A Generic Approach for Combining Linguistic and Context Profile Metrics in Ontology Matching*

DuyHoa Ngo, Zohra Bellahsene, Remi Coletta

LIRMM, Univ. Montpellier 2
34392 Montpellier, France
{firstname.name@lirmm.fr}

Short paper

Abstract. Ontology matching is needed in many application domains. In this paper, we present a machine learning approach for combining metrics, which exploits various linguistic and context profiles features in order to discover mappings between entities of different ontologies. Our approach has been implemented and the experimental results over Benchmark and Conference test cases on OAEI 2010 campaign[1] demonstrate its effectiveness and efficiency in terms of quality of matching and flexibility.

Keywords: Ontology matching, matcher combination, context profile, linguistic metrics

1 Introduction

Numerous similarity metrics have been proposed so far for ontology mapping. According to [3], element metrics can be categorized in three groups: terminological, structural and semantic matching-based techniques. Metrics in the first group exploit text features such as name, labels and comments to calculate the similarity score between entities; whereas metrics of the last two groups exploit the hierarchy and semantic relations features.

Despite the fact that metrics in the first group are less semantic than that of the second and third, they are widely used in most of the matchers. During the matching process, mappings discovered by these metrics can be used as input mappings to other metrics of the second and third groups. Obviously, the more precise results terminological metrics are the more accurate results structural and semantic metrics have. Therefore, the aim of designing a high performance quality matcher exploiting terminological features becomes an importance task.

Due to the various types of heterogeneity of data sources, there is no single best metric overall matching scenarios. It is beneficial and necessary to combine several methods for improving matching quality. However, it is very difficult and

* Supported by ANR DataRing ANR-08-VERSO-007-04.

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

time consuming even for experts to find a good combination. Therefore, the use of supervised and machine learning approache is a promising way in order to reduce the required manual effort.

According to these necessity, we aim to build a high quality ontology matcher which utilizes machine learning approach to combine terminological similarity metrics. This matcher will be a premise for us to work in next step with structural and semantic matching methods.

The main contributions of this paper are the following. (i) We propose metrics dealing with terminological and context profile features of entities in ontology. (ii) We propose to use decision tree model to combine similarity metrics and strategies to select metrics and training data for the learning process. (iii) Experimental results performed on the benchmark and conference tests of OAEI 2010 campaign show that our system achieved stable and good results in comparison with other participants.

This paper is organized into 5 sections. Section 2 presents similarity metrics working with terminological and context profile features. Next, Section 3 contains our approach for combining similarity metrics. In Section 4 we describe the setting of experiments and show experimental results. Finally, we conclude and plan future work in Section 5.

2 Feature Extraction and Similarity Metrics

2.1 Similarity Metrics for Terminological Features

Terminological features of an entity consist of text information encoded in itself in ontology such as entities' URI(name space and local name), labels and comments. Terminological metrics can be categorized in two main groups: string-based and linguistic-based. String-based metrics take advantage of similar characters from two strings, whereas, linguistic-based metrics compare the meaning of strings.

Most of string-based metrics (e.g. Levenstein, SmithWaterman, JaroWikler, Qgrams, MongeElkan, etc.) are taken from open-source libraries SeconString[2] and SimMetric[3]. Additionally, we also implemented other string-based metrics such as Equality, Prefix, Suffix, Longest Common SubString [3] and Stoilois [8]. To deal with linguistic features, we implemented Lin, JiangConrath and Wu-Palmer [6] metrics working on WordNet[4] dictionary.

Because ontologies are designed by different people, consequently, names or labels indicating even to the same object or concept may be heterogeneous. For example, *"MscThesis"* and *"Ms.dissertation"* are different but they both indicate a master's thesis. Due to the heterogeneity of naming convention, primitive string-based or linguistic-based metrics mentioned above may be not sufficient.

[2] http://secondstring.sourceforge.net
[3] http://sourceforge.net/projects/simmetrics
[4] http://wordnet.princeton.edu

In order to deal with the heterogeneity problem, we perform analyses on terminological features of entities and propose solutions for each case.

