteristics of the ontologies, which enabled ASMOV to recognize correct alignments even when some information such as lexical similarity was absent. In addition, the use of a semantic validation process enables the algorithm to reject invalid mappings, and improves the overall precision and recall by 5% and 4% respectively.

The main weakness of the algorithm, in its current implementation, is its inability or inefficiency when processing large ontologies such as the anatomy and the food ontologies. We are currently working in mechanisms to improve the performance of the algorithm itself and of its implementation.

3.2 Discussions on the Way to Improve ASMOV

The mapping validation is source dependent, making the alignment process a directional one. Let’s consider two ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$; what the alignment produces when $\mathcal{O}_1$ is the source and $\mathcal{O}_2$ is the target may be different than the one obtained when the reverse occurs. As our future work, we intend to improve the mapping validation process so that it does not favor the source ontology. Also, the use of Jena as a parser seems not to be ideal, especially when the ontologies are large. For our implementation, we had to bypass some of the methods of Jena that forced calls to its reasoner and caused performance issues. We are currently investigating the use of other parsers such as the OWL-API or more powerful ones, as well the use of RDF data stores. Although ASMOV will always converge in linear time, the amount of time needed for convergence may be too great when dealing with large ontologies. Also, the use of a checksum to stop the iterative process may cause the algorithm to converge prematurely. Thus, the convergence aspect of ASMOV needs also to be revisited. As stated earlier, ASMOV is to be used as an integration tool; consequently, the confidence values need to be accurate. This accuracy is dictated by the weights which need to be optimum. Therefore, extensive testing of the weighted calculations need to be done to improve the accuracy of ASMOV. In its current state, the user interaction component of ASMOV has not been implemented yet; ASMOV will be extended to be able to present the user with a graphical interface, facilitating system-user interaction.

3.3 Comments on the OAEI 2006 Test Cases

The testing phase of ASMOV was done using the benchmark tests, which were crucial in identifying coding issues and wrong assumptions made in the design phase. In future campaigns, we would like see a benchmarking of larger ontologies so that systems can address scalability issues. Also a benchmark test in different domains such as the biomedical domain (anatomy track) would be useful for systems targeting such domains.

4 Conclusion

In this report, we provided a brief description of an automated alignment tool named ASMOV and analyzed its performance at the 2007 Ontology Alignment Evaluation Initiative campaign. The test results show that ASMOV is effective in the ontology alignment realm, and because of its flexibility, it performs well in multiple ontology domains such as bibliographic references (benchmark tests) and the biomedical domain (anatomy test). We concluded the paper by indicating the strengths and weaknesses of ASMOV, and by stating the direction of our future work.

Acknowledgments. This work is funded by the National Institutes of Health (NIH) under grant R43RR018667. The authors also wish to acknowledge the contribution of Mr. Thomas Taylor and Mr. Patrick Shironoshita of INFOTECH Soft, Inc.

References

1. Euzenat, J. State of the art on ontology alignment. http://www.inrialpes.fr/exmo/cooperation/kweb/heterogeneity/deli/. August, 2004.
2. Euzenat J. and Valtchev P. Similarity-based ontology alignment in OWL-lite. *In Proc. 15th ECAI*, Valencia (ES), 2004, 333-337.
3. Hu, W., Cheng, G., Zheng D., Zhong X., and Qu, Y: The Results of Falcon-AO in the OAEI 2006 Campaign. International Workshop on Ontology Matching (OM 2006).
4. Jean-Mary Y., Kabuka, M. ASMOV: Ontology Alignment with Semantic Validation. Joint SWDB-ODBIS Workshop on Semantics, Ontologies, Databases, September 2007, Vienna, Austria, 15-20
5. Jena from HP Labs http://www.hpl.hp.com/semweb/
6. Rahm, E. and Bernstein, P. A. A survey of approaches to automatic schema matching. The VLDB Journal, 2001, 10:334-350.
7. Unified Medical Language System (UMLS) htt://umlsks.nlm.nih.gov/
8. WordNet http://wordnet.princeton.edu/

Appendix: Raw Results

The OAEI 2007 tests were carried out on a PC running Windows XP Professional with a dual-core Intel Pentium processor (2.8 GHz) and 3 gigabytes of memory.

Matrix of Results

The following table includes the results of ASMOV in the benchmark series of tests. It illustrates the performance of the system in terms of precision (Prec.), recall (Rec.) and processing time (Time). The processing is calculated as follows: ontology parsing time + ASMOV computational time + time involved in the generation of the alignment. For the benchmark tests, the following configuration was used:

- The lexical Similarity is calculated using WordNet and Levenshtein Distance.
- The semantic distance between words was calculated using Lin’s equation.
- The threshold used to stop the iteration process was set to 1.0.
- The threshold indicating that similarity measures have not changed was set to 0.0.
- The valid mappings were the ones that had a confidence value greater or equal to 50 % of the best calculated confidence value.
- The weights associated with missing features were re-distributed proportionally so the similarity measure stayed uniformed.

Note that the same setting was used to run the directory and the consensus tests.

#	Name	Prec.	Rec.	Time (hh.mm.ss.mms)
101	Reference alignment	1.00	1.00	0.4.7.734
102	Irrelevat ontology	NaN	NaN	0.1.6.828
103	Language generalization	1.00	1.00	0.0.55.015
104	Language restriction	1.00	1.00	0.0.54.563
201	No names	1.00	1.00	0.0.52.687
202	No names, no comments	0.88	0.88	0.0.40.985
203	No comments	1.00	1.00	0.0.35.203
204	Naming conventions	1.00	1.00	0.1.0.453
205	Synonyms	1.00	1.00	0.1.15.203
206	Translation	1.00	0.99	0.0.55.937
207		1.00	0.99	0.0.53.563
208		1.00	1.00	0.0.37.093
209		0.92	0.90	0.1.4.578
210		0.97	0.95	0.0.47.532
221	No specialisation	1.00	1.00	0.0.59.687
222	Flatenned hierachy	1.00	1.00	0.0.55.766
223	Expanded hierarchy	1.00	1.00	0.1.3.719
224	No instance	1.00	1.00	0.0.39.281
225	No restrictions	1.00	1.00	0.0.53.0
228	No properties	1.00	1.00	0.0.38.906
230	Flatenned classes	0.99	1.00	0.1.1.438
231		1.00	1.00	0.0.53.844
232		1.00	1.00	0.0.39.875
233		1.00	1.00	0.0.41.672
236		1.00	1.00	0.0.23.797
237		1.00	1.00	0.0.40.75
238		1.00	1.00	0.0.54.297
239		0.97	1.00	0.0.39.11
240		0.97	1.00	0.0.43.171
241		1.00	1.00	0.0.24.766
246		0.97	1.00	0.0.24.156
247		0.94	0.97	0.0.28.312
248		0.86	0.82	0.0.50.922
249		0.89	0.89	0.0.36.328
250		0.91	0.30	0.0.22.281
251		0.83	0.77	0.1.27.407
252		0.87	0.87	0.0.40.734
253		0.85	0.81	0.0.46.203
254		0.83	0.30	0.0.22.031
257		0.91	0.30	0.0.22.36
258		0.82	0.76	0.1.25.047
259		0.87	0.87	0.0.42.828
260		0.78	0.24	0.0.22.407
261		0.91	0.30	0.0.25.578
262		0.83	0.30	0.0.22.281
265		0.77	0.34	0.0.22.734
266		0.91	0.30	0.0.25.485
301	BibTeX/MIT	0.93	0.82	0.0.50.343
302	BibTeX/UMBC	0.68	0.58	0.1.20.563
303	Karlsruhe	0.75	0.86	0.2.42.141
304	INRIA	0.95	0.96	0.0.53.406

DSSim - Managing Uncertainty on the Semantic Web

Miklos Nagy[1], Maria Vargas-Vera[2] and Enrico Motta[2]

[1] Knowledge Media Institute (Kmi)
The Open University
Walton Hall, Milton Keynes, MK7 6AA, United Kingdom
`mn2336@student.open.ac.uk;e.motta@open.ac.uk`
[2] Department of Information Systems
Poznan University of Economics
al. Niepodleglosci 10, 60-967 Poznan, Poland
`maria@kie.ae.poznan.pl`

Abstract. Managing uncertainty on the Semantic Web can potentially improve the ontology mapping precision which can lead to better acceptance of systems that operate in this environment. Further ontology mapping in the context of Question Answering can provide more correct results if the mapping process can deal with uncertainty effectively that is caused by the incomplete and inconsistent information used and produced by the mapping process. In this paper we introduce our algorithm called "DSSim" and describe the improvements that we have made compared to OAEI 2006.

1 Presentation of the system

1.1 State, purpose, general statement

The problem of mapping two ontologies effectively and efficiently is a necessary precondition to integrate information on the Semantic Web. In recent years different research communities have proposed[1] a wide range of methods for creating such mappings. The proposed methods usually combine syntactic and semantic measures by introducing different techniques ranging from heuristics to machine learning. While these methods perform well in certain domains the quality of the produced mappings can differ from domain to domain depending on the specific parameters defined in the methods e.g. tuning similarity threshold. Considering Question Answering systems like AQUA [2, 3] which answers queries over heterogeneous sources described by their ontologies, it is very important how its mapping algorithm performs in terms of mapping precision. Our objective is to produce a method that does not depend on any fine tuned internal parameters for a specific domain or does not assume having large amount of data samples a-priory for machine learning or Bayesian probability assessment[4]. Our hypothesis is that the correctness of different similarity mapping algorithms is always heavily dependent on the actual content and conceptual structure of these ontologies which are different even if two ontologies have been created on the same domain but with different purpose. Therefore from the mapping point of view these ontologies will always contain inconsistencies, missing or overlapping elements and different conceptualisation of the same terms which introduces a considerable amount of uncertainty into the mapping process.

1.2 Specific techniques used

Our proposed method works with two ontologies, which contain arbitrary number of concepts and their properties.

$$O_1 = \{C_1, .., C_n; P_1, .., P_n; I_1, .., I_n\}$$
$$O_2 = \{C_1, .., C_m; P_1, .., P_m; I_1, .., I_m\}$$

where O represents a particular ontology, C, P and I the set of concepts, properties and instances in the ontology.

In order to assess similarity we need to compare all concepts and properties from O_1 to all concepts and properties in O_2. Our similarity assessments, both syntactic and semantic produce a sparse similarity matrix where the similarity between C_n from O_1 and C_m in O_2 is represented by a particular similariy measure between the i and j elements of the matrix as follows:

$$SIM := (s_{i,j})_{n\times m}$$
$$1 \leq i \leq n \, and \, 1 \leq j \leq m$$

where SIM represents a particular similarity assessment matrix, s is a degree of similarity that has been determined by a particular similarity e.g. Jaccard or semantic similarity measure. We consider each measure as an ”expert” which assess mapping precision based on its knowledge. Therefore we assume that each similarity matrix is a subjective assessment of the mapping what needs to be combined into a coherent view. If combined appropriately this combined view provides a more reliable and precise mapping that each separate mapping alone. However one similarity measure or some technique can perform particularly well for one pair of concepts or properties and particularly badly for another pair of concepts or properties, which has to be considered in any mapping algorithm.

In our ontology mapping method we assume that each expert carries only partial knowledge of the domain and can observe it from its own perspective where available prior knowledge is generally uncertain and subjective. In order to represent subjective probabilities in our system we use the Dempster-Shafer theory of evidence [7], which provides a mechanism for modeling and reasoning uncertain information in a numerical way, particularly when it is not possible to assign belief to a single element of a set of variables. Consequently the theory allows the user to represent uncertainty for knowledge representation, because the interval between support and plausibility can be easily assessed for a set of hypothesizes. Missing data (ignorance) can also be modeled by Dempster-Shafer approach and additionally evidences from two or more sources can be combined using Dempster's rule of combination. The main advantage of the Dempster-Shafer theory is that it provides a method for combining the effect of different learned evidences to establish a new belief by using Dempster's combination rule.

The following elements have been used in our system in order to model uncertainty:

Frame of Discernment(Θ) :finite set representing the space of hypothesizes. It contains all possible mutually exclusive context events of the same kind.

$$\Theta = \{H_1, ..., H_n, ...H_N\} \tag{1}$$

In our method Θ contains all possible mappings that have been assessed by the particular expert.

Evidence:available certain fact and is usually a result of observation. Used during the reasoning process to choose the best hypothesis in Θ. We observe evidence for the mapping if the expert detects that there is a similarity between C_n from O_1 and C_m in O_2.

Belief mass function (m): is a finite amount of support assigned to the subset of Θ. It represents the strength of some evidence and

$$\sum_{A \subseteq \Theta} m_i(A) = 1 \tag{2}$$

where m$_i$(A) is our exact belief in a proposition represented by A that belongs to expert *i*. The similarity algorithms itself produce these assignment based on different similarity measures. As an example consider that O_1 contains the concept "paper" which needs to be mapped to a concept "hasArticle" in O_2. Based on the WordNet we identify that the concept "article" is one of the inherited hypernyms of "paper", which according to both JaroWinkler(0.91) and Jaccard(0.85) measure [8] is highly similarity to "hasArticle" in O_2. Therefore after similarity assessment our variables will have the following belief mass value:

- $m_{\text{expert1}}(O_1 \{paper, article, communication, publication\},$
  $O_2 \{hasArticle\}) = 0.85$
- $m_{\text{expert2}}(O_1 \{paper, article, communication, publication\},$
  $O_2 \{hasArticle\}) = 0.91$

In practice we assess up to 8 inherited hypernyms similarities with different algorithms (considered as experts) which can be combined based on the combination rule in order to create a more reliable mapping. Once the combined belief mass functions have been assigned the following additional measures can be derived from the available information.

Belief: amount of justified support to A that is the lower probability function of Dempster, which accounts for all evidence E_k that supports the given proposition A.

$$belief_i(A) = \sum_{E_k \subseteq A} m_i(E_k) \tag{3}$$

An important aspect of the mapping is how one can make a decision over how different similarity measures can be combined and which nodes should be retained as best possible candidates for the match. To combine the qualitative similarity measures that have been converted into belief mass functions we use the Dempster's rule of combination and we retain the node where the belief function has the highest value.

Dempster's rule of combination:Suppose we have two mass functions $m_i(E_k)$ and $m_j(E_{k'})$ and we want to combine them into a global $m_{ij}(A)$. Following Dempster's combination rule

$$m_{ij}(A) = m_i \oplus m_j = \sum_{E_k E_{k'}} m_i(E_k) * m_j(E_{k'}) \qquad (4)$$

where *i* and *j* represent two different experts.

The belief combination process is computationally very expensive and from an engineering point of view, this means that it not always convenient or possible to build systems in which the belief revision process is performed globally by a single unit. Therefore, applying multi agent architecture is an alternative and distributed approach to the single one. Our algorithm takes all the concepts and its properties from the different external ontologies and assesses similarity with all the concepts and properties in the query graph.

1.3 Adaptations made for the evaluation

Our mapping algorithm which is originally based on multi agent architecture has been re-implemented as a standalone mapping process which uses the common WordNet dictionary which is considered more general knowledge then originally we assume in or architecture. Originally our mapping process receives query fragments from the AQUA system where the query fragments contain several concept names and their properties. For the evaluation we modified our mapping process so we consider the individual concept or property names as query fragments which contain less information about the possible mapping then the query fragments that we originally receive from the AQUA system.

Creating the particular ontology mappings in the context of question answering is ideally an iterative process where the users are involved in the loop as well. In a real case scenario the users pose different questions that contain both concepts and properties of a particular domain. This information then can be used to query the different ontologies, create mapping between its concepts and properties that can be used to answer the particular query. For the OAEI 2006[6] we have implemented an iterative closed loop which creates the mapping without any human interaction. Based on this implementation we have modified our process for the OAEI 2007 which works as follows:

1. We take a concept or property from ontology 1 and consider (refer to it from now) it as the query fragment that would normally be posed by a user. Our algorithm consults WordNet in order to augment the query concepts and properties with their hypernyms.
2. We take syntactically similar concepts and properties to the query graph from ontology 2 and build a local ontology graph that contains both concepts ans properties together with the close context of the local ontology fragments.
3. Different similarity and semantic similarity algorithms (considered as different experts in evidence theory) are used to assess quantitative similarity values (converted into belief mass function) between the nodes of the query and ontology fragment which is considered as an uncertain and subjective assessment.

4. Then the similarity matrixes are used to determine belief mass functions which are combined using the Dempster's rule of combination. Based on the combined evidences we select those mappings in which we calculate the highest belief function.
5. The selected concepts are added into the alignment.

The overview of the mapping process is depicted on figure 1.

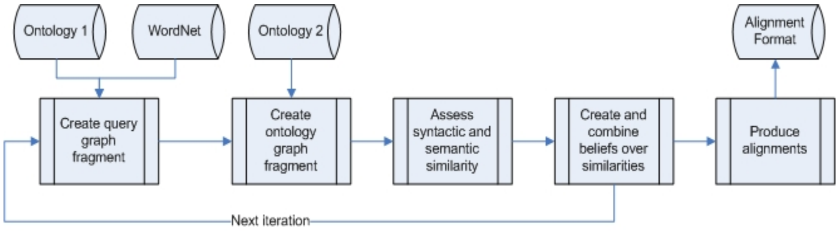


Fig. 1. The iterative mapping process

In order to avoid a complex graph of relationships in the query and the ontology fragments we need to define a reasonable limit on the number of hypernyms, which are extracted from the WordNet. To define such a limit is also desirable when we carry out the belief combination since all extracted terms represent a variable where each similarity value needs to be combined with the Dempster's rule of combination. The combination rule implies that the problem space increases exponentially with the number of variables therefore the proper definition of this limit can considerably affect the scalability of our system.

1.4 Link to the system and parameters file

http://kmi.open.ac.uk/people/miklos/OAEI2007/tools/DSSim.zip

1.5 Link to the set of provided alignments (in align format)

http://kmi.open.ac.uk/people/miklos/OAEI2007/results/DSSim.zip

2 Results

2.1 Benchmark

Based on the results of the benchmarks we have improved our algorithm compared to the OAEI 2006 results in terms of recall. The improvement was achieved by introducing instance level comparisons for the classes and properties. Nevertheless there is considerable room for improvement since we did not achieve similar results compared to the best performing systems for the tests 248-266 in terms of recall. The performance of our algorithm has also been improved considering the execution time. In general the benchmarks are excellent for improving the algorithm since we can calculate the recall and precision any time which helps a lot evaluating the impact of a particular improvement. This year we have tried to make use of the *rdfs:label* tags but it did not improve the recall or precision.

2.2 Anatomy

The anatomy test has proved quite manageable considering the execution time. During our experiments our algorithm has always created the mappings within 2 hours. We had to use the *rdfs:labels* for the comparison but we could not make use of the *oboInOwl:Synonym* tags. The usage of the labels has introduced complexity retrieving hypernyms from WordNet since it is quite challenging to split the label into terms that can be used for querying the WordNet.

2.3 Directory

The directory test as well has been managable in terms of execution time. In general the large number of small scale ontologies made it possible to verify some mappings for some cases. The tests contain only classes without any labels but in some cases different classes have been combined into one class e.g. “News_and_Media” which introduces certain level of complexity for determining synomyms using any background knowledge.

2.4 Food

The food test was extremely challenging due to the large number of concepts in the ontologies. We had to split up the original files into 8 parts and carry out the mapping one by one. Additionally we have developed a SKOS parser which can create smaller OWL chunks from the SKOS and run the mapping algorithm on it. As a result of this split we could not consider *rdfs:subClassOf* relationships between classes since it cannot be guaranteed that we would find the super classes in the same ontology chunk. The run time was around 1 week even though 2 parallel processors were used to run the mapping algorithm. During the SKOS OWL conversion we did not consider *skos:ConceptScheme* elements.

2.5 Environment

The environment test was the extension of the food test therefore it represented similar complexity in terms of run time performance. Nevertheless the GEMET SKOS contain smaller number of concepts compared to the food ontologies but we had to split up the ontologies into 2 separate parts. This implies that the *rdfs:subClassOf* relations have also not been considered which might have a negative impact on the mapping precision and recall. The mapping was also carried out on 2 parallel processors and the run time was around 2 days.

2.6 Library

The library test was also large therefore we also had to split it into 2 parts. The cumulative run time was around 1 day. Additionally not all labels were available in English therefore we have used the original Dutch labels. The implication is that we could not determine hypernyms from WordNet which might impact our mapping precision negatively.

3 General comments

3.1 Comments on the results

Most of the benchmark tests proved that when different similarity assessments have to be combined handling uncertainty can lead to a high precision rate which is a definite strength of our system. Another strength of our system is that the produced mappings are not very dependent on the structure and hierarchy of the concepts and properties in the ontology (see group 2xx). The reason is our mapping algorithm takes mainly concepts (classes) and properties (object and data type) to capture the specific restrictions in the particular ontologies and converts them into directed graph fragments. As a consequence our method is not heavily dependent on subclass, sub property, disjointness or equivalency relationships among classes and properties hence on the logical constraints imposed by the ontology language itself. Additionally the query terms are extended with their inherited hypernyms from WordNet so the uncertainty can be distributed sufficiently that can lead to a large number of possibly valid choices. However since Dempser's combination rule is computationally expensive operation we need to reduce the problem space therefore the number of additional variables per query fragment. This can lead to the loss of valuable information and consequently more irrelevant mappings.

3.2 Discussions on the way to improve the proposed system

Based on the results we have identified the following improvement possibilities that can further improve our system:

1. We need to consider Natural Language descriptions where available in the ontologies. This can lead to a definite improvement of precision for the particular mapping.
2. We need to further exploit the properties of the instances or individuals in the ontologies. This can lead to a definite improvement of recall for the particular mappings.
3. The possible application of additional multi lingual background knowledge can provide added value for improving both recall and precision of the system.

3.3 Comments on the OAEI 2007 procedure

The OAEI procedure and the provided alignment api works very well out of the box for the benchmarks, anatomy and directory tracks. However for the food, environment and library track we have developed an SKOS parser which can be integrated into the alignment api. Our SKOS parser convert SKOS file to OWL which is then processed using the alignment api. Additionally we have developed a chunk SKOS parser which can process SKOS file iteratively in chunks avoiding memory problems.

3.4 Comments on the OAEI 2007 test cases

We have found that most of the benchmark tests can be used effectively to test various aspects of an ontology mapping system since it provides both real word and generated/modified ontologies. The ontologies in the benchmark are conceived in a way that allows anyone to clearly identify system strengths and weaknesses which is an important advantage when future improvements have to be identified. However, our system did not perform as well as we first expected probably due to the fact that most of the classes and properties in the ontologies are organized in a rather flat hierarchy so in our system the semantic similarity component did not influence the overall mappings considerably. However, in order to make use of a large group of tests (248-266) our system had to consider individuals or instances of the classes.

3.5 Comments on the OAEI 2007 measures

For our system the precision measure was the most important of all because this gives us the possibility to draw constructive conclusions on how the uncertainty handling can influence the precision of the system. The additional measures like recall and fallout can be used effectively for identifying where do we need to make further improvements in our system.

3.6 Proposed new measures

Besides the traditional measures it would be useful as well to introduce a measure that expresses the difficulty to create the particular mapping. E.g. there is a considerable difference in the level of difficulty between creating mapping with the reference ontology itself (101 to 101) and real word ontology (101 to 304). Additionally this measure then could be used to assess the how the particular system can handle mappings that involves complex comparison operations.

4 Conclusion

Based on the experiments of both the OAEI 2006 and 2007 we had a possibility to realise a measurable evolution in our ontology mapping algorithm and test it with 6 different mapping tracks. Our main objective to improve the mapping precision with managing the inherent uncertainty of any mapping process and information in the different ontologies on the Semantic Web can only be achieved if different mapping algorithms can be qualitatively compared and evaluated. Therefore participating in the Ontology Alignment Evaluation Initiative is an excellent opportunity to test and compare our system with other solutions and helped a great deal identifying the future possibilities that needs to be investigated further.

Acknowledgements: This research project has been supported by a Marie Curie Transfer of Knowledge Fellowship of the European Community's Sixth Framework Programme under the contract number MTKD-CT-2004-509766 (enIRaF).

References

1. Choi N.,Song I-Y., Han H.(2006). A survey on ontology mapping. SIGMOD Record 35(3): 34-41.
2. Vargas-Vera M., and Motta E. (2004). An Ontology-driven Similarity Algorithm. *KMI-TR-151*, Knowledge Media Institute, The Open University, UK.
3. Vargas-Vera M.,Motta E., and Domingue J. (2003). AQUA: An Ontology-Driven Question Answering System. *AAAI Spring Symposium, New Directions in Question Answering*, Stanford, USA.: AAAI Press.
4. P. Rong,Ding Z.,Yu Y., and Peng Y. (2005). A Bayesian Network Approach to Ontology Mapping. In *Proceedings of the Fourth International Semantic Web Conference*,Galway, Ireland.
5. Nagy M.,Vargas-Vera M., and Motta E. (2005). Multi-agent Ontology Mapping Framework in the AQUA Question Answering System. the Fourth International Mexican Conference on Artificial Intelligence (MICAI-2005), Lecture Notes in Artificial Intelligence LNAI 3789, Gelbukh, A de Albornoz and H. Terashima (Eds), pp. 70-79, Monterrey Mexico.
6. Nagy M.,Vargas-Vera M., and Motta E. (2005). DSSim-ontology Mapping with Uncertainty In *Proceedings of the 1st International Workshop on Ontology Matching (OM-2006)* , Athens, Georgia, USA.
7. Shafer G. (1976). A Mathematical Theory of Evidence.: Princeton University Press.
8. Cohen W., Ravikumar P., and Fienberg S. (2003). A Comparison of String Distance Metrics for Name-Matching Tasks, In *Proceedings of Information Integration on the Web (IIWeb 2003)*, Accapulco, Mexico.

Appendix: Raw results

Matrix of results

#	Name	Prec.	Rec.	Time
101	Reference alignment	1	1	00.00.10.37
102	Irrelevant ontology	0	NaN	00.00.03.57
103	Language generalization	1	1	00.00.03.76
104	Language restriction	1	1	00.00.03.09
201	No names	1	0.16	00.00.03.16
202	No names, no comments	1	0.16	00.00.05.12
203	No comments	1	1	00.00.04.92
204	Naming conventions	0.96	0.91	00.00.03.23
205	Synonyms	0.94	0.33	00.00.03.44
206	Translation	0.97	0.39	00.00.04.38
207		0.97	0.39	00.00.04.36
208		0.95	0.9	00.00.04.18
209		0.91	0.32	00.00.03.19
210		0.97	0.39	00.00.04.26
221	No specialisation	1	1	00.00.04.15
222	Flatenned hierarchy	1	1	00.00.02.88
223	Expanded hierarchy	1	1	00.00.02.98
224	No instance	1	1	00.00.03.06
225	No restrictions	1	1	00.00.02.85
228	No properties	1	1	00.00.02.98
230	Flattened classes	0.97	1	00.00.01.59
231	Expanded classes	1	1	00.00.03.17
232		1	1	00.00.02.95
233		1	1	00.00.02.85
236		1	1	00.00.01.60
237		1	1	00.00.01.60
238		1	1	00.00.02.88
239		0.97	1	00.00.03.01
240		0.97	1	00.00.01.70
241		1	1	00.00.01.75
246		0.97	1	00.00.01.62
247		0.97	1	00.00.01.68
248		1	0.16	00.00.01.75
249		1	0.16	00.00.04.50
250		1	0.27	00.00.04.39
251		1	0.17	00.00.01.75
252		1	0.16	00.00.04.50
253		1	0.16	00.00.04.74
254		1	0.27	00.00.04.41
257		1	0.27	00.00.01.70
258		1	0.17	00.00.01.68
259		1	0.16	00.00.04.37
260		0.9	0.31	00.00.04.63
261		1	0.27	00.00.01.92
262		0.9	0.31	00.00.01.70
265		0.8	0.24	00.00.01.70
266		0.82	0.3	00.00.01.92
301	Real: BibTeX/MIT	0.85	0.6	00.00.03.44
302	Real: BibTeX/UMBC	0.85	0.8	00.00.02.69
303	Real: Karlsruhe	0.96	0.92	00.00.04.12
304	Real: INRIA	0.98	0.64	00.00.03.03

Falcon-AO: Results for OAEI 2007

Wei Hu, Yuanyuan Zhao, Dan Li, Gong Cheng, Honghan Wu, and Yuzhong Qu

School of Computer Science and Engineering, Southeast University,
Nanjing 210096, P.R. China
{whu,yyzhao,dli,gcheng,hhwu,yzqu}@seu.edu.cn

Abstract. In this paper, we present a brief overview of Falcon-AO (version 0.7): a practical ontology matching system with acceptable to very good performance, a flexible architecture, and a number of unique features. We also show some preliminary results of Falcon-AO for this year's OAEI campaign: evaluation on seven different matching tasks.

1 Presentation of the system

As an infrastructure for Semantic Web applications, Falcon is a vision of our research group. It desires for providing fantastic technologies for finding, aligning and learning ontologies, and ultimately for capturing knowledge by an ontology-driven approach. It is still under development in our group.

1.1 State, purpose, general statement

As a prominent component of Falcon, Falcon-AO starts as an automatic ontology matching system to help enable interoperability between (Semantic) Web applications using different but related ontologies. It has since become a very practical and popular tool for matching Web ontologies expressed in RDFS or OWL. To date, Falcon-AO is continually being improved and elaborated, and the latest version is 0.7.

1.2 Specific techniques used

Falcon-AO is implemented in Java, and it is an open source project under the Apache license. Fig. 1 exhibits the architecture of Falcon-AO (version 0.7). It consists of five components: the *Repository* to temporarily store the data during the matching process; the *Model Pool* to manipulate ontologies and to construct different models for different matchers; the *Alignment Set* to generate and to evaluate exported alignments; the *Matcher Library* to manage a set of elementary matchers; the *Central Controller* to configure matching strategies and to execute matching operations. Furthermore, Falcon-AO provides a graphical user interface (GUI) to make it easily accessible to users.

Due to space limitation, we only provide a brief overview of Falcon-AO's features in this paper. For more details, we refer the reader to the technical papers [1–6], and our website: http://iws.seu.edu.cn/projects/matching/.

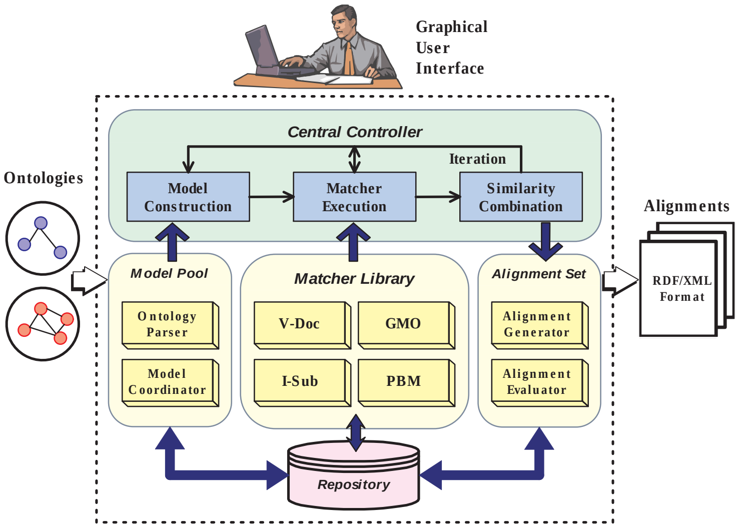


Fig. 1. System Architecture

- V-Doc, I-Sub, GMO, PBM: The four distinguishing elementary matchers make up the core matcher library of Falcon-AO. V-Doc [5] discovers alignments via revealing the usage (context) of domain entities in ontologies (i.e. the neighboring information). I-Sub [6] is a light-weighted matcher based on a refined string comparison technique (i.e. considering both commonality and difference). GMO [2] measures the structural similarity between RDF bipartite graphs based on propagating similarities between domain entities and statements. PBM [3] follows the divide-and-conquer idea to partition large ontologies into small blocks and construct mappings between the blocks for further matching with V-Doc, I-Sub, and GMO.

- Model Coordinator [4]: 21 coordination rules are used to eliminate useless axioms and reduce structural heterogeneity between the ontologies to be matched. Specifically, Falcon-AO can apply different coordination rules to different matchers. As an example, for I-Sub, Falcon-AO only removes ontology headers. As another example, for GMO, Falcon-AO uses the *rdfs:member* property to express the relationship between a list and each of its members, instead of using RDF collection vocabularies (*rdf:first*, *rdf:rest* and *rdf:nil*).

- Alignment Generator [1, 4]: Alignments are generated in terms of the observation of the linguistic comparability (LC) and the structural comparability (SC). LC is calculated by examining the proportion of the candidate alignments against the minimum number of the domain entities in the two ontologies; while SC is computed by comparing the built-in vocabularies in the two ontologies as well as estimating the correct alignments found by GMO in the portion of the reliable ones from V-

Doc and I-Sub. Falcon-AO uses these two kinds of comparability to automatically adjust the thresholds for selecting and combining alignments.

1.3 Adaptations made for the evaluation

We do not make any specific adaptation in the OAEI 2007 campaign. All the alignments outputted by Falcon-AO are uniformly based on the same parameters.

1.4 Link to the system and parameters file

The latest version of Falcon-AO can be downloaded from our website:
http://iws.seu.edu.cn/projects/matching/res/falcon.zip.

1.5 Link to the set of provided alignments (in align format)

The alignments for all the tasks in this year's OAEI campaign are available at:
http://iws.seu.edu.cn/projects/matching/res/falcon_2007.zip.

2 Results

In this section, we present the results of Falcon-AO (version 0.7) for the tasks provided by the OAEI 2007 campaign.

2.1 *Benchmark*

The *benchmark* task can be divided into five groups: #101–104, #201–210, #221–247, #248–266 and #301–304. The results of Falcon-AO are reported on each group in correspondence. The summary of the average performance of Falcon-AO (version 0.7) on the *benchmark* task is depicted in Table 1. For more details, please link to Appendix.

Table 1. Summary of the average performance on the *benchmark* task

	1xx	2xx	3xx	Average	H-mean	Time
Precision	1.00	0.93	0.89	0.93	0.92	300s
Recall	1.00	0.80	0.77	0.81	0.86	

#101–104 Falcon-AO performs perfectly on this group. Please pay attention to #102, Falcon-AO can automatically detect the two candidate ontologies are completely different, because both the linguistic comparability and the structural comparability between them are extremely low.

#201–210 Although in this group, some linguistic features of the candidate ontologies are discarded or modified, their structures are quite similar. So GMO takes much effect on this group. For example, in #202, 209, and 210, only a small portion of alignments are found by V-Doc or I-Sub, the rest are all generated by GMO. Since GMO runs much slower, it takes Falcon-AO more time to exploit all the alignments.

#221–247 The structures of the candidate ontologies in this group are altered. However, Falcon-AO discovers most of the alignments from the linguistic perspective via V-Doc and I-Sub, and both the precision and recall are pretty good.

#248–266 Both the linguistic and structural characteristics of the candidate ontologies are changed significantly, thus the tests in this group are the most difficult ones in the *benchmark* task. In some cases, Falcon-AO rarely finds any alignments. But indeed, in these cases, it is really hard to recognize the correct alignments due to lack of clues.

#301–304 Four real-life ontologies of bibliographic references are taken in this group. The linguistic comparability between the two candidate ontologies in each test is high while the structural comparability is medium. It indicates that the outputs of Falcon-AO are mainly from V-Doc or I-Sub. Alignments from GMO with high similarities are also reliable to be integrated.

2.2 *Anatomy*

The *anatomy* real world case is to match the Adult Mouse Anatomy (denoted by *mouse*) and the NCI Thesaurus describing the human anatomy (tagged as *human*). *mouse* has 2,744 classes, while *human* has 3,044 classes. Falcon-AO firstly partitions *mouse* and *human* into 122 and 14 blocks respectively, and then finds 16 block mappings based on 1,139 anchors. After further running elementary matchers on such block mappings, 715 alignments are finally generated as output. The whole process spends about 12 minutes. The summary of the performance is exhibited in Table 2.

Table 2. Summary of the performance on the *anatomy* task

	Precision	Recall	Time
Mouse vs. Human	0.96	0.59	12m

2.3 *Directory*

The *directory* task consists of Web sites directories like Google, Yahoo! or Looksmart. To date, it includes 4,639 tests represented by pairs of OWL ontologies, where classification relations are modeled as *rdfs:subClassOf* relations. Falcon-AO is quite efficient in this task, and it only takes less than 2 minutes (110 seconds) to complete all the tests. The average performance is summarized in Table 3.

Table 3. Summary of the average performance on the *directory* task

	Precision	Recall	Time
Directory	0.55	0.61	110s

2.4 *Food*

The *food* task includes two SKOS thesauri – *AGROVOC* and *NALT*. Since Falcon-AO focuses on Web ontologies expressed in RDFS and OWL, we have to adopt two OWL version ontologies transformed by campaign organizers in this task. *AGROVOC* owns 28,439 classes, while *NALT* owns 42,326 classes. Falcon-AO partitions them into 442 and 235 blocks, respectively. Supported by 15,353 anchors, Falcon-AO discovers 154 block mappings and conducts elementary matchers on them. Finally, 15,300 alignments are generated, where 14,615 alignments hold equivalence relationships, 685 ones hold subsumption relationships (558 broad relationships and 127 narrow relationships). The whole process costs nearly 6 hours. The performance is shown in Table 4.

Table 4. Summary of the performance on the *food* task

	Precision	Recall	Time
AGROVOC vs. NALT	0.84	0.45	5.75h

2.5 *Environment*

Three SKOS thesauri are collected in this task – a new thesaurus named *GEMET* (5,284 classes, is partitioned into 17 blocks), and the two ones in the *food* task. When matching *GEMET* and *AGROVOC*, Falcon-AO discovers 18 block mappings via 1,352 anchors, and generates 1,384 alignments, including 1,360 alignments hold equivalence relationships and 24 ones hold narrow relationships. The whole process spends nearly 33 minutes. When matching *GEMET* and *NALT*, Falcon-AO discovers 22 block mappings via 1,230 anchors, and generates 1,374 alignments, including 1,353 alignments hold equivalence relationships and 21 ones hold narrow relationships. The whole process spends about 1.2 hours. The performance is shown in Table 5.

Table 5. Summary of the performance on the *environment* task

	Precision	Recall	Time
GEMET vs. AGROVOC	0.88	0.39	33m
GEMET vs. NALT	0.86	0.30	1.2h

2.6 *Library*

Participants of this task are expected to match two Dutch thesauri used to index books from two collections held by the National Library of the Netherlands (KB). *Brinkman* owns 5,221 classes, and *GTT* owns 35,194 classes. Falcon-AO firstly partitions them into 232 and 2,211 blocks respectively, and then exploits 223 block mappings based on 3,641 anchors. After further running elementary matchers on such block mappings, 3,697 alignments are finally generated as outputs, where 3,661 alignments hold equivalence relationships, and 36 ones hold subsumption relationships (including 23 broad relationships and 13 narrow ones). The whole process spends about 40 minutes. The summary of the performance is exhibited in Table 6.

Table 6. Summary of the performance on the *library* task

	Precision	Recall	Time
Brinkman vs. GTT	0.97	0.87	40m

2.7 *Conference*

91 matching tasks are generated from 14 different ontologies with respect to conference organization. Falcon-AO takes 160 seconds to complete all the tests. Some statistics of the average performance are summarized in Table 7.

Table 7. Summary of the performance on the *conference* task

	Precision	Recall	Time
Conference	0.73	0.57	160s

3 General comments

In this section, we present some comments on Falcon-AO's results as well as the OAEI 2007 test cases.

3.1 Comments on the results

Here, we would like to make a rough comparison between Falcon-AO's results in this year and the results in the OAEI 2006 campaign (see Fig. 2). It can be seen that on these four tasks, Falcon-AO has more or less improvement.

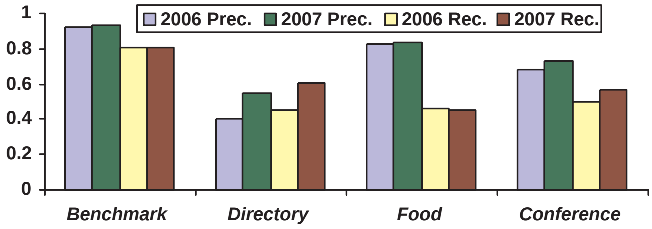


Fig. 2. Results in 2007 vs. Results in 2006

3.2 Comments on the OAEI 2007 test cases

The proposed matching tasks cover a large portion of real world domains, and the discrepancies between them are significant. Doing experiments on these tasks are helpful to improve algorithms and systems. In order to enhance applicability, we list some warnings as well as our modifications occurring in our experiment procedure, which might aid organizers to correct the problems in the future: (i) the prefix "rdfs" is not bound in "gemet_oaei2007.owl" in the *environment* task; and (ii) the encoding is inappropriate in the *library* task, and our modification is replacing "utf-8" by "iso-8859-1".

4 Conclusion

Ontology matching is an important way to establish interoperability among (Semantic) Web applications using different but related ontologies. We implement a practical system for ontology matching called Falcon-AO. From the experimental experience in the OAEI 2007 campaign, we conclude that Falcon-AO (version 0.7) performs quite well and balancing on most of tasks. In the future, we look forward to making a stable progress towards building a comprehensive ontology matching system.

Acknowledgements

The work is funded by the NSFC under Grant 60573083. We would like to thank the following people for their valuable comments to Falcon-AO: Hai Leong Chieu, Antoine Isaac, Michel Klein, Adam Saltiel, Rubén Tous, Shenghui Wang, Songmao Zhang.

References

1. Hu, W., Cheng, G., Zheng, D., Zhong, X., and Qu, Y.: The results of Falcon-AO in the OAEI 2006 campaign. In Proc. of the ISWC 2006 workshop on Ontology Matching. (2006) 124–133
2. Hu, W., Jian, N., Qu, Y., and Wang, Y.: GMO: A graph matching for ontologies. In Proc. of the K-CAP 2005 workshop on Integrating Ontologies. (2005) 41–48
3. Hu, W., Zhao, Y., and Qu, Y.: Partition-based block matching of large class hierarchies. In Proc. of the 1st Asian Semantic Web Conference (ASWC'06). (2006) 72–83

4. Jian, N., Hu, W., Cheng, G., and Qu, Y.: Falcon-AO: Aligning ontologies with Falcon. In Proc. of the K-CAP 2005 workshop on Integrating Ontologies. (2005) 85–91
5. Qu, Y., Hu, W., and Cheng, G.: Constructing virtual documents for ontology matching. In Proc. of the 15th International World Wide Web Conference (WWW'06). (2006) 23–31
6. Stoilos, G., Stamou, G., and Kollias, S.: A string metric for ontology alignment. In Proc. of the 4th International Semantic Web Conference (ISWC'05). (2005) 623–637

Appendix: Raw Results

Tests are carried out on an Intel Core 2 Duo 2.13GHz desktop machine with 2GB DDR2 memory under Windows XP Professional operating system and Java 1.6 compiler.

Matrix of Results

In the following table, the results of Falcon-AO (v0.7) in the *benchmark* test are shown by precision (Prec.), recall (Rec.) and machine processing time. Here, the machine processing time is the sum of the time in model construction, matcher execution, similarity combination and results evaluation.

#	Name	Prec.	Rec.	Time
101	Reference alignment	1.00	1.00	2.7s
102	Irrelevant ontology	NaN	NaN	1.9s
103	Language generalization	1.00	1.00	1.2s
104	Language restriction	1.00	1.00	1.2s
201	No names	1.00	0.95	1.2s
202	No names, no comments	0.87	0.87	24.9s
203	No comments	1.00	1.00	0.7s
204	Naming conventions	0.98	0.98	1.2s
205	Synonyms	1.00	0.98	1.2s
206	Translation	1.00	0.93	1.1s
207		0.98	0.91	1.1s
208		1.00	1.00	0.7s
209		0.79	0.78	24.2s
210		0.81	0.80	24.3s
221	No specialization	1.00	1.00	1.1s
222	Flattened hierarchy	1.00	1.00	1.1s
223	Expanded hierarchy	1.00	1.00	1.1s
224	No instance	1.00	0.99	0.9s
225	No restrictions	1.00	1.00	1.1s
228	No properties	1.00	1.00	0.5s
230	Flattened classes	0.94	1.00	1.1s
231	Expanded classes	1.00	1.00	1.2s
232		1.00	0.99	1.0s
233		1.00	1.00	0.5s
236		1.00	1.00	0.4s
237		1.00	0.99	0.9s
238		1.00	0.99	1.1s
239		1.00	1.00	0.5s
240		1.00	1.00	0.6s
241		1.00	1.00	0.4s
246		1.00	1.00	0.4s
247		1.00	1.00	0.5s
248		0.85	0.84	23.5s
249		0.87	0.87	23.6s
250		1.00	0.27	0.4s
251		0.56	0.56	27.2s
252		0.71	0.71	26.3s
253		0.85	0.84	23.0s
254		1.00	0.27	0.5s
257		1.00	0.27	0.4s
258		0.54	0.54	26.4s
259		0.70	0.70	25.6s
260		1.00	0.31	0.4s
261		0.89	0.24	0.5s
262		1.00	0.27	0.4s
265		1.00	0.31	0.4s
266		0.89	0.24	0.5s
301	Real: BibTeX/MIT	0.91	0.82	0.9s
302	Real: BibTeX/UMBC	0.90	0.58	0.4s
303	Real: Karlsruhe	0.77	0.76	0.7s
304	Real: INRIA	0.96	0.93	15.9s

LILY: The Results for the Ontology Alignment Contest OAEI 2007

Peng Wang, Baowen Xu

School of Computer Science and Engineering, Southeast University, China
{pwangseu@gmail.com, bwxu@seu.edu.cn}

Abstract. This paper presents the results of LILY, which is an ontology mapping system, for OAEI 2007 campaign. To accurately describe what the real meaning of an entity in the original ontology is, LILY extracts a semantic subgraph for each entity. Then it exploits both linguistic and structural information in semantic subgraphs to generate initial alignments. If necessary, using these initial results as input, a subsequent similarity propagation strategy could produce more alignments, which often can not be obtained by the previous process. The preliminary results of the experiments for four tasks (i.e. benchmark, directories, anatomy and conference) are presented. The discussion of the results and future work of LILY are also given.

1 Presentation of the system

Currently more and more ontologies are distributedly used and built by different communities. Many of these ontologies would describe similar domains, but using different terminologies, and others will have overlapping domains. Such ontologies are referred to as heterogeneous ontologies, which is a major obstacle to realize semantic interoperation. Ontology mapping, which captures relations between ontologies, aims to provide a common layer from which heterogeneous ontologies could exchange information in semantically sound manners.

LILY is a system for solving the issues related to heterogeneous ontologies. One important function of LILY is to match heterogeneous ontologies. LILY uses the *semantic subgraph* to describe the meaning of an entity. Then linguistic and structural similarity algorithm and similarity propagation strategy are exploited to create the alignments between ontologies.

1.1 State, purpose, general statement

When LILY is used to find alignments between heterogeneous ontologies, it tries to utilize all useful information to discover the correct matching results. Currently it does not use any external knowledge such as WordNet. The matching process consists of three main steps: (1) *Extracting semantic subgraph* LILY tries to use a semantic subgraph to represent the real meaning for a given entity in an ontology. A semantic subgraph, which is also a subgraph of the original ontology, is extracted by a variant algorithm based on the connection subgraphs discovery algorithm [1]. (2) *Computing alignment similarity* Through analyzing the literal and structural information in the semantic subgraphs, LILY computes the similarity confidences between entities from different ontologies. (3) *Similarity propagation* In most cases, LILY can find satisfactory alignment results after the second process. If few alignment results are got, a strategy will decide whether to take similarity propagation process. The similarity propagation could produce more alignments that can not be found in the previous processes. The matching process is shown in Fig. 1.

LILY is still being improved and enhanced, and the lasted version is V1.2.

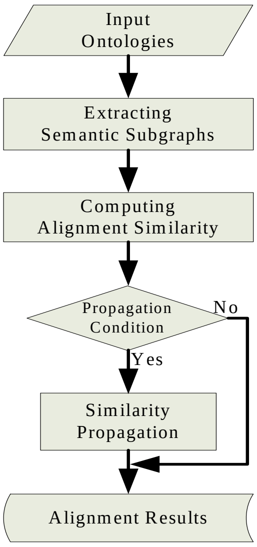


Fig. 1. Matching process

1.2 Specific techniques used

LILY aims to provide high quality alignments between concept/property pairs. The main specific techniques used by LILY are as follows.

