BOwL: Exploiting Boolean Operators and Lesk Algorithm for Linking Ontologies*

Viviana Mascardi
viviana.mascardi@unige.it

Angela Locoro
angela.locoro@unige.it

DISI - Università di Genova
Via Dodecaneso, 35
16146 Genova, Italy

ABSTRACT

BOwL applies word sense disambiguation techniques for tagging ontology entities with WordNet words. Boolean operators that appear in names of ontology entities are interpreted based on their semantics and are used during the ontology matching stage accordingly. Experimental results are shown, demonstrating the feasibility of the approach.

Categories and Subject Descriptors

I.2.4 [**Knowledge Representation Formalisms and Methods**]: Semantic networks

General Terms

Algorithms

Keywords

Ontology Tagging, Ontology Matching

1. INTRODUCTION

In many emerging scenarios including the Linked Open Data, virtual enterprises, social and collaborative environments just to cite few, ontologies created by different authors need to be linked together in order to ensure the interoperability of the applications based upon them.

One way for achieving this goal is to label the entities (classes, properties, and instances) of each ontology with tags drawn from a shared vocabulary, and use these tags as a means for giving a shared semantics to all of them. This approach would allow to derive links among ontologies just by looking at the tags they share.

In our past research, we explored this direction exploiting upper ontologies as the shared vocabulary [5]. However, few large enough upper ontologies exist and the contextual knowledge that they provide to understand the meaning of entity names is not as wide as the knowledge stored in a thesaurus or a vocabulary.

In this paper we discuss the results of using WordNet instead of upper ontologies for tagging ontology entities and for linking them. Boolean operators that appear in the entity names are also taken into account.

BOwL, which derives from *BOole with Lesk*, has been tested on a subset of the OAEI 2010 data set, a reference benchmark in the ontology matching research field, and on a small data set developed by the authors. On our own data set, built ad hoc for demonstrating *BOwL*'s features, the aggregated precision, recall and F-measure [3] reach **100%**, outperforming all the other algorithms that we run and that include string-based and WordNet-based ones, Lily – one of the OAEI 2009 best performing tools –, and AgreementMaker – one of the best performing tools in the last two OAEI editions.

On the 30 OAEI 2010 tests that we run, the aggregated result is **80%** precision, **87%** recall, and **83%** F-measure. These values are comparable to those of state-of-the-art tools, demonstrating the suitability of our approach not only on our own benchmark, but also on ontologies created by third parties.

We discuss the semantic tagging and ontology linking algorithms in Section 2, and the results we obtained in Section 3. We assume that the reader has some background knowledge on ontology matching, WordNet, and the Lesk algorithm for word sense disambiguation.

2. ALGORITHMS

Semantic tagging

BOwL exploits Lesk algorithm [4] (in the adapted version proposed by Satanjeev Banerjee and Ted Pedersen [1], which uses WordNet) for tagging each word belonging to an entity name with its most likely sense. We name these sense-carrying tags "semantic tags". Conjunctions, disjunctions and negations that appear in identifiers are labeled with "Boolean tags".

During the matching stage, semantic and Boolean tags are exploited for obtaining effective mappings: reliable semantic tags (namely, those obtained by disambiguating a word in a context considered rich and informative enough) are used during the ontology matching stage for identifying homonyms which do not share the same meaning, whereas Boolean tags are exploited for matching composite entity names as if they were Boolean propositions.

SAC'12 March 25-29, 2012, Riva del Garda, Italy.

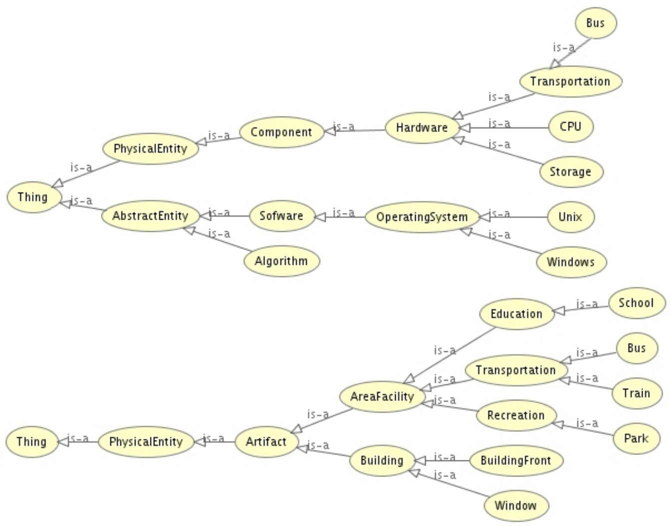

Figure 1: Ontologies from our benchmark, with synonyms.

