Sentence Simplification Based Ontology Mapping

Lu Lu
School of Mathematics &
Computational Science
GuangDong University of Business Studies
GuangZhou 510320, China
lulu@gdcc.edu.cn

Nandan Parameswaran
School of Computer Science
and Engineering
University of New South Wales
Sydney , Australia, 2031
paramesh@cse.unsw.edu.au

Abstract

Ontology mapping plays an important role in interoperability over ontologies. Many researchers have proposed algorithms and tools for (semi-)automatically mapping one concept to another concept. Among them, WordNet is widely used as the domain knowledge support in the mapping process. To our knowledge, however, most of them only use synonym, hypernym and hyponym relations in WordNet and the actual meanings provided in natural English (as gloss) are often ignored. In this paper, we treat the concepts(c) as English words (w) and propose an ontology mapping technique where we use the meanings of the words as given in Wordnet (in English) for semantic mapping by constructing their parse trees first and simplifying them for computing similarity measures. Our experimental results show that our method performs better in Recall and F1-Measure than many techniques reported in the literature.

Introduction

With the development of Semantic Web, more and more ontologies are written in RDF (Klyne et al. 2004) or OWL (Patel-Schneider et al. 2004). Unfortunately ontologies themselves are distributed and heterogeneous. Therefore, ontology mapping, or ontology alignment, has become an important challenge for ensuring interoperability across ontologies.

Recently several algorithms and methodologies have been reported and tools have been built for (semi-)automatic ontology mapping. To evaluate ontology matching technologies, since 2004, OAEI[1] has been organizing evaluation campaigns consisting of four tracks gathering six data sets and different evaluation modalities, and providing opportunities for evaluating ontology mapping systems.

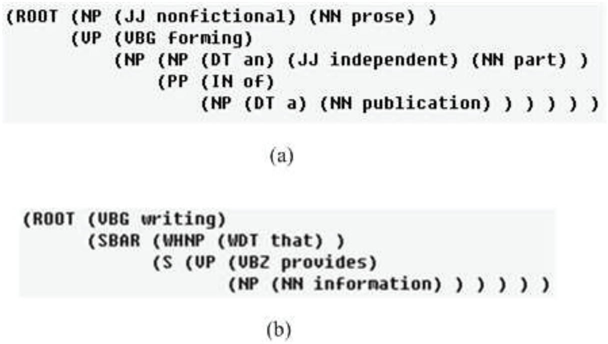
Ontology mapping techniques are commonly divided into four broad categories (Euzenat et al. 2007) (Isaac et al. 2007): lexical (detecting similarities between labels of concepts), structural (using the structure of the ontologies), background knowledge based mapping, and instance-based mapping (using classified instance data). To achieve better mapping results, many ontology mappers use WordNet to provide domain knowledge support for lexical, structural mapping. DSSim (Nagy et al. 2008), ASMOV (Yves et al. 2008) are such ontology mappers. To our knowledge, most of them use a method called semantic distance (Budanitsky et al. 2001), which calculates the link between the synset (synonym of the word) of the two given words in WordNet while deriving the similarity measure between the words. The link represents hypernym or hyponym relation. For example, *meeting* is the hypernym of *conference*, that is, *conference* is a kind of *meeting*. The semantic distance between *meeting* and *conference* is 1. Similarly, *gathering* is the hypernym of *meeting*, and therefore, the distance between *gathering* and *conference* is 2. Based on such a distance concept, many measures of similarity were proposed, such as (Lin. 1998). The problem here is that when researchers use only use synonym, hypernym and hyponym relations of the word in WordNet, they do not take the meaning of the words (the definition of the words provided by WordNet in natural English) into account in their mapping techniques. The definition of words in WordNet is called gloss and in this paper we use gloss in our mapping technique. In fact, a gloss is not a complete sentence. For example, a gloss of *document* is “writing that provides information”. However, we can regard it as “document is writing that provides information” and therefore we will treat a gloss as a sentence (and hence the term sentence in the title of this paper).