Firstly, labels and names of entities usually are compound of tokens. The whole strings may be not matched but their tokens may be highly similar. Therefore, a pre-processing procedure is needed to split a string into proper tokens. Afterward, tokens can be compared by primitive string and linguistic metrics.

Secondly, tokens may exist in various types of morphological forms of a word. To deal with this issue, we need a thesaurus or dictionary. In our system, we propose a generic algorithm to combine string and linguistic metrics at token level as follows:

In the Algorithm 1, function MorphologicalForms takes a token as input and finds all possible senses and morphological forms of token existing in Wordnet dictionary. For example, MorphologicalForms(*"published"*) returns { verb: *"publish"* , adjective: *"published"*}; MorphologicalForms(*"publishing"*) returns { noun: *"publishing"*, verb: *"publish"*}. Because two obtained sets of senses have a common {verb: *"publish"*}, therefore token *"published"* and token *"publishing"* are similar.

Algorithm 1: COMPUTE SIMILARITY BETWEEN TWO TOKENS

Input: $token_1, token_2$ two tokens,
$dictMetric$ a linguistic metric,
$stringMetric$ a string metric
Output: $score$ a numerical value

$MF_1 \leftarrow MorphologicalForms(token_1)$
$MF_2 \leftarrow MorphologicalForms(token_2)$
if $(MF_1 \neq \emptyset) \wedge (MF_2 \neq \emptyset)$ **then**
$score \leftarrow \max_{(pos_i, st_i) \in MF_1, (pos_j, st_j) \in MF_2}(dictMetric(st_i, st_j) \mid pos_i = pos_j)$
else $score \leftarrow stringMetric(token_1, token_2)$

Next, similarity score between two set of tokens is computed by two modifications of MongeEklan algortihms. One is proposed in [5] and the second is our proposal:

$$sim_m(a, b) = \frac{1}{|a|} \sum_{i=1}^{|a|} \big(sigmoid(max\{sim(a_i, b_j)\}_{j=1}^{|b|})\big) \quad (1)$$

Here, $sigmoid(x) = \frac{1}{1+e^{-10\times(x-0.5)}}$ is a promoted function which makes the higher similar tokens is more informative than the lower ones. The idea and effectiveness of using promoted function can be seen in [5] for more detail.

Thirdly, name of entity may be an abbreviation (e.g. *"Misc."* instead *"Miscellaneous"*), an acronym (e.g. *"SW"* instead *"Semantic Web"*) or even a sequence of symbols which is not understandable. To deal with these cases, we expect that entities provide some human-readable labels in annotation information. In our approach, a local name is treated as a label of entity. The similarity measure between two entities based on their labels can be formulated as follows:

$$sim(e_i,\, e_j) \;=\; max_{(l_p \in labels(e_i),\, l_q \in labels(e_j))}(sim(l_p,\, l_q)) \quad (2)$$

For example, class **Chapter** in ontology #101 and class **dzqndbzq** in ontology #201[5] have the same label *"BookPart"*. Therefore two classes are matched.

An entity may also have several comments. They usually consist of a long descriptive text. Therefore, calculating similarity of two comments by comparing word by word is not a good choice. In our approach, we use comments in building text profile for each entity. Calculating similarity based on entities' profiles are explained in detail in the next section.