Semantic subgraph An entity in a given ontology has its specific meaning. In our ontology mapping view, capturing such meaning is very important to obtain good alignment results. Therefore, before similarity computation, LILY first describes the meaning for each entity accurately. The solution is inspired by the method proposed by Faloutsos et al. for discovering connection subgraphs [1]. It is based on electricity analogues to extract a small subgraph that best captures the connections between two nodes of the graph. Ramakrishnan et al. also exploits such idea to find the informative connection subgraphs in RDF graph. We modify the method for extracting an *n-size* subgraph for a node or edge in an ontology graph. The subgraphs can give the precise descriptions of the meanings of the entities, and we call such subgraphs semantic subgraphs. The details of the semantic subgraph extraction process will be reported elsewhere.

Alignment similarity computation The similarity computation is based on the semantic subgraphs, i.e. all the information used in the similarity computation is come from the semantic subgraphs. LILY uses two kinds of descriptions to interpret the concepts and properties. The first is the basic description, which is a document consisting of the identifier, label and comments. The second is the semantic description. A semantic description of a concept contains the information about class hierarchies, related properties and instances. A semantic description of a property contains the information about hierarchies, domains, ranges, restrictions and related instances. For the descriptions from different entities, we calculate the similarities of the corresponding parts. Finally, all separate similarities are combined with the experiential weights. The descriptions collect the linguistic and structural information of entities. Therefore, for the regular ontologies, LILY can find satisfactory alignments in most cases.

Similarity propagation When the ontologies lack of regular and clear literal descriptions, the above method just returns few alignments. LILY uses similarity propagation strategy to compensate for it. Compared with other similarity propagation methods such as similarity flood [3] and SimRank [4], our similarity propagation defines stronger propagation condition and is based on the semantic subgraphs. The propagation graph is not stable, but is incremental during propagation process. Using similarity propagation can find more alignments that cannot be found in the previous process. However, the similarity propagation is not always perfect. When more alignments are discovered, more incorrect alignments would also be introduced by the similarity propagation. So it requires a strategy to determine when to use the similarity propagation.

Automatic threshold selection The previous processes will return a similarity matrix, which represents the confidence level between entities from two ontologies. It is need a threshold to filter the low confidence values and keep high confidence ones. However, the threshold is usually set up manually, that cannot adapt to all matching situations. LILY treats the similarity matrix as an image, and then uses the classic image threshold selection algorithm to find a threshold automatically. There are many image thresholding methods [5]. After comparing the effectiveness of a variety of thresholding algorithms, we choose the maximum entropy approach to calculate the threshold [6]. After filtering, final 1-1 alignments are generated using the stable marriage strategy.

1.3 Adaptations made for the evaluation

In the evaluation, the size of semantic subgraph would influence on the alignment results. We set *5-size* semantic subgraphs for most test cases. When the ontologies lack of regular literals, we set *10* to *35-size* semantic subgraphs for capturing more structural information. For large scale ontologies, we just set *2* to *3-size* semantic subgraphs for the purpose of reducing the time of extracting semantic subgraphs.

1.4 Link to the system and the set of provided alignments

A demo version of LILY and the alignment results for OAEI2007 campaign are available at http://ontomappinglab.googlepages.com/oaei2007.

2 Results

In this section, we will analyze the performances and problems during taking the four alignment tracks.

2.1 benchmark

The benchmark tests the performance of matching system during various ultimate situations.

101-104 This test set contains same, irrelevant, ontology language generalization and restriction ontologies. LILY plays well for these test cases. But for the irrelevant ontology 102, LILY returns several alignments because it cannot decide whether the two ontologies are irrelevant, so it tries to find any possible alignments.

201-210 In the test cases, the structure of ontology is preserved, but the labels and identifiers are replaced by random names, misspellings, synonyms and foreign names. The comments have been suppressed in some cases. LILY can produce good results for this test set. Even without right labels and comments information, LILY can find most correct alignments through making use of other information such as instances. Using few alignment results obtained by the basic methods as inputs, the similarity propagation strategy will generate more alignments.

221-247 The test cases can be divided into two subgroups: 221-231 and 232-247. The first subgroup contains 11 kinds of modifications, such as the hierarchy is flattened or expanded, and individuals, restrictions and datatypes are suppressed. Due to the labels and comments are preserved, the modifications have little influence on our system. LILY can find most correct alignments using the labels and comments information. In the second subgroup, the modifications are the combinations of the ones used in 221-231. LILY can obtain good results for 232-247 as well.

248-266 This is the most difficult test set. All labels and identifiers are replaced by random names, and the comments are also suppressed. LILY uses the information from the semantic subgraphs to look for alignments. However, no enough information is provided in the ontologies, and the similarity computation process can only find few alignments. Subsequently, using these initial results as input, LILY exploits the similarity propagation algorithm to discover more alignments. In our experiments, too smaller and too bigger size semantic subgraph can not produce good alignments. *10-35* is a suitable size range in our experience. In 254 and 262, since almost all literal and structure information are suppressed, the similarity propagation can not find more results, so LILY just can produce limit results. When some structure information is preserved, similarity propagation will play a role and can generate more alignment results.

301-304 This test set are the real ontologies. For LILY just can find equivalent alignment relations, the inclusion alignments can not be generated. For 301-302, LILY finds most correct alignments, but it also returns some wrong results. The alignment results for 303 are far from satisfactory. We think the reason might be that 303 is no individuals and with shallow class hierarchy, and there are no direct connections between the classes and properties. Without the external knowledge, LILY can not produce good results for 303. 304 has similar structure and vocabularies to the reference ontology 101, so LILY outperforms other three ontologies.

2.2 anatomy

The anatomy track consists of two real large-scale biological ontologies. Handling such ontologies is a big problem for LILY, because extracting semantic subgraphs would need long time and large memory space. Even though LILY sets up small size semantic subgraphs for this matching task, it needs about 4 days to create the alignment results. For the purpose of producing the alignments in time, the principal technique advantages of LILY are discards in this alignment task.

2.3 directory

The directory track requires to matching two taxonomies describing the web directories. Except the class hierarchy, there is no other information in the ontologies. Therefore, LILY will utilize the hierarchy information to decide the alignments. There are three alignment tasks. The first is matching the 4640 small ontologies pairs. The second task is matching a 10% sample ontology pair. LILY completes the two tasks smoothly. The third task is required to match two large-scale taxonomies. LILY takes 8 days to produce the alignments. Similar to the anatomy track, we just set up the small size semantic subgraphs to assure that the alignment results can be generated in time.

2.5 conference

This track contains 14 real-case ontologies about conference. For a given ontology, we compute the alignments with itself, as well as with other ontologies. For we treat the equivalent alignment is symmetric, we get 105 alignment files totally. The heterogeneous character in this track is various. It is a challenge to generate good results for all ontology pairs in this test set.

3 General comments

3.1 Comments on the results

During the OAEI campaign, we are aware of the strengths and weaknesses of LILY.

Strengths For normal size ontologies, if they have regular literals or similar structures, LILY can achieve satisfactory alignments. The reason lies in two aspects: (1) The semantic subgraphs could represent the real meanings of the concepts or properties, that avoids introducing the unnecessary and noise information to the matching processes. (2) The similarity propagation strategy could compensate for the linguistic matching methods, and it can produce more alignments when ontologies lack of linguistic information.

Weaknesses LILY has two obvious weaknesses. (1) *Processing large scale ontologies* LILY cannot work well for large scale ontologies. Semantic subgraph extraction process and similarity propagation process could take terrible time for large scale ontologies. (2) *Efficiency* LILY needs to extract semantic subgraphs for all concepts and properties. It is a time-consuming process. In similarity propagation, the propagation graph would become large that will also need more time for propagating the similarities.

3.2 Discussions on the way to improve the proposed system

In OAEI07, we find the efficiency is an outstanding problem for LILY. In the matching process, most of time is spent on extracting semantic subgraphs and similarity propagation. The two processes usually account for 80% time in the full matching process. In addition, LILY's time complexity is $O(kn^2)$, where n is the number of entities and k is the average time for calculating an alignment. Therefore, it is very slow when run the large scale ontology matching task. Even we completed two large scale ontology matching tasks (directory and anatomy), we had to use the basic parameters. It causes that some advanced methods in LILY can not be utilized. To sum up, improving the efficiency and finding suitable methods to handle large scale ontologies are the near future work for LILY.

3.3 Comments on the OAEI 2007 test cases

More real ontologies should be added to the test cases. The real ontologies could be better than the ones designed manually for testing the performance of a matching system.

The large scale ontology alignment task is a challenge for some ontology matching systems such as LILY. For the sake of fairness, currently, all reference alignment results for large scale ontologies matching tasks are unknown to all participants. We suggest that it was necessary to provide an open large scale ontology matching task for all researchers. That would be benefit to finding efficient methods for matching large scale ontologies. In addition, different matching systems could compare their results based on such open large scale ontologies.

4 Conclusion

We briefly introduce our ontology matching tool LILY. The matching process and the special techniques used by LILY are presented. The preliminary alignment results are carefully analyzed. Finally, we summarized the strengths and the weaknesses of LILY.

References

1. Faloutsos, C., McCurley, K. S., Tomkins, A.: Fast Discovery of Connection Subgraphs. In the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington (2004).
2. Ramakrishnan, C., Milnor, W. H., Perry, M., Sheth, A. P.: Discovering Informative Connection Subgraphs in Multirelational Graphs. ACM SIGKDD Explorations, Vol. 7(2), (2005)56-63.
3. Melnik, S., Garcia-Molina, H., Rahm, E.: Similarity Flooding: A Versatile Graph Matching Algorithm and its Application to Schema Matching. In the 18th International Conference on Data Engineering (ICDE), San Jose CA (2002).
4. Jeh, G., Widom, J.: SimRank: A Measure of Structural-Context Similarity. In the 8th International Conference on Knowledge Discovery and Data Mining (SIGKDD), Edmonton, Canada, (2002).
5. Sezgin, M., Sankur, B.: Survey over image thresholding techniques and quantitative performance evaluation, Journal of Electronic Imaging, Vol. 13(1): (2004)146–165.
6. Kapur, J. N., Wong, A. K. C.: A new method for gray-level picture thresholding using the entropy of the histogram. Computer Vision, Graphics and Image Processing, Vol. 29(2), (1985)273-285.

Appendix: Raw results

The final results of benchmark task are as follows.

Matrix of results

#	Name	Prec.	Rec.
101	Reference alignment	1.00	1.00
102	Irrelevat ontology	0.00	NaN
103	Language generalization	1.00	1.00
104	Language restriction	1.00	1.00
201	No names	1.00	1.00
202	No names, no comments	1.00	0.80
203	No comments	1.00	1.00
204	Naming conventions	1.00	1.00
205	Synonyms	1.00	0.99
206	Translation	1.00	0.99
207		1.00	0.99
208		1.00	1.00
209		0.92	0.91
210		1.00	0.91
221	No specialisation	1.00	1.00
222	Flatenned hierachy	1.00	1.00
223	Expanded hierarchy	1.00	1.00
224	No instance	1.00	1.00
225	No restrictions	1.00	1.00
228	No properties	1.00	1.00
230	Flatenned classes	0.94	1.00
231	Expanded classes	1.00	1.00
232		1.00	1.00
233		1.00	1.00
236		1.00	1.00
237		1.00	1.00
238		0.98	0.98
239		0.97	1.00
240		0.97	1.00
241		1.00	1.00
246		0.97	1.00
247		0.94	0.97
248		1.00	0.77
249		1.00	0.80
250		0.85	0.67
251		0.96	0.74
252		0.94	0.76
253		0.97	0.75
254		1.00	0.27

257		0.85	0.67
258		0.76	0.74
259		0.94	0.75
260		0.62	0.45
261		0.61	0.42
262		1.00	0.27
265		0.86	0.41
266		0.64	0.42
301	BibTeX/MIT	0.89	0.80
302	BibTeX/UMBC	0.82	0.65
303	Karlsruhe	0.58	0.69
304	INRIA	0.91	0.97

OLA in the OAEI 2007 Evaluation Contest

Jean François Djoufak Kengue[1], Jérôme Euzenat[2], and
Petko Valtchev[1]

[1] LATECE, Université du Québec À Montréal, Canada
[2] INRIA Rhône-Alpes and LIG, Montbonnot, France
djoufak.jean_francois@courrier.uqam.ca, jerome.euzenat@inrialpes.fr,
valtchev.petko@uqam.ca

Abstract. Similarity has become a classical tool for ontology confrontation motivated by alignment, mapping or merging purposes. In the definition of an ontology-based measure one has the choice between covering a single facet (e.g., URIs, labels, instances of an entity, etc.), covering all of the facets or just a subset thereof. In our matching tool, OLA, we had opted for an integrated approach towards similarity, i.e., calculation of a unique score for all candidate pairs based on an aggregation of all facet-wise comparison results. Such a choice further requires effective means for the establishment of importance ratios for facets, or weights, as well as for extracting an alignment out of the ultimate similarity matrix. In previous editions of the competition OLA has relied on a graph representation of the ontologies to align, OL-graphs, that reflected faithfully the syntactic structure of the OWL descriptions. A pair of OL-graphs was exploited to form and solve a system of equations whose approximate solutions were taken as the similarity scores. OLA2 is a new version of OLA which comprises a less integrated yet more homogeneous graph representation that allows similarity to be expressed as graph matching and further computed through matrix multiplying. Although OLA2 lacks key optimization tools from the previous one, while a semantic grounding in the form of WORDNET engine is missing, its results in the competition, at least for the benchmark test suite, are perceivably better.

1 Introduction

Ontologies, i.e., explicit conceptualizations of a domain involving representations of domain concepts and relations, are now the standard way to approach data heterogeneity on the Web and insure application *interoperability*. However, the existence of independently built ontologies for the same domain is a source of heterogeneity on its own and therefore calls for the design of methods and tools restoring interoperability through ontology *matching*.

Similarity has become a classical tool for ontology matching. In the definition of an ontology-based similarity measure one has the choice between covering a single facet (e.g., URIs, labels, instances of an entity, etc.), covering all of the facets or just a subset thereof. Typically, a distinction is made between the way ontology entities are named and the way these are related to other entities within the ontology, the former being termed depending on the context "lexical", "linguistic", "terminological", etc. while the latter is usually qualified as "structural". Structural similarity measurement is performed as a form of graph matching whereas lexical one boils down to either string comparison or matching of representations of the semantic of terms in entity names.

Our matching tool OLA [1, 2] targets OWL-DL (formerly OWL-LITE) ontologies. It applies an integrated approach towards similarity, i.e., calculation of a unique score for all entity pairs based on an aggregation of all facet-wise comparison results (facets here stand for the relationships between OWL entities). Computation is *exhaustive* on entity descriptions meaning that all facets are covered. The similarity is defined through a category-sensitive[3] yet universal operator that basically computes a linear combination of facet similarities. As facets mostly represent entities of their own, the similarity definition gets circular and hence cannot be directly computed. OLA considers such definitions as equations to solve and approximates their solutions through an iterative fixed-point-bound process. As initial values are based on name comparison while iterations basically perform similarity exchange between pairs of neighbor entities, OLA similarity is a trade-off between the aforementioned structural and lexical aspects.

Previous participation of OLA in the alignment competitions [3, 4], despite globally positive outcome, have put the emphasis on a certain lack of homogeneity among the computational mechanisms at different levels of the similarity model that harm efficiency. These were traced back to the somewhat overloaded structure of the OL-graphs, the graph-based representation of OWL ontologies that was used to support the similarity computation.

Restoring homogeneity and improving efficiency was the motivation behind the OLA2 version that is developed jointly by UQÀM and INRIA. It introduced a flattened version of the OL-graph model where at most one scalar value is admitted in vertices while all the remaining information is in the edges. This allowed the iterative re-calculation of the similarity scores to be modeled as matrix operation without losing the valuable properties of the result nor the process.

To that end, the ontology graphs are combined into a product-like construct, the *match graph*, where vertices and edges are products of counterpart elements from the ontology graphs. Similarity computation represents an iterative value propagation across the match graph starting with initial values yielded by name comparison. The innovation is the matrix product used in re-calculations: the adjacency matrix of the match graph is used as the multiplicative factor leading to a fixed point. The resulting method is a step further towards structure-domnated similarity computation as it encompasses all relationships of an ontology entity whereas the previous version tended to disregard non-descriptive relationships (e.g., the one between a OWL class and a relation whose range the class represents).

Our new method has outperformed the initial version of OLA on both competition tests (benchmark) and efficiency, although many of the optimizations from previous years have not been implemented in it. More dramatically, its modular implementation eases future improvements.

---

[3] Entity categories, e.g., OWL class, property, object, data type, value, etc., compare to meta-classes of language meta-model

2 System Overview

OLA is an open-source tool, implementing the OLA algorithm (for *O*WL-*L*ite *A*lignment [2]), jointly developed by teams at University of Québec at Montréal and INRIA Rhône Alpes.

2.1 General purpose statement

The primary goal behind the OLA tool design is to perform alignment of ontologies expressed in OWL [1, 2], with an emphasis on OWL-DL (formerly OWL-LITE). The system offers similarity-based alignment on graph-like ontology representations. Beside alignment, it features a set of auxiliary services supporting the manipulation of alignment results.

2.2 Ontology graph model

Traditionally, an ontology is viewed as a set of *entities* and a set of *relationships* between those entities. This view underlies the translation of the ontologies into a graphs structure where entities become vertices and relationships edges. Yet in the new settings, beside language entity categories such as *classes*, *objects*, *relations*, *properties*, *property instances*, and *data types*, *data values*, less traditional ones are considered, i.e., *tokens* (including comments on entities) and *cardinalities*. The list of relationships is accordingly completed: together with previously existing in the OL-graph format *subsumption*, *instantiation*, *attribution*, *domain*, *range*, *restriction*, *valuation*, and *all* relationships, we exploit *card* and *name*. Fig. 1 provides an overview of the meta-model for the ontology graph format.

Both vertices and edges in the graph are labeled by their respective entity category/relationship.

As a support for similarity computation, the product of both ontology graphs, or their *match* graph, is composed. The vertices of the match graphs are pairs of vertices from opposite ontology graphs. In its basic version, the match graphs comprises the cartesian product of both vertex sets, i.e., same category is not required for vertices v_1 and v_2 to form a product vertex. Clearly, product vertices correspond to the variables of the equation system in the previous version of OLA. They embed a "weight" value which stands for the similarity and is computed iteratively (see below).

In contrast, match graph edges require strict correspondence: An edge labeled l exists between compound vertices (v_1, v_2) and (v'_1, v'_2) iff there exist an edge labeled l between v_1 and v'_1 in the first graph another one between v_2 and v'_2 in the second graph. Edges in the match graph are also weighted yet their weights are effective as they correspond to the weights on neighboring sets in the OL-graphs in version one.

Similarity model Similarity between entities of the initial graphs is reflected by the weight or importance index of the corresponding match graph node. The underlying computational model is the value propagation as described in [5] (and used in a range of alignment methods starting with [6]) across the graph. We recall that, the adjacency

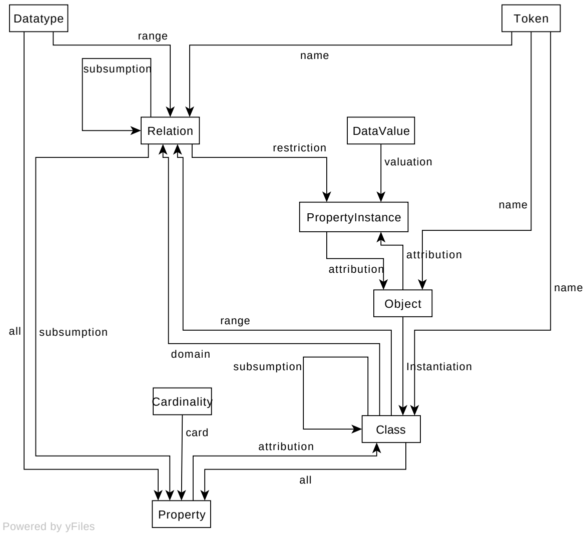


Fig. 1. Relationships between ontology entities with respect to the cluster to which they belong

matrices of the initial graphs are used to produce a larger matrix M reflecting both the inbound and the outbound neighbors of a match vertex. M is then used as a multiplier for the similarity vector V. Thus, starting with initial values, typically 1 for all entity pairs, V evolves according to the following iterative dependency: $V_k = M \times V_{k-1}$ ($k = 1, 2, 3...$), until a fixed point is reached.

Yet the model has been adapted – and even somewhat spoiled – as to compute the similarity defined in the reference OLA version. Recall that for a category X together with the set of relationships it is involved in, $\mathcal{N}(X)$, the similarity measure $Sim_X : X^2 \rightarrow [0, 1]$ is defined as follows:

$$Sim_X(x, x') = \sum_{\mathcal{F} \in \mathcal{N}(X)} \pi_{\mathcal{F}}^{X} MSim_Y(\mathcal{F}(x), \mathcal{F}(x')).$$

The function is normalized, i.e., the weights $\pi_{\mathcal{F}}^{X}$ sum to one, $\sum_{\mathcal{F} \in \mathcal{N}(X)} \pi_{\mathcal{F}}^{X} = 1$. Moreover, the set functions $MSim_Y$ compare two sets of nodes of the same category and extract a maximal pairing thereof that further optimizes the total similarity (see [2] for details).

In order to simulate the above family of functions, the graph-based model introduces first-class weights on relationship sets adjacent to a match vertex. Hence the adjacency matrix of the match graph that is central to the value propagation is not purely Boolean: values between 0 and 1 appear.

Moreover, the above equation is modified to reflect the evolving contribution of neighbor nodes:

$$V_k = M_k \times V_{k-1} \text{ (} k = 1, 2, 3... \text{ and } M_k = f(M_{k-1}, V_{k-1}) \text{)}$$

Here M_k reflects possible changes in pairings between contributing vertices within a set of neighbors, a recalculation that is done at each iteration in basic OLA. Hence f involves the current solution vector, as well.

Further adaptation of the original method is the initialization of V_0 with the results from name comparison for entities in a match vertex. This corresponds strictly to the initialization of the equation system in OLA.

The last adaptation completely changes the ideology of the method as it plays against the very basic principle of propagation: at each step a value at a vertex is replaced by a combination of the values of all its neighbors. Our understanding is that this is a major reason for the convergence of the computed values only for even steps.

In our model, the stabilizing role of name similarity (which is computed only once) has been secured by a representation trick. In fact, the vertices representing the sources of stable similarity, i.e., *token*, *cardinality*, *data type*, etc. are provided with a local looping edge while fixing their weights till the end of the process. Hence the respective OWL entity node that is identified by the token gets the same value at each iteration.

The above process provenly converges towards a solution vector V_∞.

Past optimizations A number of optimizations have been implemented within the system mainly aimed at making the weights – matching – similarity scheme more flexible. First, mechanisms for weight adaptation, both at entity and ontology level have been designed. The goal is to insure that the absence of a specific facet locally, i.e., for an entity pair, or globally, i.e., for all pairs, does not result in unbalanced similarity scores. An extension thereof based on simple statistics provides the basis for an even further adaptation of initial facet weights that in a way reflective of the relative importance of each facet. The nature of the name measure to use, i.e., string-based or term-based, is heuristically determined based on similar reasoning.

Many of the optimizations could not be implemented in the current version. Yet a new optimization could be designed to help offset the impact of meaningless names. It consists in replacing the label of a token vertex with the set of labeled paths that head towards that vertex.

Link to OLA:
`https://gforge.inria.fr/frs/?group_id=271`

Link to alignments and parameters file :
`http://ola.gforge.inria.fr/results/OAEI-2007-OLA.zip`

3 Results of Execution on Test Cases

3.1 Benchmarks

#101-104:

- Language variations

- **Mean Precision =** 1.00 and **Mean Recall =** 1.00

#201-204:

- Alteration of names and/or suppresion of comments
- **Mean Precision =** 0.92 and **Mean Recall =** 0.92

#205-210:

- Synonyms and/or foreign language
- **Mean Precision =** 0.90 and **Mean Recall =** 0.90

#221-223:

- Alteration of specialization hierarchy
- **Mean Precision =** 0.99 and **Mean Recall =** 1.00

#224-228:

- Absence of instances, properties and/or restrictions
- **Mean Precision =** 1.00 and **Mean Recall =** 1.00

#230-231:

- Classes expanded or flattened
- **Mean Precision =** 0.96 and **Mean Recall =** 1.00

#232-247:

- Alteration/suppresion of specialization hierarchy
- Suppression of some properties and/or instances
- **Mean Precision =** 0.97 and **Mean Recall =** 1.00

#248-266:

- Alteration of all names
- Suppression of all comments
- Alteration of specialization hierarchy (in most cases)
- Suppression of some instances and/or properties
- **Mean Precision =** 0.77 and **Mean Recall =** 0.51

#301-304:

- Real-world ontologies
- **Mean Precision =** 0.63 and **Mean Recall =** 0.73

3.2 Conference

- We align every possible couple of ontologies
- We ran OLA2 with the set of parameters used for the benchmarks test case

3.3 Directory

We ran OLA2 with the set of parameters used for the benchmarks test case.

3.4 Others

OLA2 was unable to run because of the large size of matrices extracted from ontology graphs.

4 General Comments

4.1 Results

OLA2 significantly improves on the previous version of OLA. This may be seen on the benchmark results which are better. The additional benefits of the new implementation are extensibility and modularity of code.

Yet the method heavily relies on similarity of vertice labels (entity names or paths). A look on the tests where OLA scored poorly reveals that.

- #201-204 & #248-266: choice among entities having similar roles within their respective ontology graphs (e.g., test #253),
- #205-210 & #301-304: lack of semantic string distance and language translator,
- #221-223 & #230-231 & #232-247: no explicit inheritance edge between classes and properties

4.2 Future Improvements

1. Factorization of ontology graphs to run OLA on large-size ontologies [7, 8];
2. Search for the set of optimal of weight values;
3. Integration of semantic string distances [9] within the OLA matching process;
4. Integration of explicit inheritance edges among classes and among properties.

5 Conclusion

OLA2 is arguably better now than two years ago. The progress on real-world ontologies (30X), a class the previous version had difficulties dealing with, is encouraging. Yet more encouraging is the fact that these results have been obtained with very few adaptation tricks. In this respect, our next target will be the weight computing mechanisms of previous OLA.

Acknowledgments

This work was partly supported by the NSERG Discovery grant of the third author. The second author has been supported by the European integrated project NeOn (IST-2005-027595) and the RNTL project Web-Content. A short term visit by the first author at EXMO research team, INRIA Rhône-Alpes, France was jointly supported by the NSERG and NeOn grants.

References

1. Euzenat, J., Valtchev, P.: An integrative proximity measure for ontology alignment. In: Semantic Integration Workshop, Second International Semantic Web Conference (ISWC-03). (2003)
2. Euzenat, J., Valtchev, P.: Similarity-based ontology alignment in OWL-Lite. In: European Conference on Artificial Intelligence ECAI-04. (2004) 333–337
3. Euzenat, J., Loup, D., Touzani, M., Valtchev, P.: Ontology alignment with OLA. In Sure, Y., Corcho, O., Euzenat, J., Hughes, T., eds.: Proceedings of the 3rd ISWC'2004 workshop on Evaluation of Ontology-based tools (EON). (2004) 59–68
4. Euzenat, J., Guégan, P., Valtchev, P.: OLA in the OAEI 2005 alignment contest. In Ashpole, B., Euzenat, J., Ehrig, M., Stuckenschmidt, H., eds.: Proceedings of the K-CAP Workshop on Integrating Ontologies 2005. (2005) 61–71
5. Blondel, V.D., Heymans, A.G.M., Senellart, P., Dooren, P.V.: A measure of similarity between graph vertices. with applications to synonym extraction and web searching. SIAM Review **46** (2004) 647–666
6. Hu, W., Jian, N., Qu, Y., Wang, Y.: GMO: A graph matching for ontologies. In: K-CAP Workshop on Integrating Ontologies. (2005) 43–50
7. Papadopoulos, C., Voglis, C.: Drawing graphs using modular decomposition. In Healy, P., Nikolov, N.S., eds.: Graph Drawing. Volume LNCS 3843., Springer (2005) 343–354
8. Ryan, M.R.K.: Graph decomposition for efficient multi-robot path planning. In: IJCAI. (2007) 2003–2008
9. Euzenat, J., Shvaiko, P.: Ontology matching. Springer-Verlag, Heidelberg (DE) (2007)

OntoDNA: Ontology Alignment Results for OAEI 2007

Ching-Chieh Kiu[1], Chien–Sing Lee[2]

Faculty of Information Technology, Multimedia University,
Jalan Multimedia, 63100 Cyberjaya, Selangor. Malaysia.
[1]cckiu@mmu.edu.my, [2]cslee@mmu.edu.my

Abstract. OntoDNA is an automated ontology mapping and merging system that utilizes unsupervised data mining methods, comprising of Formal Concept analysis (FCA), Self-Organizing map (SOM) and K-means incorporated with lexical similarity, namely Levenshtein edit distance. The unsupervised data mining methods are used to resolve structural and semantic heterogeneities between ontologies, meanwhile lexical similarity is used to resolve lexical heterogeneity between ontologies. OntoDNA generates a merged ontology in concept lattice that enables visualization of the concept space based on formal context. This paper briefly describes the OntoDNA system and discusses the obtained alignment results on some of the OAEI 2007 dataset. The paper also presents strengths and weaknesses of our system and the method to improve the current approach.

1 Presentation of the system

1.1 State, purpose, general statement

OntoDNA is an automated ontology mapping and merging tool that provides a scalable environment for interoperating ontologies between information sources. OntoDNA aims to offer contextual and robust ontology mapping and merging through hybrid unsupervised clustering techniques, which comprises of Formal Concept Analysis (FCA) [1], Self-Organizing Map (SOM) and K-Means clustering [2] incorporated with a lexical measurement, Levenshtein edit distance [3]. OntoDNA generates a merged ontology in concept lattice form that enables visualization of the concept space based on formal context.

1.2 Specific techniques used

Ontology is formalized as a tuple $O := (C, S_C, P, S_P, A, I)$, where C is concepts of ontology and S_C corresponds to the hierarchy of concepts. The relationship between the concepts is defined by properties of ontology, P whereas S_P corresponds to the hierarchy of properties. A refers to axioms used to infer knowledge from existing knowledge and I instances of concept [4]. The OntoDNA resolves heterogeneous ontologies by capturing ontological concepts (C) and its ontological elements (S_C, P, S_P, A) [5].

The OntoDNA utilizes FCA to capture the properties and the inherent structural relationships among ontological concepts of heterogeneous ontologies. The captured structures of ontological concepts act as background knowledge to resolve semantic interpretations in similar (synonymy) or different contexts (polysemy).

The unsupervised clustering techniques, Self-Organizing Map (SOM) and K-Means are used to overcome the absence of prior knowledge to discover the structural and semantic heterogeneities between ontologies. SOM organizes ontological elements, clustering more similar ontological concepts together. The clusters of the ontological concepts are derived from the natural characteristics of the ontological elements. Meanwhile K-Means is used to reduce the problem size of the SOM map for efficient semantic heterogeneous discovery in different contexts.

The OntoDNA relies on lexical similarity to resolve lexical heterogeneity by both ontological concept and property names. The lexical similarity, Levenshtein edit distance with the threshold value 0.8 [5] is applied to discover lexical similarity. Prior to the discovery of the degree of lexical similarity, linguistic processing such as case normalization, blank normalization, digit normalization, namespace prefixes elimination, link stripping, and stopword filtering are applied to normalize ontological elements.

The OntoDNA automated ontology mapping and merging framework is depicted in Figure 1. The terms used in the OntoDNA framework are defined as follows:

– Source ontology O_S: Source ontology is the local data repository ontology
– Target ontology O_T: Target ontology refers to non-local data repository ontology
– Formal context K_S and K_T: Formal context K_S is the formal context representation of the conceptual relationship of the source ontology O_S, meanwhile formal context K_T is the formal context representation of the conceptual relationship of the target ontology $O_{T.}$
– Reconciled formal context ${}_RK_S$ and ${}_RK_T$: Reconciled formal context ${}_RK_S$ and ${}_RK_T$ are formal context with normalized intents of source and target ontological concepts' properties.
– The ontological elements $O := (C, S_C, P, S_P, A)$: C is concepts of ontology and S_C corresponds to the hierarchy of concepts. P is properties of ontology, and S_P corresponds to the hierarchy of properties. A refers to axioms.

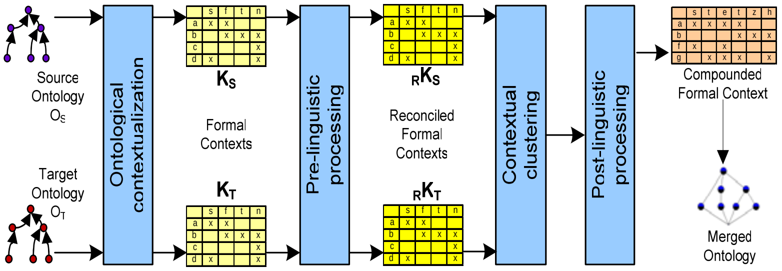


Figure 1. OntoDNA's framework

The OntoDNA algorithmic framework implementation of the automated mapping and merging process illustrated in Figure 1 is explicated below [5] [6]:

Input **:** Two ontologies that are to be merged, O_S (source ontology) and O_T (target ontology).

Step 1 **:** ***Ontological contextualization***

The conceptual pattern of O_S and O_T is discovered using FCA. Given an ontology $O := (C, S_C, P, S_P, A)$, O_S and O_T are contextualized using FCA with respect to the formal context, K_S and K_T. The ontological concepts C are denoted as G (objects) and the rest of the ontology elements, S_C, P, S_P and A are denoted as M (attributes). The binary relation $I \subseteq G \times M$ of the formal context denotes the ontology elements, S_C, P, S_P and A corresponding to the ontological concepts C.

Step 2 **:** ***Pre-linguistic processing***

String normalizations are applied to transform attributes in K_S and K_T prior to lexical similarity mapping. The mapping rules (Map_Rule 1 and Map_Rule 2) (Table 1) are applied to reconcile intents in K_S and K_T. The reconciled formal context ${}_RK_S$ and ${}_RK_T$ are output as input for semantic similarity discovery in the next step.

Step 3 **:** ***Contextual clustering***

SOM and k-means are applied for semantic similarity mapping based on the conceptual pattern discovered in the formal context. First, the formal context ${}_RK_T$ is trained by SOM. This is followed by k-means clustering to reduce the problem size of the SOM clusters as validated by the Davies-Bouldin index. Subsequently, the formal concepts ${}_RK_S$ are fed to the trained SOM. The source ontological concepts are assigned to the same cluster as their Best Matching Units (BMUs) in the target ontology.

Step 4 **:** ***Post-linguistic processing***

The mapping rules (Map_Rule 1 and Map_Rule 2) (Table 1) are applied to discover semantic similarity between ontological concepts in the clusters. The ontological concepts of the target ontology are updated to the source ontology based on merging rules (Merge_Rule 1 and Merge_Rule 2) (Table 1).

Output : Merged ontology in a concept lattice is formed.

Mapping Rules
Given source ontological element $O_{elementSi}$ and target ontological element $O_{elementTj}$, apply lexical similarity measure (*LSM*) to map the target ontology O_T to the source ontology O_S at threshold value, *t*, where elements *i* and *j* = 1, 2, 3, …, n. ***Map_Rule 1:*** map ($O_{elementTj} \rightarrow O_{elementSi}$), if $\text{LSM}(O_{elementSi}, O_{elementTj}) \geq t$; the target ontological element, $O_{elementTj}$ is mapped to (integrated with) the source ontological element, $O_{elementSi}$ and the naming convention and structure of the source ontological element, $O_{elementSi}$ are preserved. ***Map_Rule 2:*** merge ($O_{elementTj} \rightarrow O_S$), if $\text{LSM}(O_{elementSi}, O_{elementTj}) < t$; the target ontological element, $O_{elementTj}$ is merged (appended) to the source ontology and the naming convention and structure of the target ontological element, $O_{elementTj}$ are preserved.
Merging Rules
Given the source ontology O_S in a reconciled formal context $k = (G, M, I)$ and target ontology O_T in a reconciled formal context $l = (H, N, J)$. The source ontology is the base for ontology merging. ***Merge_Rule 1:*** If Map_Rule 1 or Map_Rule 3 is true, the intents of $O_{elementTj}$ (ontological concepts) and its object-attribute relationship $J \subseteq H \times N$ is aligned (appended) into formal context *k*. ***Merge_Rule 2:*** If Map_Rule 2 is true, and formal context *k* is defined by $(O_{extentS1}, O_{intentS1}) \leq (O_{extentS2}, O_{intentS2}) :\Leftrightarrow O_{extentS1} \subseteq O_{extentS2}$ $(\Leftrightarrow O_{intentS1} \subseteq O_{intentS2})$ the intents of $O_{elementTj}$, its object-attribute relationship $J \subseteq H \times N$ and its *subconcept - superconcept relation* of $O_{elementTj}$ among other concepts are aligned into formal context *k*, whereas the structural relationships of the appended concept is updated with the target ontology as the base.

Table 1. Ontology mapping and merging rules

1.3 Adaptations made for the evaluation

There is no special adaptation for the tests in the Ontology Alignment Evaluation Initiative (OAEI) 2007 campaign. However, a small program is written to translate our native alignment format in the form that is required by the OAEI contest. The URI for benchmark ontology 302 has been manually replaced in order to output the alignment file.

1.4 Link to the system, parameters file and to the set of provided alignments

The OntoDNA system and the alignment results in a ZIP file organized as presented can be downloaded from http://pesona.mmu.edu.my/~cckiu/OAEI2007.htm.

2 Results

The OAEI 2007 campaign provides four ontology tracks, which consist of benchmark, anatomy, directories and thesauri and conference. Due to the ontologies' file size, we manage only to run the alignment tests on the benchmark, directories and conference tracks. In this section, we discuss the results on the benchmark track followed by the experimental outcomes on other tracks.

2.1 Comparison track: benchmark

The benchmark track consists of 51 alignment tests. The alignment results can be divided into five categories for discussion, i.e. Tests 101 – 104, Tests 201 – 210, Tests 221 – 247, Tests 248 – 266 and 301 – 304. The full result of all the alignment tests can be referred in the Appendix.

Tests 101 – 104: The alignment tests consist of the reference alignment, irrelevant ontology, language generalization and language restriction. Overall performance of OntoDNA in the tests is good. The OntoDNA has no problem handling the language generalization (test 103) and language restriction (test 104) features in the tests. The average precision and recall achieved by the OntoDNA are 0.94 and 1.00 respectively as shown in Table 2.

Test	Name	Prec.	Rec.	Time (sec)
101	Reference alignment	0.94	1.00	6.53
102	Irrelevat ontology	NaN	NaN	169.83
103	Language generalization	0.94	1.00	6.36
104	Language restriction	0.94	1.00	6.14
	Average	0.94	1.00	47.22

Table 2. Alignment result for Tests 101 – 104

Tests 201 – 210: The alignment tests manipulate names and comments. Since the OntoDNA relies on the name of classes and properties to resolve lexical heterogeneity, this has resulted in very poor performance in terms of precision and recall for tests 201 and 202 as the name of the labels are not provided. The alignment results on tests 206, 207 and 210 are also poor as the name of the labels are in French translations, and OntoDNA does not understand non-English translations. In addition, as the OntoDNA does not use any thesaurus for resolving lexical similarity, it can't perform well in tests 205 and 209 as illustrated in Table 3.

Test	Name	Prec.	Rec.	Time (sec)
201	No names	0.11	0.01	9.77
202	No names, no comments	0.11	0.11	9.13
203	No comments	0.94	1.00	6.17
204	Naming conventions	0.93	0.84	8.25
205	Synonyms	0.57	0.12	9.31
206	Translation (name)	0.69	0.23	8.61
207	Translation (name and comments)	0.69	0.23	8.52
208	Naming conventions, no comments	0.93	0.84	7.05
209	Synonyms, no comments	0.57	0.12	8.72
210	Translation, no comments	0.69	0.23	8.45
	Average	0.62	0.37	8.40

Table 3. Alignment result for Tests 201 – 210

Tests 221 – 247: The alignment tests manipulate hierarchy. The overall performance of the OntoDNA is good with any kind of hierarchy manipulation (no specialization, flattened hierarchy and expanded hierarchy). However, the OntoDNA alignment results for tests 228, 233, 236, 239, 240, 241, 246 and 247 are poor when the properties are suppressed from the tests as displayed in Table 4.

Test	Name	Prec.	Rec.	Time (sec)
221	No specialisation	0.93	0.76	6.38
222	Flatenned hierachy	0.94	1	7.69
223	Expanded hierarchy	0.94	1	8.69
224	No instance	0.94	1	6.16
225	No restrictions	0.94	1	6.14
228	No properties	0.53	0.27	4.95
230	Flatenned classes	0.91	1	5.97
231	Expanded classes	0.94	1	6.50
232	No specialisation, no instance	0.93	0.76	6.42
233	No specialisation, no properties	0.53	0.27	4.97
236	No instance, no properties	0.53	0.27	4.89
237	Flatenned hierachy, no instance	0.94	1	5.94
238	Expanded hierachy, no instance	0.94	1	8.61
239	Flatenned hierachy, no properties	0.5	0.31	4.94
240	Expanded hierachy, no properties	0.5	0.27	7.06
241	No specialisation, no instance, no properties	0.53	0.27	5.44
246	Flatenned hierachy, no instance, no properties	0.5	0.31	5.03
247	Expanded hierachy, no instance, no properties	0.5	0.27	6.86
	Average	0.75	0.65	6.26

Table 4. Alignment result for Tests 221 - 247

Tests 248 – 266: The alignment tests manipulate hierarchy, labels and comments. The precision and recall of the tests achieved by the OntoDNA are very poor as the names and properties are suppressed as shown in Table 5. The results have proven that the OntoDNA ~~is~~ strictly relies on ontological concepts and properties name for mapping and merging the ontologies.

Test	Name	Prec.	Rec.	Time (sec)
248	No names, no comments, no specialisation	0.11	0.01	9.23
249	No names, no comments, no instance	0.11	0.01	9.23
250	No names, no comments, no properties	0	0	5.95
251	No names, no comments, flatenned hierachy	0.11	0.01	8.89
252	No names, no comments, expanded hierachy	0.11	0.01	12.66
253	No names, no comments, no specialization, no instance	0.11	0.01	9.28
254	No names, no comments, no specialization, no properties	0	0	6.30
257	No names, no comments, no instance, no properties	0	0	5.91
258	No names, no comments, flatenned hierachy, no instance	0.11	0.01	9.95
259	No names, no comments, expanded hierachy, no instance	0.11	0.01	13.2
260	No names, no comments, flatenned hierachy, no properties	0	0	5.86
261	No names, no comments, expanded hierachy, no properties	0	0	8.13
262	No names, no comments, no specialization, no instance, no properties	0	0	6.09
265	No names, no comments, flatenned hierachy, no instance, no properties	0	0	5.75
266	No names, no comments, expanded hierachy, no instance, no properties	0	0	7.88
	Average	0.05	0.00	8.29

Table 5. Alignment result for Tests 248 - 266

Tests 301 – 304: The alignment tests consist of real bibliographic ontologies. The average precision and recall on the tests are 0.90 and 0.69 respectively achieved by the OntoDNA (Table 6). The results in the tests show that the OntoDNA is a viable automated ontology mapping and merging tool to resolve the heterogeneity of the real ontologies from disparate information sources.

Test	Name	Prec.	Rec.	Time (sec)
301	BibTeX/MIT	0.88	0.69	5.84
302	BibTeX/UMBC	0.9	0.4	5.53
303	Karlsruhe	0.9	0.78	9.95
304	INRIA	0.92	0.88	6.77
	Average	0.90	0.69	7.02

Table 6. Alignment result for Tests 301 - 304

2.2 Expressive ontology: anatomy

We are not able to perform the alignment test on this ontology track due to the large size of the ontology files.

2.3 Directories and thesauri

In this ontology track, there are four ontology alignment tests, i.e., directory, food, environment and library. We manage only to perform the alignment test on the directory ontologies. The alignment tests are not run on food, environment and library ontologies due to the large size of the ontology files.

The directory track is the real world case of websites directory consisting of 4640 alignment tests. Each of the alignment tests contains source and target ontologies. The ontologies are taxonomic ontologies as each of the ontologies contains only classes with superclass-subclass relationships. Since the organizers do not provide the alignment results, we expect feedback on the OntoDNA performance on the directory alignment tests from the organizers.

2.4 Consensus workshop: conference

The conference track consists of 14 real conference ontologies from conference organizations. We have performed 182 alignment tests by aligning an ontology to other ontologies (14 x 13) in the track. Since the alignment results are not provided by the organizers, we expect feedback on the OntoDNA performance on the conference alignment tests from the organizers.

3 General comments

In this section, we summarize the strengths and weaknesses of the OntoDNA and discuss the methods to improve the OntoDNA algorithm.

3.1 Comments on the results

The OntoDNA is an automated ontology mapping and merging tool. All the parameters such as threshold value used for the lexical similarity discovery and the clustering parameters used for structural and semantic similarity discovery are predetermined based on the experimental results on numerous datasets [5][6]. Thus the OntoDNA is a viable tool for mapping and merging ontologies without requiring prior knowledge of the source and target ontological elements.

The limitation of the OntoDNA is the system strictly relies on the name of the ontological concepts and properties to resolve the heterogeneity of ontologies. Thus if the given ontology does not contain the name of the ontological concepts and properties, the OntoDNA is not able to discover lexical similarity for resolving the structural and semantic heterogeneous between the source and target ontologies.

However, given the name of the ontological concepts and properties, the tests results have confirmed that the OntoDNA is an effective system for mapping and merging real ontologies without human intervention in the mapping and merging processes.

3.2 Discussions on the way to improve the proposed system

Based on the tests results, the OntoDNA may need to consider other ontological elements as core elements for mapping and merging. Thus, the structural approach and logic approach can be extended into the OntoDNA algorithm to discover the alignment between source and target ontologies when the ontological concepts and properties name are suppressed (absent). A multi-strategy approach combining linguistic, structural and logic approaches with specific threshold value might also improve OntoDNA's performance.

4 Conclusion

The participation in the OAEI 2007 campaign enables us to identify the strengths and weaknesses of the OntoDNA algorithm and also the methods to improve the OntoDNA algorithm. The presented alignment results show that the OntoDNA has performed well in both ontological concept and property names for mapping and merging ontologies automatically.

References

1. Ganter, B., Wille, R.: Applied Lattice Theory: Formal Concept Analysis, http://www.math.tudresden.de/~ganter/psfiles/concept.ps, (1997).
2. Vesanto, J., Alhoniemi, E.: Clustering of the Self-Organizing Map. IEEE Transactions on Neural Networks, vol. 11(3), (2000) 586-600.
3. Do, H., Melnik, S., Rahm, E.: Comparison of schema matching evaluations. In Proc. of the second int. workshop on Web Databases (German Informatics Society) (2002).
4. Gruber, T.R.: A Translation Approach to portable Ontology Specifications. Knowledge Acuquisition, vol. 5, (1993) 199-220.
5. Kiu, C. C., Lee. C. S.: Ontology Mapping and Merging through OntoDNA for Learning Object Reusability. Educational Technology & Society, 9(3), (2006) 27-42.
6. Lee. C. S., Kiu, C. C.: A concept-based graphical – neural approach to ontological interoperability. WSEAS Transactions on Information Science and Applications, 2 (6), (2005) 761-770.

Appendix: Raw results

All the alignment tests are carried out using a notebook with Core Duo T2250 1.73 GHz processor and 1GB RAM in Window XP environment. The precision and recall on the alignment tests with machine processing time in hh.mm.ss.mms format are presented here.