We use the Stanford Parser (Klein D. 2003) to process the glosses of the reference and the target ontologies. Based on the parse-tree paths, we split a tree into a set of subtrees and simplify the subtrees by applying a series of transformation rules. The rules are designed to bring the glosses toward common format so that they can be compared. Such sentences simplification process is the main contribution of our paper, and it is different from previous work in NLP area (The comparison is discussed in **Related Work** section). In our mapping process, we use the structure of the ontologies to improve the precision of our mapping results. Experimental results show that our method performs better in Recall and F1-Measure as compared to other approaches (The details are presented in the **Evaluation** session).

An Intuitive Example

In this section, we present an intuitive example to explain the basic ideas of our method where we illustrate how we

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



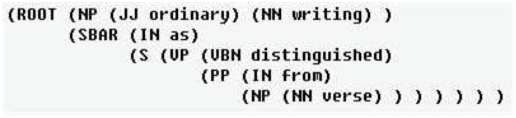
```
(ROOT (NP (JJ nonfictional) (NN prose) )
      (VP (VBG forming)
          (NP (NP (DT an) (JJ independent) (NN part) )
              (PP (IN of)
                  (NP (DT a) (NN publication) ) ) ) ) )
```

(a)

```
(ROOT (VBG writing)
      (SBAR (WHNP (WDT that) )
            (S (VP (VBZ provides)
                   (NP (NN information) ) ) ) ) )
```

(b)

Figure 1: (a) is the parse tree of the gloss of article; (b) is the parse tree of the gloss of document



```
(ROOT (NP (JJ ordinary) (NN writing) )
      (SBAR (IN as)
            (S (VP (VBN distinguished)
                   (PP (IN from)
                       (NP (NN verse) ) ) ) ) ) )
```

Figure 2: the parse tree of the gloss of prose

can map the concept *article* to the concept *document* (This is known as non-trivial mapping[2]).

Example 1: Mapping *article* to *document* [3]. We first retrieve the definitions of the given concepts from the WordNet and build parse trees for them as shown in Figure 1.

As can be seen from Figure 1, the gloss of one sense of the word *article* in WordNet is "nonfictional prose forming an independent part of a publication", and the gloss of one sense of the word *document* is "ordinary writing as distinguished from verse". In Figure 1 (a), we can simplify the first subtree (NP (JJ nonfictional) (NN prose)), and rewrite it as (NN prose). The second subtree (VP (VBG forming)(NP (NP (DT an) (JJ independent) (NN part))(PP (IN of)(NP (DT a) (NN publication))))) defines what (NN prose) is. To make this example simple, we only discuss the first subtree of the parse tree here. Thus, if *document* is similar to *article*, it should be similar to *prose*. In Figure 1(b), the first subtree is (VBG writing), and *VBG* here can be considered as *NN*. Thus, the POS tag of (NN prose) and (NN writing) is *NN*, and the trees are structurally comparable now. Since *prose* is not the same as *writing*, then we search the gloss of *prose* in WordNet (See Figure 2). Similarly, the first subtree (NP (JJ ordinary) (NN writing)) can be rewritten as (NN writing), which is the same as (NN writing) in the syntax tree of *document*. Thus we can conclude that *article* is similar to *document* roughly. (More details, regarding how to compare other subtrees, how to calculate the similarity weight, etc, are presented in **Sentences Simplification for Ontology Mapping** section.)

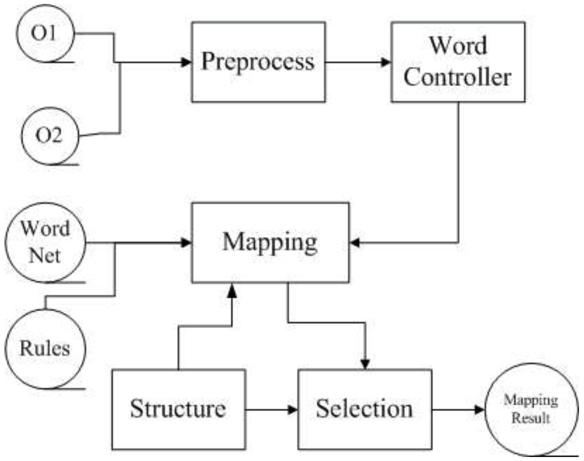


Figure 3: Framework

Preliminaries

In this section, we give a short overview of OWL-DL. The full description of the language syntax and semantics is available at http://www.w3.org/TR/owl-semantics/.