2.2 Similarity Metrics for Context Profile Features

In order to take advantage of relations information in ontology, we build variety types of text profile for each entity from its context. We divide context profiles of entities into three groups: IndividualProfile, SemanticProfile and ExternalProfile. Let us demonstrate how to build these profiles following a fragment of ontology in Fig.1

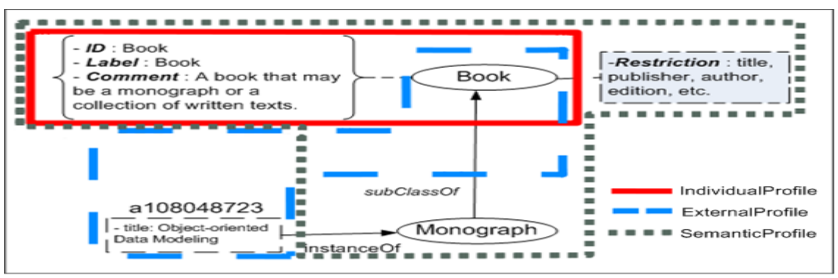

Fig. 1. Three types of context profile of class **Book**

The **IndividualProfile** of an entity is simply a string concatenation of its local name, labels and comments. *For example*: Individual profile of class **Book** is "Book Book A book that may be a monograph or a collection of written texts".

The **SemanticProfile** of an entity is an union of individual profiles of itself with individual profiles of its neighbors. Neighbors of a class consist of its sub-classes and all restricted properties. Neighbors of a property consist of its sub-properties, classes included in domain and range. *For example*: Semantic profile of class **Book** is created from individual profiles of {**Monograph, title, publisher, author, edition**}.

The **ExternalProfile** of an entity is created from texts taken from ontology instances. An external profile of a class is a string concatenation of texts of all instances belonging to either this class or its descendants. An external profile of a property is a string concatenation of value data corresponding to this property in all instances. *For example*: External profile of class **Book** is created from text value of instance **a108048723**. Therefore, external profile of class **Book** is "Object-oriented Data Modeling".

[5] http://oaei.ontologymatching.org

Having context profile for every entity, a similar technique described in [7] is used to compute similarity scores between entities. Let $sim_{IProfile}(e_i, e_j)$, $sim_{SProfile}(e_i, e_j)$ and $sim_{EProfile}(e_i, e_j)$ are similarity scores between entities (e_i, e_j) calculated by IndividualProfile, SemanticProfile and ExternalProfile respectively. To combine all of types of context profiles of entities, we propose the following generic formula:

$$sim(e_i, e_j) = \boldsymbol{f}(sim_{IProfile}(e_i, e_j), sim_{SProfile}(e_i, e_j), sim_{EProfile}(e_i, e_j)) \quad (3)$$

Where $\boldsymbol{f}$ may be *weighted average, max, etc.* If the combination function returns only similarity score achieved by SematicProfile, then our context profile metric is similar to metrics used in [1, 7]. If the combination function returns only similarity score achieved by ExternalProfile, then our context profile is similar to the instance-based metric. This property makes our context profile metric more flexible.

3 Combining Similarity Metrics with Decision Tree Model

We have implemented a system named YAM++ - an extension of [2], which is based on decision tree model to combine our proposed similarity metrics above. In our approach, a decision tree is a tree whose non-leaf nodes are the similarity metrics, leaf nodes values are either 1.0 ore 0.0 indicating if there is a match or not. At a non-leaf node, a similarity value of to-be-matched entities is computed by the similarity metric in ongoing node. The returned value is compared with condition values on outgoing edges from current node in order to decide which child node will be reached. This process will start at root node and iterate until a leaf node is reached. The value of destination leaf node indicates whether the two entities should match or not. See [2] for more detail of the advantages of using decision tree model.

4 Experiments and Evaluations

4.1 Selection of Metrics and Training Data

In our system, similarity metrics are divided in three main groups: (i) **name metrics** exploit name feature; (ii) **label metrics** exploit label feature; (iii) **context metrics** exploit different types of context profiles. The selection of the most representative metrics for each group is based on the hypothesis "A good feature subset is one that contains features highly correlated with the class" [4]. The correlation value is calculated by Pearson's formula[6] between similarity scores obtained by a metric and values provided by experts for each test in Benchmark datasets. Finally, the similarity metrics having the highest average values in each of three groups above are selected.