Matrix of results

#	Name	Prec.	Rec.	Time
101	Reference alignment	0.94	1.00	00:00:06:53
102	Irrelevat ontology	NaN	NaN	00:02:49:83
103	Language generalization	0.94	1.00	00:00:06:36
104	Language restriction	0.94	1.00	00:00:06:14
201	No names	0.11	0.01	00:00:09:77
202	No names, no comments	0.11	0.11	00:00:09:13
203	No comments	0.94	1.00	00:00:06:17
204	Naming conventions	0.93	0.84	00:00:08:25
205	Synonyms	0.57	0.12	00:00:09:31
206	Translation (name)	0.69	0.23	00:00:08:61
207	Translation (name and comments)	0.69	0.23	00:00:08:52
208	Naming conventions, no comments	0.93	0.84	00:00:07:05
209	Synonyms, no comments	0.57	0.12	00:00:08:72
210	Translation, no comments	0.69	0.23	00:00:08:45
221	No specialisation	0.93	0.76	00:00:06:38
222	Flatenned hierachy	0.94	1.00	00:00:07:69
223	Expanded hierarchy	0.94	1.00	00:00:08:69
224	No instance	0.94	1.00	00:00:06:16
225	No restrictions	0.94	1.00	00:00:06:14
228	No properties	0.53	0.27	00:00:04:95
230	Flatenned classes	0.91	1.00	00:00:05:97
231	Expanded classes	0.94	1.00	00:00:06:50
232	No specialisation, no instance	0.93	0.76	00:00:06:42
233	No specialisation, no properties	0.53	0.27	00:00:04:97
236	No instance, no properties	0.53	0.27	00:00:04:89
237	Flatenned hierachy, no instance	0.94	1.00	00:00:05:94
238	Expanded hierachy, no instance	0.94	1.00	00:00:08:61
239	Flatenned hierachy, no properties	0.50	0.31	00:00:04:94
240	Expanded hierachy, no properties	0.50	0.27	00:00:07:06
241	No specialisation, no instance, no properties	0.53	0.27	00:00:05:44
246	Flatenned hierachy, no instance, no properties	0.50	0.31	00:00:05:03
247	Expanded hierachy, no instance, no properties	0.50	0.27	00:00:06:86
248	No names, no comments, no specialisation	0.11	0.01	00:00:09:23
249	No names, no comments, no instance	0.11	0.01	00:00:09:23
250	No names, no comments, no properties	0.00	0.00	00:00:05:95
251	No names, no comments, flatenned hierachy	0.11	0.01	00:00:08:89
252	No names, no comments, expanded hierachy	0.11	0.01	00:00:12:66
253	No names, no comments, no specialization, no instance	0.11	0.01	00:00:09:28
254	No names, no comments, no specialization, no properties	0.00	0.00	00:00:06:30
257	No names, no comments, no instance, no properties	0.00	0.00	00:00:05:91
258	No names, no comments, flatenned hierachy, no instance	0.11	0.01	00:00:09:95
259	No names, no comments, expanded hierachy, no instance	0.11	0.01	00:00:13:20
260	No names, no comments, flatenned hierachy, no properties	0.00	0.00	00:00:05:86
261	No names, no comments, expanded hierachy, no properties	0.00	0.00	00:00:08:13
262	No names, no comments, no specialization, no instance, no properties	0.00	0.00	00:00:06:09
265	No names, no comments, flatenned hierachy, no instance, no properties	0.00	0.00	00:00:05:75
266	No names, no comments, expanded hierachy, no instance, no properties	0.00	0.00	00:00:07:88
301	BibTeX/MIT	0.88	0.69	00:00:05:84
302	BibTeX/UMBC	0.90	0.40	00:00:05:53
303	Karlsruhe	0.90	0.78	00:00:09:95
304	INRIA	0.92	0.88	00:00:06:77

OWL-CM : OWL Combining Matcher based on Belief Functions Theory

Boutheina Ben Yaghlane[1] and Najoua Laamari[2]

[1] LARODEC, Université de Tunis,
IHEC Carthage Présidence 2016 Tunisia
`boutheina.yaghlane@ihec.rnu.tn`
[2] LARODEC, Université de Tunis,
ISG Tunis Tunisia
`laamarinajoua@yahoo.fr`

Abstract. In this paper we propose a new tool called OWL-CM (OWL Combining Matcher) that deals with uncertainty inherent to ontology mapping process. On the one hand, OWL-CM uses the technique of similarity metrics to assess the equivalence between ontology entities and on the other hand, it incorporates belief functions theory into the mapping process in order to improve the effectiveness of the results computed by different matchers and to provide a generic framework for combining them. Our experiments which are carried out with the benchmark of Ontology Alignment Evaluation Initiative 2007 demonstrate good results.

1 Presentation of the system

1.1 State, purpose, general statement

Semantic heterogeneity has been identified as one of the most important issue in information integration [5]. This research problem is due to semantic mismatches between models. Ontologies which provide a vocabulary for representing knowledge about a domain are frequently subjected to integration.

Ontology mapping is a fundamental operation towards resolving the semantic heterogeneity. It determines mappings between ontologies. These mappings catch semantic equivalence between ontologies. Experts try to establish mappings manually. However, manual reconciliation of semantics tends to be tedious, time consuming, error prone, expensive and therefore inefficient in dynamic environments, and what's more the introduction of the Semantic Web vision has underscored the need to make the ontology mapping process automatic.

Recently, a number of studies that are carried out towards automatic ontology mapping draw attention to the difficulty to make the operation fully automatic because of the cognitive complexity of the human. Thus, since the (semi-) automatic ontology mapping carries a degree of uncertainty, there is no guarantee that the outputted mapping of existing ontology mapping techniques is the exact one.

In this context, we propose a new tool called OWL-CM (OWL Combining Matcher) with the aim to show how handling uncertainty in ontology mapping process can improve effectiveness of the output.

1.2 Specific techniques used

On the one hand OWL-CM uses the Dempster-Shafer theory of evidence [11] to deal with uncertainty inherent to the mapping process, especially when interpreting and combining the results returned by different matchers. On the other hand it uses the technique of similarity measures in order to assess the correspondence between ontology entities. For the OWL-CM tool contest we have proposed an architecture (see figure 1) that contains four components. The *transformer* takes as input two ontologies (O_1 and O_2) and constructs for each one a database (DB_1 and DB_2). The database schema meets a standard schema that we designed based on some axioms of RDF(S) and OWL languages. The *filters* decide on result mappings. Whereas *simple matchers* and *complex matchers* assess the equivalence between entities.

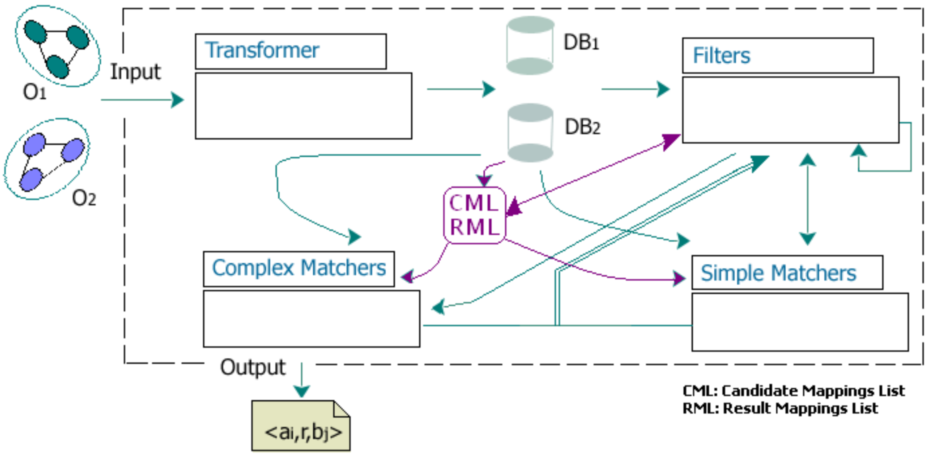


Fig. 1. OWL-CM Architecture.

The corresponding algorithm that we have implemented follows four steps (see figure 2). The first step called pre-mapping is mainly devoted to convert each one of the input ontologies O_1 and O_2 into a database (DB_1 and DB_2). The following three ones allow performing sequentially the iteration about concepts mapping, followed by the iteration about object properties mapping, and ended by the iteration about datatype properties mapping. Each iteration is based on some methods belonging to four categories of tasks namely initialization, screening, handling uncertainty, and ending. The algorithm requires as input two ontologies to be mapped and two databases that have to be declared as

ODBC data source systems. It outputs three lists of result mappings which are produced sequentially, each one is returned close of the corresponding iteration of mapping. The total result is returned in the form of a file.

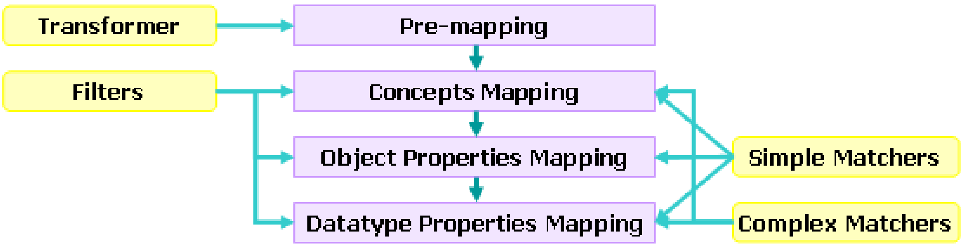


Fig. 2. OWL-CM Algorithm.

1.2.1 Preliminary concepts

The following list draws up some of the preliminaries that are used by our approach.

1. **Candidate Mapping**: We define a *candidate mapping* as a pair of entities $({e_i}^1, {e_j}^2)$ that is not yet in map.

2. **Result Mapping**: We define a *result mapping* as a pair of entities that had been related, $\langle {e_i}^1, \equiv, {e_j}^2 \rangle$ denotes that entity ${e_i}^1$ is equivalent to entity ${e_j}^2$, whereas $\langle {e_i}^1, \perp, {e_j}^2 \rangle$ denotes that the two entities are not equivalent.

3. **Similarity measure**: The *similarity measure*, *sim*, is a function defined in [3] based on the vocabularies ε_1 of the ontology O_1 and ε_2 of the ontology O_2 as follows:

$$sim\colon \varepsilon \times \varepsilon \times O \times O \rightarrow [0..1]$$

   - *sim(a, b) = 1 ⇔ a = b: two objects are assumed to be identical.*
   - *sim(a, b) = 0 ⇔ a ≠ b: two objects are assumed to be different and have no common characteristics.*
   - *sim(a, a) = 1: similarity is reflexive.*
   - *sim(a, b) = sim(b, a): similarity is symmetric.*
   - *Similarity and distance are inverse to each other.*

   A similarity measure function assesses the semantic correspondence between two entities based on some features. In table 1, we draw up the list of similarity measures employed depending on the type of entities to be mapped. Furthermore, we distinguish between two types of similarity: the *syntactic one* assessed by the measures that evaluate distance between strings (e.g., String similarity and String equality) and the other measures dedicated to assess *semantic similarity* (e.g., String synonymy, Explicit equality and Set similarity).

4. **SEE (Semantic Equivalent Entity)**: Depending on the type of entities, we formally define the *semantic equivalence* between two entities as follows:
   Definition (SEE) .
   An entity ${e_j}^2$ is semantically equivalent to an entity ${e_i}^1$ such that $({e_i}^1, {e_j}^2) \in \{C^1 \times C^2\}$, i.e., $\langle {e_i}^1, \equiv, {e_j}^2 \rangle$, if at least one of the following conditions is true:
   $sim_{expeql}({e_i}^1, {e_j}^2) = 1$, or
   $\forall\ sim_k$, with $k \neq expeql, sim_k({e_i}^1, {e_j}^2) = 1$

   An entity ${e_j}^2$ is semantically equivalent to an entity ${e_i}^1$ such that $({e_i}^1, {e_j}^2) \in \{{R_c}^1 \times {R_c}^2 \cup {R_d}^1 \times {R_d}^2\}$, i.e., $\langle {e_i}^1, \equiv, {e_j}^2 \rangle$, if:
   $\forall\ sim_k, sim_k({e_i}^1, {e_j}^2) = 1$

Table 1. Features and Measures for Similarity

Entities to be compared	No.	Feature (f)	Similarity measure
Concepts: C	1	(label, C1)	$sim_{strsim}(C1, C2)$
	2	(sound (ID), C1)	$sim_{streql}(C1, C2)$
	3	(label, C1)	$sim_{strsyn}(C1, C2)$
	4	(C1,equalTo, C2) relation	$sim_{expeql}(C1, C2)$
	5	(C1,inequalTo, C2) relation	$sim_{expineq}(C1, C2)$
	6	all (direct-sub-concepts, S1)	$sim_{setsim}(S1, S2)$
Relations: R_c	7	(sound (ID), R1)	$sim_{streql}(R1, R2)$
	8	(domain, R1)∧(range, R1)	$sim_{objeql}(R1, R2)$
	9	(domain, R1)∧(range, R1)	$sim_{objineq}(R1, R2)$
	10	all (direct-sub-properties, S1)	$sim_{setsim}(S1, S2)$
Relations: R_d	11	(sound (ID), R1)	$sim_{streql}(R1, R2)$
	12	(domain, R1)∧(range, R1)	$sim_{objeql}(R1, R2) \wedge$ $sim_{streql}(R1, R2)$
	13	(domain, R1)	$sim_{objineq}(R1, R2)$
	14	all (direct-sub-properties, S1)	$sim_{setsim}(S1, S2)$

5. **USEE (Uncertain Semantic Equivalent Entity)**: We extend the definition of SEE to USEE in order to be used throughout the process of handling uncertainty when performing and combining matchers.
   Definition (USEE) . An entity that we said to be uncertain and semantically equivalent to an ontological entity $e \in O_1$ is a pair (Θ, m), where:
   $\Theta = E$, $E \in \{C^2, {R_c}^2, {R_d}^2\}$
   m is a belief mass function (See Section 1.2.2).

1.2.2 Handling uncertainty

The Dempster-Shafer theory of evidence [11] presents some advantages that encourage us to choose among other theories. In particular, it can be used for the problems where the existing information is very fragmented, and so the information can not be modelled with a probabilistic formalism without making arbitrary hypotheses. It is also considered as a flexible modelling tool making it possible to handle different forms of uncertainty, mainly the ignorance. Moreover, this theory provides a method for combining the effect of different beliefs to establish a new global belief by using Dempster's rule of combination.

The belief mass function *m(.)* is the basic concept of this theory ([11], [12]). It assigns some belief mass in the interval [0,1] to each element of the power set 2^{Θ} of the frame of discernment Θ. The total mass distributed is 1 and the *closed world hypothesis* (i.e. $m(\emptyset) = 0$) is generally supported. In our work, $\Theta \in \{C^2, {R_c}^2, {R_d}^2\}$. The letter Φ in table 2 is the set of all candidate mappings.

Table 2. Frame of Discernment and Candidate Mappings Set.

$$\begin{array}{ll}
\begin{array}{|c|ccc|} \hline & {e_1}^2 & \dots & {e_m}^2 \\ \hline \end{array} & \Rightarrow \Theta \\
\left.\begin{array}{|c|ccc|} {e_1}^1 & ({e_1}^1, {e_1}^2) & \dots & ({e_1}^1, {e_m}^2) \\ \dots & \dots & \dots & \dots \\ {e_n}^1 & ({e_n}^1, {e_1}^2) & \dots & ({e_n}^1, {e_m}^2) \\ \hline \end{array}\right\} & \Phi
\end{array}$$

In order to discover USEEs, we use n functions called matchers $(matcher_k)^3$. A matcher compared to a "witness" that brings evidence in favor or against an advanced hypothesis. Matchers produce USEEs in order to support uncertainty. Some matchers are reliable than others. This is reflected in the confidence that is assigned to each matcher. The confidence is expressed like the mass that is distributed to Θ. For instance, if $matcher_1$ has a confidence of .6, then the masses assigned to the subsets should be normalized to sum .6, and .4 should be always affected to Θ.

We use Dempster's rule of combination to aggregate the produced USEEs. Figure 3 illustrates the architecture that we propose to discover USEEs. In addition, this theory makes it possible to express total ignorance. For instance, if the set that contains the entities having the same sound as the entity in question is empty, then the matcher $matcher_2$ will return a belief mass function $m(\Theta) = 1$.

1.3 Adaptations made for the evaluation

Our mapping algorithm has been recently conceived so to speak that our tool OWL-CM is in an alpha version and we evaluate it for the first time.

---

[3] The index k is the No. of the matcher in the table 1.

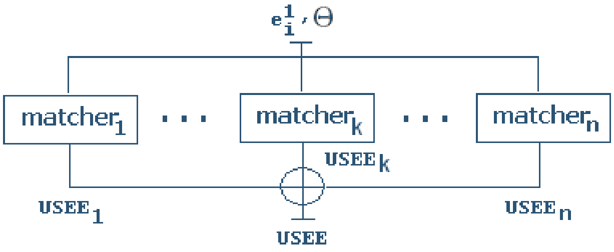


Fig. 3. Architecture for discovering USEEs.

2 Results

The tests have been carried out with the data of the benchmark of Ontology Alignment Evaluation Initiative 2007. Our experiments are restricted to the following metrics that evaluate the goodness of the algorithm output and which are derivatives of well-known metrics from the information retrieval domain [6]: *Precision*, *Recall*, and *FMeasure*. The mapping algorithm has been implemented in java and been updated so that it returns the results in the required format.

2.1 Tests 101-104

Our results (see result Figure 4) show that our mapping algorithm enabled us to achieve 100% precision and 100% recall in the tests 101, 103 and 104. The test 102 also shows the performance of the algorithm.

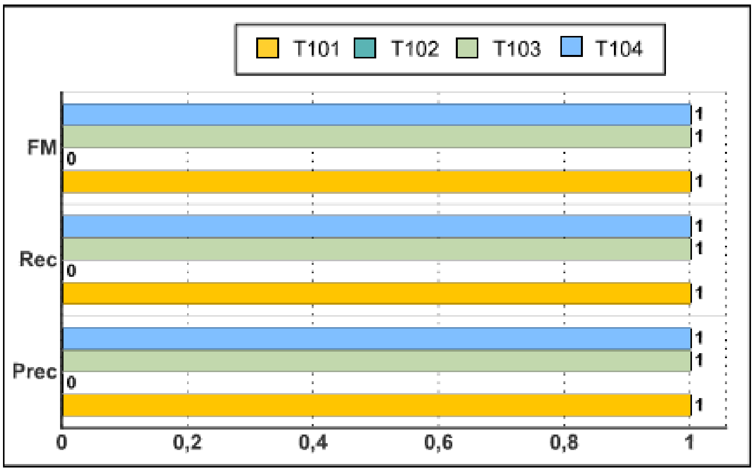


Fig. 4. Results of Tests 101-104.

2.2 Tests 201-204

The ontology 201 does not contain names and the ontology 202 contains neither names nor comments, so we will not consider the results of these tests. In fact, our algorithm considers concept and property IDs (identified by the "*rdf:ID*" tag) as well as their labels (extracted from "*rdfs:label*" tag), therefore the only information that can be used to create these result mappings in the test 201 is comments, but our algorithm does not use it. Although the performed tests are not worth considering, even though they reveal a higher precision (see result Figure 5).

Concerning the tests 203 and 204, our mapping algorithm creates the mapping with high precision (see result Figure 5). Recall values are also considerable.

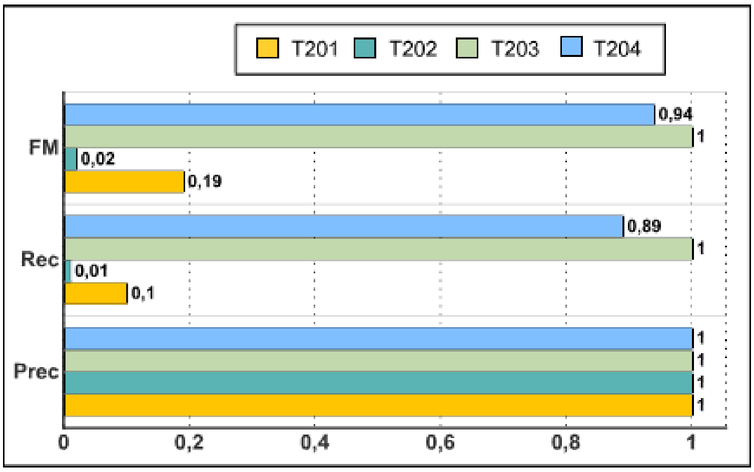


Fig. 5. Results of Tests 201-204.

2.3 Tests 205-210

Before starting the commentary, we note that ontologies 205, 206, 207, 209 and 210 contain doubloons in rdf-ID feature (e.g., there are two datatype properties with the rdf-ID "issue" in the ontologies 205 and 209). However, our algorithm does not allow this as it considers the rdf-ID to be the attribute that identifies the entity in the database during the pre-mapping step. So, in order to don't miss these tests, our algorithm only mapped the parts of ontologies that it was able to convert.

Since our algorithm does not make use of comments in the mapping, we group the tests according to alterations relating to names. Thus we distinguish three behaviors of the algorithm (see Figure 6):

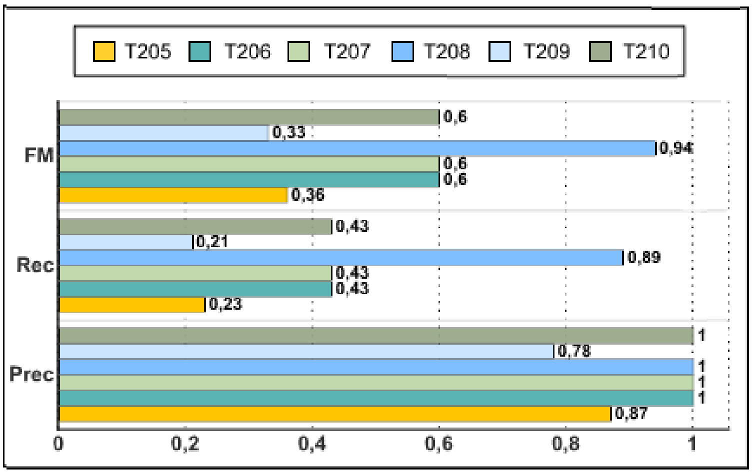


Fig. 6. Results of Tests 205-210.

– Both ontologies 205 and 209 were mapped with good precision but the recall scale is ever such low. Concerning the test case 205 we explain the weakness of the recall by the fact that the searching for Wordnet synonyms, which is the function of some matcher, is made based on full labels. The percentages of precision and recall of the second test case are a bit lower than the ones in the first test case. This note goes to show that the matchers, which deal with labels, have a part in the success of mapping.
– The algorithm generated quite good mappings for the ontologies 206, 207, and 210 with extreme precision and quite satisfactory recall. The results depicted in Figure 6 show that the precision and recall are the same for the three tests which can be explained by some reasons. On the one hand, the fact of keeping or suppressing comments does not have effect on the produced mappings at all as the algorithm doesn't make use of this information. On the other hand, since the labels are translated to French, so the matchers, which deal with labels, are faced with a situation of total ignorance. We conclude that the difference in language between ontologies affects the mapping.
– The test case 208 is similar to the test case 204 where the name of each entity is replaced by another one with different conventions.

2.4 Tests 221-247

Different categories of alteration have been carried out in each of these test cases. The precision and recall percentages of ontology mapping during these tests (see results in Figure 7) are equal or close to 100%. This result confirms that our algorithm takes both syntactic and semantic similarity into account.

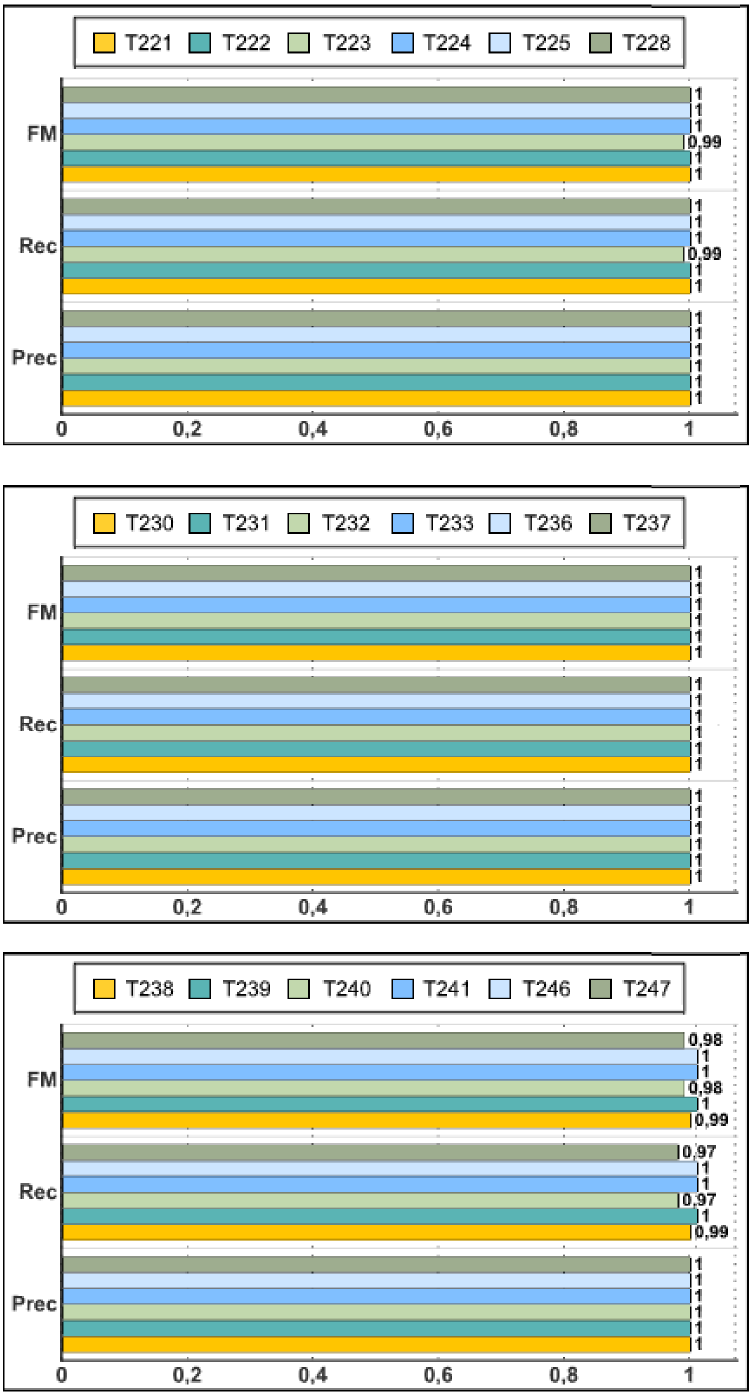


Fig. 7. Results of Tests 221-247.

2.5 Tests 248-266

As names of entities from reference ontology have been replaced by random strings and as our algorithm considers only concept and property IDs as identified by the *"rdf:ID"* tag, the ontologies of these tests were not mapped at all by our algorithm. What's more, the only information that the algorithm can make use of it to create mappings, except in tests 248, 253, 254 and 262, is the specialization hierarchies of classes and properties, which are described, respectively, through the tag "*rdfs:subClassOf*" and the tag "*rdfs:subPropertyOf*". However, since the matchers that make use of this information are complex, therefore these tests have not produced any result mapping.

2.6 Tests 301-304

Before starting the analysis of results, we note that we reduced the three collections of result mappings (col-301: from 61 to 39, col-302: from 48 to 26, col-304: from 76 to 74). This is due to, among other raisons, the fact that the three collections contain some concepts and properties that are matched with the "<" relation while our algorithm only uses the "=" relation.
The result mappings produced by the algorithm are with high precision (see Figure 8). The recall is high for the test 302 and relatively good for the test 304, but the ontology 301 was mapped with weak recall. More in detail, the weakness in the recall of the test 301 is in the mapping of datatype properties. This is due to some reasons that affect the execution of some matchers, such as the difference in the hierarchies between the ontologies in the test 301. Concerning the ontology 303, it was not mapped at all by the algorithm. In fact local entities of this ontology are identified by "*rdf:about*" tag while our algorithm makes use of the tag "*rdf:ID*" to identify local entities and makes use of the tag "*rdf:about*" only when identifying external entities.

3 General comments

3.1 Comments on the results

Since the main goal of this work is to strengthen the precision of the ontology mapping with developing an approach that deals with uncertainty inherent to the mapping process, the means of the three metrics are encouraging (see appendix raw results).

3.2 Discussions on the way to improve the proposed system

The results obtained with our OWL-CM tool turned out to be good, especially as this proposed version of the system is yet an alpha one which is still subject to improvements. In our future work, we will tend to investigate different horizons that we classify into three categories:

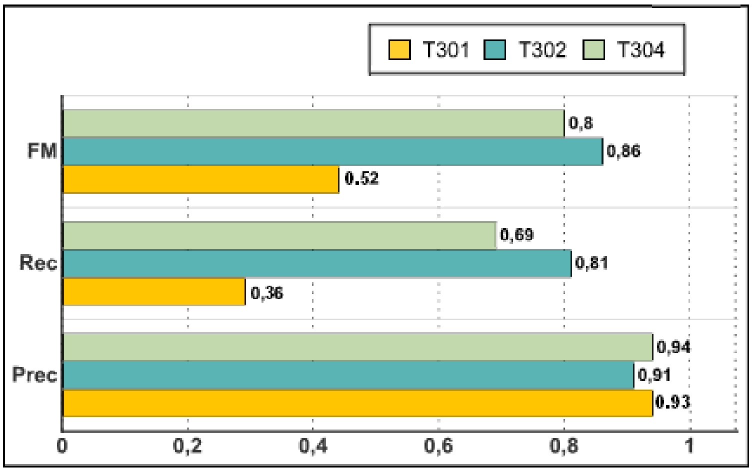


Fig. 8. Results of Tests 301-304.

1. **OWL-CM Improvements**: OWL-CM can be enhanced by different additive elements that were been revealed during the experimental study, i.e. when a full label does not have synonyms, search synonyms based on parts of the label.
2. **OWL-CM Tool Efficiency**: At this time we have exclusively worry about improving the effectiveness of the approach and left the efficiency to be investigated further.
3. **OWL-CM Tool Extensions**: OWL-CM can be extended so that it becomes, among others, able to map ontologies that differ in their language. We can use for example a translation tool.

3.3 Comments on the OAEI test cases

Concerning the benchmark, it is satisfactory since it was served as an experiment bed to assess both strong and weak points of our algorithm and gives an idea of the prospects for improving the algorithm effectiveness. But, it doesn't present some tests to interpret the use of some similarity measures that are based on the explicit assertions such as the following one:
Explicit Equality: it checks whether a logical assertion already forces two entities to be equal. In an OWL ontology, this assertion is expressed by using the axiom “owl:sameAs”. We refer to these assertions as “equalTo”.

$$\mathrm{sim}_{expeql}(a,b) := \begin{cases} 1 & \exists \text{ assertion (a,“equalTo”,b)}, \\ 0 & \text{otherwise.} \end{cases}$$

4 Conclusions

Semantic ontology mapping is an immensely rich area of research. Recently, researchers have brought attention to the fact that the mapping process can be modelled as decision-making under uncertainty. So we have intended to apply handling uncertainty in ontology mapping. We have proposed a new framework called OWL-CM, which sets the foundations for the architecture of discovering mappings under uncertainty. We have designed an algorithm for ontology mapping, based on the guidelines already established, and implemented it. The results obtained with our algorithm turned out to be good. From the experimental study, different horizons have been revealed and can be investigated in our future work.

References

1. Besana P. (2006a), *A framework for combining ontology and schema matchers with Dempster-Shafer*, In Proceedings of the International Workshop on Ontology Matching (OM-2006), Athens, Georgia, USA, pages 196-200.
2. Besana P. (2006b), *Dynamic ontology mapping: a first prototype*, SSP Talks, School of Informatics, University of Edinburgh.
3. Bisson G. (1995), *Why and How to Define a Similarity Measure for Object Based Representation Systems*, In Towards Very Large Knowledge Bases, IOS Press, Amsterdam, pages 236-246.
4. Castano, S., Ferrara, A., and Messa, G. (2006), *Results of the HMatch Ontology Matchmaker in OAEI 2006*, In Proceedings of the International Workshop on Ontology Matching (OM-2006), Athens, Georgia, USA, pages 134-143.
5. Convent. (1986), *Unsolvable problems related to the view integration approach*, In Pro- ceedings of ICDT'86, volume 243 of LNCS, Rome, Italy, Springer, pages 141-156.
6. Do, H., Melnik, S., and Rahm, E. (2002), *Comparison of schema matching evaluations*, In Web, Web-Services, and Database Systems, Springer LNCS 2593, pages 221–237.
7. Ehrig, M. and Staab, S. (2004), *Qom - quick ontology mapping*, The Semantic Web Proceedings ISWC04, Springer-Verlag LNCS 3298, pages 289-303.
8. Hu, W., Cheng, G., Zheng, D., Zhong, X., and Qu, Y. (2006), *The Results of Falcon-AO in the OAEI 2006 Campaign*, In Proceedings of the International Workshop on Ontology Matching (OM-2006), Athens, Georgia, USA, pages 124-133.
9. Nagy, M., Vargas-Vera, M., and Motta, E. (2006), *Dssim-ontology mapping with uncertainty*, In Proceedings of the International Workshop on Ontology Matching (OM-2006), Athens, Georgia, USA, pages 115-123.
10. Niedbala, S.(2006), *OWL-CtxMatch*, In Proceedings of the International Workshop on Ontology Matching (OM-2006), Athens, Georgia, USA, pages 165-172.
11. Shafer G. (1976), *A mathematical theory of evidence*, Princeton University Press.
12. Smets P. (1988), *Belief functions*, In non-standard logics for automated reasoning, Academic Press, pages 253-286.

Appendix: Raw results

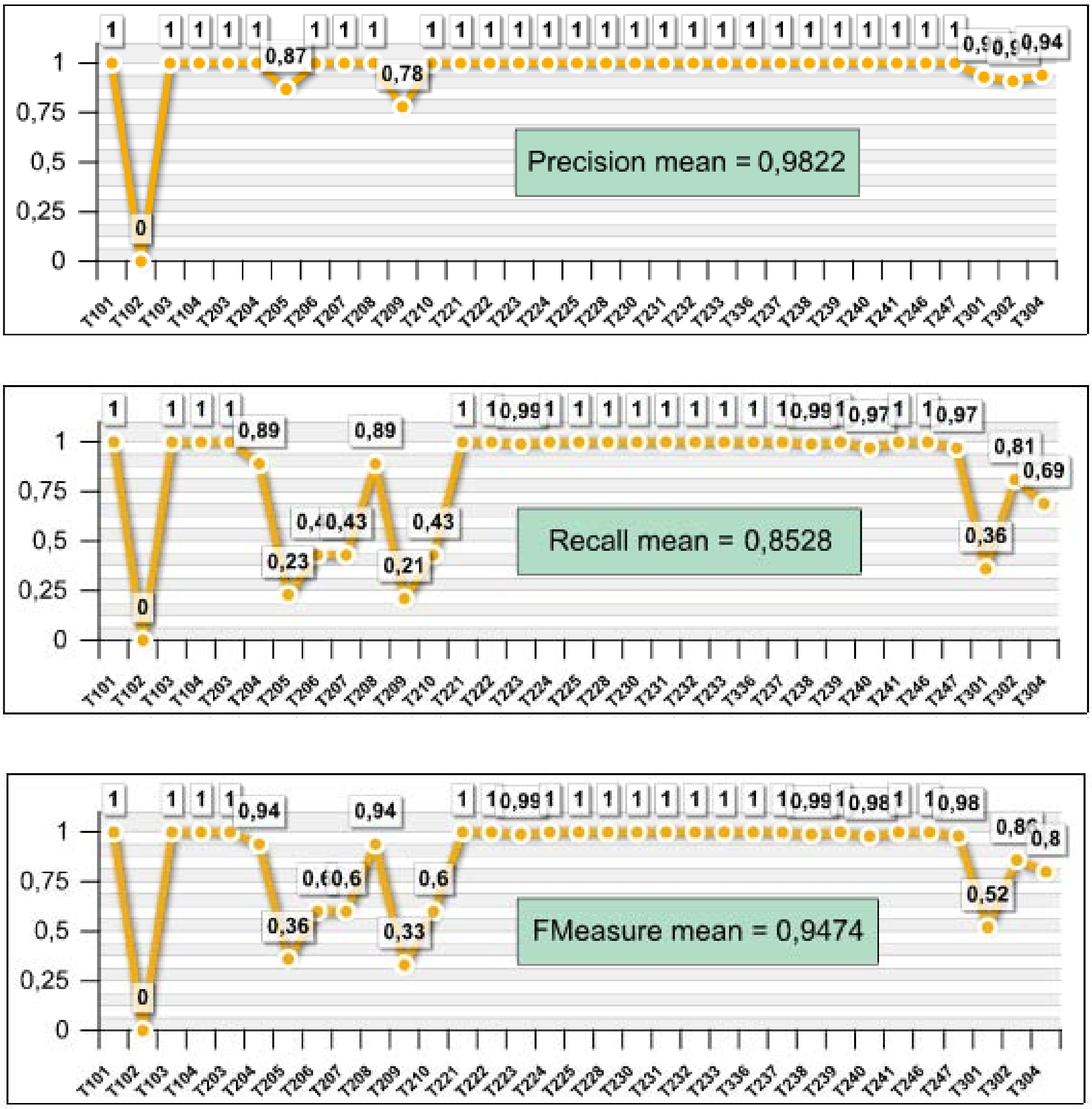


Fig. 9. Overview of Tests' Results.

The PRIOR+: Results for OAEI Campaign 2007

Ming Mao, Yefei Peng

University of Pittsburgh, Pittsburgh, PA, USA
{mingmao,ypeng}@mail.sis.pitt.edu

Abstract. Ontology mapping is to find semantic correspondences between similar elements of different ontologies. It is critical to achieve semantic interoperability in the WWW. This paper summarizes the results of the PRIOR+ participating at OAEI campaign 2007. The PRIOR+ is a generic and automatic ontology mapping tool, based on propagation theory, information retrieval technique and artificial intelligence model. The approach utilizes both linguistic and structural information of ontologies, and measures the profile similarity of different elements of ontologies in a vector space model (VSM). Furthermore, the PRIOR+ adaptively aggregate different similarities according to the harmony of similarity matrix. Finally the PRIOR+ deals with ontology constraints using interactive activation and competitive neural network. The preliminary results of benchmark task are presented, followed by a discussion. Some future works are given at the end.

1 Presentation of the system

1.1 State, purpose, general statement

The World Wide Web (WWW) now is widely used as a universal medium for information exchange. Semantic interoperability among different information systems in the WWW is limited due to information heterogeneity, and the non semantic nature of HTML and URLs. Ontologies have been suggested as a way to solve the problem of information heterogeneity by providing formal and explicit definitions of data. They may also allow for reasoning over related concepts. Given that no universal ontology exists for the WWW, work has focused on finding semantic correspondences between similar elements of different ontologies, i.e., *ontology mapping*. Automatic ontology mapping is important to various practical applications such as the emerging Semantic Web [3], information transformation and data integration [2], query processing across disparate sources [7], and many others [4].

Ontology mapping can be done either by hand or using automated tools. Manual mapping becomes impractical as the size and complexity of ontologies increases. Fully or semi-automated mapping approaches have been examined by several research studies, e.g., analyzing linguistic information of elements in ontologies [15], treating ontologies as structural graphs [12], applying heuristic rules to look for specific mapping patterns [8] and machine learning techniques [1]. More comprehensive surveys of ontology mapping approaches can be found in [9][14].

This paper proposes a new generic and scalable ontology mapping approach, the PRIOR+ approach. The architecture of the PRIOR+ is shown in **Fig. 1**. The PRIOR+ takes advantage of propagation theory, information retrieval technique and artificial intelligence model to solve ontology mapping problem. It utilizes both linguistic and structural information of ontologies, and measures the profile similarity of different elements of ontologies in a vector space model (VSM). Finally, the PRIOR+ adaptively aggregates different similarities according to the harmony of the matrix and deals with ontology constraints using interactive activation network.

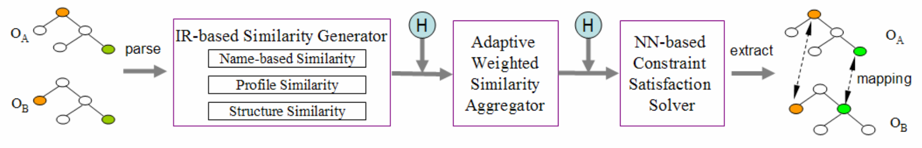


Fig. 1. The architecture of the PRIOR+ approach

1.2 Specific techniques used

The PRIOR+ is extended from the PRIOR [10][11]. In addition to the profile similarity and the edit distance of elements' name used in the PRIOR, the PRIOR+ considers structure similarity as well and adaptively aggregate different similarities based on their harmony. Furthermore, the PRIOR+ has a brand new NN-based Constraint Satisfaction Solver.

1.2.1 Similarity Generation

The similarity generation model aims to generate the similarity of both linguistic and structural information of ontologies. The details of calculating profile similarity and the edit distance of elements' name have been presented in the PRIOR [10][11]. To calculate the structure similarity of two elements, various structural features are extracted, e.g. the number of its sub-elements, the number of its direct property, the depth of the element to the root etc. Afterwards, the difference between these structural features are calculated and normalized to represent its structure similarity. The outputs of the similarity generation model are three similarity matrixes. Each matrix denotes a kind of similarity of two ontologies.

1.2.2 Harmony Estimation

The heterogeneities of information result in differences between ontologies, either from a linguistic view or structural view. Therefore, given two ontologies, it is critical to estimate the difference between ontologies, and then to adjust mapping strategies according to the difference. Here we define a term called *harmony* to represent the similarity between ontologies. Three types harmony of ontologies, i.e. name harmony, profile harmony and structure harmony, are calculated based on the similarity matrixes output from similarity generation model.

Ideally, if two ontologies are very similar in either linguistic or structural view, two true should-be-mapped elements should own a similarity equal to 1 or larger than the similarity of all other cells standing in the same row and column of those two elements in the corresponding similarity matrix. Therefore, the harmony of ontologies can be defined using Equation 1, where h_k denotes different types of harmony (i.e., name harmony, profile harmony and structure harmony), E_{O_1} and E_{O_2} denote the number of elements in ontologies, O_1 and O_2, $CMAX_{M_k}$ denotes the number of cells that own the highest similarity in its corresponding row/column in similarity matrix M_k.

$$h_k = \frac{\#CMAX_{M_k}}{\min(\#E_{o_1}, \#E_{o_2})} \qquad (1)$$

The different harmony of ontologies are used as weights to adaptively aggregate name similarity, profile similarity and structure similarity output from similarity generation model. Finally, the harmony of the aggregated similarity is estimated using the same way. The final harmony, h_f, will decide the necessity of NN-based Constraint Satisfaction Solver. If $h_f > c$ (c is an experience number), the cells having largest similarity in each row/column will be output to NN-based Constraint Satisfaction Solver as refined hypotheses. Otherwise, all cells in the final similarity matrix will be output.

1.2.3 NN-Based Constraint Satisfaction Solver

Constraint satisfaction problem (CSP) [16] arises as an intriguing research problem in ontology mapping due to the characteristics of ontology itself and its representations. The hierarchical relations in RDFS, the axioms in OWL and the rules in SWRL result in different kinds of constraints. For example, "if concept A matches concept B, then the ancestor of A can not match the child of B in the taxonomy" and "two classes match if they have owl:sameAs or owl:equvalentClass relations". To improve the quality of ontology mapping, it is critical to find the best configuration that can satisfy such constraints as much as possible.

CSPs are typically solved by a form of search, e.g. backtracking, constraint propagation, and local search [16]. The interactive activation network is first proposed to solve CSPs in [13]. The network usually consists of a number of competitive nodes connected to each other. Each node represents a hypothesis. The connection between two nodes represents constraint between their hypotheses. Each connection is associated with a weight. For example, we have two hypotheses, H_A and H_B. If whenever H_A is true, H_B is usually true, then there is a positive connection from node *A* to node *B*. Oppositely if H_A provides evidence against H_B, then there is a negative connection from node *A* to node *B*. The importance of the constraint is proportional to the strength (i.e. *weight*) of the connection representing that constraint. The state of a node is determined locally by the nodes adjacent to it and the weights connecting to it. The state of the network is the collection of states of all nodes. Entirely local computation can lead the network to converge to a global optimal state.

In the context of ontology mapping, a node in an interactive activation network represents a hypothesis that element E_{1i} in ontology O_1 can be mapped to element E_{2j} in ontology O_2. The initial activation of the node is the similarity of (E_{1i}, E_{2j}) output from the adaptive similarity aggregation model. The activation of the node can be updated using the following simple rule, where a_i denotes the activation of *node i*, written as n_i, net_i denotes the net input of the node.

$$a_i(t+1) = \begin{cases} a_i(t) + net_i(1 - a_i(t)), & net_i > 0 \\ a_i(t) + net_i a_i(t), & net_i < 0 \end{cases} \qquad (2)$$

The net_i comes from three sources, i.e. its neighbors, its bias, and its external inputs, as defined in Equation 3, where w_{ij} denotes the connection weight between n_i and n_j, a_j denotes the activation of node n_j, $bias_i$ denotes the bias of n_i, ei_i denotes the external input of n_i, which is a function of the confidence of a mapping, *istr* and *estr* are constants that allow the relative contributions of the input from internal sources and external sources to be readily manipulated. Note that the connection matrix is symmetric and the nodes may not connect to themselves, i.e., $w_{ij}=w_{ji}$, $w_{ii}=0$.

$$net_i = istr \times (\sum_j w_{ij} a_j + bias_i) + estr \times (ei_i) \qquad (3)$$

Furthermore, the connections between nodes in the network represent constraints between hypotheses. For example, the constraint that “only 1-to-1 mapping is allowed” results in a negative connection between nodes (E_{1i}, E_{2j}) and (E_{1i}, E_{2k}), where $k \neq j$. Moreover, “two elements match if their children match”, results in a positive connection between nodes (E_{1i}, E_{2j}) and (E_{1k}, E_{2t}), where E_{1k} and E_{2t} are the children of E_{1i} and E_{2j} respectively. Finally, the complexity of the connections may be very large because of complex constraints.

1.3 Adaptations made for the evaluation

We didn’t make any specific adaptations for the tests in the OAEI campaign 2007. All the mappings output by the PRIOR+ are based on the same set of parameters.

1.4 Link to the system and parameters file

The PRIOR+ is available at: http://www.sis.pitt.edu/~mingmao/om07/.

1.5 Link to the set of provided alignments (in align format)

The result file can be downloaded from http://www.sis.pitt.edu/~mingmao/om07/priorplus.zip

2 Results

In this section we present the results of the PRIOR+ in OAEI campaign 2007. All tests are run on a stand-alone PC running Ubuntu 6.0.6 operating system. The PC has Intel Dual Core 1.8 Hz processor, 1.5G memory, 100GB Serial ATA hard disk and SUN JAVA VM 1.6.0.

2.1 Benchmark

The benchmark track is the only track that opens its ground truth for participants. According to the different characteristics of ontologies, most parameters of the PRIOR+ are tuned on it. The full result of all tests can be found in Appendix[1].

The results show that Test 101, 103 and 104 are perfect because all names, comments and instances of classes and properties are the same. Test 201-210 are very structurally similar as the reference ontology, therefore the structural harmony plays an important role in deciding the final similarity of the elements of ontologies. Test 221-247 have high linguistic similarity with reference ontology, and thus the PRIOR+ obtained good performance on it. Test 248-266 are both linguistic and structural different with reference ontology. Even with the usage of the structural information, the PRIOR+ has some improvement compared with the PRIOR. The recall of these tests is still a little bit low. The reason why the PRIOR+ did not work well in these tests is under investigation. The 301-304 are real world ontologies, which have more impact when evaluating the mapping approach. The PRIOR+ also gained good results in all these tests.

Meanwhile, test 202, 209, 210, 248-266 and real case 302 and 303 demonstrate the effectiveness of using the interactive activation network to solve constraint satisfaction problem in ontology mapping.

2.2 Other Tracks

The web directory, anatomy and food track are all blind tracks that means no ground truth is available for participants to analyze the performance of the proposed approach. Therefore, please refer to the final results published by OAEI for further information.

---

[1] The data presented is slightly different from what we submitted to the OAEI campaign 2007 after improving the PRIOR+ approach.

3 General comments

3.1 Discussions on the way to improve the proposed system

Parameter tuning is an important issue in the implementation of neural network in our future work. Another possible improvement is to integrate auxiliary information and Web information for ontology mapping. For example, auxiliary information such as WordNet can be used to process synonyms. The co-occurrence of two elements returned by search engines can contribute to identify their semantic relation.

3.2 Comments on the OAEI 2006 test cases

Currently most tests in the campaign are blind. It will be better for OAEI to provide a small part of ground truth in some tests, such as anatomy, for participants to explore machine learning techniques. Meanwhile, in web directory task, some loops existing in the test cases have been broken randomly in the implementation of the PRIOR+.

4 Conclusion

In this paper, we present the PRIOR+, a generic ontology mapping tool, and its results in OAEI campaign 2007. The PRIOR+ integrates propagation theory, information retrieval technique and the interactive activation network to solve ontology mapping problem. The preliminary result of the PRIOR+ in benchmarks tests is promising.

References

1. Doan, A., J. Madhaven, et al. (2003). "Learning to Match Ontologies on the Semantic Web." VLDB Journal **12**(4): 303-319.
2. Dou, D., D. McDermott, et al. (2005). "Ontology Translation on the Semantic Web." Journal on Data Semantics (JoDS) **II**: 35-57.
3. Ehrig, M. (2006). Ontology Alignment: Bridging the Semantic Gap (Semantic Web and Beyond). ISBN-038732805X. Springer. 2006.
4. Euzenat, J., Bach, T., et al. (2004). State of the art on ontology alignment, Knowledge web NoE.
5. Euzenat, J et al. (2006). Results of the Ontology Alignment Evaluation Initiative 2006. In Proceedings of ISWC 2006 Ontology Matching Workshop. Atlanta, GA.
6. Felzenszwalb, P. F. and Huttenlocher, D. P. (2006). Efficient belief propagation for early vision. International Journal of Computer Vision, Vol. 70, No. 1.
7. Gasevic, D. and M. Hatala (2005). "Ontology mappings to improve learning resource search." British Journal of Educational Technology.