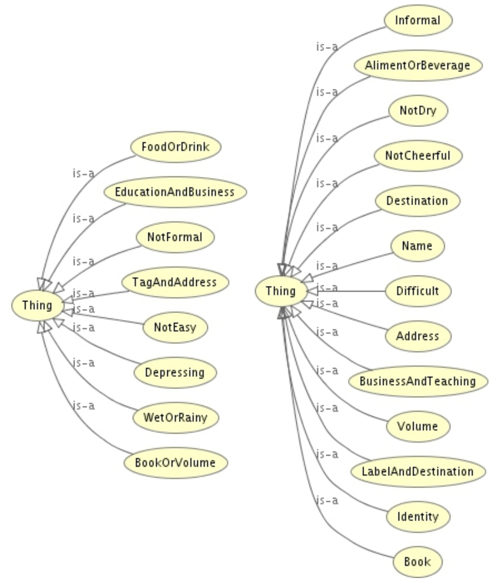

Figure 2: Ontologies from our benchmark, with Boolean operators.

For each entity e belonging to ontology o, *BOwL* performs three independent steps:

1. tokenization of the entity name, stemming of the constituent tokens, and association of Boolean tags and of words belonging to WordNet with tokens;
2. identification of the entity's context given by the WordNet words associated with the "neighboring entities" (if e is a class, neighboring entities are classes c for which object properties exist whose domain is c and range is e or vice-versa, as well as classes related via the `subClassOf` hierarchy; if e is a property, neighboring entities are classes c defined as the domain or range of e; if e is an instance, neighboring entities are classes to which the instance belongs to), plus the words belonging to the entity comment;
3. association of semantic tags with tokens that belong to the entity name by applying Lesk algorithm.

Ontology linking

Given ontologies o and o' to be matched, the correspondences of entities in o with entities in o' based on their semantic and Boolean tags are identified.

The *BOwL* matching algorithm generates the $\mid o \mid \times \mid o' \mid$ pairs (e, e') of homogeneous (namely, both classes, both properties, both individuals) entities $e \in o$ and $e' \in o'$[1] and runs the algorithm described below on all of them.

The algorithm deals with 9 basic cases that depend on the entity name's features (simple or composed), on the presence of Boolean operators in it, on whether or not the context can be considered informative enough to help making a correct word sense disambiguation (based on our experiments, we empirically define an informative context as one that contains at least 3 words).

Because of space constraints, we will only discuss 5 cases out of the 9 ones, providing examples mainly drawn from our own benchmark (two pairs of ontologies in our benchmark are represented in figures 1 and 2 respectively).

Homonyms detection in simple entity names. Two entities whose names consist of a single word tagged with identical semantic tags, and whose context is considered informative enough, are matched only if the senses of their semantic tags are the same.

Example. Figure 1 shows two ontologies each containing entities named *Bus*. The semantic tags associated with the *Bus* entities are different, because in one case the context led to selecting "an electrical conductor that makes a common connection between several circuits" (WordNet sense number 102924713) as correct meaning, whereas in the other case it led to the selection of "a vehicle carrying many passengers; used for public transport" (WordNet sense number 102924116). Since in both cases the context is considered informative enough, entity $Bus \in o$ and entity $Bus \in o'$ are recognized as having homonyms names and are not matched despite to their syntactic identity.

Synonym detection in simple entity names. Two entities having simple names tagged with different semantic WordNet tags, are matched either if their sense is the same, or if they are synonym according to WordNet.

Example. In test 302 of the OAEI 2010 competition, our algorithm matched data property *issue* in ontology 101, having domain $Conference \cup Reference$ and range `string`, with data property *number* in ontology `http://ebiquity.umbc.edu/ontology/publication.owl`, having domain *Publication* and range `string`. *Issue* and *number* share the same sense according to WordNet, namely "one of a series published periodically", if used in a bibliographic context. According to the reference alignment provided by OAEI for that test, unfortunately, the correspondence <*issue, number*> is wrong, probably because of structural features of the matched ontologies that *BOwL* does not take into account, and that were instead considered by the OAEI knowledge engineers. This and other differences on what is semantically equivalent in a given context according to WordNet, and what is equivalent according to the OAEI reference alignments, explain the presence of some false positives in the tests we run on OAEI ontologies, that lower the precision of our algorithm on that benchmark.