[6] http://en.wikipedia.org/wiki/Correlation_and_dependence

Next, training data for learning process are selected from Benchmark datasets. It is based on our heuristic: a training data is representative with the respect to a feature if this feature is highly correlated to the class. For each test in Benchmark datasets, our system computes the average correlation coefficient for all selected metrics above. Then, tests having the highest average of correlation values are selected to build training data.

4.2 Experimental Evaluations

Two experiments were designed as follows: (i) The first experiment shows the effectiveness of our proposed metrics in different scenarios over Benchmark datasets. (ii) The second experiment presents the performance quality of our approach on Conference datasets and shows the comparison results with other participants on OAEI 2010 campaign.

Result on Benchmark datasets According to terminological features described in Benchmark datasets, we select three representative groups of tests as follows: **(i) TestGroup1** contains various types of naming convention using in designing real ontologies. The typical selected tests are: **#104** (identical string), **#204** (different naming conventions) and **#205** (synonym words). **(ii) TestGroup2**: Names and labels of entities in test ontologies of this group are substituted by random meaningless strings. To discover mappings, we should take advantage from other features. We select tests **#201**, **#201-2**, **#201-4**, **#201-6** and **#201-8** for this group. Names and labels of entities in these tests are replaced by random strings with proportion **100%**, **20%**, **40%**, **60%** and **80%** respectively. However, they support annotation information and data instances for entities. **(iii) TestGroup3**: Test ontologies in this group are similar to ontologies in the second group except that they do not support annotation information for entities. We select tests **#202**, **#202-2**, **#202-4**, **#202-6** and **#202-8** for this group. Tests in the first group are suitable for name and label

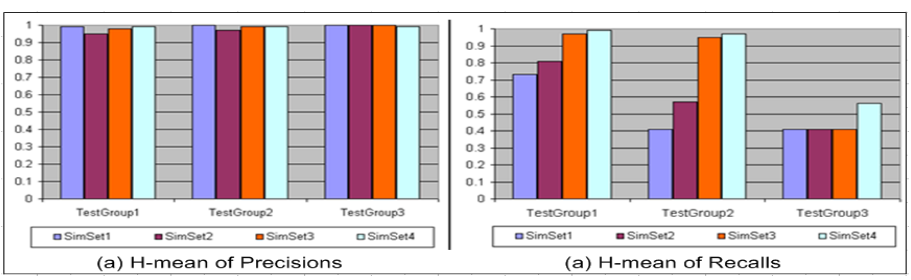

Fig. 2. H-mean of Precision and Recall on different scenarios

metrics. Tests in the second and the third groups are suitable for context metrics. In order to see how our metrics are appropriate to these scenarios above, we perform experiments on the following selected sets of similarity metrics: **(i) SimSet1**: Only name metrics; **(ii) SimSet2**: adding label metrics to **SimSet1**; **(iii) SimSet3**: adding Semantic context profile metric to **SimSet2**; **(iv) SimSet4**: adding full context profile metric (Semantic and External context profiles) to **SimSet2**; After running 10 times with different selected training datasets, the harmonic mean values of Precision and Recall for each scenario are shown in Figure.2.

Generally, our system achieves very high precisions (≈**1.0**) in all scenarios (Fig.2a). This trend is exactly what we expected because our similarity metrics focus to high accuracy in calculating similarity scores between entities. Besides, the recall increases step by step thanks to adding metrics exploiting new features (Fig.2b). Now, we look at particular scenarios in detail.

In the **TestGroup1**, our string, linguistic and their combination metrics work well. Based on these metrics, our system discovered most of commonly real patterns used to name entities (e.g., synonym, different naming conventions). However, there are other real patterns requiring more knowledge of domain, for example (*"Academic"*, *"StudentReport"*), (*"LectureNotes"*, *"CourseMaterial"*). That is why using only string and linguistic metrics (in SimSet1 and SimSet2) our system obtained good recall (≈**0.8**) but not ideal (**1.0**). Thanks to context metrics (in SimSet3 and SimSet4), more mappings have been discovered.