8. Hovy, E. (1998). Combining and standardizing large-scale, practical ontologies for machine translation and other uses. In Proceedings of the 1st International Conference on Language Resources and Evaluation (LREC), Granada, Spain.
9. Kalfoglou, Y. and M. Schorlemmer (2003). "Ontology mapping: the state of the art." The Knowledge Engineering Review 18(1): 1-31.
10. Mao, M. and Peng, Y. (2006). PRIOR System: Results for OAEI 2006. In Proceedings of ISWC 2006 Ontology Matching Workshop. Atlanta, GA.
11. Mao, M., Peng, Y. and Spring, M. (2007) A Profile Propagation and Information Retrieval Based Ontology Mapping Approach, In Proceedings of SKG 2007.
12. Melnik, S., H. Garcia-Molina, et al. (2002). Similarity flooding: a versatile graph matching algorithm and its application to schema matching. Proc. 18th International Conference on Data Engineering (ICDE).
13. McClelland, J. L. and Rumelhart, D. E. (1988). Explorations in Parallel Distributed Processing: A Handbook of Models, Programs, and Exercises. The MIT Press.
14. Noy, N. (2004). "Semantic Integration: A Survey of Ontology-Based Approaches." SIGMOD Record **33**(4): 65-70.
15. Qu, Y., Hu, W., and Cheng, G. (2006). Constructing virtual documents for ontology matching. In Proceedings of the 15th International Conference on World Wide Web.
16. Tsang, E. (1993). Foundations of Constraint Satisfaction: Academic Press.

Appendix: Raw results

Matrix of results

algorithm		prior+	
Test #	Precision	Recall	F-Measure
101	1	1	1
103	1	1	1
104	1	1	1
201	1	1	1
202	0.9756	0.82	0.894
203	1	1	1
204	1	1	1
205	0.9688	0.96	0.964
206	1	0.99	0.995
207	1	0.99	0.995
208	1	0.96	0.979
209	0.8919	0.68	0.772
210	0.9634	0.81	0.883
221	1	0.98	0.99
222	1	0.96	0.978
223	1	1	1
224	1	1	1

225	1	1	1
228	1	1	1
230	0.9351	1	0.966
231	1	1	1
232	1	1	1
233	1	1	1
236	1	1	1
237	1	1	1
238	1	1	1
239	0.9667	1	0.983
240	0.9706	1	0.985
241	1	1	1
246	0.9667	1	0.983
247	0.9706	1	0.985
248	0.9143	0.66	0.767
249	1	0.84	0.91
250	0.8065	0.76	0.781
251	0.9531	0.66	0.777
252	0.8904	0.67	0.765
253	0.913	0.65	0.759
254	1	0.27	0.429
257	0.6774	0.64	0.656
258	0.9219	0.63	0.752
259	0.8904	0.67	0.765
260	0.7895	0.52	0.625
261	0.4333	0.39	0.413
262	1	0.27	0.429
265	0.7368	0.48	0.583
266	0.5	0.45	0.476
301	0.9259	0.82	0.87
302	0.9677	0.63	0.76
303	0.82	0.84	0.828
304	0.9136	0.97	0.943
H-mean	0.9577	0.87	0.912

Result of Ontology Alignment with RiMOM at OAEI'07

Yi Li, Qian Zhong, Juanzi Li, and Jie Tang
Department of Computer Science and Technology, Tsinghua University
{ly, zhongqian, ljz, tangjie}@keg.cs.tsinghua.edu.cn

Abstract. In this report, we give a brief explanation of how RiMOM obtains the ontology alignment results at OAEI'07 contest. RiMOM integrates different alignment strategies: edit-distance based strategy, vector-similarity based strategy, path-similarity based strategy, background-knowledge based strategy, and three similarity-propagation based strategies. Each strategy is defined based on one specific ontological-information. In this contest, we, in particular, study how the different strategies (or strategy combination) perform for different alignment tasks. We found that: 1) on the directory data set, the path-similarity based strategy seems to outperform the others and 2) on the anatomy and food data sets, the background-knowledge based strategy has several distinct advantages. This report presents our results based on the evaluation. We also share our thoughts on the experiment design, showing specific strengths and weaknesses of our approach.

1. PRESENTATION OF THE SYSTEM

Ontology alignment is the key point to reach interoperability over ontologies. In recent years, much research work has been conducted for finding the alignment of ontologies [1] [4].

We have studied different strategies for ontology alignment and implemented them in a tool called RiMOM [5]. Each strategy is defined based on one kind of ontological information. In total, there are more than seven strategies implemented in RiMOM, we investigate the difference between the strategies and study which strategy will obtain the best performance on a specific alignment task. This introduces a very interesting (also critical) research issue: how to find a best strategy or an optimal strategy combination given an alignment task, called strategy selection.

1.1 State, purpose, general statement

For simplifying the following description, we here define the notations used throughout the report.

Ontology: An ontology O is composed of concepts C, properties/relations R, instances I, and Axioms A^O. We here use capital letter to indicate a set and lowercase letter (e.g., $c \in C$) to indicate one element in the set. Sometimes, for further simplification, we use entity e to indicate either c or r.

Ontology alignment: given an alignment from ontology O_1 to O_2, we call ontology O_1 as source ontology and O_2 as target ontology. We call the process of finding the alignment from O_1 to O_2 as (Ontology) alignment discovery or alignment finding.

Challenges for automating ontology alignment include: 1) how to automatically find alignments of high quality; 2) how to find the alignments efficiently; 3) what is the difference between the various alignment strategies and which one should be used for a specific task; 4) how to deal with the alignment of large scale ontology; 5) how to ease parameterizing, as the accuracy of alignments may vary largely with different parameters; 6) how to make full use of the user interaction.

In this campaign, we focus on dealing with the problems of 1), 2), and 3) with our system RiMOM.

1.2 Specific techniques used

There are six major steps in a general alignment process of RiMOM:

1) Similarity factors estimation. Given two ontologies, it estimates two similarity factors, which respectively approximately represent the structure similarity and the label similarity of the two ontologies. The two factors are used in the next step of strategy selection.

2) Strategy selection. The basic idea of strategy selection is that if two ontologies have high label similarity factor, then RiMOM will rely more on linguistic based strategies; while if the two ontologies have high structure similarity factor, then we will employ similarity-propagation based strategies on them. See Section 1.2.1 for details. Strategy selection by the two factors is mainly used on the benchmark data set. For the directory, anatomy, and food data set, we chose the strategies manually.

3) Single strategy execution. We employ the selected strategies to find the alignment independently. Each strategy outputs an alignment result.

4) Alignment combination. It combines the alignment results obtained by the selected strategies. The combination is conducted by a linear-interpolation method.

5) Similarity propagation. If the two ontologies have high structure similarity factor, RiMOM employs a similarity propagation process to refine the found alignments and to find new alignments that cannot be found using other strategies.

6) Alignment refinement. It refines the alignment results from the previous steps. We defined several heuristic rules to remove the “unreliable” alignments.

1.2.1 Similarity factors estimation

Our preliminary experiments show that the multi-strategy based alignment does not always outperform its single-strategy counterpart. For a new, unseen mapping task, we propose to use two similarity factors to determine which strategy should be used.

Given two ontologies: source ontology O_1 and target ontology O_2, we calculate two approximate similarity factors: structure similarity factor and label similarity factor.

We define structure similarity factor as: $F\_SS = \frac{\#common\_concept}{\max(\#nonleaf\_c_1, \#nonleaf\_c_2)}$, where *#nonleaf_c*$_1$ indicates the number of concepts in O_1 that has sub concepts. Likewise for *#nonleaf_c*$_2$. *#common_concept* is calculated as follows: if concepts $c_1 \in O_1$ and $c_2 \in O_2$ have the same number of sub concepts and they are in the same depth from the concept “owl:Thing”, we add one to *#common_concept*. After enumerated all pair, we obtain the final score of *#common_concept*. Intuition of the factor is that the larger the structure similarity factor, the more similar the structures of the two ontologies are.

The label similarity factor is defined as: $F\_LS = \frac{\#same\_label}{\max(\#c_1, \#c_2)}$, where $\#c_1$ and $\#c_2$ respectively represent the number of concepts in O_1 and O_2. *#same_label* represents the number of pairs of concepts $\{(c_1, c_2)|c_1 \in O_1$ and $c_2 \in O_2\}$ that have the same label.

The two factors are defined simply and not used to accurately represent the real “similarities” of structures and labels. However, they can approximately indicate the characteristics of the two ontologies. Moreover, they can be calculated efficiently.

So far, we carried out the strategy selection by heuristic rules. For example, if the structure similarity factor *F_SS* is lower than 0.25, then RiMOM suppresses the CCP and PPP strategies. However, the CPP will always be used in the alignment process.

1.2.2 Multiple strategies

The strategies implemented in RiMOM include: edit-distance based strategy, vector-similarity based strategy, path-similarity based strategy, background-knowledge based strategy, and three similarity-propagation based strategies.

1. Edit-distance based strategy (ED)

Each label (such as concept name or property name) is composed of several tokens. In this strategy (ED), we calculate the edit distance between labels of two entities. Edit distance estimates the number of operations needed to convert one string into another. We define ($1-\#op/max\_length(l(e_1), l(e_2))$) as the similarity of two labels, where #op indicates the number of operations, $max\_length(l(e_1), l(e_2))$ represents the maximal length of the two labels.

2. Vector-similarity based strategy (VS)

We formalize the problem as that of document similarity. For an entity *e*, we regard its label, comment, and instances as a ‘document’ and calculate the similarity between an entity pair. Specifically, the ‘document’ is tokenized into words. Then we remove the stop words and employ stemming on the words and view the remains as features to generate a feature vector. We also add some other general features which prove to be very helpful. For a concept, the features include: the number of its sub concepts, the number of properties it has, and the depth of the concept from “OWL:Thing”. Next, we compute the cosine similarity between two feature vectors. The advantage of this strategy is that it can easily incorporate different information (even structural information) into the feature vector.

3. Path-similarity based strategy (PS)

We define path as the aggregation of the entity labels from “OWL:Thing” to the current entity. A path-similarity measure between two entities e_1 and e_2 is defined as:

$$sim_p(e_1, e_2) = \max(sim(l(e_1), PL(e_2)), sim(PL(e_1), l(e_2)))$$

where $PL(e_2)$ is the path of e_2. $sim(l(e_1), PL(e_2))$ is the similarity between the label of entity e_1 and the path of entity e_2. It is estimated by averaging similarities between the label of e_1 and each label in the path of e_2.

4. Background-knowledge based strategy (BK)

We also try to make use of background knowledge to enhance the performance of alignment. The idea is straightforward. In some alignment tasks, for example the food alignment task and the anatomy alignment task, the available information is limited (only concept labels are available). We utilize the available knowledge base (we used wiki pages) to help find the alignment. For each entity, we first look up in the knowledge base for its definition, and then use the description of its definition in the similarity calculation of the vector-similarity based strategy.

5. Strategy combination

For some alignment task, we need use more than one strategy to find the alignment. The strategies are employed first independently and then are combined together. A combination measure is thus defined as:

$$Map\left(e_1,e_2\right)=\frac{\sum_{k=1\ldots n} w_k \sigma\left(Map_k\left(e_1,e_2\right)\right)}{\sum_{k=1\ldots n} w_k}$$

where $e_1 \in O_1$ and $e_2 \in O_2$; $Map_k(e_1,e_2)$ is the alignment score obtained by strategy k. w_k is the weight of strategy k. σ is a sigmoid function, which is defined as $\sigma(x)=1/\left(1+e^{-5(x-\alpha)}\right)$, where α is tentatively set as 0.5.

This "independence-and-combination" fashion has the advantage of easy integrating new strategies into the alignment process.

6. Similarity-propagation based strategies

The structure information in ontologies is useful for finding the alignments especially when two ontologies share the common/similar structure. According to the propagation theory [2], we define three structure based strategies in RiMOM, namely concept-to-concept propagation strategy (CCP), property-to-property propagation strategy (PPP), and concept-to-property propagation strategy (CPP).

Intuition of the propagation based method is that if two entities are aligned, their super-concepts have higher probability to be aligned. The basic idea here is to propagate the similarity of two entities to entity pairs that have relations (e.g., subClassOf, superClassOf, siblingClassOf, subPropertyOf, superPropertyOf, range, and domain) with them. The idea is inspired by similarity flooding [3]. We extended the algorithm and adaptively used them in the three structure based strategies.

In CCP, we propagate similarities of concepts pair across the concept hierarchical structure. In PPP, we propagate similarities of property pair across the property hierarchy. In CPP, we propagate similarities of concepts pair to their corresponding property pair, and vice versa. Details of the method will be reported elsewhere.

The similarity-propagation based strategies are performed after the other strategies defined above. They can be used to adjust the alignments and find new alignments.

1.3 Adaptations made for the evaluation

Some parameters were tuned and set in the experiments. For example, for strategies combination (cf. equation 1), we set the weight of ED as 0.5 and that of VS as 1. For strategy selection, we define 0.25 as the threshold to determine whether CCP and PPP will be suppressed or not. We also define 0.2 as threshold to determine whether ED will be suppressed or not. In addition, we employed background-knowledge based strategy for food and anatomy alignment, and path-similarity based strategy for directory.

1.4 Link to the system, parameters file, and provided alignments

Our system RiMOM (including the parameters file) can be found at http://keg.cs.tsinghua.edu.cn/project/RiMOM/. For details of the approach, see [5].

The alignment results of the campaign are available at http://keg.cs.tsinghua.edu.cn/project/RiMOM/OAEI2007/.

2 Results

RiMOM has been implemented in Java. We use OWL-API to parse the RDF and OWL files. The experiments were carried out on a Server running Windows 2003 with two Dual-Core Intel Xeon processors (2.8 GHz) and 3-gigabyte memory. All the alignments outputted by RiMOM are based on the same parameters.

2.1 Benchmark

There are in total 54 alignment tasks defined on the benchmark data set. The task is to find the alignment from every ontology to the reference ontology 101. We conducted alignment on the benchmark data set in the following steps: 1) we first employ the vector-similarity based strategy. We make use of the entity labels, comments, and instances to generate a feature vector and calculate the similarity between each entity pair; 2) we utilize the similarity-propagation based strategies to refined alignment results.

We also compute the similarity factors and use the similarity factor in Step 1) (for determining whether we add special features into the feature vector) and 2) (for determining whether a propagation based strategy should be used).

In these tasks, the average precision is 0.97 and the average recall is 0.99. The average time cost is about 4 second per task.

2.2 directory

The directory ontologies are organized as a taxonomy with sub-sumption hierarchies. We obtain the alignment results in the following ways: 1) edit-distance based strategy is used to calculate the similarity between entity labels; 2) path-similarity based strategy is employed to compute the similarity between two entity paths; 3) combination of the two similarities; 4) similarity propagation (CCP) is utilized on the hierarchical structure to refine the result; and 5) pruning some found alignments. The pruning is performed using heuristic rules.

2.3 anatomy

For anatomy ontology, we utilize the background-knowledge based strategy to find the alignment. Specifically, we perform the alignment finding in the following steps: 1) we constructed a background knowledge base by using the concept definitions from UMLS [6], in total we have a base of more than 100 K terms; 2) for each entity from the source ontology, we find if there is an entity with the identical label in the target ontology. If so, we alignment them; otherwise, we look up in the background knowledge base to find the definition description of the label; 3) we use the vector-similarity based strategy to calculate the similarity. We create the feature vector using the entity label and the concept definition (if found in the knowledge base).

2.4 food

We employ the same process as that in anatomy to find alignment on the food data set by using wiki as the background-knowledge.

3 General comments

3.1 Comments on the results

An objective and comprehensive comment on strengths or weakness requires the comparison with other participants, which are not available so far (will be available before the workshop). Here, we share some thoughts about the results.

Strengths

From experimental results, we see that RiMOM can achieve high performance when the ontologies to be aligned have similar linguistic information or similar structure information. Some concluding remarks are summarized as follows:

1) Linguistic information (including label of concepts and properties) is important and help to align most of the entities.

2) Structure information can be used to improve the alignments, in particular when linguistic information is missing.

3) Strategy selection is important. In different alignment tasks, the ontologies to be aligned have different characteristics, it would be particularly helpful to find the characteristics of the ontologies and apply correspondingly strategies on them. This also introduces an interesting research issue: how to perform the strategy selection efficiently? Currently, we use two factors to select the structure strategy and to determine whether we add several features into the vector when using vector-similarity based strategy. However, it is far from an ideal solution of the strategy selection.

4) Alignment refinement is helpful. We removed the unreliable alignments.

Weakness

1) Although the preliminary experiments show that our strategy selection method can enhance the alignment finding, it is not sufficient. There are many problems needed to be solved.

2) We note that parameter setting is very important. We have found that using different parameter settings, with the exactly same approach, the alignment results may differ largely. So far, we tuned the parameters manually. It is not adaptable in particular when the ontologies are very large, which means that tuning different parameters to find the best ones is not possible.

3.2 Discussions on the way to improve the proposed system

Possible improvements are corresponded to the related weaknesses in the previous section.

1) New strategy selection by considering all the strategies and all the factors should be proposed.

2) Our thinking is to use a supervised machine learning method to find the optimal parameters based on some training data sets.

3.3 Comments on the OAEI 2007 test cases

The benchmark tests indicate very interesting general results on how the alignment approach behaves. These tests are really useful, as a good underlying test base, for evaluating and improving the alignment algorithm and system.

For future work, it might be interesting to add some tests to evaluate the cross-linguistic alignment, as for English ontology to Chinese ontology, an issue is important in practical application.

4 Conclusion

In this report, we have briefly introduced how we employed RiMOM to obtain the alignment results in OAEI'07 contest. We have presented the alignment process of RiMOM and explained the strategy defined in RiMOM. We have also described how we performed the alignment for different alignment tasks. We summarized the strengths and the weaknesses of our proposed approach and gave possible improvement for the system in the future work.

Acknolwegement

The work is supported by the National Natural Science Foundation of China under Grant No. 90604025 and No. 60703059. It is also supported by IBM Innovation funding.

References

[1] J. Euzenat. State of the art on ontology alignment. http://www.inrialpes.fr/exmo/cooperation/kweb/heterogeneity/deli/. August, 2004.

[2] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. International Journal of Computer Vision, Vol. 70, No. 1, October 2006.

[3] S. Melnik, H. Garcia-Molina and E. Rahm: Similarity Flooding: a versatile graph matching algorithm and its application to schema matching. In Proc. of 18th ICDE. San Jose CA, Feb 2002. pp. 117-128

[4] E. Rahm and P. A. Bernstein. A survey of approaches to automatic schema matching. The VLDB Journal, 2001, 10:334-350.

[5] J. Tang, J. Li, B. Liang, X. Huang, Y. Li, and K. Wang. Using Bayesian Decision for Ontology Alignment. Journal of Web Semantics, Vol(4) 4, pp. 243-262, 2006.

[6] http://umlsks.nlm.nih.gov/kss/

Appendix: Raw results

The following results were obtained in the evaluation runs.

Matrix of results

#	Name	Prec.	Rec.
101	Reference alignment	1.00	1.00
102	Irrelevat ontology	N/A	N/A
103	Language generalization	1.00	1.00
104	Language restriction	1.00	1.00
201	No names	1.00	1.00
202	No names, no comments	1.00	0.80
203	No comments	1.00	0.88
204	Naming conventions	1.00	1.00
205	Synonyms	1.00	0.99
206	Translation	1.00	0.99
207		1.00	0.99
208		0.98	0.86
209		1.00	0.84
210		0.99	0.85
221	No specialisation	1.00	1.00
222	Flatenned hierachy	1.00	1.00
223	Expanded hierarchy	1.00	1.00
224	No instance	1.00	0.99
225	No restrictions	1.00	1.00
228	No properties	1.00	1.00
230	Flatenned classes	0.94	1.00
231		1.00	1.00
232		1.00	0.99
233		1.00	1.00
236		1.00	1.00
237		1.00	0.99

238		1.00	0.99
239		1.00	1.00
240		1.00	1.00
241		1.00	1.00
246		1.00	1.00
247		1.00	1.00
248		0.99	0.78
249		1.00	0.79
250		1.00	0.55
251		0.76	0.58
252		0.85	0.70
253		0.99	0.77
254		1.00	0.27
257		1.00	0.55
258		0.76	0.57
259		0.85	0.69
260		0.93	0.45
261		1.00	0.27
262		1.00	0.27
265		0.93	0.45
266		1.00	0.27
301	BibTeX/MIT	0.75	0.67
302	BibTeX/UMBC	0.72	0.65
303	Karlsruhe	0.45	0.86
304	INRIA	0.90	0.97

SAMBO Results for the Ontology Alignment Evaluation Initiative 2007

He Tan and Patrick Lambrix

Department of Computer and Information Science
Linköpings universitet
581 83 Linköping, Sweden

Abstract. This article describes a system for ontology alignment, SAMBO, and presents its results for the benchmark and anatomy tasks in the 2007 Ontology Alignment Evaluation Initiative. For the benchmark task we have used a strategy based on string matching as well as the use of a thesaurus, and obtained good results in many cases. For the anatomy task we have used a combination of string matching and the use of domain knowledge. This combination performed well in former evaluations using other anatomy ontologies.

1 Introduction

Many ontologies have already been developed and many of these ontologies contain overlapping information. Often we would want to be able to use multiple ontologies. For instance, companies may want to use community standard ontologies and use them together with company-specific ontologies. Applications may need to use ontologies from different areas or from different views on one area. Ontology builders may want to use already existing ontologies as the basis for the creation of new ontologies by extending the existing ontologies or by combining knowledge from different smaller ontologies. Further, different data sources in the same domain may have annotated their data with different but similar ontologies. In each of these cases it is important to know the relationships between the terms in the different ontologies. It has been realized that this is a major issue and some organizations have started to deal with it. For instance, regarding anatomy ontologies there is the CARO (http://www.bioontology.org/wiki/-index.php/CARO:Main_Page) effort and earlier the SOFG effort (http://www.sofg.org/).

To deal with this issue we developed and continue developing SAMBO, System for Aligning and Merging Biomedical Ontologies. We use the term 'alignment' for defining the relationships between terms in different ontologies. We use the term 'merging' when we, based on the alignment relationships between ontologies, create a new ontology containing the knowledge included in the source ontologies. In the remainder of the paper we only discuss the alignment component of SAMBO.[1] In section 2 we describe the purpose, the framework on which SAMBO is based, the techniques used, and the adaptations made for OAEI 2007. Section 3 describes the test runs and general comments are given in section 4. The paper concludes in section 5.

---

[1] SAMBO also merges two source ontologies in OWL syntax with given alignment relationships using a reasoner.

2 Presentation of the system

2.1 State, purpose, general statement

Although several of our methods and techniques are general and applicable to different areas, when developing SAMBO, we have focused on biomedical ontologies. Research in biomedical ontologies is recognized as essential in some of the grand challenges of genomics research [2]. Further, there exist de facto standard ontologies such as GO, and much support is being provided to the community to develop and publish ontologies in the biomedical domain in a principled way through, for instance, the OBO Foundry initiative (http://www.obofoundry.org/). There are also many overlapping ontologies available in the field, many of which are available through OBO. The field has also matured enough to start tackling the problem of overlap in the ontologies and standardization efforts such as SOFG and CARO have started.

Ontologies may contain concepts, relations, instances and axioms. Most biomedical ontologies are controlled vocabularies, taxonomies, or thesauri. This means that they may contain concepts, is-a and part-of relations, and sometimes a limited number of other relationships. Therefore, we have focused on methods that are based on these ontology components. For some approaches we have also used documents about a concept as instances for that concept. We have not dealt with axioms.

2.2 Framework

SAMBO is based on the framework shown in figure 1 [5]. The framework consists of two parts. The first part (*I* in figure 1) computes alignment suggestions. The second part (*II*) interacts with the user to decide on the final alignments. An alignment algorithm receives as input two source ontologies. The algorithm includes one or several matchers, which calculate similarity values between the terms from the different source ontologies. The matchers may use knowledge from different sources. Alignment suggestions are then determined by combining and filtering the results generated by one or more matchers. By using different matchers and combining and filtering the results in different ways we obtain different alignment strategies. The suggestions are then presented to the user who accepts or rejects them. The acceptance and rejection of a suggestion may influence further suggestions. Further, a conflict checker is used to avoid conflicts introduced by the alignment relationships. The output of the alignment algorithm is a set of alignment relationships between terms from the source ontologies.

2.3 Specific techniques used

In this section we describe the matchers, and combination and filtering techniques that are available in SAMBO. These matchers and techniques were previously evaluated using test cases for aligning Gene Ontology and Signal Ontology, and for aligning Medical Subject Headings (MeSH) and the Anatomical Dictionary for the Adult Mouse (MA) [5] using the KitAMO evaluation environment [7].[2] In addition to these techniques we

---

[2] An introduction to SAMBO and KitAMO can be found in [6].

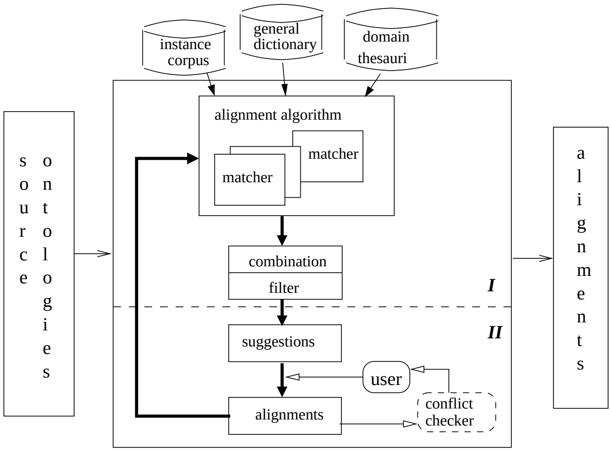


Fig. 1. Alignment framework [5].

have also experimented with other matchers [9, 11] and another filtering technique [1], some of which may be added to SAMBO in the future. We are also working on methods for recommendation of alignment strategies [10] which we intend to integrate into SAMBO in the future.

Matchers SAMBO contains currently five basic matchers: two terminological matchers, a structure-based matcher, a matcher based on domain knowledge, and a learning matcher.

Terminological matchers. The basic terminological matcher, *Term* contains matching algorithms based on the textual descriptions (names and synonyms) of concepts and relations. In the current implementation, the matcher includes two approximate string matching algorithms, n-gram and edit distance, and a linguistic algorithm. An n-gram is a set of n consecutive characters extracted from a string. Similar strings will have a high proportion of n-grams in common. Edit distance is defined as the number of deletions, insertions, or substitutions required to transform one string into the other. The greater the edit distance, the more different the strings are. The linguistic algorithm computes the similarity of the terms by comparing the lists of words of which the terms are composed. Similar terms have a high proportion of words in common in the lists. A Porter stemming algorithm is employed to each word. These algorithms were evaluated in [4] using MeSH anatomy (ca 1400 terms) and MA (ca 2350 terms). Term computes similarity values by combining the results from these three algorithms using a weighted sum. The combination we use in our experiments (weights 0.37, 0.37 and 0.26 for the linguistic algorithm, edit distance and n-gram, respectively) outperformed the individual individual algorithms in our former evaluations [4]. Further, the matcher *TermWN* is based on Term, but uses a general thesaurus, WordNet (http://wordnet.princeton.edu/), to enhance the similarity measure by looking up the hypernym relationships of the pairs of words in WordNet.

Structural matcher. The structural matcher is an iterative algorithm based on the is-a and part-of hierarchies of the ontologies. The algorithm requires as input a list of alignment relationships and similarity values and can therefore not be used in isolation. The intuition behind the algorithm is that if two concepts lie in similar positions with respect to is-a or part-of hierarchies relative to already aligned concepts in the two ontologies, then they are likely to be similar as well. For each pair of concepts (C_1,C_2) in the original list of alignment relationships the structural matcher augments the original similarity value for pairs of concepts (C'_1,C'_2) such that C'_1 and C'_2 are equivalent to, are in an is-a relationship with, or participate in a part-of relationship with C_1 and C_2, respectively. The augmentation depends on the relationship and on the distance between the concepts in the is-a and part-of hierarchies. The augmentation diminishes with respect to distance. The new similarity value can also not exceed 1. In our earlier experiments we used a maximal distance of 2 and the effect on ancestors is lower than the effect on descendants.

Use of domain knowledge. Another strategy is to use domain knowledge. Our matcher *UMLSKSearch* uses the Metathesaurus in the Unified Medical Language System (UMLS, http://www.nlm.nih.gov/research/umls/). The similarity of two terms in the source ontologies is determined by their relationship in UMLS. In our experiments we used the UMLS Knowledge Source Server to query the UMLS Metathesaurus with source ontology terms. The querying is based on searching the normalized string index and normalized word index provided by the UMLS Knowledge Source Server. We used version 2007AB of UMLS. As a result we obtain concepts that have the source ontology term as their synonym. We assign a similarity value of 1 if the source ontology terms are synonyms of the same concept and 0 otherwise.[3]

Learning matcher. The matcher makes use of life science literature that is related to the concepts in the ontologies. It is based on the intuition that a similarity measure between concepts in different ontologies can be defined based on the probability that documents about one concept are also about the other concept and vice versa. The strategy contains the following basic steps. (i) For each ontology that we want to align we generate a corpus of PubMed abstracts. In our implementation we generated a corpus of maximally 100 PubMed abstracts per concept using the programming utilities provided by the retrieval system Entrez (http://www.ncbi.nlm.nih.gov/Entrez/). (ii) For each ontology a document classifier is generated. This classifier returns for a given document the concept that is most closely related to the document. To generate a classifier the corpus of abstracts associated to the classifier's ontology is used. In our algorithm we use a naive Bayes classification algorithm (based on the code available at http://www.cs.utexas.edu/users/mooney/ir-course/). (iii) Documents of one ontology are classified by the document classifier of the other ontology and visa versa. (iv) A similarity measure between concepts in the different ontologies is computed by using the results of step (iii). The similarity is computed as

$$lsim(C_1, C_2) = \frac{n_{NBC2}(C_1, C_2) + n_{NBC1}(C_2, C_1)}{n_D(C_1) + n_D(C_2)}$$

---

[3] Observe that this is slightly different from the version reported in [5] where we used version 2005AA of UMLS and we assigned a similarity value of 1 for two terms with the exact same names, 0.6 if the source ontology terms are synonyms of the same concept, and 0 otherwise.

where $n_D(C)$ is the number of abstracts originally associated with C, and $n_{NBCx}(C_p, C_q)$ is the number of abstracts associated with C_p that are also related to C_q as found by classifier $NBCx$ related to ontology x. More details about this algorithm as well as some extensions can be found in [9].

Combinations The user is given the choice to employ one or several matchers during the alignment process. The similarity values for pairs of concepts can then be determined based on the similarity values computed by one matcher, or as a weighted sum of the similarity values computed by different matchers.

Filtering The current filtering method is threshold filtering. Pairs of concepts with a similarity value higher than or equal to a given threshold value are returned as alignment suggestions to the user.

2.4 Adaptations made for the evaluation

SAMBO is an interactive alignment system. The alignment suggestions calculated by SAMBO are normally presented to the user who accepts or rejects them. Alignment suggestions with the same concept as first item in the pair are shown together to the user. Therefore, SAMBO shows the user the different alternatives for aligning a concept. This is a useful feature, in particular when the system computes similarity values which are close to each other and there is no or only a small preference for one of the suggestions. Further, the acceptance and rejection of a suggestion may influence which suggestions are further shown to the user.

The computation of the alignment suggestions in SAMBO is based on the computation of a similarity value between the concepts. The computation of the similarity values does not take into account what the relationship of the alignment should be. However, when an alignment is accepted, the user can choose whether the alignment relationship should be an equivalence relation or an is-a relation.

As the OAEI evaluation only considers the non-interactive part of the system and the computation of the similarity values does not take the relationship into account, we had to modify the computation of the suggestions. It would not make sense to have alignment suggestions where a concept appears more than once as the user would not be able to make a choice. Therefore, we decided to filter SAMBO's alignment suggestion list such that only suggestions are retained where the similarity between the concepts in the alignment suggestion is higher than or equal to the similarity of these concepts to any other concept according to the alignment suggestion list. (In the case there are different possibilities, one is randomly chosen.)

2.5 Link to the system and parameters file

The SAMBO project page is at http://www.ida.liu.se/∼iislab/projects/SAMBO/.

2.6 Link to the set of provided alignments (in align format)

The suggested alignments are available at
http://www.ida.liu.se/∼iislab/projects/SAMBO/OAEI/2007/.

3 Results

We have provided alignment suggestions for the tasks ’benchmark’ and ’anatomy’. Tests were performed on a PC (Pentium(R) D CPU 2.80GHz 2.79GHz, RAM 0.99GB, Windows XP).

3.1 benchmark

The results for the benchmark task were obtained by using TermWN with threshold 0.6. As a preprocessing step we split names based on capital letters occurring within a name. For instance, ’InCollection’ was split into ’In Collection’. We did not use the comment field. The results may be improved using also this field.

We assume that ontology builders use a reasonable naming scheme and thus we did not tackle the cases where labels were replaced by a random one. Therefore, the recall for tests 201-202, 248-254, 257-262, 265-266 is low. For these cases we may use other kind of information in the ontology such as the comment field or the structure. We also did not focus on different natural languages (206-207, 210) or subsumption relationships (302).

Regarding the other cases we received high precision and recall except for cases 205 and 209. For 205 and 209 we had expected that using WordNet would be an advantage. Therefore, we compared the results with a run using Term (without WordNet). The differences between the results for Term and TermWN were small for all cases, including cases 205 and 209.

3.2 anatomy

The results for the anatomy task were obtained by first running UMLSKSearch and suggesting the pairs with similarity value 1 and then running Term with threshold 0.6 on the remainder of the pairs. With respect to the computation of the suggestions, this would be similar to having a matcher that returns as similarity value for a pair the maximum of the similarity value for the pair according to UMLSKSearch and the similarity value for the pair according to Term, and then using 0.6 as threshold.

4 General comments

A problem that users face is that often it is not clear how to get the best alignment results given that there are many strategies to choose from. In most systems, including SAMBO) there usually is no strategy for choosing the matchers, combinations and filters in an optimal way. Therefore, we used our experience from previous evaluations [5] to decide which matchers to use for which task. The lack of an optimization strategy is also the reason why we did not provide results for the second and third test for anatomy (optimization with respect to precision and recall, respectively). The results for precision and recall for SAMBO may be influenced by the filtering phase. Intuitively, higher thresholds lead to higher precision and lower recall, while lower thresholds usually lead to higher recall, but lower precision. However, for SAMBO as stand-alone system, there is no strategy for how to choose the threshold for optimizing precision or recall. In the future, however, this may be possible using recommendation methods for alignment strategies such as proposed in [10] that will be able to recommend matchers, combinations and filters based on the alignment task and evaluation methods.

The OAEI deals with the non-interactive part of the alignment systems. This allows for evaluating how good the alignment suggestions are. However, for some systems, such as SAMBO, the list of alignment suggestions is only an initial list and is updated after each acceptance or rejection of a suggestion.

5 Conclusion

We have briefly described our ontology alignment system SAMBO and some results of running SAMBO on the alignment tasks of OAEI.

For the benchmark task we have used TermWN and obtained good results in many cases. We expect that the results will still improve when we use more information available in the ontology, such as the comment field and the structure. Therefore, we will continue this task using Term and TermWN also on the comment field, as well as using our structural matcher. Further, in earlier tests, also our advanced filtering technique described in [1] usually improves the results of Term and TermWN.

Regarding the anatomy task we have used a combination of UMLSKSearch and Term, which performed best in former evaluations using other anatomy ontologies. We are currently also evaluating instance-based matchers.

A major problem is deciding which algorithms should be used for a given alignment task. This is a problem that users face, and that we have also faced in the evaluation. We expect that recommendation strategies [10, 8, 3] will alleviate this problem.

References

1. B Chen, H Tan, and P Lambrix. Structure-based filtering for ontology alignment. In *Proceedings of the IEEE WETICE Workshop on Semantic Technologies in Collaborative Applications*, pages 364–369, 2006.
2. F Collins, E Green, A Guttmacher, and M Guyer. A vision for the future of genomics research. *Nature*, 422:835–847, 2003.
3. M Ehrig, S Staab, and Y Sure. Bootstrapping ontology alignment methods with apfel. In *Proceedings of the International Semantic Web Conference*, pages 186–200, 2005.
4. P Lambrix and H Tan. Merging daml+oil ontologies. In J Barzdins and A Caplinskas, editors, *Databases and Information Systems - Selected Papers from the Sixth International Baltic Conference on Databases and Information Systems*, pages 249–258. IOS Press, 2005.
5. P Lambrix and H Tan. Sambo - a system for aligning and merging biomedical ontologies. *Journal of Web Semantics, Special issue on Semantic Web for the Life Sciences*, 4(3):196–206, 2006.
6. P Lambrix and H Tan. Ontology alignment and merging. In Burger, Davidson, and Baldock, editors, *Anatomy Ontologies for Bioinformatics: Principles and Practice*. Springer, 2007.
7. P Lambrix and H Tan. A tool for evaluating ontology alignment strategies. *Journal on Data Semantics, LNCS 4380*, VIII:182–202, 2007.
8. M Mochol, A Jentzsch, and J Euzenat. Applying an analytic method for matching approach selection. In *Proceedings of the Workshop on Ontology Matching*, 2006.

9. H Tan, V Jakonienė, P Lambrix, J Aberg, and N Shahmehri. Alignment of biomedical ontologies using life science literature. In *Proceedings of the International Workshop on Knowledge Discovery in Life Science Literature, LNBI 3886*, pages 1–17, 2006.
10. H Tan and P Lambrix. A method for recommending ontology alignment strategies. In *Proceedings of the 6th International Semantic Web Conference*, 2007.
11. T Wächter, H Tan, A Wobst, P Lambrix, and M Schroeder. A corpus-driven approach for design, evolution and alignment of ontologies. In *Proceedings of the Winter Simulation Conference*, pages 1595–1602, 2006. Invited contribution.

SEMA: Results for the Ontology Alignment Contest OAEI 2007[1]

Vassilis Spiliopoulos[1, 2], Alexandros G. Valarakos[1], George A. Vouros[1], and Vangelis Karkaletsis[2]

[1] AI Lab, Information and Communication Systems Engineering Department, University of the Aegean, Samos, 83 200, Greece
`{vspiliop, alexv, georgev}@aegean.gr`
[2] Institution of Informatics and Telecommunications, NCSR "Demokritos", Greece
`vangelis@iit.demokritos.gr`

Abstract. In this paper we present SEMA tool for the automatic mapping of ontologies. The main purpose of SEMA is to locate one to one equivalence correspondences (mappings) between elements (i.e., classes and properties) of two input ontologies. Towards this goal, SEMA synthesizes lexical, semantic and structural matching algorithms through their iterative execution.

1 Presentation of the system

1.1 State, purpose, general statement

Ontologies have been realized as the key technology to shaping and exploiting information for the effective management of knowledge and for the evolution of the Semantic Web and its applications. In such a distributed setting, ontologies establish a common vocabulary for community members to interlink, combine, and communicate knowledge shaped through practice and interaction, binding the knowledge processes of creating, importing, capturing, retrieving, and using knowledge. However, it seems that there will always be more than one ontology even for the same domain. In such a setting, where different conceptualizations of the same domain exist, information services must effectively answer queries, bridging the gaps between conceptualizations of the same domain. Towards this target, networks of semantically related information must be created at-request. Therefore mapping of ontologies is a major challenge for bridging the gaps between agents (software and human) with different conceptualizations.

---

[1] This work is part of research project ONTOSUM (www.ontosum.org), implemented within the framework of the "Reinforcement Programme of Human Research Manpower" (PENED) and co-financed by E.U.-European Social Fund (75%) and the Greek Ministry of Development-GSRT (25%).

Tools for the automated mapping of ontologies have achieved remarkable results but still there is lot of space for improvements when dealing with real world ontologies. Building on our experience in participating in OAEI 2006 with AUTOMS [1], we intent to further increase the precision and recall of our matching methods, and further minimize the efficiency cost, by devising enhanced techniques and combinations of methods.

This paper presents the SEMA tool for the mapping of ontologies. SEMA is built on top of AUTOMS-F [2], which a framework implemented as a Java API, aiming to facilitate the rapid development of tools for the automatic mapping of ontologies. AUTOMS-F provides facilities for synthesizing individual ontology matching methods.

The main purpose of SEMA is to locate one to one equivalence correspondences (mappings) between the elements (i.e., classes and properties) of two input ontologies, by increasing the recall of the mapping process and achieving a fair balance between precision and recall. SEMA combines lexical, semantic and structural matching algorithms: A semantic matching method exploiting Latent Dirichlet Allocation model (LDA) [3], requiring no external resources, in combination with the lexical matcher COCLU (COmpression-based CLUstering) [4] and a matching method that exploits structural features of the ontologies by means of simple rules. This combination of approaches contributes towards automating the mapping process by exploiting lexical, structural and semantic features of the source ontologies, resulting to increased recall and precision. It must be emphasized that the aggregation of the mappings produced by the individual methods is performed through their iterative execution as described in [5, 6].

It must be pointed that the experience gained by participating in the OAEI contest helped us towards the following aspects: (i) We increased the precision and recall of SEMA by iteratively combining the individual matching methods, (ii) we improved AUTOMS-F framework by adding more facilities towards the synthesis of individual matching methods, (iii) we noticed the fact that tools such as SEMA tend to fail to notice subsumption relations between elements of distinct ontologies, since they assess only equivalences between them, and finally, (vi) we managed to improve the execution time of matching methods, such as the one based on LDA.

1.2 Specific techniques used

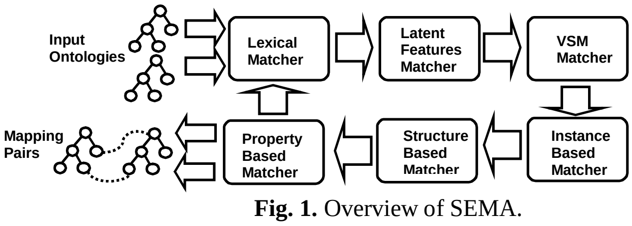


Fig. 1. Overview of SEMA.

SEMA combines six matching methods, executed in a predefined sequence, as depicted in Fig. 1. Each method in sequence exploits the results of the previous methods, aiming to find additional mapping element pairs. This policy is applied as the competition is restricted to the discovery of one-to-one mappings, and in order to exploit the complementary nature of the combined methods. As already pointed, SEMA iteratively executes the overall mapping method, providing the mappings computed during an iteration as input to the next iteration, until no change in the matching pairs arises. In more detail, the ontology mapping problem is modeled as an iterative process that finds the most nearest reachable fixed point of a vector function, as presented in similar approaches in the literature ([5], [6] and [7]).

The following paragraphs present the matching methods in the order of their execution:

Lexical matcher: As said, SEMA uses a lexical matcher implementing the COCLU lexical similarity approach [4].

COCLU was originally proposed as a method for discovering typographic similarities between sequences of characters over an alphabet (ASCII or UTF character set), aiming to reveal the similarity of classes instances' lexicalizations during ontology population [4]. It is a partition-based clustering algorithm which divides data into clusters and searches the space of possible clusters using a greedy heuristic. Each cluster is represented by a model, rather than by the collection of data assigned to it. The cluster model is realized by a corresponding Huffman tree which is incrementally constructed as the algorithm dynamically generates and updates the clusters by processing one string (instance's surface appearance) at a time. The use of a model classifies the algorithm to the conceptual or model based learning algorithms. To decide whether a new string should be added in a cluster (and therefore, that it lexicalizes the same class/property as the other strings in the cluster do) the algorithm employs a score function that measures the compactness and homogeneity of a cluster. This score function, Cluster Code Difference (*CCDiff*), is defined as the difference of the summed length of the coded string tokens that are members of the cluster, and the length of the cluster when it is updated with the candidate string. This score function groups together strings that contain the same set of frequent characters according to the model of a cluster (e.g., Pentium III and PIII).

According to the above, COCLU takes as input two strings and returns their similarity. The local name, label or comment of an OWL class or property, considered as a string, belongs in a particular cluster when its *CCDiff* is below a specific threshold and it is the smallest between the *CCDiff's* of the given string and all existing clusters. Based on our experience with COCLU, the similarity threshold (ranging in [0,1]) was set to 0.986. A new cluster is created if the candidate string cannot be assigned to any of the existing clusters. As a result, it is possible to use the algorithm even when no initial clusters are available.

Matching pairs of ontology elements are generated according to the following rules:

1. A pair of ontology elements is a matching pair if the similarities of their local names, and labels, and comments are greater than the threshold value.
2. If two *ontology elements* do not match according to rule 1, then the elements are considered to match if the similarities of either their local names, or labels, or comments are greater than the threshold value.
3. If two *ontology elements* do not match according to rule 1 and rule 2, then their labels are substituted by their synonyms found in WordNet (in case *they have* WordNet entries) and rules 1 and 2 are repeated using the synonyms.

Latent Features Matcher: The second matching method applied is the semantic matching one detailed in [3]. This method aims at discovering and exploiting *latent features* that reveal the intended meaning of ontology elements. This is a contextual approach to the ontology mapping problem, where at first, ontology elements are transformed into vectors according to specific rules [3] that exploit elements' vicinity (e.g., labels, comments, instances, properties, super/sub elements, domain and range of properties etc.) with respect to the semantics of the specifications. The vectors' length corresponds to the number of the distinct words in both input ontologies and each entry holds the frequency of each word in the vicinity of the corresponding element. Exploiting this representation of ontology elements, the method computes *latent features* that express the intended meaning of ontology elements. This is done by applying the reverse generative process of the Latent Dirichlet Allocation (LDA) [8] model. Doing so, each element is represented as a distribution over *latent features*, and similarities between elements' pairs of the two ontologies is computed by means of the Kullback-Leibler divergence [9] measure. This measure estimates the divergence of distributions over latent features i.e., the divergence of elements' approximated intended meaning.

The major advantages of this approach are as follows: The use of latent features helps to deal with problems of imprecise and vague ontology elements' descriptions, as well as with cases of polysemy and synonymy. Also, the proposed approach does not presuppose the existence of any external resource, as it exploits words in the vicinity of ontology elements.

Vector Space Model (VSM) Matcher: The third method is a standard Vector Space Model [10] based technique where ontology elements are represented as vectors of weights. Each weight corresponds to a word and is being calculated using the TF/IDF measure. The similarity between two vectors is being computed by means of the cosine similarity measure. Element pairs with cosine similarity above a predefined threshold (0.2 in our experiments) are returned as matched pairs. The rules used for the extraction of words from ontology elements are the same as in the *latent features'* based matcher.

Instance Based Matcher: The fourth mapping method of SEMA is a lexical matching method exploiting the instances of classes. Specifically, two classes are considered to match if the percentage of their mapped instances is above a predefined threshold (10% in our experiments). Two instances match if the percentage of their matched properties is above a predefined threshold (10% in our experiments). Two properties of two distinct classes' instances match if their values are assessed to match by the COCLU lexical matcher.

Structural Based Matcher: The fifth method of SEMA is a structural matching method, which utilizes the mappings produced by the above described matching methods. According to this method, if two classes have at least a pair of matched super classes and a pair of matched sub class, then they are also considered to match.

Property Based Matcher: Similarly to the previous structural method, this one utilizes the properties' mappings produced by the other methods in order to locate new matching pairs of classes. Specifically, two classes are considered to match, if the percentage of their mapped properties is above a predefined threshold (90% in our experiments).

Iterative Execution: The above mentioned matching methods, performed in the specified sequence, compute matching pairs by taking into account the vicinity of each ontology element, as well as its features (local name, label and comments). The vicinity includes the elements directly related to this element, together with their features. However, performing iteratively, the vicinity of each element can be extended to include its matching element in the target ontology.

This introduces a recursive dependency, which as it is pointed in [6], requires non-standard computational means. This problem has been approached by Bisson [5] and Euzenat et al [6].

As it has been proposed in [6], given the recursive nature of these computations and aiming to compute the intended meaning of classes, we can still find the intended meaning of each class through an iterative process that finds the most nearest reachable fixed point of a vector function. Based on this work, SEMA, aiming to compute matching pairs of the input ontologies, performs an iterative computation as follows:

```
Repeat the following process until there is no change in the mapping pairs between the input ontologies.
1. For each element $E$ do the following:
1.1. For each element in the vicinity of $E$
    In case there is no mapping element associated to this element compute the initial mapping based on its features.
1.2. Repeat the following until there is no change in the mapping computed for the element $E$
        1.2.1 Compute the mapping of $E$ using the mappings of elements in its vicinity
        1.2.2 Re-compute the mappings of elements in its vicinity changing only the mapping for $E$
```

Performing iteratively, SEMA improves its precision, as it manages to propagate mappings to elements' vicinity, while also achieving a high degree of individual methods' combination: The results of each method feed the input of the other methods in the next iteration. In the current SEMA version the methods that contribute to the iterative computation are the VSM matcher, the property based matcher and the structure based matcher. The latent features matcher is executed only in the first iteration due to its requirements of high computational resources.