An OWL-DL ontology consists of several sets of entities: classes, properties, individuals (instances) and axioms. Classes denote sets of individuals. For example, *article* and *document* in Example 1 are classes. "Ontology Mapping - An Integrated Approach" is an instance of the class *article*. Properties relate individuals to other information. For example, in ("Ontology Mapping - An Integrated Approach", writtenBy, "Marc Ehrig"), *writtenBy* is a property. Axioms are used to provide information about classes and properties. For example, ConferenceMinutes $\sqsubseteq$ $\forall$talks.Communication, ConferenceMinutes and Communication are classes, talks is the property. In general, all information in OWL can be expressed in the form of a triple (subject, predicate, object).

In this paper, we use the noun and verb dictionary in WordNet to map classes and properties in ontologies. For mapping purpose, the RDF tags rdfs:label and rdfs:comment are very useful, especially for the benchmark track of OAEI 2007. If two classes have the same values of rdfs:label or rdfs:comment, then these two classes are regarded as the equivalent classes. We also use these relations in our mapping.

Sentences Simplification for Ontology Mapping

In this section, we first describe of our method, and explain the various phases in our framework.

Framework

Figure 3 shows the various phases involved in our method. O1 and O2 are the reference and target ontologies. In the preprocess module, we normalize the names of classes and properties. For example, "conference_document" is rewritten as "conference document". It involves a combination of words (we will refer to words such as this as compound

[2] The trivial mapping means the words are the same or similar, for example *write* is equivalent to *write*, *write* is equivalent to *writing*

[3] This mapping is from OAEI 2007, *conference track*.

words), and WordNet does not provide gloss for compound words. In word controller module, we propose some heuristic methods for mapping with compound words. For example, while mapping “conference document” to “meeting article”, we compare “document” with “article” first, then “conference” with “meeting”. The mapping module simplifies the parse tree of the gloss from WordNet, using “Rules” modules, and calculates the similarity weight between the reference ontology and the target ontology. In OWL ontology, the structure of the ontology provides some hints for mapping. Finally, based on the similarity weight and the ontology structure, the selection module produces the mapping results.

For ease of understanding, we explain the mapping and word controller module first, and then structure and selection modules.

Mapping and Word Controller Module

(1) Mapping process

In our mapping process, we first compare the rdfs:label/rdfs:comment (we describe these tags in the section **Preliminary**) from the given ontologies. If they are not the same, then before we simplify the gloss of the ontologies, we use the synset (a set of the synonyms of the word) and hypernym word of the ontology provided by WordNet to speed up the mapping process. For example *document* in Example 1, it has four senses in WordNet. For each sense, it has a synset, and thus, for example, if we find article in the synset { written document, papers ,text file }, then we can conclude *document* is the equivalent word of *article*. Otherwise, we will search their first hypernym where the semantic distance (we introduced this concept in **Introduction**) is 1. For example, *representation* is a first hypernym word of *document*. It means *document* is a kind of *representation*. If we find *article* in the first hypernym words of *document*, then we can conclude *article* is the super class of *document*. To avoid wandering too far away from originally given concepts (*document*, *article*), we stop at their first hypernym of them. (Note that, we compare *article* with *document*’s hypernym, or compare *document* with *article*’s hypernym, but we do not compare *article*’s hypernym with *document*’s hypernym.)

If we cannot find the target concept and the reference concept in their synset and hyernym words, we simplify their glosses. The details are discussed below.

(2) Sentences simplification and word controller

The method to simplify sentences is based on pattern matching. We analyzed the glosses in WordNet, and manually generated some rules (See Figure 4 below.). These rules form the pattern.