Conjunction and disjunction of non-synonym terms. If entity e is named $\text{Token}_{1.1}\text{OpToken}_{1.2}$ (`Op` being `And` or

[1]Since *BOwL* does not exploit structural features of the entities to be matched, but only linguistic ones, all the pairs must be considered.

`Or`, or their syntactical variants) and $\text{Token}_{1.1}$, $\text{Token}_{1.2}$ are not synonym, it can be matched with entity e' only if e' has name $\text{Token}_{2.1}\text{OpToken}_{2.2}$ and tokens can be pairwise matched according to the *BOwL* matching algorithm.
Example. The two correspondences <*EducationAndBusiness, BusinessAndTeaching*> and <*FoodOrDrink, AlimentOrBeverage*> among entities in the two ontologies represented in Figure 2 are correctly returned by the *BOwL* algorithm.

Conjunction and Disjunction of synonym terms. If entity e is named with a conjunction (resp. a disjunction) of tokens $\text{Token}_{1.1}$ and $\text{Token}_{1.2}$ that are either synonyms or similar according to WordNet, then e's name is not a proper conjunction (resp. disjunction). Any entity e' whose name is equivalent either to $\text{Token}_{1.1}$ or to $\text{Token}_{1.2}$, should be matched with e.
Example. In Figure 2, the correspondences <*BookOrVolume, Book*> and <*BookOrVolume, Volume*> are both correct, since *Book* and *Volume* are synonyms according to WordNet, and hence *BookOrVolume* reduces to any of them.

Negation. If entity e is named NotToken_1 or NonToken_1, it can be matched with a simple entity e' whose name is Token_2, if Token_1 and Token_2 are antonymous according to WordNet.
Example. In Figure 2, correspondences <*NotFormal, Informal*>, <*NotEasy, Difficult*>, and <*Depressing, NotCheerful*> are correctly found.

3. EXPERIMENTAL RESULTS

In order to evaluate the feasibility of our algorithm, we have applied it to 30 pairs of ontologies from the OAEI 2010 benchmark (`http://oaei.ontologymatching.org/`) and to 6 pairs of small ontologies (less than 15 entities) that involve both ontologies from the OAEI 2010 data set and ontologies developed by ourselves. We manually created the reference alignment for each of our pairs.

Among the almost one hundred ontologies belonging to the OAEI 2010 benchmark, we selected 30 of them to be matched with the OAEI 101 ontology because the remaining ones either have scrambled labels like *sqndsqgy* (example taken from test 201), or have labels in languages other than English. Since the *BOwL* algorithm operates at the semantic level and assumes that names associated with ontology entities are meaningful English words, it made little sense to run it on ontologies where entities are named with random strings or are written in French: clearly, it would inevitably fail on such tests.

The results of running *BOwL* on the 30 OAEI 2010 ontologies are shown in Figure 3 and are comparable to that of state-of-the-art tools. The reader may find the results of the tools that joined the OAEI 2010 competition at `http://oaei.ontologymatching.org/2010/results/benchmarks/index.html`.

Our own benchmark consists of ontologies aimed at demonstrating the major features of our approach, like the ones shown in Figures 1 and 2. We run the original *BOwL* algorithm and a set of existing ontology matching methods on our benchmark. Besides the substring, n-gram, SMOA, and language-based methods provided by the Alignment API, `http://alignapi.gforge.inria.fr/`, and indicated with **subst**, **SMOA**, **n-gram** and **WN** in Figure 4, we run the **Lily**

	BOwL		
Test	Prec.	Rec.	FMeas.
101	0.80	1.00	0.89
103	0.80	1.00	0.89
104	0.80	1.00	0.89
203	0.80	0.89	0.84
204	0.78	0.86	0.82
205	0.58	0.27	0.37
208	0.78	0.75	0.77
209	0.57	0.26	0.35
221	0.80	1.00	0.89
222	0.79	1.00	0.89

	BOwL		
Test	Prec.	Rec.	FMeas.
223	0.80	1.00	0.89
224	0.80	1.00	0.89
225	0.80	1.00	0.89
228	0.94	1.00	0.97
230	0.77	0.99	0.87
231	0.80	1.00	0.89
232	0.80	1.00	0.89
233	0.94	1.00	0.97
236	0.94	1.00	0.97
237	0.79	1.00	0.89

	BOwL		
Test	Prec.	Rec.	FMeas.
238	0.80	1.00	0.89
239	0.91	1.00	0.95
240	0.92	1.00	0.96
241	0.94	1.00	0.97
246	0.91	1.00	0.95
247	0.92	1.00	0.96
301	0.77	0.17	0.28
302	0.79	0.48	0.60
303	0.70	0.67	0.68
304	0.78	0.86	0.82
H-mean	0.80	0.87	0.83

Figure 3: Results on 30 OAEI 2010 tests.