1.3 Adaptations made for the evaluation

Since SEMA is built on top of AUTOMS-F framework [2], as already mentioned above, there was no need for particular adaptations in order to run SEMA on the benchmark test cases and output the resulting mappings in the requested format. AUTOMS-F provides specific classes and methods for this purpose.

1.4 Link to the system and parameters file

http://iit.demokritos.gr/~vspiliop/SEMA.zip

1.5 Link to the set of provided alignments (in align format)

http://iit.demokritos.gr/~vspiliop/SEMA_results.zip

2 Results

Results produced by SEMA are grouped and discussed below. SEMA is implemented as a stand-alone Java programme and was executed on a dual core Ubuntu Linux PC (2 x 2.44 GHz cores, 1.5GB memory). Although, two cores where available no multi-threading mechanism was exploited.

2.1 Benchmark

2.1.1 Tests 101 to 104

In these test cases the target ontologies have no major differences (except test 102) from the common source ontology. All mapping pairs are produced by the COCLU lexical matching method. In 102, where the target ontology is irrelevant to the source ontology, no mappings are returned.

Tests	H-mean Precision	H-mean Recall
101-104	1.0	1.0

2.1.2 Tests 201 to 210

In these test cases ontology elements' features such as local names, labels and comments in the target ontologies have been changed in various ways (e.g., using uppercase letters, underscores, translating in foreign language, using synonyms, random strings or being suppressed). It is evident from the table below that the recall harmonic mean value drops significantly, comparing to the previous test category. This is due to tests 202, 209 and 210, where except from the fact that many source ontology elements' labels have been replaced by elements having completely different lexicalizations (foreign language, synonyms or random strings) in the target ontology, the comments are suppressed, limiting the common features of the input ontologies. However, even in test 202, where comments are suppressed and ontologies do not share much of lexical information, SEMA manages to achieve 74% precision and 30% recall. This is achieved by exploiting a small fragment of instances' common lexical information and by propagating similarity through the iterative execution of methods. However, as we will see in test cases 248-266 the propagation is not so effective and improvements should be considered.

It must be noticed that the exploitation of WordNet by the lexical matcher helps SEMA to perform 82% in terms of precision and 61% in terms of recall, even when no comments are available and labels are replaced by synonyms (test 209).

Tests	H-mean Precision	H-mean Recall
201-210	0.91	0.80

2.1.3 Tests 221 to 247

In these tests the changes made in the target ontology concern the hierarchy, the properties, the instances and the number of properties defined in classes. As we can see from the overall results, there is a minor decrease in terms of both precision and recall (comparing to test case 101). This performance is due mainly to test case 230 where SEMA performs 75% in terms of precision and 100% in terms of recall. The methods that introduce false mappings in this test (in order of negative influence) are the property based matcher, the VSM matcher and the latent features matcher. The first is due to the introduction of much more properties in the target ontology. The second is due to the noise introduced in the feature vectors representing classes and properties. More specifically, if elements different in meaning contain common words (resulting to similar representation vectors), then the VSM matcher may introduce false positives. Although, the latent features matcher is more tolerant to such noise, as it statistically generates latent features that focus on the significant statistical correlations between the elements of the two input ontologies, still it generates some false positives. It must be also pointed that as the latent features matcher is executed before the VSM matcher, it acts as a filter against such phenomena (since elements matched by the latent features matcher are not tested by the VSM matcher).

Tests	H-mean Precision	H-mean Recall
221-247	0.96	0.99

2.1.4 Tests 248 to 266

These are the most difficult tests of the benchmark track since local names, labels, and comments have been removed or replaced by random strings and only in some cases a fragment of the lexical features of instances is not altered from the target ontologies. SEMA relies totally on its instance based matcher to locate mappings between classes and properties. The iterative execution of SEMA does not manage to propagate similarity efficiently and restrain itself mainly in correcting some false positive mappings. By examining the raw results we observe that SEMA performs indifferently to the (non) existence of a class hierarchy. It must be pointed that the lexical matching method contributes only one mapping, i.e., the “lastName = lastName”. Concerning the recall values, they range in [25, 31], while the precision values range in [65, 100].

Tests	H-mean Precision	H-mean Recall
248-266	0.75	0.27

2.1.5 Tests 301 to 304

In the real world tests SEMA performs relatively satisfactory in terms of recall, but its precision definitely needs improvement. The main reason for the low precision is the low threshold value (0.2) of the VSM matcher. The threshold has been tuned in order to maximize the overall (101-304) precision and recall values. But this threshold is not the optimum if the evaluation is narrowed to tests 301-304. On the other hand, as it has been explained, the latent features matcher has a better discriminating ability and introduces very few false positives.

Tests	H-mean Precision	H-mean Recall
301-304	0.67	0.79

2.2 Anatomy, Directory and Food

We were not able to run these tests due to technical problems in parsing these ontologies.

2.3 Conference

Several experiments were performed using various ontologies provided in the Conference track. The main lesson learned is that SEMA tends to “confuse” subsumption relations between elements of different ontologies with equivalence ones. For example, when mapping ekaw.owl to Conference.owl there where such cases, such as “Research_Topic=Topic” or “Assigned_Paper=Paper”. This is mainly due to the low threshold value (0.2) of the Vector Space Model mapping method. On the other hand, in other ontology pairs (e.g. ekaw.owl – crs_dr.owl) there is no such a phenomenon: all mappings assessed to be equivalences are indeed equivalences.

3 General comments

3.1 Comments on the results

The major advantages of SEMA are:
(i) The extensive exploitation of all linguistic features of the input ontologies through different methods.
(ii) The aggregation of results and methods’ combination via iteration, leading to the correction of false positive mappings of previous iterations and simultaneously performing propagation of similarities.
(iii) The use of latent features as a way to overcome the problems introduced by the phenomena of synonymy and polysemy, and finally,
(vi) The efficient and effective lexical similarity assessment performed by the COCLU lexical matcher.

The major weaknesses of SEMA are:
(i) Its dependence on threshold values. This leads SEMA to “confuse” subsumption relations with equivalence mappings (as noticed in the real world cases of the Consensus Workshop Track). This phenomenon has been also reported by others in the OAEI 2006 contest [11].
(ii) The inability of SEMA to parse large ontologies.

3.2 Discussions on the way to improve the proposed system

The ways to improve SEMA directly derive from its weaknesses, as follows:
(i) SEMA’s parsing abilities must be enhanced in order to be tested in all tracks of the OAEI contest.
(ii) Advanced mapping policies must be specified, so as to exploit input ontologies’ characteristics, deciding on the mapping methods to be applied. We believe that this will lead to an improvement of the precision values of SEMA.
(iii) We plan to introduce a matcher that locates subsumption relations [12] as a way to filter subsumption relations that are “confused” as equivalence relations.

3.3 Proposed new measures

We believe that systems should be able to locate not only equivalences between elements of distinct ontologies, but also other types of relations such as subsumption (inclusion) ($\sqsupseteq$ or $\sqsubseteq$), mismatch ($\perp$) and overlapping ($\sqcap$) and still be able to discriminate among them. This is of particular importance in the case of real world ontologies, where different conceptualizations make the complete alignment of ontologies very difficult. As a result, a new measure could be the number of false positive pairs assessed to belong in the equivalence relation, while belonging to an other relation (e.g., to the subsumption relation).

4 Conclusion

Our participation in the OAEI 2007 contest with SEMA tool has been a significant experience. We have actually been able to identify pros and cons of our tool, and improve several of its aspects. The organizers’ feedback and the comparison with the other tools will also contribute to future improvements of the tool.

References

1. K. Kotis, et al. AUTOMS: Automating Ontology Mapping through Synthesis of Methods, OAEI 2006 contest, Ontology Matching International Workshop, USA, 2006
2. A. Valarakos, V. Spiliopoulos, K. Kotis, G. Vouros. AUTOMS-F: A Java Framework for Synthesizing Ontology Mapping Methods. I-KNOW 2007 Special Track on Knowledge Organization and Semantic Technologies 2007 (KOST '07) September 5, 2007, Graz
3. V. Spiliopoulos, G. Vouros, V. Karkaletsis. Mapping Ontologies Elements using Features in a Latent Space. The 2007 IEEE / WIC / ACM International Conference on Web Intelligence (WI 07), Silicon Valley, USA
4. Valarakos A., Paliouras A., Karkaletsis G., Vouros G. A name-Matching Algorithm for supporting Ontology Enrichment. In Proceedings of SETN’04, 3rd Hellenic Conference on Artificial Intelligence, Samos, Greece (2004)

5. Gilles Bisson. Learning in FOLwith similarity measure. In Proc. 10th American AAAI conference, San-jose (CA US), 1992
6. Jérôme Euzenat, Petko Valtchev, Similarity-based ontology alignment in OWL-Lite, Ramon López de Mantaras, Lorenza Saitta (eds), Proc. 16th european conference on artificial intelligence (ECAI), Valencia (ES), pp333-337, 2004
7. Vouros A. and Kotis K.: Extending HCONE-merge by approximating the intended interpretations of concepts iteratively. 2nd European Semantic Web Conference, Heraklion, Crete May 29 – June 1, 2005. Proceedings, Series: Lecture Notes in Computer Science, vol. 3532, Asunción Gómez-Pérez, Jérôme Euzenat (Eds.), Springer-Verlag
8. Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet Allocation. Journal of Machine Learning Research, 3, 993-1022
9. Lin, J. (1991). Divergence measures based on Shannon entropy. IEEE Transactions on Information Theory, 37(14), 145–51
10. Raghavan, V.V. and Wong, S.K.M. A Critical Analysis of Vector Space Model for Information Retrieval. JASIS 37(5) 1986, 279-287
11. O. Svab, V. Svatek, H. Stuckenschmidt: A Study in Empirical and 'Casuistic' Analysis of Ontology Mapping Results. In Proceedings of ESWC, 2007
12. V. Spiliopoulos, A. Valarakos, G. Vouros, V. Karkaletsis: Learning Subsumption Relations with *CSR*: A Classification-based Method for the Alignment of Ontologies. The Second International Workshop on Ontology Matching, Korea, 2007 (accepted as poster)

Appendix: Raw results

Matrix of results

#	Name	Prec.	Rec.	Time (sec)
101	Reference alignment	1	1	10
102	Irrelevat ontology	NaN	NaN	36
103	Language generalization	1	1	9
104	Language restriction	1	1	8
201	No names	0,92	0,98	9
202	No names, no comments	0,74	0,3	11
203	No comments	1	1	5
204	Naming conventions	0,95	0,96	8
205	Synonyms	0,93	0,96	9
206	Translation	0,94	0,97	11
207		0,92	0,97	11
208		0,89	0,8	6
209		0,82	0,61	9
210		0,81	0,47	10
221	No specialisation	1	1	8
222	Flatenned hierachy	0,96	1	9
223	Expanded hierarchy	0,97	0,98	10
224	No instance	1	1	7
225	No restrictions	1	1	8

228	No properties	1	1	16
230	Flatenned classes	0,75	1	8
231		1	1	8
232		1	1	7
233		1	1	16
236		1	1	12
237		0,96	1	6
238		0,97	0,97	9
239		0,94	1	16
240		1	1	22
241		1	1	12
246		0,94	1	12
247		0,94	0,94	12
248		0,73	0,3	11
249		0,73	0,28	12
250		1	0,27	14
251		0,65	0,26	11
252		0,65	0,25	11
253		0,71	0,28	11
254		1	0,27	15
257		1	0,27	14
258		0,66	0,25	12
259		0,68	0,26	14
260		1	0,31	14
261		1	0,27	18
262		1	0,27	14
265		1	0,31	14
266		1	0,27	18
301	BibTeX/MIT	0,7	0,75	8
302	BibTeX/UMBC	0,62	0,6	5
303	Karlsruhe	0,55	0,8	9
304	INRIA	0,76	0,93	6

Simple library thesaurus alignment with SILAS

Roelant Ossewaarde[1]

Linguistics Department
University at Buffalo, the State University of New York
rao3@buffalo.edu

Abstract. This paper describes a system written in C which employs the instance-based approach to ontology alignment of library thesauri. It computes relatedness relations between subsets of a library catalogue. Even a very basic method to calculate confidence in the found relation yields usable alignment results.

1 Presentation of the system

1.1 State, purpose, general statement

The system in this paper (SILAS - Simple Instance-based Library-thesaurus Alignment System) is an example of 'instance-based ontology matching' [3]; it measures the similarity between subsets annotated with words from different ontologies to match up the concepts described by these words.

Definitions An *ontology* is taken to be an organization of concepts C by some set of ontological relations. An *agent* is a human or computer program that employs an ontology. A *concept* is a particular agent's conceptualization of an element of the domain of discourse. A concept is denoted by one or more words, which may be shared between agents. In the specific case of the library task, the goal is to find concepts in one ontology that are equivalent or related to concepts in the other one.

A thesaurus is typically used in a library to create a subset of books - we assume that for each word $w_i \in W$ that denotes a concept $c_i \in C$, there is a subset $B_i \in B$ of books that is related to that concept. For the purpose of ontology alignment, a subset B_i is assumed to represent the meaning of the concept c_i described by word w_i in that database. Terminology: subset B_i represents the *extensional meaning*, or *extension*, of the concept c_i. For example, the concept ROWING may be described by the Dutch word *roeien* and the English word *rowing*. The extension of the concept ROWING is formed by the set of books which have their subject tagged as *roeien* or *rowing*.

Assumptions SILAS relies on the following assumptions about the relationship between subsets and concepts:

1. Assume a concept c_A (denoted by word w_A) in ontology A maps onto a concept c_B (denoted by word w_B) in ontology B. Then the subset of books described by word w_A will generally be the same as the subset of books described by word w_B of B.

For example, if both ontology A and B have a term for the concept *rowing*, a librarian who works with ontology A will generally tag the same books as a librarian who works with ontology B.

2. Converse: consider two partitionings of the set of books, one tagged as about concept c_A and the other tagged as about concept c_B. The more overlap the two subsets show, the stronger the semantic relation between concepts c_A and c_B. If there is enough overlap, the concepts are said to be the same.
   For example, if there is a subset of books tagged with the words from ontology A for the concept *rowing* in that ontology, and there is also a subset of books tagged with the words from ontology B for the concept *rowing*, and the subsets contain the exact same members, then the two concepts are probably equivalent.
3. It is assumed that for every concept in ontology A, there is a somewhat equivalent concept in ontology B.

The two ontologies used for alignment were the ones provided by the organizers, ontology ‘GTT’ with 32530 concepts and ontology ‘Brinkman’ with 4845 concepts.

The approach The system identifies overlapping subsets and computes a very simple metric to predict semantic relatedness between the concepts of which the subsets are extensions. The library track was particularly suited for this approach because it provides both a pair of ontologies, roughened and shaped by actual use, and a large set of datapoints actually described by the ontologies.

For this particular task, the collection of the library of the University of Amsterdam was used to provide the datapoints. This library has a collection of about 4 million books, of which about 90% is annotated using either of the two thesauri.

Systems such as AUTOMS [4] and Falcon-AO [2] use WordNet senses as an intermediary between concepts in two ontologies. But external semantic annotation using resources such as WordNet is limited to languages for which such resources exist, and is limited to domains for which such resources are formulated.

SILAS is based on the condition that no semantic knowledge from outside the ontology is available at runtime. The meaning of the ontological concepts is in some way coded through its use; the extension of a concept represents its meaning. For example, if two different librarians with two different thesauri make a subset of all books on rowing and label the subset with their respective terms for that sport, the system should find that the used terms probably describe the same concept, because they both have the same subset as their extension.

1.2 Specific techniques used

Confidence scores The overlap between sets may be relevant, and it may not. In the latter case, it is noise that needs to be filtered out. As an example, the category ‘antillianen’ (*Antillians*) shows some overlap with the category ‘anticonceptie’ (*contraceptives*). Perhaps there have been included in the subset publications on the use of contraceptives among Antillians, but that overlap should clearly not be taken as evidence that Antillians and contraceptives are in any way semantically related.

For each overlapping pair of sets, a confidence score is calculated, which expresses the following characteristics:

1. Two sets are more likely to be aligned if they have more elements in common, instead of less.
2. Two sets, one of which large, the other small, are more likely to aligned if their average portion of joint elements is more, instead of less.
3. Two sets are more likely to be aligned if they are identified by the same words. (*lexical booster*)

The confidence score is implemented by ranking properties of the overlapping sets A and B using a variation of the often used Jaccard similarity measure:

1. Let confidence = 0 and x = $|A \cap B|$ (the number of elements in the intersection of A and B);
   (a) if $((x/|A|) + (x/|B|))/2 > 0.05$, confidence +1, if > 0.15, additional +1.
   (b) i. if $x \geq 5$ and $x/|A| \geq 0.3$, confidence +4; otherwise
       ii. if $x/|A| \geq 0.05$, confidence + 1, if $x/|A| \geq 0.2$, additional +1.
   Similarly so for set B.
2. if the name of the concept described by subset A is identical to the name of the concept described by subset B, confidence +5. (*lexical booster*)

This yields a confidence score S between 0 and 15. If $3 \geq S \geq 5$, two concepts are related; if $6 \geq S \geq 15$, two concepts are equivalent. If there are no alignment candidates with a confidence score high enough for equivalence, but there is an alignment candidate with the same name, this candidate is chosen to be the equivalent concept. Related concepts have been assigned the relation `broadMatch`, equivalent relation have been assigned the relation `exactMatch`. The confidence score is then mapped onto a scale $\{0 \ldots 1\}$.

The rationale for the lexical booster is that it is obvious that two concepts with the same name are probably equivalent. However, lexical similarity alone is not enough for concepts to be judged related. Its use is mainly for ranking alignment candidates: if there is a lower-ranked alignment candidate with a similar name, it gets promoted to the top ranking. The numbers used in the algorithm were initially chosen on the basis of intuition and adjusted after hand-checking the results for 5% of all categories. The algorithm becomes more selective if the thresholds for relatedness and equivalence are raised.

Alignment procedure The procedure of alignment is as follows:

1. The words describing ontological concepts in the two ontologies were translated into two sets search terms T.
   A software tool `skostool` was developed in C, on the basis of the open-source RDF-parsing library `libraptor`. The tool enables easy browsing, searching and manipulation of the concepts in the ontologies, and provides a C API-interface to the files provided by the organizers. In particular, it can compute hierarchical measures that we intend to use in subsequent versions of our alignment protocol.

2. Using automated spiders, for each $t_i \in T$, all records tagged with term t_i were retrieved into subset $B_i \in B$.
3. Each subset corresponding to a concept from ontology GTT was then compared to all subsets corresponding to concepts from ontology Brinkman.
   Each comparison yields an overlap score for each pair $\{B_i, B_j\}$, where B_i is a subset according to ontology GTT and B_j is a subset according to ontology Brinkman. If the overlap between the two sets is sufficiently high so that the confidence score is greater than 0, B_j is considered an *alignment candidate* for B_i.
   The overlap is determined by finding the intersection $B_i \cap B_j$, using ISBN as a unique identifier for each $b \in B_i(B_j)$. The comparison function was coded in C. The runtime on a linux PC is about 3 hours.
4. For each B_i, the alignment candidates were scored according to the confidence measurement system. If the confidence exceeds certain thresholds, the concepts of which subsets B_i and B_j are extensions are judged *related* or *equivalent*.

1.3 Link to the system and parameters file

The system and its documentation is available from `http://www.buffalo.edu/~rao3/oaei2007`. The system requires a POSIX-compliant C-compiler to build, and makes use of the open source libraries `libxml` and `libraptor`. Two memory leak bugs were fixed in `libraptor` in order to make it work flawlessly in combination with `libxml`; patches have been submitted to the maintainer of `libraptor`.

1.4 Link to the set of provided alignments (in align format)

The alignments, both in human readable form and in XML-format are available from `http://www.buffalo.edu/~rao3/oaei2007`.

2 Results

Because participation was limited to the library track (scored blind), no formal results can be reported yet. Informal review by the author of the system's performance shows that with the current parameters, in most cases both the precision and recall of the algorithm are high enough to be satisfactory. Some of the judgements are subjective (is 'pottery' really equivalent to 'ceramic arts'?); fellow linguists judged more than 90% of the relations as 'acceptible as correct' in an informal evaluation of the first 500 concepts.

3 General comments

3.1 Comments on the results

There are a few benefits of the instance-based approach which make it interesting for further study.

First, the approach is independent of outside semantic knowledge at runtime. No sources such as WordNet or FrameNet are required to generate the mappings. The assumption is that humans, in this case librarians, have already assigned meaning to a term by formulating its extension, ie. by identifying a subset of books on that subject, and it would be a waste not to use that knowledge in mapping.

Second, the system is blind as to the exact terms used to describe concepts. The two ontologies may as well have been given in completely unrelated languages; as long as they are both used to describe the same set of data points, the algorithm will find matches. The advantage of this can easily be seen in a situation of academic libraries in different countries; each library would use its own localized thesaurus to describe the proceedings of this conference, and if all works well, the algorithm would pick up the relevant localized terms and align them.

3.2 Discussions on the way to improve the proposed system

The system presented is in its infant stages. The following are a few possible improvements for SILAS v2.

- The system currently cannot compute hierarchical relations. We'd like to implement that, for example using the MedOnt / MedCount algorithms described in [1] or a similar methodology. For example, given a set of alignment candidates from one ontology, the hierarchy between the different candidates may be reconstructed using the original ontology and factored in while scoring the confidence rating, perhaps using the semantic similarity between nodes of the ontology as computed in the DSSim-system [5].
- The system regards hierarchical relations as non-transitive. In other words, if concept c_1 taxonomically dominates concept c_2 (ex. *science* describes the superset of *linguistics*), a book tagged as subject c_2 would not also occur in the extension of concept c_1. If the relation is considered transitive, as it perhaps should (although [3] suggests a decrease in performance in this particular task once hierarchical information is taken into account), the recall of the alignment improves. However the precision decreases significantly, due to the fact that the confidence ratings used for this task cannot adequately distinguish a higher and a lower hierarchical concepts. Because by definition the extension of the concept SCIENCE includes all scientific concepts, such as BIOLOGY and LINGUISTICS, each of these concepts would show overlap with SCIENCE and would be considered an alignment candidate. Until the algorithm can be restricted to only choose alignment candidates in roughly the same hierarchical area, transitive relations would degrade the quality of the system.
- The lexical booster is very simple. It is currently based on simple pattern matching, and does not detect plural-singular alternations and such. A better way of computing lexical similarity may provide a more fine-grained confidence measure and a better way of using lexical information in ranking alignment candidates.

4 Conclusion

SILAS uses a very naive, simple, number-crunching approach. Official results have not been made available yet, but a glance over the computed alignments shows at least a satisfactory performance, given the low complexity of the methodology. It makes effective use of a source of information - the annotated database - that is often ignored, and as such may provide a starting point for combinations with other methodologies.

References

1. Alistair E. Campbell and Stuart C. Shapiro. Algorithms for ontological mediation. In S. Harabagiu, editor, *Usage of WordNet in Natural Language Processing Systems: Proceedings of the Workshop*, pages 102–107, New Brunswich, NJ, 1998. COLING-ACL.
2. Wei Hu, Gong Cheng, Dongdong Zheng, Xinyu Zhong, and Yuzhong Qu. The results of Falcon-AO in the OAEI 2006 campaign. In Shvaiko et al. [6].
3. Antoine Isaac, Lourens Van der Meij, Stefan Schlobach, and Shenghui Wang. An empirical study of instance based ontology matching. ISWC 2007 + ASWC 2007, 2007.
4. Konstantinos Kotis, Alexandros Valarakos, and George Vouros. AUTOMS: Automated Ontology Mapping through Synthesis of methods. In Shvaiko et al. [6].
5. Miklos Nagy, Maria Vargas-Vera, and Enrico Motta. Dssim-ontology mapping with uncertainty. In Shvaiko et al. [6].
6. Pavel Shvaiko, Jérôme Euzenat, Natalya Noy, Heiner Stuckenschmidt, Richard Benjamins, and Michael Uschold, editors. *Proceedings of the 1st International Workshop on Ontology Matching (OM-2006)*, 2006.

SODA: an OWL-DL based ontology matching system

Sami Zghal[1,2], Sadok Ben Yahia[2], Engelbert Mephu Nguifo[1], and Yahya Slimani[2]

[1] CRIL CNRS FRE 2499, Artois University, IUT of Lens
Rue de l'Université - S.P. 16, 62307 Lens Cedex, France
{zghal, mephu}@cril.univ-artois.fr
[2] Computer Science Department, Faculty of Sciences of Tunis, Tunisia
Campus Universitaire, 1060 Tunis, Tunisia
{sadok.benyahia, yahya.slimani}@fst.rnu.tn

Abstract. This paper describes SODA a novel ontology alignment method for the OWL-DL format. SODA uses a new approach that consists in computing local and semantic similarities among ontological elements.

1 Presentation of the system

SODA [1] (Structural Ontology OWL-DL [2] Alignment), is a new approach that aligns two OWL-DL ontologies using similarity measures [3]. Both OWL-DL ontologies are transformed in two corresponding graphs DL-GRAPH which describe all information in the ontologies. SODA uses the DL-GRAPH to align the two ontologies. It operates into successive steps. The first step, computes local similarity by means of linguistic and structural similarities, whereas the second one computes the semantic similarity. Figure 1 depicts the architecture of SODA system.

1.1 Specific techniques used

Each OWL-DL ontology to be aligned is transformed into a non oriented graph called DL-GRAPH. All the information belonging to OWL-DL ontology are faithfully mapped into the DL-GRAPH. Nodes of the proposed graph represent classes, properties and instances. The DL-GRAPH nodes represent six types (named also categories) of entities that may exist in an OWL-DL ontology: *i.e.*, concepts, instances of concepts, data types, values of data types and class properties (object nature and data type nature). Connections between the graph nodes map the relationships between the entities in an OWL-DL ontology. It is worthily noted that an OWL-GRAPH describes all the semantic relations between different entities of an ontology. A graph DL-GRAPH allows to represent four kinds of links: *specialization*, *attribution*, *instantiation* and *equivalence*. DL-GRAPHS are exploited by the alignment model SODA. Similarity measures are used to compare the components of the graphs in order to obtain the correspondence between them. Nodes and links of the two graphs are compared to get out the correspondence between different ontological entities using similarity measures. The output algorithm is an RDF file containing all the correspondences between the entities and the similarity measure values.

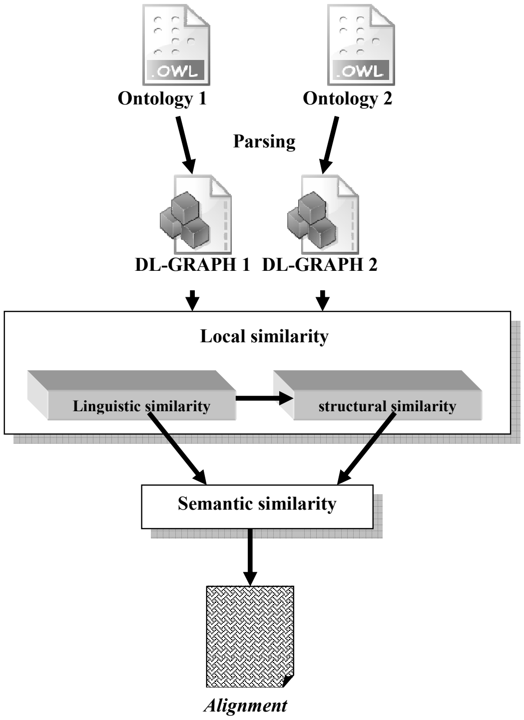


Fig. 1. Architecture of SODA

SODA explores the structure of DL-GRAPH to compute the similarity values between the nodes of both ontologies. The alignment model associates to each category of nodes an aggregation function. This function takes in consideration all the similarity measures and the structure of couple of nodes to be matched. This aggregation function explores all descriptive information of nodes.

SODA operates into two successive steps: local and semantic. The first step, implemented via PHASE1_LINGSIM (see Algorithm 1) and PHASE2_STRUCTSIM (see Algorithm 2) functions, computes the local similarity (linguistic and structural one). The second step, *c.f.* the PHASE3_SEMSIM function (see Algorithm 3), computes the semantic similarity. Table 1 summarizes the notations that are used in the description of our algorithms.

Local similarity

The computation of the local similarity is carried out in two phases. The first phase allows to compute the linguistic similarity for each couple of node of the same category. The second phase allows to compute the structural similarity using the structure of neighbors of the nodes to be aligned.

- O_1,O_2: two OWL-DL ontologies for alignment
- V_{LS}: linguistic similarity vector
- V_{SS}: structural similarity vector
- V_{VSEMS}: semantic similarity vector
Each node of the ontology is characterized by:
- **Type**: node type
- **Name**: node name
Each element of the vectors V_{SL}, V_{SS} et V_{VSEM} is characterized by:
- **Node 1**: the node of ontology O_1
- **Node 2**: the node of ontology O_2
- **Sim**: the similarity value

Table 1. Algorithm notations

Algorithm 1 (*c.f.*, PHASE1_LINGSIM function) computes the linguistic similarity measure. The name of properties and instances are used to compute linguistic similarity. For classes, the computation of linguistic similarity integrates also comments and labels. The computation of linguistic similarity is done only once for each node of the same category. JARO-WINKLER or MONGE-ELKAN [4] functions are used to compute the linguistic similarity. JARO-WINKLER measure is more adapted for short strings, like those representing names and labels [4]. Besides, MONGE-ELKAN measure is better indicated for long strings, *e.g.* the comments [4]. PHASE1_LINGSIM function computes the linguistic similarity of couple of nodes of both considered ontologies. PHASE1_LINGSIM function takes as input the two ontologies O_1 and O_2 and the linguistic similarity function $Funct_{LS}$. COMPUTELINGSIM function (*c.f.*, line 8 of Algorithm 1) takes as an input two nodes, $Node_1$ et $Node_2$, and linguistic similarity function. PHASE1_LINGSIM function returns as an output linguistic similarity value Sim_L. This function implements the JARO-WINKLER or the MONGE-ELKAN measures. Linguistic similarity of the different couples of nodes are used after that in the computation of the structural similarity.

Structural similarity is computed by using linguistic similarity of the couple of nodes to align and the neighborhood structure. Adjacent neighbor nodes of the entities are grouped by category, *c.f.* PHASE2_STRUCTSIM. This function takes as input two ontologies O_1 and O_2 to align, linguistic similarity vector V_{LS} and weights associated for each category Π_C. EXTRACTNODES function, (*c.f.*, lines 9 - 11 of Algorithm 2), allows to extract for each node, its neighbors and to put them in V_{Node_i}, where $Node_i$ is a node of O_1 or O_2. V_{Node1} and V_{Node_2} vectors and weights associated for each category, Π_C, are used by the COMPUTESTRUCTSIM function (*c.f.*, line 13 of Algorithm 2) to compute the structural similarity, Sim_S. To work out, the following "*Match-Based similarity*" [5, 6] is used to compute similarity between two categories (one in the first ontology and the other in second one):

$$MSim(E, E') = \frac{\sum_{(i,i') \in Paires(E,E')} Sim(i, i')}{Max(|E|, |E'|)},$$

```
Function : PHASE1_LINGSIM
Data:
  1. $O_1$ and $O_2$ : two ontologies to align
  2. $Funct_{LS}$ : linguistic similarity function
Results: $V_{LS}$ : linguistic similarity vector
Begin
    /*Parse all nodes of the ontology $O_1$*/
    forall ($Node_1 \in O_1$) do
        /*Parse all nodes of the ontology $O_2$*/
        forall ($Node_2 \in O_2$) do
            If $Node_1$.type=$Node_2$.type then
                $Sim_L$ = COMPUTELINGSIM($Node_1$,$Node_2$)
                /*Add $Node_1$,$Node_2$ and $Sim_L$ to $V_{LS}$*/
                Add(($Node_1$,$Node_2$,$Sim_L$),$V_{LS}$)
    return($V_{SL}$)
End
```

Algorithm 1: PHASE1_LINGSIM

where E et E' represent two sets of nodes belonging to the same category in O_1 and O_2. This function uses the local similarities of the couple (i, i') already computed. Structural similarity is computed by aggregating the "*Match-Based similarity*" of each group of adjacent neighborhood nodes by category. A weight is attributed for each group to have a normalized structural similarity. Each category has the same weight which is equal to 1 over the number of groups (categories). Structural similarity, Sim_S, is computed as follows:

$$Sim_S = \sum_{(E,E')\in(V_{Node_1},V_{Node_2})} \Pi_{(E,E')} MSim(E,E').$$

The structural similarity, Sim_S, is normalized since $\sum(\Pi_{(E,E')}) = 1$. Values of the linguistic similarity vector, *i.e.* V_{LS}, and structural similarity vector, *i.e.* V_{SS}, already obtained are combined to compute the semantic similarity.

Semantic similarity

The semantic similarity is a combined similarity measure of linguistic and structural similarities (local similarity). Algorithm 3 takes as input the two ontologies to be aligned, the two similarity vectors (the linguistic one, V_{LS}, and the structural one, V_{SS}). It takes also the weights attributed to the linguistic similarity and structural similarity (Π_L and Π_S =). Optimal weight values can be determined through several experiments. Algorithm 3 outputs semantic similarity vector, V_{SEMS}. EXTRACTSIM function (*c.f.*, lines 9 - 11 of Algorithm 3) extracts corresponding values of similarities of the nodes ($Node_1$ or $Node_2$) from similarity vectors (V_{LS} or V_{SS}). Each couple of entities, $Node_1$ and $Node_2$, of the same category, the semantic similarity is computed as follows (*c.f.*, line 13 of Algorithm 3):

$$Sim_{SEM}(e_1, e_2) = \mathit{\Pi}_L Sim_L(e_1, e_2) + \mathit{\Pi}_S Sim_S(e_1, e_2).$$

1.2 Link to the system and parameters file

At the following URL the system SODA can be downloaded: www.cril.univ-artois.fr/˜mephu/OAEI2007/systemSODA.rar.

```
Function : PHASE2_STRUCTSIM
Data:
  1. $O_1$ et $O_2$: two ontologies to align
  2. $V_{LS}$: linguistic similarity vector
  3. $\mathit{\Pi}_C$: weights for each category of node
Results: $V_{SS}$ : structural similarity vector
Begin
    /*Parse all nodes of the ontology $O_1$*/
    forall ($Node_1 \in O_1$) do
        /*Parse the nodes of the ontology $O_2$*/
        forall ($Node_2 \in O_2$) do
            If $Node_1$.type==$Node_2$.type then
                /*Extract form $V_{Node1}$ neighbor nodes $Node_1$*/
                $V_{Node_1}$=EXTRACTNODE($Node_1$)
                /*Extract form $V_{Node2}$ neighbor nodes $Node_2$*/
                $V_{Node_2}$=EXTRACTNODE($Node_2$)
                /*Compute the structural similarity*/
                $Sim_S$ = COMPUTESTRUCTSIM($V_{Node_1}$,$V_{Node_2}$,$\mathit{\Pi}_C$)
                /*Add $Node_1$, $Node_2$ and $Sim_S$ to $V_{SS}$*/
                ADD(($Node_1$,$Node_2$,$Sim_S$),$V_{SS}$)
    return($V_{SS}$)
End
```

Algorithm 2: PHASE2_STRUCTSIM

1.3 Link to the set of provided alignments

At the following URL the contest results of SODA are available: www.cril.univ-artois.fr/˜mephu/OAEI2007/resultsSODA.rar. As the evaluation process of OAEI 2007 Challenge was over, it is not possible to comment those results here. Some details of results can be found in [1], especially for the benchmark dataset.

2 Conclusion

In this paper, a new ontology alignment system for OWL-DL format is described. Results obtained on OAEI 2007 benchmarks and directory are available on website. Evaluation of benchmarks dataset are also in [1]. SODA, finds ontological entities to be aligned using local and semantic similarities. In further research, we plan to tackle alignment of large ontologies.

```
Function : PHASE3_SEMSIM
Data:
  1. $O_1$ et $O_2$: two ontologies to align
  2. $V_{LS}$: linguistic similarity vector
  3. $V_{SS}$: structural similarity vector
  4. $\Pi_L$ and $\Pi_S$: the respective weights of linguistic and structural similarity

Results: $V_{SEMS}$ : semantic similarity vector
Begin
    /*Parse all nodes of the ontology $O_1$ */
    forall ($N\oe ud_1 \in O_1$) do
        /*Parse all nodes of the ontology $O_2$*/
        forall ($N\oe ud_2 \in O_2$) do
            If $Node_1$.type==$Node_2$.type then
                /*Extract the linguistic similarity of $Node_1$ and $Node_2$ from $V_{LS}$*/
                $Sim_L$=EXTRACTSIM($V_{LS}$,$Node_1$,$Node_2$)
                /*Extract the linguistic similarity of $Node_1$ and $Node_2$ from $V_{SS}$*/
                $Sim_S$=EXTRACTSIM($V_{SS}$,$Node_1$,$Node_2$)
                /*Compute the semantic similarity*/
                $Sim_{SEM} = \Pi_L Sim_L + \Pi_S Sim_S$
                /*Add $Node_1$, $Node_2$ and $Sim_{SEM}$ to $V_{SEMS}$*/
                ADD(($Node_1$,$Node_2$,$Sim_{SEM}$),$V_{SEMS}$)

    return($V_{SG}$)
End
```

Algorithm 3: PHASE3_SEMSIM

References

1. Zghal, S., Ben Yahia, S., Mephu Nguifo, E., Slimani, Y.: SODA : Une approche structurelle pour l'alignement d'ontologies OWL-DL. In: Proceedings of the first French Conference on Ontology (JFO 2007), Sousse, Tunisia (October 2007)
2. Smith, M.K., Welty, C., Mcguinness, D.L.: OWL: Ontology Web Language Guide. Technical report, W3C: Word Wide Web Consortium, http://www.w3.org/TR/2004/REC-owl-guide-20040210/ (February 2004)
3. Euzenat, J., Mochol, M., Shvaiko, P., Stuckenschmidt, H., Svab, O., Svatek, V., van Hage, W.R., Yatskevich, M.: Results of the ontology alignment evaluation initiative 2006. In: Proceedings of the First ESWC 2006 international workshop on ontology matching. (2006)

4. Monge, A., Elkan, C.: The field-matching problem: algorithm and applications. In: Proceedings of the 2^{nd} International Conference on Knowledge Discovery and Data Mining. (1996) 267–270
5. Touzani, M.: Alignement des ontologies OWL-Lite. Master's thesis, University of Montreal (2005)
6. Euzenat, J., Loup, D., Touzani, M., Valtchev, P.: Ontology alignement with OLA. In: Proceedings of the 3^{rd} International Workshop : Semantic Web Conference EON, Hirochima, Japan (November 2004) 341–371

Acknowledgements

This work is partially supported by the French-Tunisian project PAI CMCU 05G1412. Thanks to Amine FERJANI and Marouen KACHROUDI for implementing the method.

TaxoMap in the OAEI 2007 alignment contest

Haïfa Zargayouna, Brigitte Safar, and Chantal Reynaud

LRI, Université Paris-Sud 11, Bât. G, INRIA Futurs
2-4 rue Jacques Monod, F-91893 Orsay, France
`firstname.lastname@lri.fr`

Abstract. This paper presents our first participation in the OAEI 2007 campaign. It describes an approach to align taxonomies which relies on terminological and structural techniques applied sequentially. We performed our method with various taxonomies using our prototype, TaxoMap. Previous experimental results were encouraging and demonstrate the relevance of this alignment approach. In this paper, we evaluate the strengths and weaknesses of TaxoMap in the context of the OAEI campaign where the ontologies to align are different from taxonomies we are used to deal with.

1 TaxoMap: Context

The increasing amount of information sources available on the Web requires techniques providing integration. Ontologies define concepts relative to particular application domains. They have become central in information integration because they allow description of content of integrated sources and make the vocabulary to be used in the queries explicit. The ontology alignment task (correspondences or mappings finding) is particularly important in information integration systems because it allows several heterogeneous systems which have their own ontology to be used jointly.

Our alignment system, TaxoMap, has been designed in the setting of query answering in the food risk domain. We aimed at increasing answers delivered by a web portal thanks to information provided by others sources annotated by semantic resources. Querying the portal was supported by a global schema exploited by a query interface which had to be reused without any change.

TaxoMap was designed to discover alignments between this global and rich schema and much simpler semantic resources of other sources. The experts required that the retrieval process should not be altered by the alignment process.

The alignment process is then **oriented** from an ontology, named source ontology, (for instance an ontology associated to external resources) to a target ontology (for instance the ontology of a web portal).

Moreover, our mapping techniques are strict: only concepts that have strictly the same label are matched with an equivalence relation. The remaining concepts of the source ontology are matched with a subclass relation which denotes a proximity relation. Therefore, TaxoMap proposes essentially subclass relation mappings.

We assume that often, content of information sources is not specified a lot. Simple ontologies reduced to classification structures, i.e. taxonomies, are the only way to describe their content. Moreover, we suppose that the taxonomies that we align are heterogeneous, describing the same domain in different vocabularies and structures, the target taxonomy being well-structured whereas the source taxonomy perhaps not. In this context, the approaches that rely on OWL data representations, exploiting all the ontology language features, do not apply. To find mappings, we can only use the following available elements: the labels of concepts in both taxonomies and the structure of the target taxonomy.

We propose several alignment techniques whose aim is to discover classes of mappings between taxonomies belonging to a general methodology usable across different application areas. Classes of mappings are categorized into probable mapping and potential mapping classes (i.e. to be confirmed or refuted manually). The mapping process can be viewed as an execution of various techniques invoked in sequence, namely terminological followed by structural techniques.

Terminological techniques are based on string comparisons. They discover mappings that exploit the richness of the labels of the concepts. These techniques are efficient in the sense that they provide high-quality alignments corresponding to probable mappings. Unfortunately, they are not sufficient. Many of the mappings are undiscovered. So, we extend the terminological techniques with structural ones.

The paper is organized as follows. Section 2 describes the alignment approach and the adaptations made for the evaluation. In section 3, we present the results of the experiments we have done so far.

2 Presentation of the system

2.1 State, Purpose and General Statement

The objective of our approach is to generate mappings between taxonomies. For us, a taxonomy T is a pair (C, H_C) consisting of a set of concepts C arranged in a subsumption hierarchy H_C. A concept c is only defined by two elements: a label and subclass relationships. The label is a name (a string) that describes entities in natural language and which can be an expression composed of several words. A subclass relationship establishes links with other concepts. It is the only semantic association used in the classification. A taxonomy is generally represented by an acyclic graph where concepts are represented by nodes connected by directed edges corresponding to subclass links.

The objective is to map the concepts of the source taxonomy T_{Source} to the concepts of the target taxonomy T_{Target}. It is an **oriented process** from T_{Source} to T_{Target}. Hence, we define the mapping as follows: Given two taxonomies, T_{Source} and T_{Target}, mapping T_{Source} with T_{Target} means that: for each concept (node) c_S in T_{Source}, we try to find a corresponding concept (node), c_T in T_{Target}, linked to c_S with an equivalence or a subclass relation.

The alignment process aims at finding one-to-one mappings between single concepts and establishing two types of relationships, equivalence and subclass relationships defined as follows.

Equivalence relationships An equivalence relationship, *isEq*, is a link between a concept in T_{Source} and a concept in T_{Target} with labels assumed to be similar.

Subclass relationships Subclass relationships are usual *is-A* class links. When a concept c_S of T_{Source} is linked to a concept c_T of T_{Target} with such a relationship, c_T is be considered as a super concept of c_S.

2.2 Techniques Used

All our alignment techniques are based on Lin's similarity measure ($Sim_{LinLike}$)[1] computed between each concept c_S in T_{Source} and all the concepts in T_{Target}. This measure compares strings and has been adapted to take into account the importance of words inside expressions. Terminological and structural techniques are used (see figure 1) and are applied in sequence to maximize the efficiency of the overall alignment process. For each technique, the objective is to select the best concept in T_{Target} among many mapping candidates MC (with a similarity measure not being null), the best concept having not necessarily the highest similarity measure.

Terminological techniques are executed first. Being based on the richness of the labels of the concepts, they provide the most relevant mappings. They are performed in three steps:

- **Search for equivalents** The first relationships to be discovered are equivalence relationships, which map concepts with a similarity measure corresponding to a strong similarity (greater than a threshold which has been set to 1 in our experiments).
- **Labels inclusion** We consider inclusion of name strings for which we propose a subclass mapping between c_S and c_T if c_T is the concept in T_{Target} with the highest similarity measure and if the name string of c_T is included in the name string of c_S in T_{Source}.
- **Relative similarity** If the name string of the concept c_T of T_{Target} with the higher similarity measure is not included in the name string of c_S, but if its similarity measure is significantly highest than the measure of the others, c_T is considered as a brother of c_S and the system proposes a subclass relationship between c_S and the father node of c_T.

All of the above techniques are performed in sequence. They merely rely on the values of similarity measures and lead to mappings which are generally reliable but not always sufficient in number.

```
TaxoMap ($T_{Source}$, $T_{Target}$)
    Compute $Sim_{LinLike}$ between each concept of $T_{Source}$ and each concept of $T_{Target}$
    For each $c_S \in T_{Source}$
        Compute MC = MappingCandidates($c_S$)
        If TerminologicalMapping($c_S$, MC) then stop
        Else StructuralTechniques($c_S$, MC)

TerminologicalMapping($c_S$, MC)
    Compute $C_{max} \in$ MC
    If TestEquivalent($c_S$, $C_{max}$) then return $c_S$ is-Eq $C_{max}$
        Else If LabelInclusion ($c_S$, $C_{max}$) then return $c_S$ is-A $C_{max}$
            Else If RelativeSimilarity($C_{max}$, MC) then return $c_S$ is-a Father($C_{max}$)
```

Fig. 1. The alignment process

When terminological techniques are not sufficient, complementary techniques are used to provide additional mappings. In this regard, we propose the following complementary techniques:

- Take advantage of structural features in the target taxonomy,
- Exploit the hierarchical structure of additional background knowledge,
- Deduce new mappings from prior defined ones.

We do not detail these techniques as they were not applied in the experiments we report below. The interested reader can refer to [3, 2] for a detailed presentation of these techniques.

2.3 Adaptations made for the Evaluation

The TaxoMap prototype is written in Java and takes as input two taxonomies whose format is compliant with that of TaxoMap[1]. TaxoMap outputs a file per technique used (equivalence, inclusion, proximity, etc.). We have developed two conversion modules to link our application to the API Alignment. They are:

- *OWL2TM:* It parses ontologies in OWL or RDF and generates taxonomies in TaxoMap format where only labels and subclass relationships are taken into account. This module generates a table of correspondences which stores URIs with their associated labels.
- *TM2Align:* It generates an output from TaxoMap internal structure, having the alignment output format (RDF/XML) which regroups all the information stored in TaxoMap's output.

---

[1] A text format in which concepts are presented in their context (below their super concept) and defined by their label and their level in the hierarchy.

2.4 Link to the Set of Provided Alignments

The alignments produced by TaxoMap are available at the following URLs:
http://www.lri.fr/~haifa/benchmarks/
http://www.lri.fr/~haifa/anatomy/

3 Results

3.1 Benchmark Tests

Tests 101-104 Since our algorithm only considers labels and subclass relations and only provides mappings for concepts, the recall is low even for the reference alignment (#101-#101).

Test	Precision	Recall	F-Measure
101-101	1.00	0.34	0.5
102-101	NaN	NaN	NaN
103-101	1.00	0.34	0.5
104-101	1.00	0.34	0.5

Table 1. Results from 101 to 104

Tests 201-266 Given below are some results concerning alterations on labels and hierarchies. Alterations on properties, instances and comments have no effect on our algorithm since it ignores these descriptions.

Our algorithm relies on labels of concepts, so tests where labels were suppressed or replaced by random string or translated[2] have produced no mapping.

In most remaining tests, the precision was very high. This is encouraging since our main objective is to increase ontology mapping precision.

We considered the reference ontology #101 as the Target ontology to fit to our initial hypotheses (see section 2.1). This restriction distorts the interpretation of the results. In fact, our alignment is oriented and affects the results. We generate alignments from concepts of #X to concepts of #101 (i.e (ci_X *isEq* cj_{101}), (cn_X *isEq* cj_{101}), etc.). However reference alignments contain #101 as the first ontology to align. So to be comparable with these alignments, we changed the order of the results (cj_{101} *isEq* ci_X), (cj_{101} *isEq* cn_X), etc.). This leads to generate 1:n relations and explains the cases where the precision has deteriorated slightly.

---

[2] For official runs, we only take into account English labels. However, when this restriction is not applied, we obtain few alignments because there could exist some common roots between labels even if the language is different (it is the case between French and English).