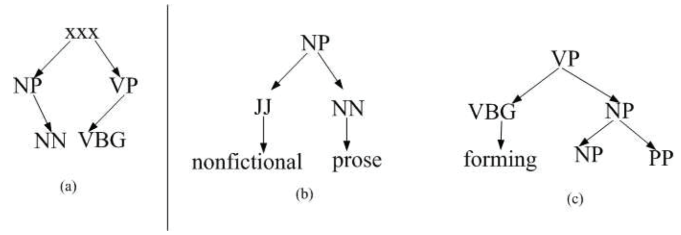


Figure 4: An example rule to separate a sentence

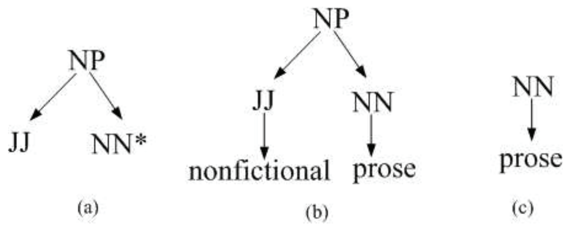
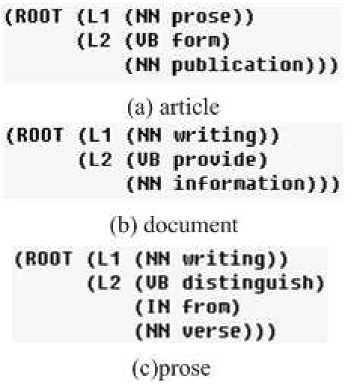


Figure 5: An example rule to simplify a sentence

These rules can be classified into two types. The first type of rules is used to decompose a parse tree into a set of subtrees. In Figure 1(a), the parse tree has been decomposed into two subtrees using the rule in Figure 4(a). This rule says if NP and VP are two sibling nodes under xxx(xxx can be NP or ROOT), and VP follows NP. NN is the right most child of NP, and VGB is the left most child of VP, then, the tree is decomposed into two subtrees as shown. The idea of decomposing a tree is based on the fact that a sentence can be divided into several sections, the later sections providing more information about the earlier sections. For example, in Figure 1(a), the later sections (forming an independent part of a publication) provide more information about what the previous section (nonfictional prose) is about. We call these sections (or subtrees) as layers in this paper. Figure 4(b) shows the first subtree of the parse tree of Figure 1(a) and Figure 4(c) shows the second subtree of the parse tree in Figure 1(a). We found that the rule in Figure 4(a) can be successfully applied to decompose parse trees of about 20900 glosses (out of the total 94633 glosses given in the noun dictionary in WordNet , that is 22% of all glosses in the noun dictionary.)

The second type of rules is used to simplify the layers above. For example, Figure 5(a) is a rule used to simplify the subtree in Figure 4(b). This rule says, if JJ and NN*(NN* means the POS tag which starts with “NN”) are the two children of NP, then the subtree is transformed into a single node (NN prose). (We also found that the rule in Figure 5(a) ap-



```
(ROOT (L1 (NN prose))
      (L2 (VB form)
          (NN publication)))
```

(a) article

```
(ROOT (L1 (NN writing))
      (L2 (VB provide)
          (NN information)))
```

(b) document

```
(ROOT (L1 (NN writing))
      (L2 (VB distinguish)
          (IN from)
          (NN verse)))
```

(c)prose

Figure 6: Modified Parse Tree

plies successfully to about 22560 glosses out of 94633, that is 24%in the noun dictionary in WordNet.) These rules can be used to simplify a subtree into a triple (subject, predicate, object) or to some components of a triple. In Figure 1(a) the parse tree is finally simplified into the tree shown in Figure 6(a). The first layer is simplified as a node (NN prose). Thus in this layer, the triple only contains the object(prose). The second layer is simplified into two components of a triple, predicate (VB form), and object (NN part of publication) (Our simplification rules always ignore the adjectives, adverbs).

The section "part of publication" can be viewed as a compound word . Its POS tag is (NP (NP (NN part)) (PP (IN of) (NP (NN publication))))). To simplify this section, we propose the following heuristic rule: if the node under IN is "of", and "of" occurs in-between two nouns (POS tag is "NN*"), the original path of the parse tree is rewritten as a node (NN part of publication). Because WordNet does not support search for compound words, we use a module called word controller, where we compare "part of publication" with other words. We use Example 2 to illustrate our idea.