	BOwL			Lily			AgrMaker			substr		
Test	Prec.	Rec.	FMeas.	Prec.	Rec.	FMeas.	Prec.	Rec.	FMeas.	Prec.	Rec.	FMeas.
500	1.00	1.00	1.00	0.75	0.33	0.46	0.50	0.11	0.18	0.67	0.22	0.33
501	1.00	1.00	1.00	1.00	1.00	1.00	0.50	1.00	0.67	0.25	1.00	0.40
502	1.00	1.00	1.00	1.00	0.50	0.67	0.50	0.50	0.50	0.20	0.50	0.29
503	1.00	1.00	1.00	0.67	1.00	0.80	0.40	1.00	0.57	0.50	1.00	0.67
504	1.00	1.00	1.00	1.00	1.00	1.00	0.50	1.00	0.67	0.25	1.00	0.40
505	1.00	1.00	1.00	1.00	1.00	1.00	0.67	1.00	0.80	0.67	1.00	0.80
H-mean	1.00	1.00	1.00	0.83	0.59	0.69	0.50	0.47	0.48	0.39	0.53	0.45
	BOwL			**SMOA**			**n-gram**			**WN**		
Test	Prec.	Rec.	FMeas	Prec.	Rec.	FMeas.	Prec.	Rec.	FMeas.	Prec.	Rec.	FMeas.
500	1.00	1.00	1.00	0.43	0.33	0.38	0.50	0.11	0.18	0.67	0.22	0.33
501	1.00	1.00	1.00	0.11	1.00	0.21	0.50	1.00	0.67	0.13	1.00	0.22
502	1.00	1.00	1.00	0.10	0.50	0.17	0.50	0.50	0.50	0.17	1.00	0.29
503	1.00	1.00	1.00	0.40	1.00	0.57	0.50	1.00	0.67	0.40	1.00	0.57
504	1.00	1.00	1.00	0.25	1.00	0.40	0.25	1.00	0.40	0.25	1.00	0.40
505	1.00	1.00	1.00	0.50	1.00	0.67	0.67	1.00	0.80	0.40	1.00	0.57
H-mean	1.00	1.00	1.00	0.26	0.59	0.36	0.47	0.47	0.47	0.27	0.59	0.37

Figure 4: Results on our own tests.

[6] and the AgreementMaker [2] (**AgrMaker** in the figure) algorithms. We could not compare *BOwL* with the two winners of the OAEI 2010 competition because either they are not available to the research community, or the available version only works on older OAEI benchmarks.

On our benchmark, *BOwL* outperforms all the other algorithms. This is not surprising, since the benchmark was developed ad hoc for letting *BOwL*'s features emerge, however this experiment demonstrates that the correct detection of homonyms that should not be matched despite their syntactic identity, and the management of “and”, “or” and “not” driven by their logical semantics, may be useful in many scenarios and are not fully supported by existing tools.

4. REFERENCES

[1] S. Banerjee and T. Pedersen. An adapted Lesk algorithm for word sense disambiguation using WordNet. In *CICLing*. Springer, 2002.

[2] I. Cruz, F. P. Antonelli, and C. Stroe. AgreementMaker: Efficient matching for large real-world schemas and ontologies. *PVLDB*, 2(2):1586–1589, 2009.

[3] H. H. Do, S. Melnik, and E. Rahm. Comparison of schema matching evaluations. In *NODe*. Springer, 2002.

[4] M. Lesk. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *SIGDOC*. ACM, 1986.

[5] V. Mascardi, A. Locoro, and P. Rosso. Automatic ontology matching via upper ontologies: A systematic evaluation. *IEEE Trans. Knowl. Data Eng.*, 22(5):609–623, 2010.

[6] P. Wang and B. Xu. Lily: ontology alignment results for OAEI 2009. In *OM2009*, 2009.