Tests 301-304 Tests with real ontologies presented coherent results with the rest of tests. The precision is high, but the recall is low because alignments concern also properties which are not taken into account by our algorithm.

Test	Precision	Recall	F-Measure
301-101	1.00	0.21	0.35
302-101	1.00	0.21	0.35
303-101	0.80	0.24	0.37
304-101	0.92	0.34	0.5

Table 2. Results from 301 to 304

3.2 Anatomy Test

We considered *nci_anatomy* as the target taxonomy as is it well structured and larger than *mouse_anatomy*. With respect to the chosen anatomy, we were only able to apply the terminological techniques due to the large size of the taxonomies.

In order to remedy the above problem, we attempted to reduce the size of taxonomies by performing a filtering phase consisting of detecting disjoint domains between the taxonomies and deleting sub-hierarchies that had no match in the terminological mapping phase. The filtering phase did not detect disjoint subcategories, common concepts between the two taxonomies are homogeneously distributed in all sub-hierarchies of T_{Target}. The two taxonomies seem to be homogeneous and no subsequent part can be deleted without deteriorating the performances of the system.

The reference alignment contains only equivalence correspondences between concepts of the ontologies. In order to have a little insight on the relevance of the extracted relations, we transformed *is-A* relations into equivalence ones with a confidence value of 0.5. This transformation seems adequate for some cases, where labels are slightly different and where the label of a concept c_S in *mouse_anatomy* is included in the label of a concept c_T in *nci_anatomy*. For example, the concepts *abducens VI nerve* and *abducens nerve* are identified as not equivalent but considered as being related by an is-A relation (*abducens VI nerve is-A abducens nerve*). When we transform this relation into an equivalence one (*abducens VI nerve isEq abducens nerve*), the alignment remains good.

This, however, is not always possible and can lead to misinterpretation of our results. It is especially serious when the subsumption relation is proposed between c_S and a father of c_T. For example, the relative similarity technique discovers a similarity between *cervicothoracic ganglion* and *thoracic ganglion*, and then infers subsumption relation between *cervicothoracic ganglion* and *sympathetic ganglion*[3] (*cervicothoracic ganglion is-A sympathetic ganglion*). This re-

---

[3] *sympathetic ganglion* is the father of *thoracic ganglion* in T_{Target}.

lation seems to be adequate, but its transformation into an equivalence relation even with a low confidence seems to be nonsense.

So, for us the absence of is-A relations in the reference mappings is unsatisfactory. The labels of concepts are almost complex terms which are composed of terms of their super-concepts. Therefore, terminological techniques seem to fit to this sort of ontologies, particularly the label inclusion technique which find subsumption relaions.

We applied a strict threshold for the two runs we submitted[4]. Equivalence relations are found between concepts when labels are strictly the same. If there is some variation, the concepts are considered to be linked by a subsumption relation. We identified 941 equivalence mappings and 879 subsumption correspondences, but almost 900 concepts were left unmapped. Among these 900 unmapped concepts, 581 have a label which includes labels from other concepts and so can be candidates for an is-A mapping with a more relaxed threshold.

Hence, our first effort will be to solve the problem of large-scale ontologies in order to test the whole proposed techniques.

4 General Comments

4.1 Strengths and Weaknesses of the Results

TaxoMap was designed for semantic resources with poor concept descriptions. The benchmark tests were not adequate for testing the robustness of the terminological mappings as the values of the recall are influenced by property mappings and the reference alignment is oriented.

The anatomy test proposes interesting taxonomies. The generated mappings seem interesting. However, the results will, again, depend on the reference alignment. As subsumption relations are not evaluated, we should have relaxed the similarity threshold.

4.2 Ways to Improve the Proposed System

Our algorithm does not take into account properties and instances and only generates mappings between concepts, all of which seems to handicap our system. However, we believe that properties and instances in ontologies can be valuable when the ontologies are constructed rigorously. But if we consider the near future of semantic web where anyone can place his ontology on the web, it can lead to numerous light ontologies with only labels and subsumption relations.

The following improvements can be made to obtain better results:

- Take into account multi-label concepts in the terminological mappings.
- Exploit comments, if present, to enforce the confidence on the extracted mapping classes.

The main issue remaining concerns the way to process large-scale ontologies. This needs techniques to split ontologies and aggregate the returned results.

---

[4] The first run returns only equivalent mappings, the second returns equivalent and subsumption correspondences.

4.3 Comments on the OAEI measures

The precision and recall measures are necessary to have a general idea of the alignment performance. Nonetheless, they need to be adapted to the context of alignment where:

- All mappings have different weights. In fact, some mappings are more difficult to find and this difficulty should be quantified.
- Subsumption relations are probably less interesting than equivalence but are important in certain contexts (for instance query expansion).
- There should be a difference between a false alignment and an approximate one. The recall and precision measures take into consideration binary relevance (a mapping is considered as correct or not). They can be adapted in order to take into account graduate relevance (0 and 1 remain as "totally irrelevant" and "totally relevant" respectively, and intermediate values are assumed with various degrees of "partial relevance").

5 Conclusion

This paper reports our first participation in OAEI campaign. Our algorithm proposes an oriented mapping between concepts. This specificity leads to a misinterpretation of our results. The assessments of alignments consider only equivalent relations so we did not have an insight on the relevance of subclass relations. Despite these difficulties, our participation in the campaign opens perspectives to ameliorate our system.

References

[1] Lin, D. : An Information-Theoretic Definition of Similarity. ICML. Madison. (1998) p.296–304

[2] Reynaud, C., Safar, B. Exploiting WordNet as Background Knowledge. The ISWC'07 Ontology Matching (OM-07). (2007) to-appear

[3] Reynaud, C., Safar, B. When usual structural alignment techniques don't apply. The ISWC'06 workshop on Ontology matching (OM-06). Athens. (2006)

X-SOM Results for OAEI 2007

Carlo Curino, Giorgio Orsi, and Letizia Tanca
{curino,orsi,tanca}@elet.polimi.it

Politecnico di Milano
Dipartimento di Elettronica e Informazione
Via Ponzio 34/5, Milano (Italy)

Abstract. This paper summarizes the results of the X-SOM tool in the OAEI 2007 campaign. X-SOM is an extensible ontology mapper that combines various matching algorithms by means of a feed-forward neural network. X-SOM exploits logical reasoning and local heuristics to improve the quality of mappings while guaranteeing their consistency.

1 Presentation of the system

Nowadays, the spreading of data intensive and community-centered web-applications has multiplied the number of available datasources accessible through the Internet. In order to effectively query and integrate this information, a shared formalism should be used, at least as a means to mediate the access to datasources. In many situations, ontologies [8] have demonstrated , to be a suitable formalism for evenly representing the content of heterogeneous datasources [15], with a well-defined semantics. In principle, it is possible to extract an ontology from a datasource, and then integrate its information content with that of other datasources, by relating their respective ontologies.

Ontology mapping is then defined as *the process of bringing two or more ontologies into mutual agreement, by relating their similar concepts and roles by means of alignment relationships*. Generally speaking, the mapping process aims at providing a *unified, consistent and coherent view over multiple conceptualizations of one or more domains of interest.*

In this paper, we briefly describe our ontology mapping tool, X-SOM [5] (eXtensible Smart Ontology Mapper), summarizing the performance obtained against the OAEI 2007 test cases.

The architecture of the X-SOM Ontology Mapper is composed by three subsystems: *Matching*, *Mapping* and *Inconsistency Resolution*.

The Matching Subsystem is constituted by an extensible set of matching modules, each of which implements a matching technique that may be invoked by the mapper according to a configurable *matching strategy*; this strategy defines also the way the matching values are combined. Each module receives as input two ontologies and returns a set of matchings, along with a similarity degree, between *homogeneous* resources (i.e., concepts with concepts, roles with roles and individuals with individuals); the produced structure is called *similarity map*.

All similarity maps produced by the Matching Subsystem are collected by the Mapping Subsystem; the various proposals are then combined by means of a feed-forward neural network in order to produce an aggregated similarity degree, starting from the single similarities computed by each module of the Matching Subsystem. Given these aggregate matching values, the Mapping Subsystem computes a set of *candidate mappings* by applying, to the set of matchings, a pair of configurable threshold values. The first threshold is called *discard threshold*; the matchings with a similarity degree lower than it are discarded a-priori. The second threshold is called *accept threshold*, and the matchings with a similarity degree greater than it are accepted as candidate mappings. The remaining matchings, whose similarity is between the two thresholds, are considered as uncertain and manually evaluated by the user.

Mapping two ontologies might produce inconsistencies [12]; for this reason, the set of candidate mappings computed by the Mapping Subsystem is handed to the Inconsistency Resolution Subsystem, responsible for guaranteeing mappings consistency. Moreover, the X-SOM consistency-checking process can be instructed to preserve the semantics of the original ontologies, in terms of concept definitions and relationships among them. The so-obtained mappings capture the *consensual knowledge* about the domain, i.e., that information which represents an added value for the system, without changing the semantics of the input ontologies and, in turn, without incurring in the need to adapt the applications built upon them.

Ontologies are often published on the Web and not accessible for modifications. For this reason, and to preserve the original representations, X-SOM mappings are stored in a separate ontology called *mapping ontology*. This ontology acts as a “bridge” between the mapped ontologies and can be used to access the global model constituted by source ontologies connected through the mappings. If needed, it is possible to store in the bridge ontology also the concept definitions needed to disambiguate some terms or to solve particular inconsistencies.

X-SOM generates subsumption and equivalence mappings between pairs of resources; they are expressed by means of RDFS and OWLS constructs, in order to maintain the highest interoperability of mapping definitions.

1.1 State, purpose, general statement

X-SOM has been designed to automatically discover useful relationships among ontological representations with the purpose of enabling ontology-based data integration and tailoring [6]. The theoretical framework used in this work is that of DL ontologies; however, the X-SOM approach is very flexible and we believe that it is possible to extend it to other ontology languages, and even to other data models such as XML and the relational model.

X-SOM is part of a wider research project named Context-ADDICT (Context-Aware Data Design, Integration, Customization and Tailoring) [1], which aims at the definition of a complete framework able to support mobile users through the dynamic hooking and integration of new, heterogeneous information sources, until a suitable, contextualized portion of the available data is delivered on their devices, in a structured and offhanded way. The whole process is widely based on ontological representations of both the application domain and datasources; this naturally leads to an ontology mapping process that should be as much automatic as possible.

1.2 Specific techniques used

In this section we describe, in more detail, the three subsystems that constitute the X-SOM architecture.
The Matching Subsystem has been designed to be extensible, to allow easy integration of future matching modules. Since this architecture makes experimenting new modules very easy, X-SOM can also be used as a framework for evaluating matching techniques. X-SOM's matching modules can be roughly classified into three families:

- *language-based:* The modules belonging to this family of algorithms compare resources by analyzing their names, labels and comments. They consider both the lexical and linguistic features as terms of comparison. The lexical modules currently implemented are: the *Jaro module*, based on Jaro String Similarity [4] and the *Levenshtein module* based on the Levenshtein string distance; To exploit linguistic similarities, we implemented a *WordNet module* that uses the WordNet [13] thesaurus, computing some distance measures like the *Leacock-Chodorow* [11].
- *structure-based:* These modules are used to compare the structures of the resources' neighborhoods. In X-SOM, we have implemented a modified version of the GMO (Graph Matching for Ontologies) algorithm [9], used to find structural similarity in ontological representations. Since the GMO algorithm is quite expensive in terms of required computational resources, we implemented a bounded-path matcher called *Walk* that reaches lower performance while requiring less resources.
- *semantics-based:* The modules belonging to this family implement algorithms that use background, contextual and prior knowledge to compute the similarity degree between two resources. At the moment, only a Google-based algorithm, described in [3], is implemented.

The Mapping Subsystem receives as input the set of similarity maps computed by the modules of the Matching Subsystem, and produces a set of candidate mappings to be verified by the Inconsistency Resolution Subsystem.
The most challenging issue is how to aggregate all the contributions coming from the various matching modules. In our setting, the problem has been modeled as the estimation of an optimal aggregation function $y = W(\mathbf{X})$ where each component $x_i \in \mathbf{X}$ is the matching degree given by the i^{th} module of the schedule with respect to a pair of resources, and y is the computed aggregate similarity.
The Mapping Subsystem is as extensible as the Matching Subsystem previously described; it allows to add new aggregation algorithms to X-SOM, by implementing a simple interface.
At the current development state of the prototype, the most effective aggregation algorithm implemented uses a three-layer, feed-forward neural network. The learning algorithm used is a standard back-propagation algorithm with cross-validation; the values for the moment and the learning-rate have been set after empirical evaluation (i.e., over 50.000 runs of the tool). Notice that the task of determining a good aggregation function is, in general, very complicated, since it is not possible to imagine a unique aggregation function that is suitable for every possible alignment situation. Even by supposing a trivial situation where the W function is approximated with a linear function (e.g., a weighted mean), the process of determining the weight of each module implies that the

user knows in advance how reliable the various techniques are.

Another interesting aspect is how to build a suitable training set for the neural network. In X-SOM, the training set is generated from a manually-aligned pair of ontologies called *reference alignment*; correct mappings generate a sample with desiderata equal to 1.0, while the others will be set to zero. Moreover, a *cleaning process* removes: duplicate samples (i.e., similar inputs and same desiderata), conflicting samples (i.e., same inputs but contradictory desiderata) and linearly dependent samples. In the situation of conflicting samples, only the ones with desiderata equal to 1.0 are kept and the reason resides in the way the desiderata are obtained. To determine if, to a set of inputs, should correspond a positive outcome, the trainer looks at the reference alignment. If the given set of inputs is generated by a correct alignment, the outcome is positive (i.e., 1.0) else, it is set to zero. When two conflicting samples are found, the trainer assumes that the one with positive outcome is correct, while the other is discarded.

It is possible that, in certain situations, a module be not able to produce a similarity degree for a given pair of resources; in this case, the value is approximated by means of an average over the similarity degrees generated by the other modules belonging to the same family.

Once the neural network has produced the aggregate similarity values, X-SOM filters them by means of two configurable thresholds: *accept* and *discard*. These thresholds also determine the level of automation of the tool, called *behavior*, which can be: *Fully-automatic*, *Conservative* or *Human-intensive*. When X-SOM acts with one of the last two behaviors (i.e., supervised behaviors), it is possible to involve the user in deciding what matchings should be accepted. In particular, with the conservative behavior, all the mapping proposals with a similarity degree between the discard and accept thresholds are submitted to the user to be evaluated. When the user does not agree with a X-SOM proposal about a pair of resources, the network trainer performs additional training steps until the result of the network agrees with the user, thus allowing fine-tuning of the network's biases. The human-intensive behavior is very similar to the previous one, only it does not discard any mapping a-priori, leaving to the user the freedom to explore all the mappings with a similarity degree under the accept threshold.

The Inconsistency Resolution Subsystem takes as input the candidate mappings from the Mapping Subsystem and produces a set of mappings, in which at least all the logical inconsistencies have been solved. Since the input ontologies are supposed to be consistent, consistency resolution is reduced to identifying those mappings that introduce a contradiction into the final model. This problem is faced in X-SOM at two different levels: *consistency check* and what we have called *semantic coherence check*. *Consistency check* locates those mappings that introduce a logical contradiction in the original ontologies. X-SOM uses an extended tableau algorithm to identify the set of mappings responsible for inconsistency and uses a set of heuristic rules, based on the similarity degree, in order to remove those mappings; since the removal of mappings leads to a loss of information, the rules try to preserve as much information as possible, in terms of logical axioms. Also the inconsistency resolution policies are affected by the tool behavior described above. When the tool acts in a supervised behavior, the inconsistent mappings are submitted to the user who selects the correct ones; wrong mappings are then removed automatically. When acting with the fully-automatic be-

havior, X-SOM removes the less probable mappings using the heuristic rules.
By *Semantic coherence check* we mean the process of verifying whether there are mappings that introduce into the model a *semantic incoherence* without introducing a logical contradiction into the T-BOX. To better explain what we mean by semantic coherence, let us introduce the notion of *local entailment*: an entailment A $\sqsubseteq$ B, in the global model, is said to be *local to an ontology O* if it involves only resources of *O*. By semantic incoherence we mean the situation in which the alignment relationships enable one or more local entailments that were not enabled within the original ontologies. This, in general, is a desirable behavior for systems that exploit ontologies; however, in certain situations, it is possible to introduce an incoherent assertion without introducing a logical contradiction into the model.
A simple example of semantic incoherence is the emergence of a cycle of subsumptions after a mapping process, which leads to a collapse of the involved concepts into a unique concept. The collapse of two concepts – which were only in a subclass relationship in the original ontologies – changes the semantics of the representation: for this reason, our algorithm removes the mappings responsible for that behavior. We consider a semantic incoherence as a possible symptom of an inconsistency; since we are interested in developing a high precision ontology mapper, we currently adopt a conservative approach that does not allow any change in the semantics of the original ontologies. The main drawback of this approach is that it is possible to lose some useful inferences on the global model.

1.3 Adaptations made for the evaluation

In order to comply with the test-cases proposed in this contest, we made two main adaptations:

- *External resources:* Using the original configuration of X-SOM, external resources (e.g., FOAF definitions) are imported and used in the mapping process. As a result, also the mappings between pairs of external resources are included in the alignment ontology produced by X-SOM. To avoid a wrong computation of performance measures, we artificially removed this kind of mappings from the output of the tool when they were not part of the reference alignment.
- *Properties comparison:* In some reference alignments, datatype properties are compared and aligned with object properties; since this kind of mapping is normally forbidden in X-SOM, we modified the matching algorithms in order to allow this kind of matching.

1.4 Link to the system and parameters file

X-SOM is an open-source project, since it also relies on existing implementations of known matching algorithms. To obtain a working copy of the X-SOM prototype, along with the source code, please send an email to `orsi@elet.polimi.it`.

1.5 Link to the set of provided alignments

http://home.dei.polimi.it/orsi/xsom-oaei07.zip

2 Results

The tests have been made with a configuration that includes:

- The WordNet-based module.
- The GMO structural module without feeding.
- The Jaro syntactical module.

The best results are obtained by aggregating the results by means of the neural network that has been trained on the `animals.owl` ontology available at the I^{3}CON Initiative website[1]. X-SOM is implemented in Java and relies on the JENA API. The behavior is fully-automatic with an accept threshold set to 60% of similarity and consistency checking turned on. The presence of consistency checking accounts for the somewhat high execution times. In this section we report the performance of X-SOM for the various OAEI tracks. The tests have been made using a Pentium IV 2.1 GHz with 2 GB of RAM. The JVM has been set with a minimum and maximum heap space of 64 MB and 2 GB respectively. The WordNet-based module relies on the JWNL API version 1.3.

2.1 Benchmark

The test cases belonging to this track can be divided into five categories:

Basics (101-104) This family analyzes the ability of a matcher to make simple alignments and to be robust to variations in the OWL dialect. On these, very simple, tests X-SOM obtains an average precision of 99% and an average recall of 98.6%.

Linguistics (201-210) The test cases belonging to this family manipulate resources' names, comments and labels in order to stress the performance of syntactic and lexical matchers. X-SOM performs quite well thanks to the Jaro and WordNet modules; some problems come out when the tool deals with test case 204 since we are not able to recognize acronyms. We have planned to add a pre-processing step in these modules in order to recognize common naming conventions thus normalizing the name of resources also considering possible compound words. This normalization will be kept internal to each module, avoiding any modification in the input ontologies. In this section, X-SOM reaches an average precision of 81.6% and an average recall of 75.4%.

Structure (221-247) These test cases stress the capabilities of the various matching algorithms of finding similar resources in ontologies with different structure. X-SOM performs very well since it reaches an average precision and recall of 99%.

Systematic (248-266) This family combines the previous techniques by removing systematically the structure or by randomizing the names of the resources. This is the hardest part of the benchmark track since X-SOM obtains useful information from the GMO module only. X-SOM obtains an average precision and recall of 26%. The hardest tests cases are 262 and 265 where the result of the matching process is empty because also the GMO module has not enough information to find the similarities.

---

[1] `http://www.atl.external.lmco.com/projects/ontology/i3con.html`

Real ontologies (301-304) In these test cases the reference ontology is aligned with real world ontologies describing the same bibliographic domain. These are the most informative tests since these ontologies includes a set of design choices that make the alignment task quite hard. X-SOM reaches an average precision of 94% and an average recall of 67%. The hardest test for X-SOM is 301 since contains many compound words with different naming conventions.

2.2 Anatomy and Food

These tracks have been the most challenging. The problem resides in their dimension that is too big for the current version of X-SOM. In order to perform the alignment, we needed to partition both ontologies using the partition algorithm implemented in the SWOOP [10] ontology editing and debugging framework. However, this procedure was not enough to reduce the ontologies to a manageable dimension; the partition algorithm used on the NCI thesaurus produced a partition with over 3200 classes that cannot be analyzed by our GMO module thus, for this reason, only lexical modules have been used.

2.3 Directory

The main problem with this track is the modularization of the test cases. The small test-cases are too small to exploit the full power of the GMO module, while the comprehensive ones are too big and exhaust the JVM heap space if not partitioned. These limitations lead the GMO module to return poor answers that, in turn, affect the final results.

3 General comments

X-SOM seems to perform quite well on the OAEI test cases; however, the main problems are represented by the aggregation function and large ontologies processing.

We recall from Section 1.2 that the X-SOM neural network is trained only once and then used for all the proposed tracks. Previous tests, performed using ontology pairs describing different domains, have shown that the learned aggregation function is substantially independent of the domain, but strongly dependent on the ontology design technique [2]. This means that, if the neural network is trained on a pair of ontologies with a rich structure, the learning algorithm will probably keep the results of the structural modules in high consideration since, in general, they are helpful for finding the correct alignments. If the same function is then used in a mapping task concerning ontologies that lack of structure, the poor results generated by the structural modules will affect the final results, lowering the whole performance of the tool. An even better performance would be achieved if the modules' schedule could be changed among different tracks.

The second problem is represented by very large ontologies that require too much memory to be processed with the X-SOM approach. A solution to this problem is the modularization of ontologies and the subsequent mapping of ontology chunks modeling the same portion of the application domain. At the moment we are not planning to address this problem within X-SOM but, for instance, we are considering to resort to modularization algorithms such as that implemented in SWOOP.

3.1 Discussions on the way to improve the proposed system

We are planning to introduce new modules able to extract and reuse the consensual knowledge that emerges in collaborative and social web-applications, in order to disambiguate some mapping situations that generally need user intervention. We are currently exploring other machine-learning techniques for the matchings combination task [7], in particular *white-box* techniques like *decision-tree learning*. At the moment, the matching strategy is determined by the user; we aim at introducing techniques to suggest a suitable strategy using a-priori analysis of the input ontologies [14], and make it adaptive during the matching process. Moreover, we are developing a clustered version of X-SOM, called *k*X-SOM, which exploits the intrinsic parallelism contained into the matching algorithms.

3.2 Comments on the OAEI 2007 procedure

The OAEI contest is well conceived, and has helped the improvement of the modules implemented in X-SOM. In our opinion, however, OAEI organizers should allow reparametrization of the tool, in order to better configure the prototype for each task.

3.3 Comments on the OAEI 2007 test cases

In our opinion, the benchmark track contains too many test cases in the systematic family (248-266). These tests reflect too “unreal” ontology design situations, while it will be more interesting to add test cases that include also complex alignments (i.e., mappings between complex definitions of concepts). In addition, it would be useful to introduce one or more test cases to analyze if different matching algorithms are able to avoid those mappings that produce contradictions in the ontological model.

3.4 Comments on the OAEI 2007 measures

Traditional precision and recall measures along with their combinations are the most suitable measures to evaluate the performance of ontology mapping tools. Moreover, it will be helpful to consider also the mappings among external resources.

3.5 Acknowledgment

We thank all the people and students that worked on X-SOM, in particular Alessandro Dalvit who executed the OAEI tests.

4 Conclusion

Ontology alignment and integration represents crucial aspects of the effort that computer science community is making to achieve systems interoperability. The OAEI contest represents a valuable opportunity to gather all the approaches and improve the current matching algorithms. Participating in the OAEI allowed the identification of the weaknesses of the X-SOM approach, in particular, problems have arisen with large ontologies and during the aggregation phase. It is our intention to address these weaknesses in order to improve our approach for OAEI 2008.

References

1. C. Bolchini, C. Curino, F. A. Schreiber, and L. Tanca. Context integration for mobile data tailoring. In *Proc. of the 7th Int. Conf. on Mobile Data Management (MDM'06)*. IEEE Computer Society Washington, DC, USA, 2006.
2. G. Orsi C. Curino and L. Tanca. X-som: Ontology mapping and inconsistency resolution. In *Proc. of the 4th European Semantic Web Conference (ESWC'07) - Poster Session*, Innsbruck, Austria, June 2007.
3. P. Cimiano and S. Staab. Learning by Googling. *ACM SIGKDD Explorations Newsletter*, 6(2):24–33, 2004.
4. W. W. Cohen, P. Ravikumar, and S. E. Fienberg. A comparison of string distance metrics for name-matching tasks. *Proc. of the 1st Int. Workshop on Information on the Web (IIWEB'03)*, pages 73–78, 2003.
5. C. Curino, G. Orsi, and L. Tanca. X-SOM: A Flexible Ontology Mapper. In *Proc. 1st Intl. Workshop on Semantic Web Architectures For Enterprises*, September 2007.
6. C. Curino, E. Quintarelli, and L. Tanca. Ontology-based information tailoring. In *Proc. 2nd IEEE Int. Workshop on Database Interoperability (InterDB 2006)*, pages 5–5, April 2006.
7. M. Ehrig, S. Staab, and Y. Sure. Bootstrapping ontology alignment methods with apfel. pages 1148–1149, 2005.
8. A. Gómez-Pérez, M. Fernández-López, and O. Corcho. *Ontological Engineering*. Springer, 2004.
9. W. Hu, N. Jian, Y. Qu, and Y. Wang. Gmo: A graph matching for ontologies. *K-Cap 2005 Workshop on Integrating Ontologies*, pages 43–50, 2005.
10. A. Kalyanpur, B. Parsia, E. Sirin, B. Cuenca-Grau, and J. Hendler. Swoop: A web ontology editing browser. *Journal of Web Semantics*, 2005.
11. C. Leacock, M. Chodorow, and G. A. Miller. Using corpus statistics and wordnet relations for sense identification. *Computational Linguistics*, 24(1):147–165, 1998.
12. J. Madhavan, P. A. Bernstein, P. Domingos, and A. Y. Halevy. Representing and reasoning about mappings between domain models. *Proc. of the 18th Nat. Conf. on AI*, 2002.
13. G. Miller. *WordNet*. MIT Press, 1998.
14. M. Mochol, A. Jentzsch, and J. Euzenat. Applying an analytic method for matching approach selection. In *Proc. of The 1st Workshop on Ontology Matching (OM2006)*.
15. H. Wache, T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Hübner. Ontology-based integration of information: a survey of existing approaches. In *Proc. of the Int. Workshop on Ontologies and Information Sharing*, pages 108–117, 2001.

Appendix: Raw results

#	Name	Prec.	Rec.	Time (sec)
101	Reference alignment	1.00	0.98	59.206
102	Irrelevant ontology	NaN	NaN	48.392
103	Language generalization	1.00	0.98	60.100
104	Language restriction	0.97	1.00	62.408
201	No names	0.81	0.81	77.133
202	No names, no comments	0.82	0.82	69.696
203	No comments	1.00	0.97	65.871
204	Naming conventions	0.99	0.69	84.651
205	Synonyms	0.72	0.71	75.767
206	Translation	0.74	0.73	71.233
207		0.69	0.68	92.888
208		0.99	0.75	66.384
209		0.70	0.69	71.624
210		0.70	0.69	70.693
221	No specialisation	1.00	0.99	61.508
222	Flatenned hierarchy	1.00	0.98	74.516
223	Expanded hierarchy	1.00	0.98	91.145
224	No instance	1.00	0.98	58.956
225	No restrictions	1.00	0.98	45.680
228	No properties	1.00	1.00	17.235
230	Flattened classes	0.99	0.97	58.191
231	Expanded classes	1.00	0.97	59.549
232		1.00	0.99	54.451
233		1.00	1.00	16.366
236		1.00	1.00	15.356
237		1.00	0.98	57.156
238		1.00	0.99	67.714
239		0.97	1.00	16.436
240		0.97	1.00	22.687
241		1.00	1.00	14.682
246		0.97	1.00	14.521
247		0.97	1.00	21.739
248		0.75	0.75	71.264
249		0.60	0.60	67.246
250		0.18	0.18	20.118
251		0.45	0.45	72.949
252		0.49	0.49	98.557
253		0.54	0.54	68.091
254		0.03	0.03	19.909
257		0.12	0.12	19.319
258		0.32	0.32	78.671
259		0.33	0.33	100.424
260		0.03	0.03	19.963
261		0.03	0.03	34.414
262		0.00	0.00	19.336
265		0.00	0.00	18.999
266		0.03	0.03	32.471
301	Real: BibTeX/MIT	0.91	0.49	42.561
302	Real: BibTeX/UMBC	1.00	0.58	23.537
303	Real: Karlsruhe	0.96	0.73	70.359
304	Real: INRIA	0.96	0.87	48.097

A Survey and Categorization of Ontology-Matching Cases

Zharko Aleksovski[1,2], Willem Robert van Hage[2,3], and Antoine Isaac[2]

[1] Philips Research, Eindhoven,
[2] Vrije Universiteit, Amsterdam
[3] TNO Science & Industry, Delft
{zharko,wrvhage,aisaac}@few.vu.nl

Abstract. Methodologies to find and evaluate solutions for ontology matching should be centered on the practical problems to be solved. In this paper we look at matching from the perspective of a practitioner in search of matching techniques or tools. We survey actual matching use cases, and derive general categories from these. We then discuss the value of existing techniques for these categories.

1 Introduction

For an application developer to know which ontology matching system best suits his needs, application requirements have to be taken into account. Recently, innovative work carried out in the KnowledgeWeb network of excellence [1] has analyzed the requirements of usage scenarios and proposed a case-based recommendation method: a given application is profiled along different dimensions – input, usage, etc. This profile is then compared to a characterization of different matching tools, to determine which tool best fits the case that is considered. This method is based on a characterization of existing matching solutions obtained by the carefully crafted benchmark test of the Ontology Alignment Evaluation Initiative[4] (OAEI). However, the focus of the OAEI has been mainly on comparing techniques for research. As a result, the categories that are used are not straightforwardly linked to real-world cases.

To give answers for application developers it is necessary to build better links between application specifications and matching systems. A possible way to do so is to position each new matching case with respect to an matching-oriented categorization of applications built from the characterization of these cases in terms of matching requirements and performances of different techniques. [1] actually initiates such an effort: the authors gather use cases, point at the typical operations (data transformation, ontology merging, etc.) and elicit some matching quality requirements for them (correctness, completeness). Yet, the cases considered there are abstract scenarios. In this paper, we intend to provide a better application grounding for case-based recommendation by turning to lessons learned from concrete cases. Our contribution consists in a categorization-oriented survey of existing ontology matching cases trying to give answers to the following questions: (i) what are the different kinds of cases in which

---

[4] `http://oaei.ontologymatching.org`

ontology matching has been deployed so far? (ii) Can we observe common aspects, leading to a classification of these cases? (iii) Are there matching techniques that have been observed to perform better on specific sets of cases?

We first analyse existing documented ontology-matching use cases, highlighting their main requirements (Section 2). We notice the emergence of four different categories of use cases, depending on their purpose, the data they deal with, and their priorities regarding matching qualities: data migration, question answering, serendipity in browsing, and unified view over collections (Section 3). Section 4 extends these considerations towards the realm of ontology-matching tools, by showing how specific matching techniques perform better for specific classes. Section 5 concludes the paper.

2 Ontology-Matching Cases

The importance of ontology matching was identified through various scenarios which require its solution. Such scenarios, for example described in [2], are: agent communication, emergent semantics, P2P information sharing, personal information delivery, etc. However, these stand for *possible* uses of matching. To carry out the analysis grounding our case categorization and recommendations for matching techniques, we have instead investigated examples of reported matching applications. Such cases had to: (i) provide information on the actual alignment, (ii) report on which techniques can be used to solve the matching problem, and (iii) clearly describe how the correspondences will be used in an application. As a consequence we did not take examples from real-time matching cases like negotiation, where the mapping data is usually generated dynamically, or cases such as the one reported in [3] where the application of the correspondences is not explicitly defined.

The cases we have selected are: *MACS*, *Agricultural Thesauri*, *Renardus*, *STITCH browser*, *WebDewey*, *Intensive Care*, *the High Performance Knowledge Base*, *the Unified Medical Language System*, *Internet Music Schemas* and *Internet Directories*. We detail here only one example, and refer the reader to the companion webpage, `http://www.few.vu.nl/~aisaac/iswc2007/cases`, for the other descriptions.

Intensive Care In this use case the alignment is needed for data-migration purposes. The alignment is directly applied for classification reasoning. Two Amsterdam hospitals, OLVG and AMC, own controlled unstructured vocabularies for registering reasons for patient admission to the intensive care units. The vocabularies are lists of terms and every time a patient enters intensive care she is assigned one of these terms. Correspondences between OLVG and AMC classes are required in order to migrate the patient data from OLVG to the AMC vocabulary. The OLVG vocabulary contains 1,399 terms and AMC 1,460 [4]. To test the performance of automatic matching techniques, a gold standard was created by a medical expert for a sample of 200 OLVG classes. For 37% of OLVG terms in the corpus the expert found no correspondences, for 36% he found correspondences with large lexical similarity between the corresponding terms, and for the remaining 37% he found correspondences with no lexical overlap, which would require the use of some kind of background knowledge. An example of lexical correspondence is Brain tumor to Braintumor, and example of a correspondence that requires background knowledge is Heroin intoxication to Drugs overdosis.

3 Descriptions of Problem Types

Precision versus Recall Three tasks determine the time cost of applying ontology matching: (i) preparation and actual alignment, (ii) assessing correspondences and (iii) adding correspondences. The goal of an automatic system is to reduce the amount of time a user spends on these tasks. Usually, better performance for one task means worse performance for another. Good representation standards and a fast system optimize on the first task, a system with high Precision optimizes on the second task, and a system with high Recall optimizes on the third task. For some cases assessing a single correspondence is time consuming (*e.g.* if concepts are imprecisely defined). For others it is very time consuming to find a missing correspondence (*e.g.* if ontologies are huge). Hence, each case has its own optimal combination of Precision and Recall.

Complexity of representation Each use case requires a different level of knowledge-representation complexity. Some use cases only need basic semantic structures, others need rich ontologies with many different properties and use of logical axioms.

Four categories of use-cases Some of the use cases we presented use the alignment for a similar purpose. Some use the alignment primarily to enrich the descriptions of the data, while others use them primarily to enlarge the data collections. Further, we have noticed that use cases with the same problem type often have similar performance requirements for the ontology matching and use a similar level of knowledge-representation complexity. We propose to categorize the use cases according to the following four types of problems.

Question answering. This problem type is characterized by an emphasis on precise results and the need for highly complex knowledge sources. The use cases we found aim at providing detailed factual information about the data. For example, *"What is the connection between trombose and mortality?"* or *"Who was the president of the United States of America in 1965?"*. As opposed to the real-time question answering described in [1] the goal of these use cases is not to provide a complete list of answers, but one very precise answer (*cf.* [5]). Use cases of this type are the *High-Performance Knowledge Base* case and (partly) the *Unified Medical Language System* case.

Unified view over collections. This problem type is characterized by a balanced need for Precision and Recall and knowledge sources of medium complexity. Examples of such sources include traditional thesauri and thesauri with added relations, such as artist-style links, part-whole, or tool-action. The use cases we found aim at providing unified access to heterogeneous collections that are usually maintained by different authorities. Use cases of this type are the *STITCH browser* case, *MACS*, *Renardus*, the *Agricultural Thesauri* case and, to a lesser extent, *WebDewey*.

Serendipity in browsing. This problem type is characterized by an emphasis on Recall and relatively simple knowledge structures, such as taxonomies. The use cases we found aim at joining taxonomies to enlarge the collection, to provide users with instances they did not know before. Use cases of this type are the *Internet Music Schemes* and the *Internet Directories* cases.

Data migration. Like Question Answering, this problem type emphasizes the need for precise results, not necessarily requiring complex knowledge sources. The use case of this type that we found, the *Intensive Care* case, aims at re-classifying existing instances into classes from a newer schema.

4 Techniques that solve the Mapping Problem in the Use Cases

Observing the cases leaves an open question: which techniques can actually produce the required correspondences? Here, we consider four different types of matching techniques: lexical – based on lexical comparisons of labels and glosses, structural – using the structure of the ontologies, background knowledge – using additional external knowledge, and instance-based – using classified instance data.

Question answering. In these cases the ontologies are usually vast and complex. They have substantial lexical overlap, but the use of different naming conventions and of the same names in different contexts prevents straightforward lexical detection of the correspondences. As reported in the UMLS and HPKB examples, the first problem can be overcome by detecting the patterns used in naming, and then normalizing the names so that lexical techniques can find the correspondences. The second problem, detecting the context, can be solved by taking into account the domain of the ontologies and using their structure. In UMLS, for instance, if a concept Kidney is found in a classification of diseases or is a subconcept of concept Diseases, then it surely refers to problems related to kidneys. Practical cases indicate thus that lexical and structural techniques are good candidate solutions for question answering matching use cases.

Unified view over collections. Here, naming conventions and modeling decisions may differ, but lexical matching solves a large part of the problem. The vocabularies to align can have quite a broad coverage or shared domain and concerns. Often – *e.g.*, when jargon differs – background knowledge is however needed. If different languages are used it even becomes crucial, either in the form of a multilingual "rosetta stone" or of a translation service. Structure-based techniques are generally of much less use: *e.g.*, the semantic link that come in thesauri can be used for meaning disambiguation, but this makes them a secondary source for matching information, not reported to contribute significantly in the examples. Furthermore, as the principles and coherence of the structure can vary from one thesauri to another, these techniques might prove unreliable. Finally, the instance data found in several collections can prove very useful, as the meaning of manipulated concepts is assumed to be ultimately given by the items that are categorized with their help.

Serendipity in browsing. Here, lexical methods are reported to perform poorly, which is caused by two problems: ambiguity in naming concepts and lack of standardized criteria for classifying instance data. Among the artists shared by two portals of the music case, only 38% of the ones classified in the Rock genre in one dataset fall in the Rock category defined by the other. The first problem can be approached by considering the context in which the concepts appear. The second problem can only be solved by matching the instance data. Actually, when matching based on instance data, one has to consider the instances in the subclasses of a given class. As reported in the Internet Directory case, this makes big difference in the performance of the alignment.

Data migration. In the *Intensive care* case, the vocabularies have no structure. Furthermore, there is no substantial, explicitly shared instance data, since the goal is to transfer the instance data itself from one system to the other. This leaves two options for a solution: lexical techniques and background knowledge. These have actually turned out to be sufficient to solve the problem in this specific example.

5 Conclusions and Future Work

Focusing on how to find a good matching method for a given application, we have surveyed a number of real-world ontology matching use cases and proposed a categorization of them in four groups, based on the applicative purpose of the alignment. We have then positioned each category in way compatible with a principled (benchmark-based) profiling of different matching techniques. These can then be selected based on their matching the criteria coming with a given application.

More descriptions of realistic use cases (where the alignment is applied in practice) are clearly needed to complement the analysis presented here, especially to get a better coverage of new innovative scenarios still being investigated now, like semantic web agent communication. It would also be interesting to investigate some cases coming from the database domain, as our survey is quite biased towards alignment cases for description *vocabularies* – as opposed to description *structures*. Such accounts exist [6, 7] but the database research community, as the semantic web one, seems to have put more effort on describing tools and methods than cases [8, 9].

A particular emphasis shall be put on revealing application-specific limitations of matching techniques, as when dealing with specific naming schemes or underspecified structural links. Better consideration of such application-specific constraints is necessary for future benchmarking efforts. This way, the ontology-matching research community could also fully benefit from the surveying effort.

Acknowledgments
We would like to thank Frank van Harmelen, Guus Schreiber, Lourens van der Meij, Stefan Slobach (VU), Hap Kolb (TNO), Merlijn Sevenster, Warner ten Kate (Philips Research), Margherita Sini (FAO), Lori Finch (NAL). Our work is partly funded by NWO, the Netherlands Organisation for Scientific Research (STITCH project) and the Virtual Laboratories for e-Science (VL-e) project.

References

1. Euzenat, J., Ehrig, M., Jentzsch, A., Mochol, M., Shvaiko, P.: Case-based recommendation of matching tools and techniques. KnowledgeWeb Project deliverable D1.2.2.2.1 (2007)
2. Euzenat, C.J.: D2.2.3: State of the art on ontology alignment (2004)
3. Zhang, S., Bodenreider, O.: Aligning representations of anatomy using lexical and structural methods. AMIA Symposium (2003) 753–757
4. Aleksovski, Z., Klein, M., ten Kate, W., van Harmelen, F.: Matching unstructured vocabularies using a background ontology. In: Proc. of EKAW. (2006)
5. Voorhees, E., Tice, D.: The TREC-8 question answering track evaluation. In: Proc. of TREC-8. (1999)
6. Clifton, C., Housman, E., Rosenthal, A.: Experience with a combined approach to attribute-matching across heterogeneous databases. In: IFIP Conference on Data Semantics. (1997)
7. Rosenthal, A., Seligman, L., Renner, S.: From semantic integration to semantics management: case studies and a way forward. Sigmod Records Special Section **33**(4) (2004)
8. Rahm, E., Bernstein, P.A.: A Survey of Approaches to Automatic Schema Matching. VLDB Journal **10**(4) (2001)
9. Doan, A., Halevy, A., Renner, S.: Semantic integration research in the database community: A brief survey. AI Magazine **26**(1) (2005)

Exploiting WordNet as Background Knowledge

Chantal Reynaud, Brigitte Safar

LRI, Université Paris-Sud, Bât. G, INRIA Futurs
Parc Club Orsay-Université - 2-4 rue Jacques Monod, F-91893 Orsay, France
{chantal.reynaud, brigitte.safar}@lri.fr

Abstract. A lot of alignment systems providing mappings between the concepts of two ontologies rely on an additional source, called background knowledge, represented most of the time by a third ontology. The objective is to complement others current matching techniques. In this paper, we present the difficulties encountered when using WordNet as background knowledge and we show how the *TaxoMap* system we implemented can avoid those difficulties.

1 Introduction

In order to identify mappings between the concepts of an ontology, called the source ontology (O_{Src}), with concepts of another one, called the target ontology (O_{Tar}), a lot of recent works use additional descriptions, called background knowledge, represented by a third ontology O_{BK} [1,2,9,4,7,8,10]. The common objective is to complement current matching techniques which may fail in some cases. Some works as [1,2,10] assume that ontology alignment can rely on a unique and predefined ontology that covers a priori all the concepts of the ontologies to be matched. Conversely, other works [9] suppose that there does not exist a priori any suitable ontology. Hence, their idea is to dynamically select online available ontologies. In this paper, we present the difficulties encountered when using WordNet as background knowledge, in particular the misinterpretation problem coming from the different senses of a term, and how the *TaxoMap* system we implemented avoids these difficulties. The solution that we propose aims at limiting the meanings of the terms involved in a match. Experimental results are given and the increase of precision obtained with a limitation of the senses of the terms is shown.

2 Use of WordNet

WordNet is an online lexical resource for English language that groups synonym terms into synsets, each expressing a distinct concept. The term associated with a concept is represented in a lexicalized form without any mark of gender or plural. Synsets are related to each other with terminological relations such as hypernym relations. WordNet can be used for ontology matching in several ways. A first way is to extend the label of a concept with the synonyms in WordNet belonging to the synset of each term contained in the label [3]. Another way is to exploit WordNet restricted to a concept hierarchy only composed of hypernym relations. Given two nodes in this hierarchy, equivalence relation can be inferred if their distance is lower than a given threshold [4]. Other works compute similarity measures [5,6,8]. This last approach leads to relevant results when the application domains of the ontologies to be mapped are very close and targeted. Conversely, results are much less satisfactory when application domains are larger. Indeed a term can belong to several synsets. This leads to misinterpretations and false positive mappings.

We illustrate this problem with results coming from experiments performed with TaxoMap [8] on the taxonomies Russia-A (O_{Tar}) and Russia-B (O_{Src}) loaded from the Ontology Matching site [11]. Both taxonomies describe Russia from different view points but Russia-B contains extra information on the means of transport. Fig.1 represents the WordNet subgraph that is exploited in the search of the terms of O_{Tar} (grey circles in Fig.1) the most similar to the terms in O_{Src} (white circles in Fig.1) that denote vehicles. As no term in O_{Tar} correspond to means of transport, all terms in O_{Src} that refer to vehicles will be related to ‘Berlin’, a term belonging to three synsets respectively corresponding to a city in Germany, a musician and a kind of car.

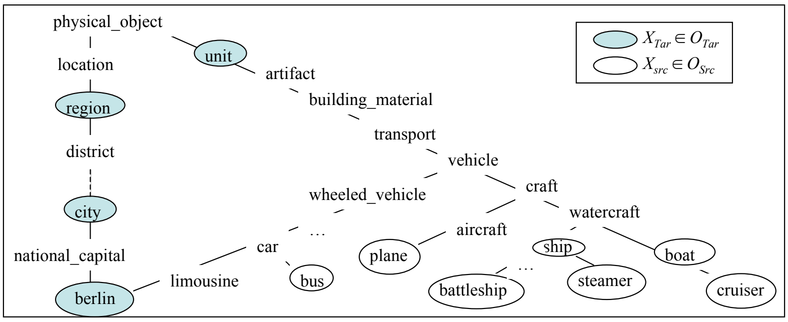


Figure 1. WordNet sub-graph.

To avoid this problem, the *TaxoMap* system relies on a limitation of the senses of the terms in WordNet. It performs a two-step process: a sub-tree is first extracted from WordNet, which only corresponds to the senses assumed to be relevant to the domain of the involved ontologies. Second, mappings are identified in this sub-tree.

2.1 Extraction of a sub-tree relevant to the domain from WordNet

The extraction of a sub-tree starts with a manual phase. If the application domains of the ontologies to be mapped are close and targeted, an expert has to identify the concept, noted $root_A$, that is the most specialized concept in WordNet which generalizes all the concepts of both ontologies. If the target ontology is relative to several distinct application domains**,** the expert has to identify several root nodes in order to cover all the topics. Then, the extraction of the relevant sub-tree needs the search of relations between all the concepts in O_{Tar} and in O_{Src} not yet mapped and $root_A$. Hypernyms of each concept are looked for in the WordNet hierarchy until $root_A$, or one of the WordNet roots, is reached. For example, a search on cantaloupe will result in these two following derivation paths:

Path 1: cantaloupe → sweet melon → melon → gourd → plant → organism → Living
Path 2: cantaloupe → sweet melon → melon → edible fruit → green goods → food

The paths from the invoked terms to the $root_A$ (*food* in the example) will only be selected because they represent the only accurate senses for the application. That way, a sub-graph, denoted T_{WN}, is obtained. It is composed of the union of all the concepts and relations of the selected paths (cf. Fig. 2). The T_{WN}'s root is the concept the most general in the application, $root_A$, leaf nodes correspond to the concepts of the ontologies to be mapped (circles in Fig.2), middle nodes have been extracted from WordNet but possibly belong to one of the two ontologies too.

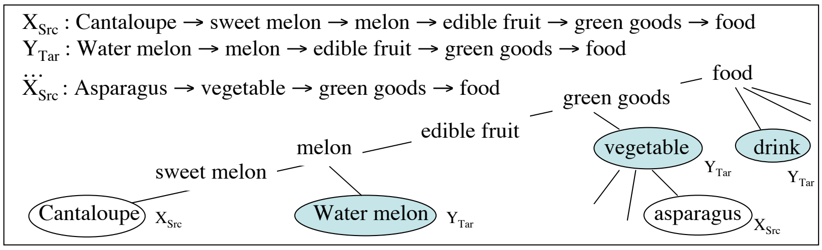


Figure 2. An example of sub-graph T_{WN} where the root is food

In the Russia experiment, the chosen roots (*Location, Living Thing, Structure* and *Body of Water*) covering all the topics of O_{Tar} are not hypernyms of terms in O_{Src} relative to vehicles and no derivation path computed from these terms is retained. Missing terms are preferred over misinterpretations. Recall of matching process will be smaller but precision higher.