In the **TestGroup2**, with name metrics only, our system achieves **0.41** for recall. Although this value is low but it is the expected number. Let see the description on test ontologies in this group. The average number of altered names of entities is $(20\% + 40\% + 60\% + 80\% + 100\%)/5 = 60\%$. It mean that the maximum number of mappings found by name metrics is $100\% - 60\% = 40\%$. This number is in line with the recall value (**41%**) obtained by our system. Similarly to TestGroup1, by adding new metrics (labels, contexts), our system discovered more mappings.

In the **TestGroup3**, recalls achieved by running with SimSet1, SimSet2 and SimSet3 are the same (**0.41**). That is because the test ontologies in this group do not support any annotation information. Only when running with SimSet4, thanks to ExternalContext profile, our system discovered more mappings.

Results on Conference datasets In order to evaluate the performance of our approach with matching scenarios in another domain which is independent with training data, we select Conference datasets for testing. After running 10 times with different selected training datasets, we obtain precision (**0.75**), recall (**0.52**) and f-measure (**0.61**) in harmonic mean. Fig.3 shows the performance quality of our system among participants in OAEI 2010 campaign.

Most of participants need to set a confidence threshold for finding mappings. Threshold values in the Fig.3 are found for the optimal f-measure value for matchers. Like CODI system, we do not need to set threshold to our system. We obtain the second position (under CODI) in term of harmonic mean F-measure among all participants in campaign OAEI 2010.

Matcher	Confidence threshold	Precision	Recall	F-measure
AgrMaker	0.66	0.53	0.62	0.58
AROMA	0.49	0.36	0.49	0.42
ASMOV	0.22	0.57	**0.63**	0.60
CODI	*	**0.86**	0.48	**0.62**
Ef2Match	0.84	0.61	0.58	0.60
Falcon	0.87	0.74	0.49	0.59
GeRMeSMB	0.87	0.37	0.51	0.43
SOBOM	0.35	0.56	0.56	0.56
YAM++	*	**0.75**	**0.52**	**0.61**

Fig. 3. Optimal results of participants in Conference track

5 Conclusion and Future Work

In this paper, we have proposed new similarity metrics exploiting both terminological and context profile features. We also proposed a machine learning approach to combine these similarity metrics. Experiments over OAEI datasets show that our proposed metrics work effectively. Our system achieved high position among participants of OAEI 2010 campaign in Conference track. Additionally, our combining approach is automatic, flexible and extensible. In the future work, we plan to integrate structural and semantic methods to our system in order to improve its performance.

References

1. Isabel F. Cruz, Flavio Palandri Antonelli, and Cosmin Stroe. Agreementmaker: Efficient matching for large real-world schemas and ontologies. *Proceedings of The Vldb Endowment*, 2:1586–1589, 2009.
2. Fabien Duchateau, Zohra Bellahsene, and Remi Coletta. A flexible approach for planning schema matching algorithms. In *OTM Workshops*, pages 249–264, 2008.
3. Jérôme Euzenat and Pavel Shvaiko. *Ontology matching.* Springer-Verlag, Heidelberg (DE), 2007.
4. Mark A. Hall. Correlation-based feature selection for discrete and numeric class machine learning. In *ICML*, pages 359–366, 2000.
5. Sergio Jimenez, Claudia Becerra, Alexander Gelbukh, and Fabio Gonzalez. Generalized mongue-elkan method for approximate text string comparison. In *CICLing '09*, pages 559–570, 2009.
6. Feiyu Lin and Kurt Sandkuhl. A survey of exploiting wordnet in ontology matching. In *IFIP AI*, pages 341–350, 2008.
7. Yuzhong Qu, Wei Hu, and Gong Cheng. Constructing virtual documents for ontology matching. In *World Wide Web Conference Series*, pages 23–31, 2006.
8. Giorgos Stoilos, Giorgos B. Stamou, and Stefanos D. Kollias. A string metric for ontology alignment. In *ISWC Conference*, pages 624–637, 2005.