Example 2, Mapping *conference document* to*meeting article*; and mapping *abstract of paper* to *abstract* ;

In the case of mapping *conference document* to *meeting article*, our heuristic rule is to compare the last word first(*document* and *article*). If the similarity weight is higher than a threshold W0, then we compare the previous word (*conference* and *meeting*). In the case of mapping *abstract of paper* to *abstract*, we compare the first word "abstract" in (*abstract of paper*) with the word (*abstract*). There are some exceptions, for example "some/any/every/all/part... of NN", in which case we skip "some/any/every/all/part...", and compare NN with the target concept. Thus in Figure 6(a), (NN part of publication) is rewritten as (NN publication). And the second layer is simplified into two nodes (VB form)(NN publication) which contains the predicate and the object.

Similarly, Figure 1(b) is simplified to Figure 6(b), and Figure 6 (c) is the simplified version of Figure 1(c). In Figure 6(c), (IN from) is part of the predicate (VB distinguish). From Figure 6, we see that, the structures of first layer of these trees are the same (POS tag NN). Thus the first layer of these trees are comparable. The structure of second layer of Figure 6(c) and Figure 6(a) is considered as different in our method. If the noun/verb in the corresponding layer is different, we search them in the noun/verb dictionary in WordNet. For example, "prose" in Figure 6(a) and "writing" in Figure 6(b) are different, we first search for "prose", then for "writing". This time we only consider the noun/verb in the first layer of the tree. For example, the noun in first layer of Figure 6(c) is the same as "writing". If they are different, then the similarity weight of "prose" and "writing" is 0. (We explain below the notion of similarity weight.)

(3) Similarity weight

In this section, we use Figure 7 to illustrate our method to calculate the similarity weight given two sentences. If the structures of the corresponding layers in the two sentences are different, then the weight of this layer is 0.

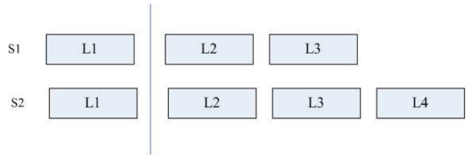


Figure 7: Similarity weight example (s1, s2 are two glosses, L1, L2, L3,L4 are layers)

Suppose the weight of the first layer is W1, and the number of the layers in the two sentences are n and m (n<m). Then the weight of the layers(W(Li)) for the shorter sentence is:

$W(L1)=W1; W(Li)= (1-W1)/(n-1),(2\leq i\leq n)$

the weight of layers(W(Li)) for other sentence is:

$W(L1)=W1; W(Li)=(1-W1)/(n-1),(2\leq i\leq n);$

$W(Li)=0,(n<i\leq m)$

For example, in s2, the weight each of L2 and L3 is (1-W1)/(2), and the weight of L4 is 0. But if the weight of the first layer is 0, then the total weight of the sentence is also 0. In one layer, if its weight is W, it contains a triple, and there are n components in the triple. Then the weight of each component is W/n. For example, if the layer contains (predicate, object), then the weight of predicate, as well as the weight of object is W/2. For subject and object, if the noun is a compound word, and the weight of the noun is W, then the weight of the first word in the mapping sequence is W2*W. If minimal length of two compound words is n, then the weight of other words in the shorter compound word is (1-W2*W)/(n-1) . We will give the value of W0, W1 and W2 in the **Evaluation**.

Finally, if s1 and s2 (in Figure 7) are similar, and they are the glosses of the words A and B, then A is equivalent to B. If s1 is the gloss of hypernym of A, then A is the super class of B (or B is the subclass of A).