2.2 Mappings identification

Identification of relevant mappings consists of discovering, for each concept in O_{Src}, the closest concept in O_{Tar} that is its ancestor and that belongs to its derivation path rooted in $root_A$. So, from the sub-graph in Fig.2, the mapping (asparagus *isA* vegetable) can be derived. That process does not allow the discovery of mappings for cantaloupe because none of its ancestor nodes is a concept in O_{Tar}. All are middle nodes coming from WordNet. However, it should be very interesting to map cantaloupe to Watermelon because they are two specializations of melon, so very close semantically. Such a mapping can be derived using a similarity measure between nodes of T_{WN}. There is evidence that correspondences discovered thanks to such measures cannot be used to derive "semantic" mappings (as *isA* or *Eq* relation) which have a clear semantic and which could then be automatically exploited [8]. But there is also evidence that it would be a great pity not to exploit discovered information. So, we propose to retain such relations which will be labelled '*isClose*' as "potential" mappings for which an expert evaluation will be necessary.

Consequently, the choice in *TaxoMap* is to discover, for each concept X_{Src} in O_{Src} not yet mapped, the concept Y_{Sim} in O_{Tar} that is the most similar according to a similarity score. From that correspondence we derive the potential mapping (X_{Src} *isClose* Y_{Sim}). Then we extract, as we mentioned before, the set of semantic mappings in T_{WN}. If a concept Y_{Sim} is linked to the same concept X_{Src} both in a semantic and in a potential mapping, only the semantic mapping is retained. For example, the concept vegetable in O_{Tar} being the concept the most similar to the concept asparagus in O_{Src}, we derive the potential mapping (asparagus *isClose* vegetable). However, the semantic mapping (asparagus *isA* vegetable) can also be derived. We will then consider that this semantic mapping will be the only retained one. On the opposite, as no semantic mapping has been derived for the concept cantaloupe, the potential mapping (cantaloupe *isClose* Watermelon) will be retained.

3 Experiments

Different experiments have been performed in the micro-biology domain and on taxonomies used for tests in the Ontology Matching community [11]. All these experiments showed that if the application domain is too large, we can not use a unique root. Indeed, in that case, the concept in WordNet which is an hypernym of all the concepts to be mapped, is too general (*entity)* and T_{WN} is too big. It is composed of all the nodes in the WordNet hierarchy without any restriction. Several meanings are mixed. This leads to the derivation of non relevant mappings.

We give results obtained with the Russia taxonomies. As we see in Tab. 1, with a single root (Entity) our technique has derived 61 *isA* and 15 *isClose* mappings among 162 terms in Russia-B not yet mapped by others techniques (370 terms were to be mapped at the beginning). As no reference mappings were delivered, the results have been evaluated manually. Only 29 out of 61 *isA* mappings and 8 out of 15 *isClose* mappings were correct. In particular, all mappings relative to vehicle are false (cf. FIG.1). A significant increase of the precision of the found mappings has been obtained when several roots have been specified. In that case, several distinct sub-trees are built in the same time, one per sub-domain. Four roots have been identified: Location, Living Thing, Structure and Body of Water. Then 35 *isA* and 11 *isClose* mappings have been derived. 29 out of 35 *isA* mappings and 9 out of 11 *isClose* mappings were correct. In particular, all geographic mappings relative to towns, countries, regions and rivers were relevant. Even though the same number of correct *isA* mappings (29) appears as the results of the two experiments, these mappings are not all the same. For example, the (alcohol *isA* drink) mapping is not identified in the second experiment because the concept drink of O_{Tar} is not covered by the chosen roots. On the opposite, the (pine *isA* plant) mapping is identified whereas without senses limitation the incorrect (pine *isA* material) mapping was found.

	With a single root (*Entity*)	With several roots
# *isA* mappings found (relevant)	61 (29)	35 (29)
# *isClose* mappings found (relevant)	15 (8)	11 (9)
Total Number of mappings (relevant)	76 (37)	46 (38)
Recall (Precision)	0,23 (0,49)	0,23 (0,83)

Table 1. Number of found mappings among 162 terms in Russia-B not yet mapped

A more precise choice of the roots would very probably increase recall. In our application context, as the identification step of the roots in WordNet can be done just in reference to O_{Tar}, this task is only performed once and the identified roots will be exploited whatever the taxonomies of information sources to be aligned with O_{Tar} might be. Hence it is worthwhile to pay attention to this identification step. Our first results are already promising. Yet we think they could be even better with a more precise choice of the roots.

4 Conclusion

So, in conclusion the use of external background knowledge can be very interesting when the context of interpretation of the involved concepts is precisely known. It allows the obtention of semantic relations and so overcomes the major limitations of syntactic approaches. WordNet is often used as background resource. However, the drawback is that it is difficult to get relevant information if the meaning of the searched terms is not known. The results of our experiments using WordNet indicate that our approach based on the definition of multiple roots is a promising solution when the domain of the background knowledge is too large. Whatever the domain, the sub-tree grouping terms relevant to the application can be extracted from WordNet with our system. We showed how semantic mappings, when they exist, can be found in this sub-tree and how, when they do not exist, meaningful proximity relationships can be found instead.

References

1. Z. Aleksovski, M. Klein, W. Ten Kate, F. Van Harmelen. Matching Unstructured Vocabularies using a Background Ontology, In *Proc. of EKAW'06,* Springer-Verlag, 2006.
2. Z. Aleksovski, M. Klein, W. Ten Kate, F. Van Harmelen. Exploiting the Structure of Background Knowledge used in Ontology Matching». In *Proceedings of ISWC'06 Workshop on Ontology Matching (OM-2006),* Athens, Georgia, USA, November 2006.
3. T. L. Bach, R. Dieng-Kuntz, F. Gandon. On Ontology matching Problems – for Building a Corporate Semantic Web in a Multi-Communities Organization, in *Proc. of ICEIS* (4), 2004.
4. F. Giunchiglia, P. Shvaiko, M. Yatskevich. Discovering Missing Background Knowledge in Ontology Matching, In *Proc. of ECAI 06.*
5. Y. Kalfoglou, B. Hu. Crosi Mapping System (CMS) Result of the 2005 Ontology Alignment Contest. In *Proc of K-Cap'05 Integrating Ontologies WS*, pp. 77-85, Banff, Canada, 2005.
6. T. Pedersen, S. Patwardhan, J. Michelizzi. WordNet::Similarity - Measuring the Relatedness of Concepts, In *Proc. of AAAI-04*, July 25-29, San Jose, CA, 2004.
7. C. Reynaud, B. Safar. When usual structural alignment techniques don't apply. In *Proc. of ISWC'06 Ontology Matching WS, (OM-2006),* Poster, 2006.
8. C. Reynaud, B. Safar. Structural Techniques for Alignment of Taxonomies: Experiments and Evaluation, In TR 1453, LRI, Université Paris-Sud, Juin 2006.
9. M. Sabou, M. D'Aquin, E.Motta. Using the Semantic Web as Background Knowledge for Ontology Mapping, In *Proc. of ISWC'06 Ontology Matching WS (OM-2006),* 2006.
10. S. Zhang, O. Bodenreider. NLM Anatomical Ontology Alignement System. Results of the 2006 Ontology Alignment Contest. In *Proc. of ISWC'06 Ontology Matching* 2006.
11. http://www.atl.external.lmco.com/projects/ontology/ontologies/russia/

Towards Correspondence Patterns for Ontology Mediation[*]

François Scharffe, Ying Ding, and Dieter Fensel

University of Innsbruck, Austria
`firstname.lastname@uibk.ac.at`

Abstract. We introduce in this paper correspondence patterns as a tool to design ontology alignments. Based on existing research on patterns in the fields of software and ontology engineering, we define a pattern template and use it to develop a correspondence patterns library. This library is published in RDF following a structured vocabulary. It is meant to be used in ontology alignment systems, in order to support the user or improve matching algorithms to refine ontology alignments.

1 Introduction

The semantic web contains many ontologies, and is expected to contain more and more as it will develop. When two ontologies overlap, they can be linked together in order to enable exchange of their underlying knowledge. An *alignment* between two ontologies specifies a set of *correspondences*, and each correspondence models a bridge between a set of ontologies entities. Designing ontology alignments is a tedious task. There are many ongoing efforts to develop tools such as graphical user interfaces and matching algorithms, in order to make it easier.

Most ontology alignment systems [1] are limited to detect simple equivalence or subsumption correspondences between single entities, and research concentrates on improving their quality on various datasets more than on finding more complex matches[1]. This can be explained as the problem of detecting complex matches is not a trivial one, maybe also because no standard semantic-web language is expressive enough to represent such correspondences.

However, simple correspondences are often not sufficient to correctly represent the relation between the aligned entities. The two ontologies in the following example deal with wines. In the "Wine Ontology"[2], which main class is $Wine$, the class $BordeauxWine$ represents instances of a popular french wine grown up in the region around Bordeaux. In the "Ontologie du Vin"[3] a similar wine

---

[*] This work was partially funded by the European Commission under the Knowledge Web NoE (FP6-507482) and SEKT project

[1] See the Ontology Alignment Evaluation Initiative 2007 evaluation criteria. http://oaei.ontologymatching.org/2007/

[2] http://www.w3.org/TR/2003/CR-owl-guide-20030818/wine#

[3] http://www.scharffe.fr/ontologies/OntologyDuVin.wsml

is expressed as an instance of Vin with an attribute $terroir$[4] indicating the wine provenance. Classical systems are able to detect the two correspondences $Wine \equiv Vin$ stating equivalence between two wines and $BordeauxWine \subseteq Vin$ stating that $BordeauxWine$ is a narrower concept than Vin . In this case, a more precise correspondence would be $BordeauxWine \equiv Vin \wedge terroir = Bordeaux$, restricting the scope of Vin to only those instances having $Bordeaux$ as value of the $terroir$ attribute.

Going from the initial two basic correspondences to the refined one involving a condition would be easier using a pattern showing the structure of the correspondence. Inspired from design patterns in software engineering, this paper introduces correspondence patterns as helpers that facilitate the ontology alignment process. They improve graphical tools by assisting the user when creating complex correspondences, and we believe they are a first step towards matching algorithms able to detect complex matches.

2 Correspondence Pattern Template

A pattern template provides a standard way to represent patterns. Following [2], we introduce a correspondence pattern template divided in two parts that roughly correspond to the two lowest levels of Blomqvist pattern classification [3]. The first part, core of the correspondence pattern, defines the pattern using classical elements from design patterns literature. The second part presents a grounding for that pattern in a knowledge representation formalism. Many groundings can be defined given a pattern.

A Correspondence Pattern represents a correspondence between two ontologies. The core part of the pattern template contains common pattern elements found in the literature. It follows the high level classification of the GoF [4].

The grounding part represents the grounding of the pattern into a knowledge representation formalism used by a mediator executing pattern instances.

The pattern template defined above give all the elements to describe a correspondence pattern. We give next an example of pattern based on this template. We then see in Section 3 how correspondence patterns can be encoded in a machine readable format in order to facilitate their use in semantic web applications.

The following pattern can be used to model the correspondence presented in Section 1.

Name: Class By Attribute Correspondence
Also Known As: classByAttributeCorrespondence
classByAttributeMapping

Problem:

---

[4] Terroir is a French word for a particular agricultural region

A class in one ontology is related to a class in the other ontology. The scope of the class of the first ontology is restricted to only those instances having a particular value for a a given attribute.

Context: This pattern is used in case two classes have a similar but not completely overlapping extension and the intent of the first class is expressed by a particular attribute value in the target class.

Solution:
Solution description:
This pattern establishes a correspondence between a class / attribute / attribute value combination in one ontology and a class in another.

Syntax: Here comes a correspondence described using the Alignment Ontology

Example: Relate a Human to a Blue-Eyed-Person by restricting those instances of Human whose eye-color attribute is equal to "blue"

Related Patterns: Equivalent Class Correspondence, Subclass Correspondence

Degenerated Pattern: Subclass Correspondence

The patterns library defined Section 3.2 provide generic entities used to be replaced. A grounding of the pattern is given in the following in SPARQL.

Name of the target language/system SPARQL

Applicability Applicable

Purpose Instance Transformation

Example Grounding

```
CONSTRUCT { ?X rdf:type target:BlueEyedPerson }
WHERE { ?X rdf:type source:Human.
        ?X source:eyeColor xsd:String^^"Blue" }
```

The wine example Section 1 can be built by instantiating this pattern. We will show in the next section how is it done in our implementation.

The syntax element of the solution shows a correspondence described in RDF, as an instance of the Alignment Ontology [5]. Purpose of this ontology is the abstract representation of ontology alignments. Section 3 briefly introduces this ontology, and presents its extension: a correspondence patterns library.

3 Correspondence Patterns on the Semantic Web

Now that correspondence patterns are given a template, we need to consider how to implement and store them. Mediation systems need an organized library of patterns in order to retrieve them efficiently. A hierarchical organization seems preferable, following patterns specialization: from general purpose patterns such as between any two classes to very specific patterns using conditions and transformations of attributes values.

Correspondence patterns are used to align ontologies, which are themselves used to reference and reason about RDF resources on the semantic web. On the other way around, publishing patterns and correspondences as ontology instances presents many advantages. The class hierarchy automatically organizes patterns according to their specificity. The degenerated version of a pattern is thus obtained by taking its parent in the class hierarchy. In addition, patterns assert additional facts about correspondences that can be used as background information in order to construct new alignments. Let us consider the wine example shown in Section 1 solved using the `ClassByAttribute` pattern. A system trying to align $BordeauxWine$ with a third ontology can then try applying the same pattern as a similar structure is likely to appear.

Based on the template given in Section 2 we present in this section a library organizing common ontology correspondences patterns. This library is provided as an extension of the Alignment Ontology [5].

3.1 The Alignment Ontology

The Alignment Ontology is an OWL-DL ontology that models ontology alignments as sets of correspondences between ontological entities. It results from efforts [6, 7] to create an ontology alignment format, abstract from the ontology language, that could serve as an interchange format between matching algorithms, mediators and graphical user interfaces.

The ontology is available at http://www.omwg.org/TR/d7/.

3.2 A Correspondence Patterns Library

The patterns library extends the Alignment Ontology under the `Cell` class. The library contains 35 correspondence patterns at the time of writing[5]. Properties corresponding to the pattern template elements are modeled as OWL annotation properties. A pattern in the library is modeled as a class with restrictions on its possible instantiations and an example instance is given for each pattern. In other words, the deepest one goes into the pattern hierarchy, the more specific are the patterns, making retrieval of patterns easier.

Besides patterns themselves, the library provides a set of generic ontological entities used to compose patterns. These entities serve as placeholders to be filled with concrete entity names when using the pattern to make a concrete correspondence. They provide a reference structure for modeling entities in the same way patterns provide a reference structure for modeling correspondences. For example, the `ClassConditionInstance` models a class with a condition restricting its scope.

The pattern library is available at http://www.omwg.org/TR/d7/.

---

[5] This figure does not take into account all possible value transformations patterns.

We ground correspondences using transformations of the Alignment Ontology instances into the target formalism. The Alignment API [6] and the Mapping API [7] propose groundings into various semantic-web languages such as OWL, SWRL, WSML and SKOS. We are currently working on extending the implementation in order to provide SPARQL++ [8] grounding.

4 Conclusion

We introduced in this paper ontology correspondences patterns as helper to model ontology alignments. We defined a template based on the literature on design patterns and gave a library of elementary correspondence patterns. Patterns instances are described using the Alignment Ontology, a formalism to represent ontology alignments abstract from the underlying knowledge representation formalism. In order to be executed, pattern instances are grounded to the desired formalism. We are constantly refining the library given experience in research projects. We currently investigate on using correspondence patterns in order to develop a patterns based ontology matching algorithm.

5 Acknowledgements

We would like to thanks Jos de Bruijn for useful discussions about correspondence patterns.

References

1. Shvaiko, P., Euzenat, J.: A survey of schema-based matching approaches. (2005) 146–171
2. Staab, S., Erdmann, M., Maedche, A.: Engineering ontologies using semantic patterns. In O'Leary, A.P..D., ed.: Proceedings of the IJCAI-01 Workshop on E-Business & the Intelligent Web, Seattle, WA, USA, August 5, 2001. (2001)
3. Blomqvist, E., Sandkuhl, K.: Patterns in ontology engineering: Classification of ontology patterns. In: ICEIS (3). (2005) 413–416
4. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns. Addison-Wesley Pub. (1995)
5. Scharffe, F.: Omwg d7: Ontology mapping language. http://www.omwg.org/TR/d7/ (2007)
6. Euzenat, J.: An API for ontology alignment. In: Proc. 3rd international semantic web conference, Hiroshima (JP). (2004) 698–712
7. Scharffe, F., de Bruijn, J.: A language to specify mappings between ontologies. In: Proc. of the Internet Based Systems IEEE Conference (SITIS05). (2005)
8. Polleres, A., Scharffe, F., Schindlauer, R.: Sparql++ for mapping between rdf vocabularies. In: ODBASE, On the Move Federated Conferences (to appear). (2007)

Evaluating a confidence value for ontology alignment

Paulo Maio, Nuno Bettencourt, Nuno Silva, and João Rocha

GECAD - Knowledge Engineering and Decision Support Group
Institute of Engineering - Polytechnic of Porto
Porto, Portugal
{paom,nmgb,nps,jsr}@isep.ipp.pt

Abstract. Many methods for automatic and semi-automatic ontology alignment have been proposed, but they remain error prone and labor-intensive. This paper describes a novel generic process for evaluating the mappings' confidence value. This process uses rules extracted through inductive machine learning methods from the matching results proposed by others. Further, the precision and recall of the extracted rules are exploited in order to transform each rule into a mathematical formula that generates the mappings' confidence value. Mappings are then classified not as valid or invalid but through a quantitative confidence value that can be easily managed during the alignment process.

1 Introduction

Ontology alignment overcomes the information heterogeneity problem and provides mechanisms for each system to process data as if it was represented according to their internal model (ontology). The ontology alignment process aims to define an alignment between a source and target ontology [5].

The alignment specification is a very time consuming and knowledge demanding task, whose result is error prone even when domain experts are part of the process [2]. This problem is even bigger in scenarios where online alignment is required (*e.g.* e-business, e-commerce). Automatic mechanisms are necessary in order to supply the necessary consensus and speed up the interoperability process.

In last few years different methods for automatic ontology alignment have been proposed to overcome this gap, but there still remains the need to automatically combine multiple diverse and complementary alignment strategies of all indicators, *i.e.* extensional and intensional descriptions, in order to produce comprehensive, effective and efficient alignment methods [3]. Such methods need to be flexible to cope with different strategies for various application scenarios.

This paper presents a novel confidence value evaluation method based on machine learning techniques that can be easily integrated into general alignment methods like QOM [4] and PROMPT [7], or can be applied in relaxation processes required in distributed ontology alignment negotiation processes (*e.g.* [9]).

The document structure is as follows: the next section introduces our approach in order to evaluate mappings' confidence value. At the end, a brief discussion about the proposed approach is presented emphasizing the major contributions of the paper and suggesting further research and development directions.

2 Our approach

The adopted approach is based on inductive machine-learning methods. However, the extracted rules are not directly applied on classifying the mappings but serve as an input for the configuration of the system. We pursue a method that reflects the reliance of the rules upon the training dataset. In fact, the precision and recall of the extracted rules are often low, easily leading to many false positives and false negatives when applied to testing and running data sets.

The method comprises of three phases, described in the following sub-sections.

2.1 Extracting Rules

This section applies a set of machine-leaning methods (*e.g.* J48 and JRIP [1]) to the training dataset. This set (see Table 1) is comprised of several ontology mappings, in which are identified pairs of source and target ontologies' entities and the values generated by several matching algorithms [8]. The goal attribute of the learning process is the validity of the mapping.

Table 1. Partial training dataset example.

O	O'	Source Entity	Target Entity	Valid	$Matcher_1$	...	$Matcher_n$
O_1	O_2	Woman	Woman	Yes	1.00	...	0.50
O_1	O_2	HumanBeing	Hermaphrodite	No	0.13	...	0.00

The result is a set of extracted rules $SR = \{r_1, ..., r_n\}$ for each learner. A rule (r_i) can be of two types: (i) *Simple*, (Example 1) which exploits a single matching algorithm and (ii) *Complex* (Example 2), which exploits at least two different matching algorithms. Each complex rule can be split into sub-rules (sr_j). Each sub-rule establishes one and only one criteria through a unique matching algorithm.

Example 1. $valueof(StructurePlus) \geq 0.95$.

Example 2. $valueof(INRIA) \geq 0.84 \wedge valueof(Cano) \geq 0.31$.

2.2 Converting Rules into a Formula

The rules are therefore prepared to dichotomously map pairs of ontologies' entities (*i.e.* valid or invalid). This often leads to poor results evidenced by many false positives and false negatives.

For this, the proposed process disregards the rule itself, but instead evaluates its precision *i.e.* $prec(r_i)$ and recall *i.e.* $reca(r_i)$ when applied to the training set. Similar values are evaluated for each sub-rule, *i.e.* $prec(sr_j)$ and $reca(sr_j)$.

The process proceeds by combining precision and recall into a reliance value for each rule/sub-rule. For this purpose, different functions can be used, *e.g.* harmonic average $fmeasure$ (see Equation 1) and the weighted average (see Equation 2), where α allows us to trade-off between precision and recall.

$$f_\alpha(sr_j) = \frac{(1+\alpha).prec(sr_j).reca(sr_j)}{\alpha.prec(sr_j) + reca(sr_j)} : \alpha \geq 0; \tag{1}$$

$$w_\alpha(sr_j) = \alpha.prec(sr_j) + (1-\alpha).reca(sr_j) : 0 \leq \alpha \leq 1 \tag{2}$$

Furthermore, in order to (i) Abstract from the combination function and (ii) Normalize the reliance value of the sub-rule in respect to the overall rule, the $p_\alpha(sr_j)$ is defined:

$$p_\alpha(sr_j) = \begin{cases} f_\alpha(sr_j)/\sum\limits_{k=1}^{n} f_\alpha(sr_k) \\ w_\alpha(sr_j)/\sum\limits_{k=1}^{n} w_\alpha(sr_k) \\ \ldots \end{cases} \tag{3}$$

An equivalent formula is defined for evaluating the reliance of each rule (r) in respect to the set of rules (SR).

Therefore, applying rule r and p_α it is possible to evaluate the confidence value of a mapping m_i through the following function:

$$u^r_{p_\alpha}(m_i) = \sum_{j=1}^{n} valueof_{m_i}(matcherof(sr_j)).p_\alpha(sr_j) \tag{4}$$

where, $matcherof(sr_j)$ returns the name of the matching algorithm used as criteria at sub-rule j, $valueof_{m_i}(MatcherName)$ represents the matching algorithm's similarity value for m_i and n is equal to the number of sub-rules of the rule r.

2.3 Aggregating and Applying Formulas

Because each learner extracts several rules (see 2.1), several valid $u^r(m_i)$ might exist, *i.e.* there is one different $u^r(m_i)$ for each rule in the extracted set of rules. In that sense, mappings have one distinct confidence value for each rule, given by $u^r(m_i)$.

Consequently, it is necessary to choose or evaluate an unique confidence value, *i.e.* $u(m_i)$ based on all available $u^r(m_i)$. With that purpose, an aggregation function (agg) is used. A preference list over the existing rules based on the learners' additional information (*e.g.* percentage of error) and the maximum, minimum, linear average or weighted average (*e.g.* using $p_\alpha(r_i)$) are some possible agg functions. In that sense, the confidence value of a mapping, *i.e.* $u(m_i)$ is evaluate by the function presented in Equation 5:

$$u(m_i) = agg[u^{r_1}_{p_\alpha}(m_i), ..., u^{r_n}_{p_\alpha}(m_i)] \tag{5}$$

Thus, despite the mappings classification (valid or invalid), formulas deliver a quantitative confidence value ($[0 - 1]$).

This allows constraint and relaxation of the alignment requirements by changing the acceptance/rejection threshold (t_r). Therefore, given two ontologies, the suggested mappings will be those where $u(m_i) \geq t_r$, where t_r is the acceptance/rejection threshold (Example 3).

Example 3. Having $t_r = 0.8$, $u^{r_1}_\alpha(m_i) = 0.9$ and $u^{r_2}_\alpha(m_i) = 0.7$ which means that $u(m_i) = agg[0.9, 0.7]$. Thus, using maximum function as agg, m_i is considered accepted ($0.9 \geq 0.8$). On the other hand, using minimum function as agg, m_i is considered rejected ($0.7 < 0.8$).

Notice that usually t_r is exclusively defined by the user, but the training stage might provide some good hints to be used when defining the threshold.

3 Discussion

This paper presents a novel process for calculating the mappings' confidence value for ontology alignments, using the rules extracted through machine learning methods. The basic idea is to convert the extracted rules into formulas that reflect the reliance of each rule. Rules are further combined in order to generate a single value on the mapping.

Matching algorithms and their results therefore play a relevant role. Thus, a careful and strict matching algorithms selection phase is required in order to include diversification of abilities (*e.g.* hierarchy, semantics, data types and instances analysis) according to the training dataset's characteristics. Their virtues and limitations have positive and negative influence in the results obtained. Notice that due to matching algorithms' internal configuration, the similarity value between the same pair of source and target entities might be different.

Machine-learning methods play another important role in the system. In fact, they are responsible for efficiently combining matching algorithms and finding out the relevant threshold for that combination. From all initially available matching algorithms, only a few of them are combined into rules. That selection is automatically performed based on matching algorithms acting capabilities which implies no user information is needed. Extracted rules can also be updated automatically when new matching algorithms or mapped ontologies are added.

The use of more than one learner is recommended as tests show that different learners can extract different rules with similar results. Therefore, this fact can be seen as an advantage when used to avoid errors or disambiguate results.

Also, the training data set's ontologies have a special relevance. Because each ontology is related to one main knowledge domain, if training ontologies are only concerned with one matching knowledge domain (*e.g.* health care) then the learned rules and information should only be used in a similar domain. However, preliminary tests showed that learning rules and information are independent of ontologies characteristics (*e.g.* flattened hierarchy vs expanded hierarchy).

It is our conviction that the proposed approach can be easily integrated with existing automatic and semi-automatic ontology alignment tools. Also, correctness of generated alignments can be improved when combining this approach with other existing techniques as, for instance, debugging alignments with logical reasoning [6]. While evaluation results are not conclusive, they are encouraging. Ongoing research is focused in (i) the systematization of the application of the *agg* functions, α parameter depending on the requirements of the mapping scenario, and (ii) on the generalization of the proposed approach to ontology attributes and relations.

4 Acknowledgments

This work is partially supported by the Portuguese MCT-FCT project EDGAR (POSI/EIA/61307/2004).

References

- [1] Auknomi. Categorical machine learning algorithms. `http://www.auknomi.com/categorical_learners.html`, 2006.
- [2] A. Doan, J. Madhavan, P. Domingos, and A. Halevy. Ontology matching: A machine learning approach. Handbook on Ontologies, 2004.
- [3] M. Ehrig, S. Staab, and Y. Sure. Bootstrapping ontology alignment methods with APFEL. In *Proceedings of ISWC*, pages 186–200, Galway, Ireland, November 2005.
- [4] Marc Ehrig, Steffen Staab, and York Sure. Qom - quick ontology mapping. volume 3298 of *Lecture Notes in Computer Science*, pages 683–697, Hiroshima, Japan, Nov 2005. Springer.
- [5] Jérôme Euzenat. An API for ontology alignment. In *Proceedings of ISWC*, pages 698–712, 2004.
- [6] C. Meilicke, H. Stuckenschmidt, and A. Tamilin. Improving automatically created mappings using logical reasoning. In *Ontology Mapping Workshop at ISWC*, Athens, GA, USA, 2006.
- [7] Natalya F. Noy and Mark A. Musen. The prompt suite: Interactive tools for ontology merging and mapping. International Journal of Human-Computer Studies,59(6):983–1024, 2003.
- [8] Pavel Shvaiko and Jérôme Euzenat. A survey of schema-based matching approaches. *Journal on data semantics*, 4:146–171, 2005.
- [9] Nuno Silva, Paulo Maio, and João Rocha. An approach to ontology mapping negotiation. Banff (Alberta), Canada, Oct 2005.

Finding Semantic Similarity in a Biological Domain: A Human-Centered Approach

Takashi Yamauchi and Na-Yung Yu

Mail Stop 4235, Department of Psychology
Texas A&M University College Station, TX 77843 USA
tya@psyc.tamu.edu, dbskdud40@tamu.edu

A behavioral study investigated how college students judge similarity between cell pictures. The study indicates that there is a strong tendency to rely on class-inclusion relations in judgments of similarity. This means that biological concepts are likely to be organized and conceptualized with respect to class-inclusion relations even for non-experts.

1 Introduction

Concepts are assemblies of knowledge that are developed, construed, modified, and constructed by people interacting in a particular domain. This means that the concepts that people form, which are the medium of ontology matching, are necessarily influenced by the way people process, represent, and retrieve information.

In this brief article, we will illustrate how college students who do have no special training in medicine judge semantic similarity among cell pictures, and show that there is a strong predisposition for lay people to rely on class-inclusion relations to determine conceptual similarity.

2 Study 1

Fig. 1 shows variations of two different animal tissues. The two pictures placed at the top of the two frames are original cell pictures (i.e., target pictures) and those at the bottom are morphed images of the two original pictures (i.e., base pictures). In this study, participants (undergraduate students, *N*=227) were presented with 60 triads of cell pictures similar to those shown in Fig. 1 and they judged which base picture, left or right, was more similar to the target picture placed on the top.

The question of interest was the effect of labeling. We hypothesized that class-inclusion relations are particularly important for the conceptualization in the biological domain; biological concepts are arranged and understood in the context of how entities relate to one another in their taxonomical relations rather than in their concrete appearance, attributes, or properties [1] [2].

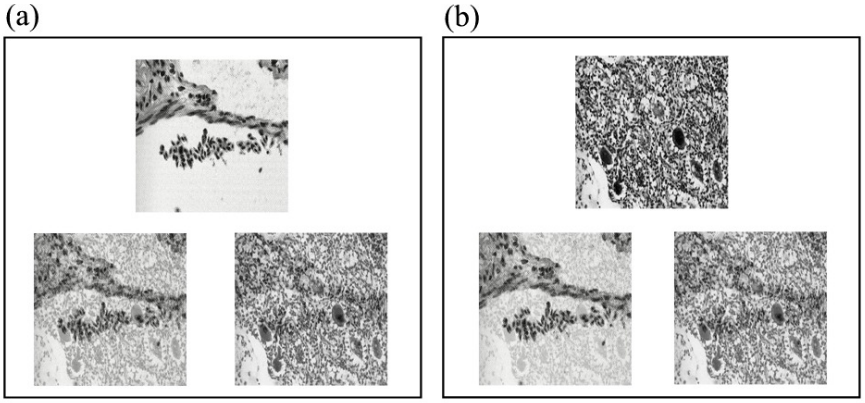


Fig. 1. Two samples of the stimulus frames used in Study 1. The base pictures (shown at the bottom) were produced by merging the two cell pictures shown at the top.

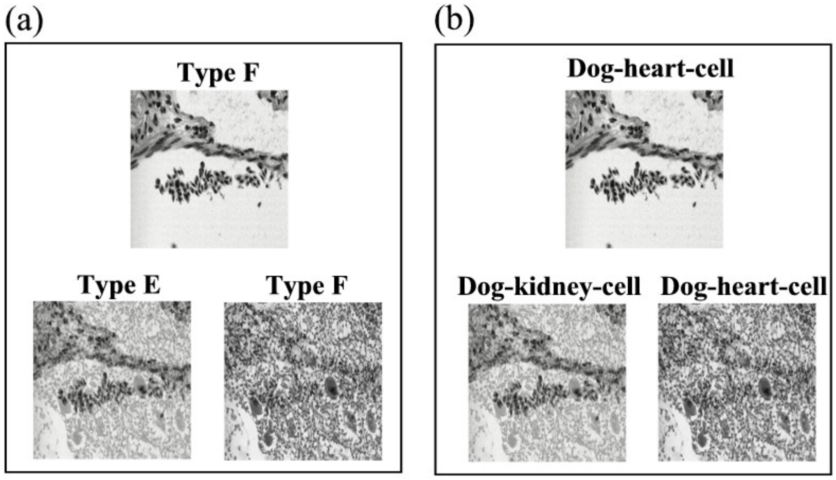


Fig. 2. Two samples of the stimulus frames in the alphabet-label and cell-label conditions in Study 1. Meaningless alphabetical labels are attached in (a) ("Type F" and "Type E"); meaningful verbal labels are attached in (c) ("Dog-heart-cell" and "Dog-kidney-cell").

To test this idea, we examined how the ontological labels attached to the cell pictures would influence participants' judgments of similarity. In one condition, no labels were attached to the pictures (control condition, Fig 1). In another condition, meaningless verbal labels were attached to the same cell pictures ("Type E" and "Type F," Fig 2a). In the other condition, fictitious yet conceptually meaningful labels were attached to the same cell pictures ("Dog-kidney-cell" and "Dog-heart-cell," Fig

2b). Given these three conditions, participants judged which cell pictures, left or right, were more similar to the target picture placed on the top.

2.1 Method

2.1.1 Materials

We produced a total of 60 triads from 5 pairs of original cell pictures. For each pair, one original picture was merged with the other original picture in different degrees, creating three groups of morphed pictures for each pair (low-, medium-, or high-level groups; see Fig. 3). These cell pictures carried different types of labels depending on the condition to which participants were assigned.

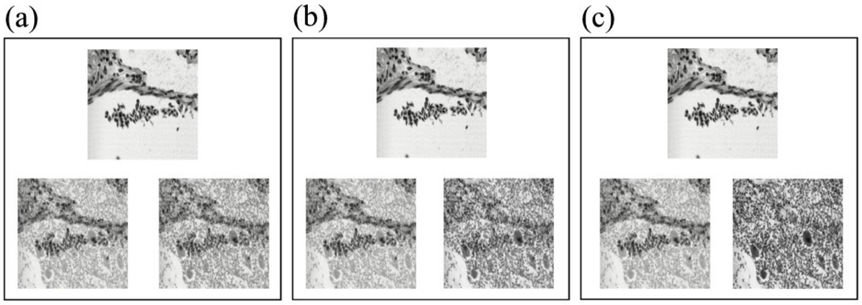


Fig. 3. Samples from three groups of morphed pictures.

2.1.2 Procedure

Sixty triads of cell pictures were presented to each participant one at a time at the center of the computer screen. Participants selected one base picture that was deemed similar to the target picture. The order of presenting the stimuli was determined randomly, and the location of placing base pictures (either left or right) was also determined randomly.

2.1.3 Design

The experiment had one between-subjects factor: (label condition; no-label, alphabet-label, cell-label). In the no-label condition, no pictures carried labels (Fig. 1). In the alphabet-label condition, the pictures carried meaningless alphabetical labels (Fig. 2a). In the cell-label condition, the pictures carried meaningful labels (Fig. 2b). Thus, the conceptual relations between the cell pictures were unclear in the no-label and alphabet-label conditions, but the conceptual relations were clear (e.g., heart vs. kidney) in the cell-label condition.

We employed two measures to assess the effects of labeling. First, we examined the proportion of participants selecting dissimilar base pictures as more similar as these cell pictures carried different kinds of labels. For example, we measured the proportion of participants selecting the base picture on the right in Fig. 2b; this base picture was less similar to the target than the other base picture (the one on the left), so measuring the proportion of selecting dissimilar base picture would tell us the extent to which labels override perceived similarity. Second, we examined the impact of labeling in two situations – one in which the target and dissimilar base pictures had the same labels (i.e., the same-label condition, Fig. 2a and 2b), and the other in which the target and dissimilar base pictures had different labels (i.e., the different-label condition). These two conditions were produced by simply swapping the assignment of the labels to the base pictures. For example, in the different-label condition, "Type E" and "Type F" given to the two base pictures in Fig 2a were swapped.

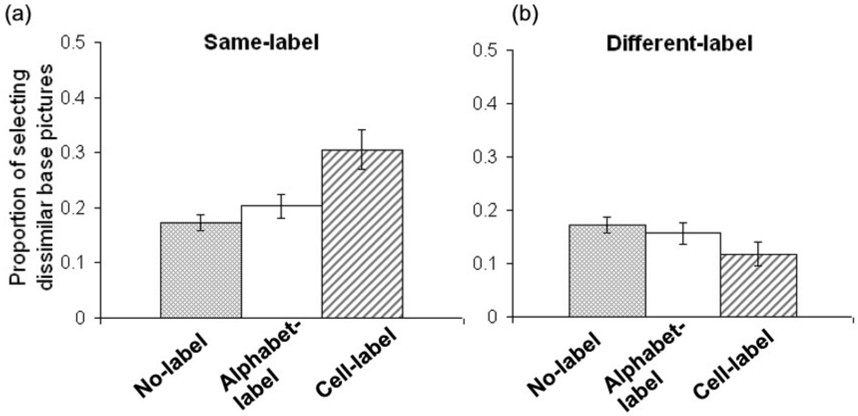


Fig. 4. A summary of the results from Study 1. The error bars represent two standard error units obtained from each condition.

2.2 Results

When the target and the dissimilar base pictures had the same label (Fig. 4a), the proportion of selecting the dissimilar base pictures increased significantly. When the target and the dissimilar base picture had different labels (Fig. 4b), the proportion of selecting the dissimilar base pictures declined substantially. This impact of labeling was present only when the pictures carried conceptually meaningful cell-labels, but not when they carried meaningless alphabetical labels.

In the same-label condition (Fig. 4a), the mean proportions of selecting the dissimilar base picture were significantly higher in the cell-label condition (*M*=0.31) than in the no-label (*M*=0.17) and alphabet-label (*M*=0.20) conditions; *F*(2, 140)=7.84, *MSE*=0.09, *p*<0.001; cell-label vs. no-label, *t*(94)=3.54, *p*<0.001, *d*=0.72; cell-label vs. alphabet label, *t*(93)=2.47, *p*=0.02, *d*=0.51. The proportions of selecting the dissimilar base pictures were not different between the no-label condition and the alphabetical label condition; *t*(93)=1.23, *p*=0.22, *d*=0.25.

Given different labels (Fig 4b), the mean proportion of selecting dissimilar base pictures was significantly lower in the cell-label condition (*M*=0.12) than in the no-label (*M*=0.17) and alphabet-label (*M*=0.16) conditions; *F*(2, 129)=4.77, *MSE*=0.02, *p*<0.05; cell-label vs. no-label, *t*(89)=2.87, *p*=0.005, *d*=0.60; cell-label vs. alphabet-label, *t*(80)=2.06, *p*=0.04, *d*=0.45.

These results suggest that the labels influenced participants' similarity judgments only when the labels were conceptually meaningful, indicating that the conceptual links between cell pictures were crucial even in the perceptual judgment of similarity of the cell pictures.

3 Discussion

Ontologies are formal descriptions of concepts developed by people, so it appears natural to study how people acquire and use concepts in a given domain in order to develop viable mapping agents. In ontology matching, "similarity" is generally assessed in multiple levels (e.g., lexical, structural, and/or relational levels). The overall similarity between ontologies is specified as a weighted sum of individual similarity measures [3] [4] [5]. However, allocating appropriate weights to these similarity factors is not trivial. Because conceptualization arises from a highly interactive environment in which different sets of goals and constraints are required, the mapping agent needs to incorporate different heuristics and background knowledge to identify adequate weights for similarity factors. The present study suggests that conceptually meaningful class-inclusion relations are crucial even for lay people in determining perceptual similarity among cell pictures.

Acknowledgements. This research was supported by the Glasscock Center Faculty Fellow Award, and a Developmental Grant by Mexican American and U.S. Latino Research Center, Texas A&M University. We would like to thank Wookyoung Jung for her valuable comments.

References

1. Yamauchi, T., Yu, N.: Category Labels versus Feature Labels: Category Labels Polarize Inferential Predictions. Mem Cognition (in press)
2. Yamauchi, T., Markman, A. B.: Inference Using Categories. J Exp Psychol Learn 26 (2000) 776-795
3. Shvaiko, P., & Euzenat, J.: A survey of schema-based matching approaches. Lecture Notes in Computer Science Vol. 3730. Springer-Verlag, Berlin Heidelberg New York (2005) 146-171
4. Noy, N. F. Semantic Integration: A Survey of Ontology-based approaches. SIGMOD Record, 33 (2004) 65-70.
5. Doan, A., Madhavan, J., Domingos, P., & Halevy, A. (2004). Ontology matching: A machine learning approach. In S. Staab & R. Studer (eds.), Handbook on Ontologies (pp. 385-403). New York: Springer

A 'Tagging' Approach to Ontology Mapping

Colm Conroy[1], Declan O'Sullivan[1], Dave Lewis[1]

[1] Knowledge and Data Engineering Group,
Trinity College Dublin
{coconroy,declan.osullivan,dave.lewis}@cs.tcd.ie

Abstract. The reasons for the lack of uptake of the semantic web amongst ordinary users can be attributed to technology perception, comprehensibility and ease of use. To address these three problems, we believe that the interfaces to ontology management tools will need to be engineered in such a way as they disappear into the background from the ordinary person's perspective. The majority of the state of the art approaches to ontology mapping relies on the user being ontologically aware. In contrast, this paper reports upon an approach to use 'tagging' as a means of ontology mapping to support ordinary people.

1 Introduction

The promise of ontologies, which is a core technology of the semantic web, is in the sharing of an understanding of a domain that can be communicated between people and application systems [fensel 2003]. Although the semantic web approach is establishing itself in certain domains, as yet no viral uptake within the mainstream internet environment has occurred [berners-lee 2006]. Instead recently collaborative tagging schemes (referred to as "folksonomies") have started to emerge within the mainstream internet community. With collaborative tagging, people publicly annotate resources with keywords that describe those resources (called tags). In contrast, ontologies provide a means to associate terms with concepts and relate concepts together, thereby providing a means to discover ambiguity, navigate over relationships, and cope with synonyms and homonyms. The reasons for the lack of uptake of the semantic web amongst ordinary users can be attributed to technology perception, comprehensibility and ease of use. To address this problem, we believe that the interfaces to ontology management tools will need to be engineered in such a way as they disappear into the background from the ordinary person's perspective. The automatic and efficient matching between the personal ontology and the models used by others (collaborative tags and/or community ontologies) can be achieved through the application of a variety of matching techniques [shvaiko 2004]. The research challenge lies in how to derive ontology mappings from the candidate matches. Fully automatic derivation of mappings is considered impossible as yet [noy 2004], and the majority of state of the art tools in the ontology mapping area [kalaynpur 2004], [aumuller 2005] and the community ontology creation area [zhdanova 2005] rely on a classic presentation of the class hierarchy of two ontologies side by side and some means for the user to express the mappings. Two key objectives we address is making ontology mapping **natural** and **transparent** as possible for the casual web user, which can be stated as:

- To represent match information in a clear and intuitive way for the user;
- To make the turning of the match information into a mapping expression intuitive for the user; which is also unobtrusive and takes place over time.

2 Initial Experiment & Findings

In our initial experiment undertaken in early 2007 we aimed to make the semantic mapping process as user friendly as possible so that ordinary people could undertake mapping of semantic models without having to be ontologically aware. With current ontology mapping systems, the ontologies are explicit, the mappings are difficult to read, hard to navigate, and are prone to data overload with too much clutter. In our solution we chose to take a Q&A approach to reduce the complexity involved. Using this approach we reduce the mapping process into piece wise comparisons. We used a fix set of question templates that use the words “similar” and “correspond” in comparing the terms of the match. We used a natural language approach to display each concept as it was our contention that it should help a non technical user in understanding the information better, since ordinary people are more accustomed to reading sentences rather than looking at graphical structures. For each concept its parents, properties and siblings are displayed. We categorised each match within one of the following groups: **valid/mostly/skipped/to-be-validated/rejected**. The main purpose of our initial experiment was to test the usability of our prototype natural language mapping tool (NL) and contrast it with a current state of the art ontology mapping tool, which we choose as the graph type mapping tool COMA++. We split the user test group into three distinct groups: ontology aware, technology aware and non-technical aware. Two different domains were used for the ontologies to help keep the experiment grounded which were music[1] and university[2]. 8 different users were asked to use both tools in different orders which allowed the experiment to be done in 4 different orders, twice per each group. For the experiment we allowed the users to use each tool and after they were finished we had a short interview with them where we asked some questions to gauge the usability of each tool. Some conclusions that we drew from this experiment were:

- Natural language can be used for representing ontological terms to ordinary people as it seems to help people read and understand the information.
- The question & answer approach helped user to navigate through the process of validating the mappings.
- Results showed that ordinary users can validate mappings effectively and efficiently even compared to ontology aware users.
- Some were still confused as to the task of the experiment which showed there is still a need to reduce the complexity even more.
- When answering mapping questions with the NL tool a lot of the wrong answers came from people getting confused with the mean of some questions, i.e. it may not be the same but it corresponds, and on the use of ontological terms like Thing and Agent.

---

[1] http://maciej.janik/test and http://www.kanzaki.com/ns/music
[2] http://annotation.semanticweb.org/iswc/iswc.owl and http://swrc.ontoware.org/ontology

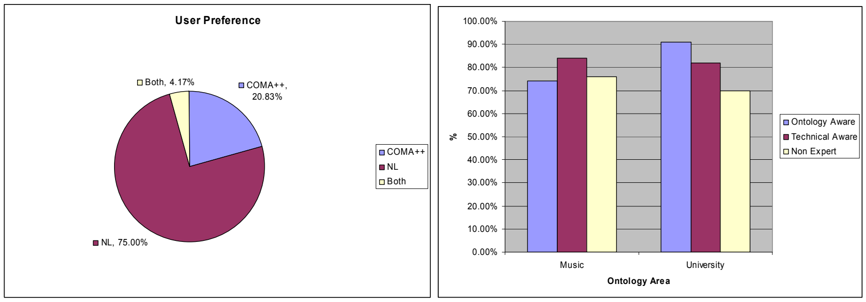


Fig 1: *Experiment results* showed that a two thirds majority preferred the NL tool (left). NL prototype overall accuracy across the 3 groups were very comparative (right).

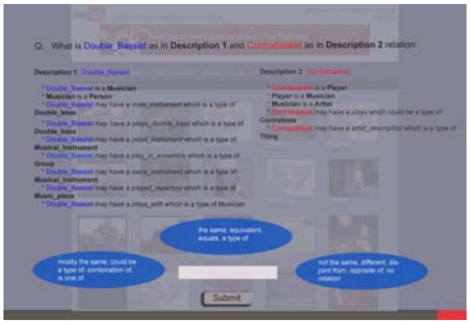
3 Turning matches into mappings using a tagging approach

One of the main outcomes of our initial experiment was the realization that ordinary users found it very restrictive to be limited to a narrow range of mapping terminology, e.g. “corresponds” and “similar to”. It is our contention that ordinary people do not like the ‘yes’ or ‘no’ strict approach but rather they would prefer the ability to ‘tag’ things with their own terms to give them their own meaning. Sites like flickr[3] have become popular because they allow the user to become more expressive by ‘tagging’ items with their own terms, it is our opinion that by allowing users to map ontologies in this fashion would allow the mapping process to become easier for the user and would lead to expressive mappings. In our current work we are allowing the user to 'tag' the mapping relationship with multiple user defined 'tags'. Each tag is then annotated based on the categories in Fig 2. A decision is then made as to which category the mapping should belong to overall. The decision making rules are configurable, for example majority rules. The ‘corresponds’ category refers to when a mapping relation is tagged with unknown ‘tags’ and the matching cannot be assigned to one of the other categories. 'Validation' questions will be used to sort the ‘correspond tags’ into other categories, based on other users’ experiences. In our current implementation our mapping tool is a Firefox browser extension[4] which will be used to display the 'tag' question by making a transparent screen over the current page the user is browsing. The information source which we plan to use in our next experiment will be information from RSS feeds and podcast feeds. Each user will have ontologies representing their interests, e.g. music, sport etc. The system will then be used by the user to map between their interests and the RSS feeds. For the experimentation we will use the same groups as in the initial experiment with different and more people tested. The experiment will occur over a period of a month with feedback captured at random times. After the experiment is concluded we will have the users fill in an online questionnaire form to gauge the user’s reaction/performance.

---

[3] http://www.flickr.com/
[4] https://addons.mozilla.org/firefox/extensions/

- Equivalent
    - The same
    - Subclass
- Equivalent Sometimes
    - Super class
    - One of
    - Union
    - Intersection
- Different
    - Different from
    - Complement of
    - Disjoint
- Corresponds



Q: What is Double_Bassist as in Description 1 and Contrabassist as in Description 2 relation

Fig 2: *The 'Tag' Categories* (left): The 4 categories have corresponding concept relationships as subcategories. *Conceptual 'tag' interface* (right): It uses same NL to represent the ontological terms as in initial experiment.

4 Design of Mapping Process to make more transparent

In the design of our tagging approach (Fig 3) we aim to make the mapping process more transparent and occur over multiple sessions so the user will see mapping not as inconvenient work but more as being part of there daily life.

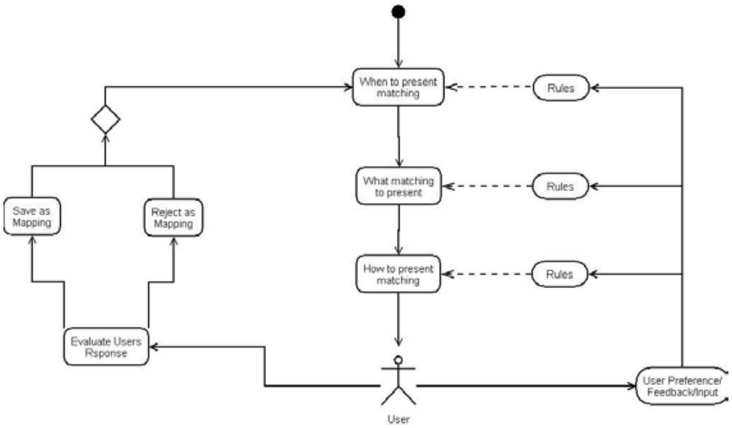


Fig 3: *User Mapping Interaction*

- **When to present matching pair:** Involves calculating the time to present a matching pair for the user, making the process occur over multiple sessions.
    - ”Just in time”, e.g. if the user is submitting a query but needs to map their own ontology to another one for the query to work.
    - When the user is perceived as being idle.
    - After a specific time period, e.g. every hour when available.
    - Threshold on number of matching pairs asked, e.g. 3 per hour.

- **What matching pair to present:** This step involves deciding which matching present is the next one to present to the user
    - Priority based on expected need (based on user interaction patterns).
    - Specific matching API, matching strategy and threshold percentage.
    - A user specified matching, e.g. the matching API misses a matching pair which the user implicitly/explicitly implies is a mapping.
- **How to present the matching pairs**: This step involves deciding the best way to display the matching pair to the user.
    - Visual type, i.e. natural language, graphically, etc...
    - Different forms of representation, i.e. NL bullet point or paragraph..
    - Filtering away information whether necessary or unnecessary.

In our next experiment we plan to address the first two points.

5 Final Remarks

It is hoped through our proposed tagging experiments we will be able to show that the mapping process can be undertaken over multiple sessions rather than one sit down session and that this method will be demonstrably better for the user in making the process as transparent as possible. We are also hoping to show that through our Firefox browser extension that the combination extension of the natural language interface with tagging will enable users to turn matches into expressive mappings in a natural manner. In summary we hope to show that a combination of natural language information presentation, 'tagging' for mapping expressivity, and process to manage the mapping generation over time, will reduce the complexity of semantic mapping, help the user be more expressive in the mappings generated, which will lead to higher user engagement in undertaking mapping and clearer understanding by the user of the benefits for the user in terms of greater access to internet resources

6 References

[aumuller 2005]	Aumüller D., Do H., Massmann S., Rahm E., “Schema and Ontology Matching with COMA++”, Proc. SIGMOD 2005 (Software Demonstration), Baltimore, June 2005.
[berners-lee 2006]	Berners-Lee T., Shadbolt N., Hall W., The Semantic Web Revisited. IEEE Intelligent Systems Vol. 21, Issue 3, pp. 96-101, 2006.
[fensel 2003]	Fensel D., “Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce”, 2nd edition, Springer-Verlag, Berlin, 2003.
[kalyanpur 2004]	Kalyanpur A., Sirin E., Parsia B., Hendler J., "Hypermedia inspired ontology engineering environment: Swoop", Proceedings of 3rd International Semantic Web Conference (ISWC-2004), Japan (Poster).
[noy 2004]	Noy N., “Semantic Integration: A Survey of Ontology-Based Approaches”, Special Issue on Semantic Integration, SIGMOD Record, Volume 33, Issue 4, pages 65-70, December 2004.
[shvaiko 2004]	Shvaiko P. and Euzenat J., "A Survey of Schema-based Matching Approaches", Submitted to Journal of Data Semantics, Long version in DIT Technical Report DIT-04-87, 2004.
[zhadonova 2005]	Zhdanova A., Towards Community-Driven Ontology Matching, Proceedings of K-CAP 2005, Banff, Canada, October 2-5, 2005.

Learning Subsumption Relations with *CSR*: A Classification-based Method for the Alignment of Ontologies[1]

Vassilis Spiliopoulos[1, 2], Alexandros G. Valarakos[1], George A. Vouros[1], and Vangelis Karkaletsis[2]

[1]AI Lab, Information and Communication Systems Engineering Department, University of the Aegean,
Samos, 83 200, Greece
{vspiliop, alexv, georgev}@aegean.gr
[2]Institution of Informatics and Telecommunications, NCSR "Demokritos", Greece
vangelis@iit.demokritos.gr

Abstract. In this paper we propose the "Classification-Based Learning of Subsumption Relations for the Alignment of Ontologies" (*CSR*) method. Given a pair of concepts from two ontologies, the objective of *CSR* is to identify patterns of concepts' features (here, properties) that provide evidence for the subsumption relation among these concepts. This is achieved by means of a classification task using decision trees. For the learning of the decision trees, the proposed method generates training datasets from the source ontologies', considering each ontology in isolation. The paper describes thoroughly the method, provides experimental results for computing subsumption relations over an extended version of the OAEI 2006 benchmarking series and discusses the potential of the method.

Keywords: ontology alignment, subsumption, supervised learning, binary classification.

1 Introduction

Although many efforts [1] aim to the automatic discovery of equivalence relations between the elements of ontologies, in this paper we conjecture that this is not enough: To deal effectively with the ontologies' alignment problem, we have to deal with the discovery of subsumption relations among ontology elements. This is particularly true, when we deal with ontologies whose conceptualizations are at different "granularity levels": In these cases, elements (concepts and/or properties) of an ontology are more generic than the corresponding elements of another ontology. Although subsumption relations between the elements of two ontologies may be deduced by the equivalence relations of other elements, in extreme cases where no equivalence relations exist, this can not be done. In any case, we conjecture that the discovery of subsumption relations between elements of different ontologies may further facilitate the discovery/filtering of equivalence relations, and vise-versa, augmenting the effectiveness of our ontology alignment and merging methods [2].

This paper presents the "Classification-Based Learning of Subsumption Relations for the Alignment of Ontologies" (*CSR*) method. *CSR* computes subsumption relations between concept pairs of two distinct ontologies by means of a classification task, using decision trees, and by exploiting equivalences between properties. Given a pair of concepts, the supervised machine learning method "locates" a hypothesis concerning their relation in a space of hypotheses, which best fits (but not restricted) to the training examples [3], generalizing beyond them. Concept pairs are represented as feature vectors of length equal to the number of the *distinct* properties of source and target ontologies: Equivalent properties (i.e., properties with equivalent meaning) correspond to the same vector component. The training examples for the learning method are being generated from the target and source ontologies.

Although other features may be used, in this paper we study the importance of concepts' properties to assessing the subsumption between concepts: This is an important first step to assessing subsumption relations among concepts, since (a) it appeals to our intuition about the importance of properties as distinguishing characteristics of classes of entities, (b) it makes the least possible commitment to the precision of any method for the discovery of equivalence relations among ontology elements, (c) it provides a basic method that can be further enhanced with other concepts' distinguishing features (e.g., concepts in a given vicinity), and can be further combined with other

---

[1] This work is part of research project ONTOSUM (www.ontosum.org), implemented within the framework of the "Reinforcement Programme of Human Research Manpower" (PENED) and co-financed by E.U.-European Social Fund (75%) and the Greek Ministry of Development-GSRT (25%).

alignment methods: This paper studies the potential of *CSR*, while leaving further enhancements and combinations for future work.

The machine learning approach has been chosen since (a) there are no evident generic rules *directly* capturing the existence of a subsumption relation between ontology elements (e.g., by means of their surface appearance) and (b) concept pairs of the same ontology provide examples for the subsumption relation, making the method self-adapting to idiosyncrasies of specific domains, and non-dependant to external resources.

Decision trees are used widely in classification problems, since they are robust to noisy data, to missing attribute values, and they are capable of learning disjunctive expressions [3]: Features that match to the subsumption computation problem. Weka's j48 [3] is the implementation of the widely used and state of the art C4.5 [4] decision tree learning algorithm that we have used in this work. C4.5 suits perfectly to problems with the above characteristics.

2 Problem Statement and Related Work

An ontology is a pair $O=(S, A)$, where S is the ontological signature describing the vocabulary (i.e., the terms that lexicalize ontology elements) and A is a set of ontological axioms, restricting the intended meaning of the terms included in the signature [5]. Considering a partition of S let us introduce the sets S_p and S_c, denoting the sets of terms lexicalizing ontology properties and ontology concepts, respectively.

Ontology mapping from a source ontology $O_1=(S_1,A_1)$ to a target ontology $O_2=(S_2,A_2)$ is a morphism $f{:}S_1 \rightarrow S_2$ of ontological signatures such that $A_2 \vDash f(A_1)$, i.e., all interpretations that satisfy O_2's axioms also satisfy O_1's translated axioms. However, instead of a function, we may articulate five different kinds of binary relations between the elements of source ontologies: Namely, equivalence ($\equiv$), subsumption (inclusion) ($\sqsupseteq$ or $\sqsubseteq$), mismatch ($\perp$) and overlapping ($\sqcap$). In this case, the ontology mapping problem is as follows: Classify any pair (C^1,C^2) of elements of the input ontologies, such that C^i is a term in S_i, $i=1,2$, to the above relations, consistently.

In this paper we deal with the *subsumption computation problem* which, given the above generic problem, is as follows: Given (a) a source ontology $O_1=(S_1, A_1)$ and a target ontology $O_2=(S_2, A_2)$ such that $S_1=S_{1c}\cup S_{1p}$ and $S_2=S_{2c}\cup S_{2p}$, and (b) a morphism $f{:}S_{1p} \rightarrow S_{2p}$ from the lexicalizations of the properties of the source ontology to the lexicalizations of the properties of the target ontology (computing properties' equivalences), classify each pair (C^1,C^2) of concepts, where C^1 is a term in S_{1c} and C^2 is a term in S_{2c}, to two distinct classes: To the "subsumption" ($\sqsubseteq$) class, or to the class "*R*". The class "*R*" denotes pairs of concepts that are not known to be related via the subsumption[2] relation, or that are known to be related via the equivalence, mismatch or overlapping relations.

Given the above stated problem, to the best of our knowledge only the *Semantic Matching* approach [6] deals with the computing of subsumption relations between concepts of ontologies. This method relies on codified knowledge contained in external dictionaries, and specifically in WordNet, transforming the available information into a propositional formula and solving a propositional satisfiability problem. Relations that do not satisfy the formula are filtered out and the remaining ones are returned in order of semantic strength.

In contrast to *Semantic Matching* method, *CSR* is a machine-learning based method that exploits the semantics of the input ontologies to assess the equivalence of properties and to generate the appropriate examples for the training of the classifier. This makes the proposed method independent from any third/external domain resource (lexicon or thesaurus).

3 The Classification-Based Learning of Subsumption Relations (*CSR*) Method

As it is shown in Fig 1, given a pair of ontologies $O_1=(S_1, A_1)$ and $O_2=(S_2, A_2)$, expressed in OWL-DL, the aim of the *CSR* method is to classify pairs of concepts (C^1,C^2), where C^1 is a term in S_{1C} and C^2 is a term in S_{2C}, either in the class "$\sqsubseteq$" – assessing the fact that the concept C^1 is subsumed by C^2 – or in the class "*R*".

---

[2] This means that a pair of concepts belonging to "*R*" may belong to the subsumption relation. In conjunction, "*R*" includes concept pairs that are not related via the subsumption relation (e.g., disjoint concepts).

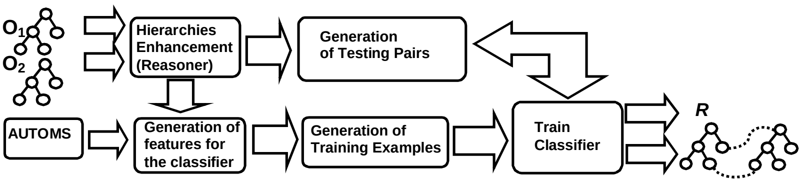


Fig. 1. Overview of the *CSR* method.

The discrete steps of the *CSR* method, as depicted in Fig. 1, are the following:

- Reasoning services are being used for inferring all facts according to ontologies' specification semantics [7]. This is a necessary step as it affects the generation of the training dataset.
- The generation of the features is performed by gathering all discrete properties from both ontologies:

Given properties' equivalencies computed by the AUTOMS mapping tool [8], each pair of concepts (C^1,C^2) is represented by a vector whose components range in {0, 1, 2, 3}. Specifically, the value of a feature is (a) "0", if the corresponding property does not appear neither in C^1 nor C^2, (b) "1", if the corresponding property appears only in C^1, (c) "2", if the corresponding property appears only in C^2, (c) "3", if the corresponding property appears in both C^1 and C^2.

It must be emphasized that (a) by "property appearance" we do not mean the occurrence of the property's lexicalization, but the occurrence of property's meaning as it is assessed by AUTOMS, and (b) feature vectors are not identical for symmetric pairs of concepts, allowing the computation of the direction of the subsumption relation.

- The sets of training examples are being generated according to the generic rules specified in the following. This step includes the balancing of the training dataset, as well.

Generating the training examples for the class "$\sqsubseteq$". This set of examples contains all the stated and inferred subsumption relations among the concepts in each of the source ontologies.

Generating the training examples for the class "***R***". According to the open world semantics, in case there is not an axiom that specifies the subsumption relation between a pair of concepts (or in case this relation can not be inferred by exploiting the semantics of the subsumption relation), then this pair does not belong to the subsumption class and it is included in the generic class "*R*". Training examples for the class "*R*" is further enriched by taking into account (a) the stated equivalence relations between concepts, and (b) by exploiting the union construct: Concepts defined as the union of others, can be substituted by any of their constituents.

The result of the above rule is the definition of four different training example categories for class "*R*", defined as follows: (a) *Concepts belonging to different hierarchies*, (b) *siblings at the same hierarchy level* for which no subsumption relation is defined or inferred, (c) *siblings at different hierarchy levels*: If any concept that is in a pair belonging in the "siblings of the same hierarchy level" category is substituted by any of its subsumees, then new pair examples are recursively generated, until the leaf concepts of the ontology are reached. Finally, (d) *concepts related via an object property* in case no subsumption relation is defined or inferred between them.

Creating a Balanced Training Dataset. It is very important for the performance of the classifier that the training examples are balanced: The number of training examples of the two classes must be equal, forcing both categories to be equally represented. This is referred as the *dataset imbalance* problem. Considering the various techniques the have been proposed towards its solution [9], we have adopted an under-sampling method:

1. All the generated training examples for the class "$\sqsubseteq$" are used.
2. Duplicate examples across different categories of class "*R*" are removed.
3. Select randomly *n/t* examples for each category of training examples of the class "*R*", where *n* is the number of examples of class "$\sqsubseteq$" and *t* is the number of different categories of class "*R*".

Given that examples are chosen randomly, the under-sampling method introduces non-determinism into the learning process. Furthermore, as shown in step 3 above, all the different types of example categories are equally present among the example pairs for class "*R*". This is of paramount importance, as a "good" classifier must learn to identify all the different types of examples.

Subsequent steps are as follows:

- The classifier is being trained using the training dataset.
- Concept pairs are being classified by the trained classifier, pruning the search space.

In order to prune the search space, the proposed algorithm firstly checks all the concepts from the first ontology with the root concepts (concepts with no subsumer) and unit concepts (root concepts with no subsumees) of the second ontology. If a pair is not classified in the class “⊑”, then the hierarchy rooted by the corresponding concept of the second ontology is not being examined by the classifier. If a pair is assessed to belong to the class “⊑”, then the concept of the first ontology is recursively being tested with the direct subsumees of the corresponding concept in the second ontology, until either a pair is assessed to belong in the class “*R*”, or until the leaf concepts are reached.

4 Experimental Results and Discussion

The testing dataset has been derived from the benchmarking series of the OAEI 2006 contest [10]. As our method exploits the properties of concepts, we do not include ontologies with no properties. Hence, the compiled corpus consists of 31 out of the 51 OAEI 2006 ontologies, and it is available at the URL http://www.icsd.aegean.gr/incosys/csr, together with the gold standard created. All benchmarks (101-304) except *R1-R4*, define the second ontology of each pair as an alteration of the same first. The benchmarks can be categorized based on their common features as follows: (a) in *A1-A5* (101-210, 237, 238 and 249) elements’ lexicalizations of the target ontologies are altered in various ways (e.g., uppercasing, underscore, foreign language, synonyms and random strings), (b) in *A6-A7* (225 and 230) restrictions are removed and/or properties are modeled in more detail, (c) in *F1-F2* (222, 237, 251 and 258) the hierarchies are flattened and/or random lexicalizations of elements are introduced, (d) in *E1-E2* (223, 238, 252 and 259) the same as *F1-F2*, but the hierarchies are expanded and (e) in *R1-R4* (301-304) target ontologies are real world ontologies.

Due to the non-determinism introduced by the under-sampling method used, for each ontology pair the experimental results have been produced by applying the *CSR* method 20 times. The set of evaluation values produced during the experiments are visualized by using boxplots [11].

Results show the precision and recall of the proposed method as it is applied in the different types of ontology pairs. Precision is the ratio *#correct_pairs_computed*/ *#pairs_computed* and recall is the ratio *#correct_pairs_computed*/*#pairs_in_gold_standard*. *CSR* is compared with a baseline classifier which is based on the Boolean Existential Model (BEM), in order to show *CSR*’s ability to generalize successfully from the training examples. The baseline classifier consults the training examples of the class “⊑”, testing whether each testing pair matches exactly to any of the training examples.

Fig. 2 and Fig. 3 depict the boxplots for the precision and recall of *CSR* for the various benchmark categories. The *CSR* method, while trying to generalize, takes into account the training examples of both classes. This, in conjunction to the fact that the feature vectors are not the optimum (due to errors in the mapping of properties), there are cases where feature vectors of examples for the class “*R*” are the same with examples for the class “⊑” (this happens for instance in cases A7, F1 and F2). This affects the discriminating ability of the classifier, resulting to low precision. These problems do not apply to the baseline classifier: in cases where the testing concept pairs are almost the same with the training pairs this classifier is quite effective. The above argument regarding the behavior of *CSR* is strengthened by the results of cases A4, A5 and A6 where the mapping of properties is more difficult than in A1 to A3 where the testing concept pairs are almost the same with the training pairs. In cases A4 to A6, the training and testing examples are not completely identical (due to the replacement of properties’ and concepts’ labels with synonyms and due to the absence of comments in some ontologies, which affects the mapping of properties): In these cases *CSR* outperforms the baseline classifier, as it manages to generalize successfully from the training examples.

Categories R1 to R4 include real world ontologies: These cases clearly show that the *CSR* method generalizes over the training examples. For example, in the test case R1 the baseline classifier fails completely in terms of both precision (0%) and recall (0%), while the *CSR* method achieves 25% precision and 89% recall. In category R3 the *CSR* method performs poorly, because properties are defined only for the root concepts. As a result, there are many training examples with 0’s in their feature vectors that prevent the classifier for generalizing properly.

To further assess the quality of the classification method, we performed ROC analysis [16] (Fig. 6). In our case, ROC analysis considers the trade off between how “good” is the classifier in classifying testing examples in the distinct classes “⊑” and “*R*”. By examining the ROC area under line values of the *CSR* method in all test cases it is obvious that the classifier is always above being “fair” and in the majority of the test cases (9/15) can be characterized as “excellent”. It must be stated that these values depict that, although the performance of the classifier in the class “*R*” is of no evident interest for the ontology alignment problem, the *CSR* method performs even better there.

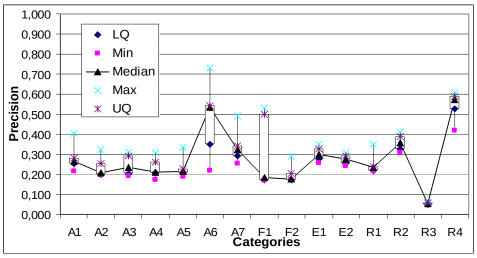


Fig. 2. Precision of *CSR* in various test cases.

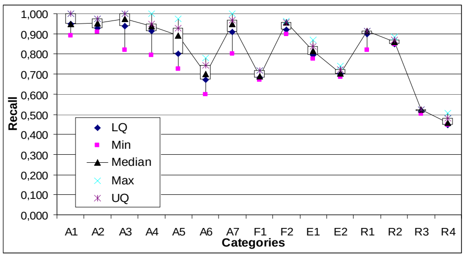


Fig. 3. Recall of *CSR* in various test cases.

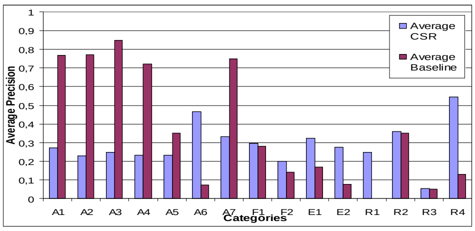


Fig. 4. Average precision of *CSR* and baseline classifier.

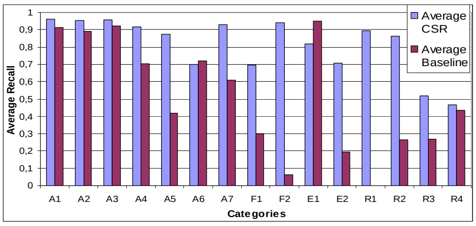


Fig. 5. Average recall of *CSR* and baseline classifier.

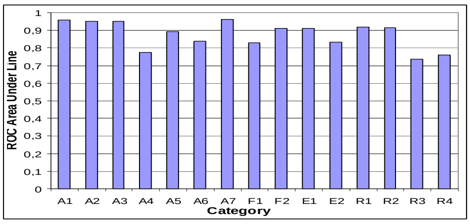


Fig. 6. ROC areas under line in all test cases.

5 Conclusions and Future Work

In this paper we propose the "Classification-Based Learning of Subsumption Relations for the Alignment of Ontologies" (*CSR*) method. Experimental results for computing subsumption relations over an extended version of the OAEI 2006 benchmarking series show the potential of the proposed method: *CSR* generalizes effectively over the training examples, showing (a) the importance of properties to assessing the subsumption relation between concepts of discrete ontologies (b) the importance of incorporating more precise property mapping methods into the process, (c) the potential to further improve the method via the incorporation of more types of features, and via its combination with other methods.

References

1. P. Shvaiko, J. Euzenat: A Survey of Schema-based Matching Approaches. Journal on Data Semantics, 2005.
2. O. Svab, V. Svatek, H. Stuckenschmidt: A Study in Empirical and 'Casuistic' Analysis of Ontology Mapping Results. In Proceedings of ESWC, 2007.
3. T. Mitchell: "Decision Tree Learning", in T. Mitchell, Machine Learning. The McGraw-Hill Companies, Inc., pp. 52-78, 1997.
4. Ross Quinlan: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo, CA (1993).
5. K. Kotis, G. Vouros, K. Stergiou: Towards Automatic Merging of Domain Ontologies: The HCONE-merge approach. Elsevier's Journal of Web Semantics (JWS). vol. 4:1, pp. 60-79 (2006), (Published Online First: 19 Okt. 2005).
6. F. Giunchiglia, M. Yatskevich, E. Giunchiglia: Efficient Semantic Matching. In Proceedings of ESWC, 2005.
7. Baader, F. Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds). Description Logics Handbook, Cambridge University Press, 2003.
8. K. Kotis, et al., AUTOMS: Automating Ontology Mapping through Synthesis of Methods, OAEI 2006 contest, Ontology Matching International Workshop, USA, 2006.
9. N. Japkowicz: The Class Imbalance Problem: Significance and Strategies. In Proceedings of IC-AI, 2000.
10. Ontology Alignment Evaluation Initiative. http://oaei.ontologymatching.org/2006/newindex.html.
11. J.W. Tukey. "Exploratory Data Analysis". Addison-Wesley, Reading, MA. 1977.

Ontology Merging using Answer Set Programming and Linguistic Knowledge

Jürgen Bock[1], Rodney Topor[2], and Raphael Volz[1]

[1] FZI Forschungszentrum Informatik, Karlsruhe, Germany
{bock, volz}@fzi.de
[2] Griffith University, Brisbane, Australia
r.topor@griffith.edu.au

Abstract. With the increasing number of ontologies available on the web, the problem of merging ontologies from different sources to interoperate applications becomes important. This paper presents a novel approach for merging of light-weight ontologies based on answer set programming (ASP) and linguistic background knowledge. ASP provides a declarative execution environment for intuitive merging rules. WordNet provides broad linguistic knowledge that is used to identify corresponding concepts. We present a semi-automatic merging algorithm, where users can choose appropriate results from a set of suggestions.

1 Introduction

Many semantics-based applications are isolated applications utilising single ontologies to improve data access and navigation.

The popularity of the Web 2.0 theme has brought high attention to so-called "mashups", where data from several applications is combined to provide novel applications. For example, `housingmaps.com` combines map data with real estate information to depict houses for sale on a map. Beyond the Web, ontology merging is also a fundamental task in enterprise data integration. In many cases data interoperation goes beyond mapping-based translation of data between applications and a recent trend in large enterprises is to create meta-databases which holds master data about the schemas (or ontologies) of all applications in the enterprise.

This paper[3] addresses ontology merging, *i.e.* creating a single, coherent ontology out of several different ones, and presents a novel approach for merging of light-weight ontologies. We present a semi-automatic merging algorithm, where users can choose appropriate results from a set of suggestions. Our techniques are based on combining answer set programming (ASP) with linguistic background knowledge, which brings several benefits for ontology merging:

---

[3] We thank Kewen Wang for useful comments and Roman Schindlauer, the main developer of dlvhex, for useful information and fruitful web chats. Early discussions with Marilyn Ford have been very helpful to identify this project.

- Correspondence detection: Linguistic knowledge is used to detect correspondences between concepts based on synonymy
- Merging options: ASP calculates several answer sets which provide merging options among which the user can choose
- Extensibility: We provide intuitive merging rules that can be easily extended to capture domain-specific extensions

While our general approach is not language specific, our implementation currently deals with WordNet [1] as a basic broad linguistic resource that is used to identify corresponding concepts. Due to our observation on the prevalence of light-weight ontologies, we concentrate on merging light-weight ontologies based on the recently proposed SKOS standard for controlled vocabularies.

The problem of ontology merging has been addressed by several authors and the use of linguistic information is common in these approaches.

The PROMPT Suite [2] is a collection of tools, available as a plugin for the Protégé ontology editor. It incorporates lexical and (optionally) linguistic knowledge to identify similar or synonymous entities. However, the PROMPT approach utilisies linguistic resources only to a limited extent and requires a high degree of user-interaction.

Ehrig and Sure [3] suggest 17 rules to gain similarity measures between a number of ontologies to be mapped. Since the rules were "manually formulated by domain experts", this approach shows how explicit encoding of intuitive rules works well for the ontology merging task.

Wang *et al.* [4] use ASP for ontology merging and alignment of expressive DL-programs [5]. The paper focuses solely on conflict resolution and maintaining of consistency throughout the merging/alignment process. We pick up the idea of using ASP, but follow a more practical approach.

2 Answer Set Programming

We choose Answer Set Programming (ASP) [6] as implementation language for several reasons. Firstly, it allows very compact encodings for complex problems, such as the graph colouring problem. Secondly, it is a purely declarative programming paradigm, which allows to formulate the problem in terms of "what" should be done, instead of "how" to compute the solution.

Intuitively, ASP programs are a set of three basic constructs: facts (*e.g.*: $light$), rules (*e.g.*: $light \leftarrow switchOn$), and constraints (*e.g.*: $\leftarrow light, daytime$). With particular respect to DLV, atoms can be default (`not` $light$) or classically negated ($\neg light$), and disjunction can occur in rule heads. Refer to Baral [7] for a formal account.

Using dlvhex as an extension to DLV, we were able to provide an external atom to access the WordNet database[4]. The external atom was designed to be useful for a wider range of applications beyond the scope of this merging algorithm.

---

[4] `http://con.fusion.at/dlvhex/download.php`

3 Ontology Merging using ASP and Linguistic Knowledge

The fundamental assumption behind our algorithm is to provide a semi-automatic merging approach, which presents a number of different possible merging solutions to the user, who can finally choose the one(s) best suitable for her needs. The ontology merging algorithm itself is designed to follow intuitive rules to make it not only easy to understand but also extensible. The rule set incorporates information given by the structure of the ontologies to be merged, as well as additional linguistic background knowledge and respects the following issues:

- Exploit *linguistic information*, since ontologies typically follow a human knowledge model and entities are labeled in natural language terms.
- Allow or forbid different *ontology structures*, such as trees or DAGs.
- Respect *explicit domain knowledge* provided by the user via certain flags or parameters.
- Provide an option for *brave merging* to further reduce the number of merging suggestions.

3.1 Formal Design

For two concepts c_1 and c_2, by $c_1 \simeq c_2$ we denote that their labels are identical or synonyms. $c_1 \prec c_2$ denotes that a label of c_1 is linguistically narrower (*i.e.* a direct hyponym or meronym in this approach) of a label of c_2. By $c_1 \leq_C c_2$ we denote that c_1 is defined narrower than c_2 in one of the input ontologies.

Concept Melding. Intuitively, two concepts of different ontologies can be melded, if any of their labels are identical or synonyms[5].

$$meld(c_1, c_2) \ \vee \ \neg meld(c_1, c_2) \quad \leftarrow \quad c_1 \simeq c_2 \tag{1}$$

$$\leftarrow \quad c_1 \simeq c_2, \ \text{not } c_1 \prec c_2, \ \text{not } c_2 \prec c_1, \ \neg meld(c_1, c_2) \tag{2}$$

Since some words can be synonyms, as well as in a linguistical narrower relation, they do not necessarily have to be melded. This can be expressed by rule (1). However, if c_1 and c_2 are not in a linguistic narrowing relation, they must be melded, which will be forced by constraint (2).

Hierarchy Restructuring. Intuitively, two concepts can be merged in a potential narrowing relation, if they are in a narrowing relation in one of the input ontologies, or if they are in different ontologies but in a linguistical narrowing relation.

---

[5] The rules presented in this section are denoted in a formal and simplified way to demonstrate basic ideas of the algorithm. They violate rule safety and other restrictions and cannot be implemented straightforwardly. Definitions of auxiliary atoms are omitted. Please refer to [8] for the full translation to the implemented set of safe rules.

For ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ and $i, j \in \{1, 2\}$

$$pot\_narr(c_2, c_1) \leftarrow c_1 \leq_C c_2 \quad (c_1, c_2 \in \mathcal{O}_i) \tag{3}$$

$$pot\_narr(c_2, c_1) \leftarrow c_1 \prec c_2 \quad (c_1 \in \mathcal{O}_i,\ c_2 \in \mathcal{O}_j,\ i \neq j) \tag{4}$$

$$m\_narr(c_1, c_2) \lor \neg m\_narr(c_1, c_2) \leftarrow pot\_narr(c_1, c_2) \tag{5}$$

Firstly, narrowing relations given by the input ontologies and linguistic information are collected in rules (3) and (4). (Without loss of generality, only atomic concepts are considered in these rules, *i.e.*, concepts that are not yet melded.) However, a reasonable merging will not contain all potential narrowing relations due to transitivity and adjustable restrictions to the final structure (see later in this section). Therefore, a second step identifies appropriate subsets of all potential narrowing relations to form the different merging proposals. The main idea is, to either pick a potential narrowing relation for the final merging, or not (rule (5)).

3.2 User Guidance

To restrict the (so far exponential) number of possible merging solutions to reasonable ones, constraints are used, which are enabled by several flags, set by the user, namely *no singles*, *one root*, *single parent*, and *always meld*.

Let $b(c)$ and $n(c)$ be atoms containing all concepts c that are chosen to be merged as a broader ($b(c)$) or narrower ($n(c)$) concept to any other. A root concept $root(c)$ is defined as a concept that is broader but not narrower than any other concept, or an isolated concept. Let $chained(c, d)$ be the transitive closure of potential narrowing relations that form a chain of at least three concepts.

The following constraints are enabled by the according flags:

$$\leftarrow \mathsf{not}\ b(c),\ \mathsf{not}\ n(c),\ no\_singles \tag{6}$$

$$\leftarrow root(c),\ root(d),\ c \neq d,\ one\_root \tag{7}$$

$$\leftarrow m\_narr(c, e),\ m\_narr(d, e),\ c \neq d,\ single\_parent \tag{8}$$

$$\leftarrow \neg meld(c_1, c_2),\ c_1 \simeq c_2,\ always\_meld \tag{9}$$

$$\leftarrow m\_narr(c, d),\ pot\_narr(c, d),\ chained(c, d),\ brave \tag{10}$$

The *no singles* constraint (6) does not allow concepts that do not occur as either a broader, or a narrower concept. The *one root* restriction (7) constrains the possible merging solutions to those where only one root concept exists. Note that *one root* implies *no singles*, since every single node is a root. Ontologies, that are organised in a tree structure require concepts to be connected only by one single incoming narrowing relation. This can be enforced by setting the *single parent* flag, which enables constraint (8). The *always meld* constraint (9) disables the generation of multiple answer set by simple disallowing meldable concepts not to be melded. This can reduce the number of merging suggestion drastically. The *brave* constraint (10) makes the algorithm greedy in terms of preferring a more nested hierarchy to a flatter one. This is achieved by preferring longer chains of narrowing relations to shortcuts of a single narrowing edge.

4 Conclusion

We have presented a novel approach to ontology merging, incorporating natural language background knowledge and a direct implementation of intuitive merging rules. For the algorithm, a number of requirements have been identified, and formally translated into declarative rules for ASP. This formal design could be implemented straightforwardly in only 56 logical lines of code (cf. [8]) for dlvhex, which we extended by a new external plug-in to deal with WordNet. The algorithm addresses the two main aspects of concept melding and hierarchy restructuring. A possible merging contains a reasonable subset of these potential narrowing relations by constraining possible solutions according to a number of (adjustable) conditions.

This work focused mainly on providing a new methodology of computing ontology mergings, rather than producing an off-the-shelf application.

Early experiments have shown promising results for various flag combinations, and further evaluations will be conducted. Future work also includes the extension and refinement of the algorithm, such as the use of weak constraints and aggregates. These would allow for extensions, such as checking for multi-level linguistic relations, ordering answer sets and discarding answer sets below a certain confidence threshold, or the use of the linguistic resource also to augment the merging results by missing concepts. Furthermore, a modularisation of the algorithm would allow for user interaction for crucial decisions in disjunctive rules.

References

1. Miller, G.A.: WordNet: A Lexical Database for English. Communications of the ACM **38**(11) (1995) 39–41
2. Noy, N.F., Musen, M.A.: The PROMPT Suite: Interactive Tools For Ontology Merging And Mapping. Int. J. Hum.-Comput. Stud. **59**(6) (2003) 983–1024
3. Ehrig, M., Sure, Y.: Ontology Mapping - An Integrated Approach. In: Proc. 1st European Semantic Web Synopsium (ESWS). (2004) 76–91
4. Wang, K., Antoniou, G., Topor, R.W., Sattar, A.: Merging and Aligning Ontologies in dl-Programs. In: Proc. 1st International Conference on Rules and Rule Markup Languages for the Semantic Web. (2005) 160–171
5. Eiter, T., Lukasiewicz, T., Schindlauer, R., Tompits, H.: Combining Answer Set Programming with Description Logics for the Semantic Web. Technical Report INFSYS RR-1843-03-13, Institut für Informationssysteme, Technische Universität Wien, A-1040 Vienna, Austria (December 2003)
6. Gelfond, M., Lifschitz, V.: Classical Negation in Logic Programs and Disjunctive Databases. New Generation Computing **9**(3–4) (1991) 365–386
7. Baral, C.: Knowledge Representation, Reasoning and Declarative Problem Solving. Cambridge University Press (2003)
8. Bock, J.: Ontology Merging using Answer Set Programming and WordNet. Honours Thesis, Griffith University (October 2006)

Ontology Hierarchies Matching by Lattices Alignment*

Hu He[1][2] and Du Xiaoyong[1][2]

[1] School of Information, Renmin University of China, 100872, Beijing, China
[2] Key Lab of Data and Knowledge Engineering, MOE, 100872, Beijing, China
`{hehu,duyong}@ruc.edu.cn`

Abstract. Ontology matching determines the correspondences between concepts and relations of related ontologies. In this paper, we put forward an ontology hierarchies matching approach based on lattices alignment. The proposed lattice-based matching algorithm can be utilized not only in matching processes between two ontologies, but also in annotation processes between an ontology and its corresponding resources. Experiments on spatiotemporal ontology annotation have been carried out which shown the applicability of the approach.

1 Motivations

Ontologies are formal, explicit specifications of share conceptualizations [1]; they provide a formal way to describe concepts and their relations for a specified domain. The Semantic Web vision [2] has greatly promoted people's interest in ontologies. More and more ontologies are put forward by different groups and individuals. As many ontologies for the same domain appearing on the Web, a quantitative evaluation method is needed to discriminate between these ontologies so that we can find the most appropriate one for specified applications.

Large-scale use of ontologies in knowledge discovery and semantic web has stimulated automatic ontology learning and population, with various machine-learning methods applied in these efforts. The evolution of these ontologies also needs to be evaluated quantitatively so that a good evolution can be distinguished from a bad one.

We advocate a lattice representation and assessment algorithm for comparing ontology hierarchies quantitatively in the paper. This approach is based on lattice alignment and can be used for ontology matchmaking, clustering, comparisons and annotations in Semantic Web enabled applications. The proposed lattice based metric can be used both in ontologies matching and ontology annotation between ontologies and their corresponding resources. It has unique advantages comparing with existing measures and algorithms as it provides a unified method which takes into account not only concepts but also the relations between concepts in ontologies.

---

* This work is supported by China NSFC programs (60496325, 60573092, 60603020).

2 Related Work

Maedche [3] et. al. consider ontologies as two-layered systems, consisting of a lexical and a conceptual layer. They use a set of ontology similarity measures and compare two ontologies separately with concepts and relations. OntoMetric [4] gets for every candidate ontology a quantitative measure of its suitability using: a multilevel framework of 160 characteristics that describe the ontology domain. The specialization of the characteristics and the assessment of the criteria of a particular ontology require considerable effort and thus limit its application for novel users. To use the method, the engineer must compare the importance of the project objectives, and study carefully the characteristics of ontologies. Brewster [5] et. al. use a vector-space model of instances (terms) in a corpus and an ontology to give a measure of the "fit" between the ontology and the corpus (domain of knowledge). The method proposed by Brewster is straightforward and easy to use. But it has some drawbacks too, the method loses structure relations information and uses only the lexical information.

The lattice metric in [7] extends Brewster's works and convert ontologies to be evaluated into lattice structures; based on an algorithm for finding identical concepts, we align two lattices to the same dimension; a traditional Vector Space Model (VSM) [6] can be used afterwards to measure the differences quantitatively. Moreover, we will show that the lattice metric approach can be utilized not only in matching processes between two ontologies, but also in annotation processes between an ontology and its corresponding resources.

3 Lattice based Ontology Hierarchies Matching

3.1 Ontology Lattice

Definition 1 Hierarchy $H(S, \leq)$**:** Suppose $(S, \prec)$ is a partially ordered set. A hierarchy $H(S, \prec)$ for $(S, \prec)$ is the Hasse diagram for $(S, \prec)$, which is a directed acyclic graph whose set of nodes is S and has a minimal set of edges such that there is a path from u to v in the Hasse diagram iff $u \prec v$.

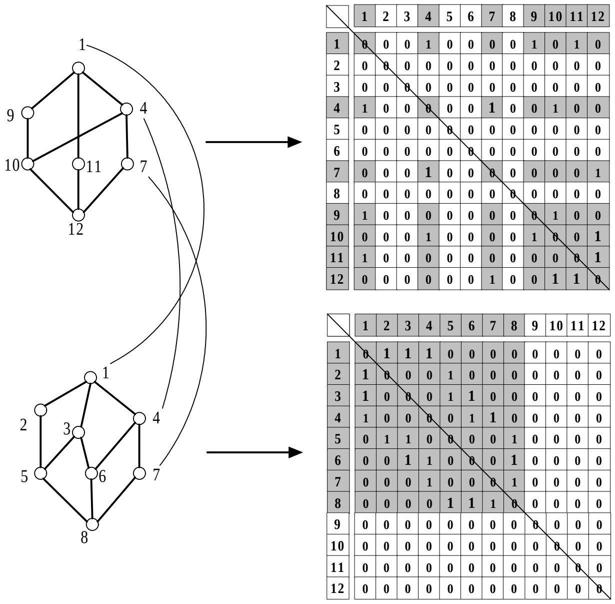
Definition 2 Ontology $O(C, R, H)$**:** A ontology is represented as $O(C, R, H)$, where C is a set of concepts $\{c_1, c_2, ..., c_i\}$, R is a set of relationships $\{r_1, r_2, ..., r_j\}$, and H is a set of hierarchies H(C,r). There is a root in H(C,r) which is the most abstract concept in C.

Definition 3 Concept Depth $depth(c)$**:** Define the depth of a concept c node(denoted as depth(c)) in a hierarchy H(C,r) of ontology $O(C, R, H)$ is the number of edges on the path from the root of O to that concept node.

Definition 4 Ontology Lattice: For any particular domain $\mathcal{D}$, and a hierarchy relation $\mathcal{H}$, we use $\prec$ to represent the $\mathcal{H}$ relation: for any two concepts C_1, C_2 satisfies $\mathcal{H}$ (C_1, C_2). We have $C_1 \prec C_2$. Then $(\mathcal{D}, \prec)$ forms an ontology lattice. Detailed information about lattice construction and alignment algorithm is referred to [7].

3.2 VSM Representation

In most cases, the matrices of two ontologies (or resource and ontology in annotation process) to be matched are not in the same dimensions; to use matrix based comparison methods such as VSM, the matrices must be transformed into the comparison space of the same dimension. After the matrices being transformed into the same dimension space(in Fig. 1), the traditional model of VSM can be used. The measure of two Ontology lattices `A` and `B` is formulated as: Diff(`A`,`B`)=(Vect(`A`) $\cdot$ Vect(`B`)) / |Vect(`A`) $\cdot$ Vect(`B`)|. The similarity between lexical entries follows the edit distance formulated by Levenshtein[8] which is a well-established method for weighting the difference between two strings.



	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	0	1	0	0	0	0	1	0	1	0
2	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0
4	1	0	0	0	0	0	1	0	0	1	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	1	0	0	0	0	0	0	0	1
8	0	0	0	0	0	0	0	0	0	0	0	0
9	1	0	0	0	0	0	0	0	0	1	0	0
10	0	0	0	1	0	0	0	0	1	0	0	1
11	1	0	0	0	0	0	0	0	0	0	0	1
12	0	0	0	0	0	0	1	0	0	1	1	0

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	1	1	1	0	0	0	0	0	0	0	0
2	1	0	0	0	1	0	0	0	0	0	0	0
3	1	0	0	0	1	1	0	0	0	0	0	0
4	1	0	0	0	0	1	1	0	0	0	0	0
5	0	1	1	0	0	0	0	1	0	0	0	0
6	0	0	1	1	0	0	0	1	0	0	0	0
7	0	0	0	1	0	0	0	1	0	0	0	0
8	0	0	0	0	1	1	1	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 1: Aligned Lattices and their VSM Representations.

4 Case Study

4.1 Spatiotemporal Ontology Annotation

Spatial and temporal information constitutes a most elementary part of our everyday life. The representation and reasoning of spatial and temporal knowledge remain an important field in artificial intelligence research. Because much of spatiotemporal information is scattered in free texts, they can not be easily extracted. NLP techniques such as lexical, part of speech, syntactic and semantic reprensentation formalisms are often used in unstructured documents analysis; but the analysis results can be read and understand only by human beings not machines, making it difficult to apply on vast amount of information on the WWW. Spatiotemporal ontology annotation is in urge need to solve these problems.

To prove the correctness and practicability of the lattice-based metric, we choose the spatiotemporal ontology as the experimental data and refine the obtained ontological structure based on Classified Chinese Library Thesaurus, which classifies the controlled vocabulary in particular domains and is adopted widely for organizing literature resources.

The lattice-based matching method is applied to evaluate the spatiotemporal ontology and its corresponding resources; for a particular spatiotemporal resource, we align the resource to the spatiotemporal ontology and construct the lattice structure representation as depicted in Fig. 2; then the resource lattices are compared with the ontology lattice to determine the most appropriate annotation concept, which is the annotation result of the current resource. In practice, we use concept depth value of 3 and choose the maximal matching concept from the ontology as the annotation result for the resource.

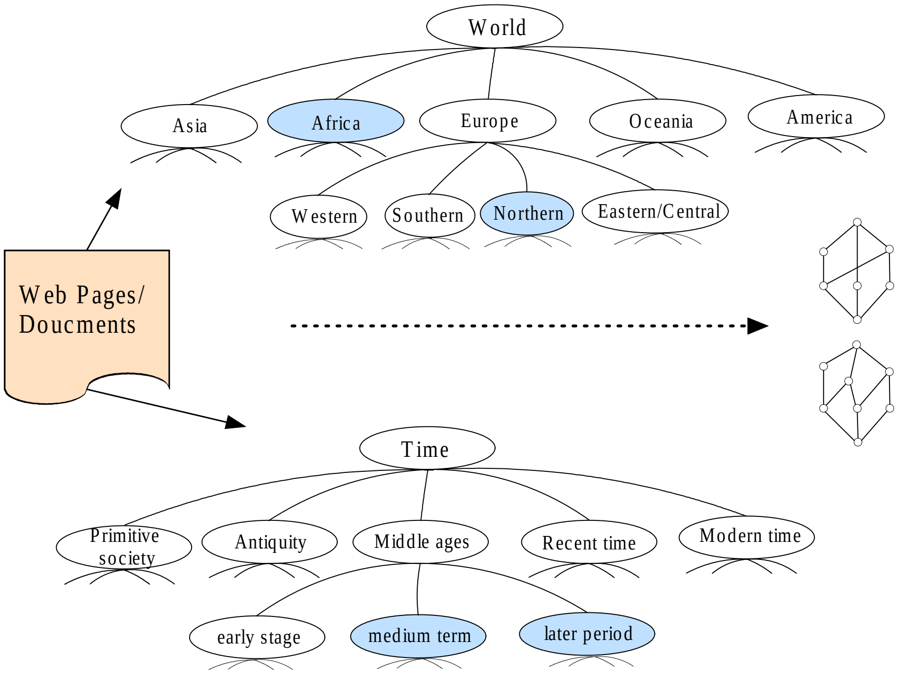


Fig. 2: The Resource Lattice Construction.

4.2 Algorithm Efficiency

Let n be the number of **A**'s nodes, and m be the number of **B**'s nodes; the matrix alignment algorithm need n*m loops of **Find-Identical-Concepts**. Let k be the average number of children nodes for **A** and **B**, the lattice comparison complexity will be n × m × 2k. We can see the algorithm is polynomial in time complexity, which indicates its' efficiency in real world applications.

5 Conclusion

We propose a unified lattice based approach for ontology hierarchies matching tasks and ontology annotation tasks. A unique quality of this measure method is that it combines the concepts and hierarchy relations into a unified structure: a lattice. By aligning two different lattices, the traditional vector space model can be used in the matching processes. The presented lattice alignment algorithm can be utilized not only in matching processes between two ontologies, but also in annotation processes between an ontology and its corresponding resources. Experiments on spatiotemporal ontology annotation have been carried out which shown the applicability of the approach.

References

1. Gruber T, (1995), Towards principles for the design of ontologies used for knowledge sharing, International Journal of Human-Computer Studies 43 (5/6): 907-928
2. Tim Berners-Lee, James Hendler and Ora Lasilla, (2001), The Semantic Web, The Scientific American.
3. A. Maedche and S. Staab, Measuring Similarity between Ontologies,(2002), In: Proc. Of the European Conference on Knowledge Acquisition and Management (EKAW 2002), Madrid, Spain
4. Lozano-Tello, A. and Gmez-Prez, A., (2004), ONTOMETRIC: A Method to Choose the Appropriate Ontology, Journal of Database Management, Special Issue on Ontological analysis, Evaluation, and Engineering of Business Systems Analysis Methods, Volumen: 15(2)
5. Christopher Brewster, Hartith Alani, Srinandan Dasmahapatra, and Yorick Wilks, (2004), Data driven ontology evaluation, in Proceedings of LREC 2004, Lisbon
6. G. Salton, M. Lesk, (1968), Computer evaluation of indexing and text prcessing, Journal of the ACM, 12(1), pp.8-36
7. H. Hu, X. Du, (2007), A Lattice Metric for Evaluating Ontology Hierarchies, In: Proceedings of International Conference of Machine Learning and Cybernetics (ICMLC 2007), pp.3880-3883, Hongkong
8. Levenshtein, I.V., (1966), Binary codes capable of correcting deletions, insertions, and reversals. Cybernetics and Control Theory, 10(8):707-710

The 6th International Semantic Web Conference and
the 2nd Asian Semantic Web Conference

November 11~15 2007
BEXCO, Busan KOREA