Structure and Selection Module

The structure of an OWL ontology often provides some hints for mapping. In this paper, we use the structure information available in the ontology in four cases: (1) Deciding the part of speech. For example, "graduate" is the subclass of "student". In the mapping process, when we search for "graduate" in WordNet, if "student" only contains the noun sense, and "graduate" contains not only noun sense, but also adjective and verb senses, then we can quickly skip the verb, and the adjective senses of "graduate", but only search for the noun sense of the word. (2) Improving the precision of mapping. For example, consider the two concepts, "abstract of paper" and "abstract". The concept "abstract" is the subclass of "paper", so we first compare "abstract" in "abstract of paper" with the single word "abstract", and then compare the "paper" in "abstract of paper" with the parent ("paper ") of the single word "abstract". (3) Selection module. The output of the mapping module are pairs of words and the similarity weight of these words. Instead of only comparing the similarity weights, if the weight is in a range [W3, W4] (W3<W4<1), we compare the words' superclasses, and the distance between the words and their parents. For example, consider the pairs of words (A, B), (A, C), where A is a concept from the reference ontology, and B and C are concepts from the target ontologies. Both the weights are in the range [W3, W4]. Suppose D is the parent of A, E is the parent of B, and D is equivalent to E. Then we choose the mapping (A, B) as the mapping result. (4) Reducing the number of comparisons. The root class(that is, concept) of OWL ontology is "Thing". We compare classes in the first level of "Thing" (the root of the ontology) firstly. For example,if A is a class in the first level of the reference ontology, B and C are the classes in the first level of the target ontology, where the similarity weight of (A, B), (A, C) are not 0, then the classes under A are only compared with the classes under B and C.

Evaluation

In this section, we examine the method we propose and report our experimental results. (All algorithms were implemented in Java.)

(1) Data set and experimental setup

There are four tracks in OAEI 2007. The organizers manually find out the mappings of ontologies for the *benchmark track* as a gold-standard. However, there is no such gold-standard in other three tracks. Thus we had to manually find out the mappings of ontologies ourselves for the *conference track* (because its background was more familiar to us). In this paper, we use the ontologies from *conference track* and *benchmark track* to evaluate our method.

The data set in *conference track* contains 14 real word ontologies. There are 974 classes and 611 properties all together in these ontologies.

In the experiments, we conducted evaluations in terms of precision, recall and F1 measure. The measures are defined as follows:

Precision(P): It is the percentage of correct discovered mappings in all discovered mappings.

Recall(R): It is the percentage of correct discovered mappings in all correct mappings.

P = #(m1 $\bigcap$ m2)/#m1; R = #(m1 $\bigcap$ m2)/#m2

F1 = 2*P*R/P+R

where m1 is the mapping discovered by our method and m2 is the mapping assigned manually (we view the manually assigned mappings as the correct mapping).

In our experiments, we used the Stanford Parser to parse all the glosses that were used in our experiments, and manually corrected the errors in the parse trees(To our knowledge, none of the existing parsers can parse the sentences in WordNet 100% correctly).

In the experiments, we find that some senses of the word in WordNet are not often used. For example, one sense of *school* is "a large group of fish". Thus, we only use the first two senses of each word. The parameters we described earlier in section Sentences Simplification for Ontology Mapping are set as: W0=0.53, W1=0.57, W2=0.76 , W3=0.41 and W4=0.72.

(2) Results and comparison

In our experiments, we compared our system with two ontology mappers Falcon, ASMOV[4] which performed well in the *conference track* of OAEI 2007. From the experimental results for the *conference track*,shown in Figure 8, we see that the improvement in Recall quality of our method is significant: 49% over Falcon and 85% over ASMOV. This means our method was able to find much more correct mappings than Falcon and ASMOV. In addition, our method also has an impressive improvement in F1: 30% and 151% for Falcon and ASMOV, respectively. However, there was no much improvement of precision in our method implying that our method also found some mappings that were incorrect. The reason is that when we compared each layers of parse trees, we compared them separately as we discussed earlier. If we had integrated all the layers together and performed the comparison, we might have achieved better results. (We plan to do this in our future work.)

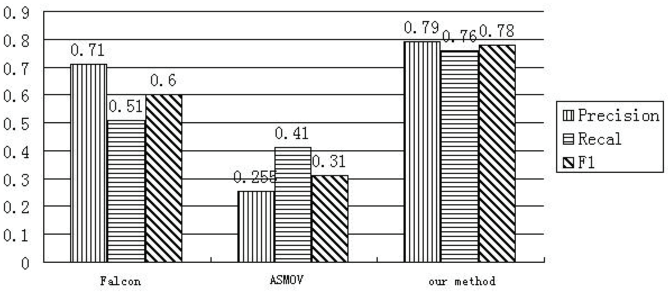
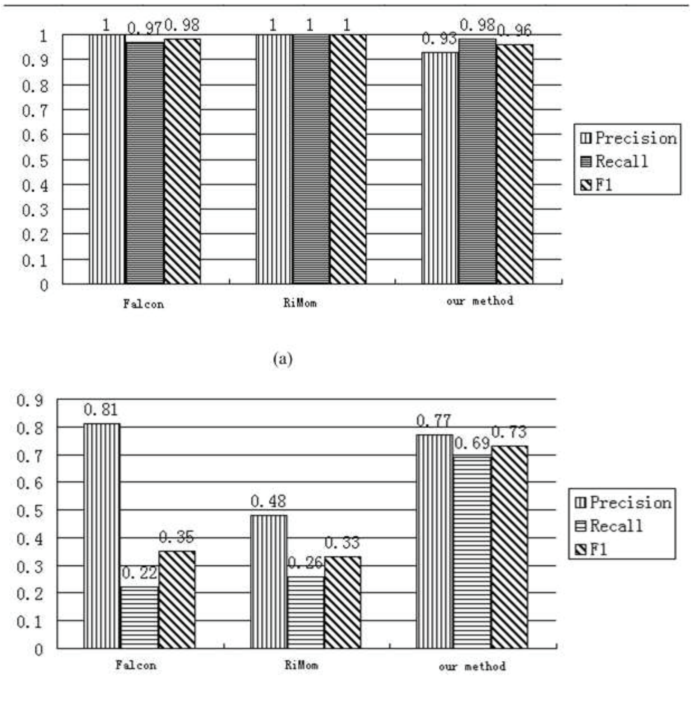


Figure 8: Evaluation-1

In the *benchmark track*, the ontology mappers that participated in the competition have already performed very well. However, such good performance was due to the OWL-DL axioms that were present in the ontology, where for example, in the case of the property axioms, the domains or ranges act as constraints to the properties or classes. Also some RDF tags, especially *rdfs:label* and *rdfs:comment* are very useful improving the precision of the mapping results. However, we find that ontology developers usually only use subclass and equivalent class relations. They do not often use *rdfs:label*, *rdfs:comment* , property axioms, etc. Thus we chose the two ontologies #101 and #205 in the *benchmark track*, removed all the *rdfs:label* and *rdfs:comment* in #101 and #205, only kept the subclass and equivalent class relations, and deleted all other axioms. (Let us call the revised versions as #101' and #205') In fact, after this change, the ontology became much closer to a real world ontology. In the experiments, we compared our system with Falcon, RiMom which performed best in the *benchmark track* of OAEI 2007. Figure 9 (a) shows the mapping results for #101 and #205, and we see that RiMom performs best. However, the performance of our method is relatively weak: the Precision is 93% and Recall is 98%. Figure 9(b) shows the mapping results for #101' and #205'. This time, the improvement in Recall quality of our method is significant high; 214% over Falcon and 165% over RiMom. In addition, our method has

[4] Flacon, ASMOV and RiMon can only find out equivalent relations in ontologies. Thus in the experiment, we consider subclass as equivalent class.



(a)

(b)

Figure 9: Evaluation-2

an impressive improvement in F1: 108% and 121% over Falcon and RiMom, respectively. From the experimental data, we also see that Falcon and Rimom perform very well when the ontologies contain more structural information and special tags, for example, *rdfs:label* or *rdfs:comment*. However, when the structural information, special tags, etc were not available, our method performs much better. When we compare the F1 in Figure 8 and F1 in Figure 9(b), we find that our method is also somewhat robust(F1 in these Figures are about 70%).

Related Works

One area of current research which has similarities with our work is on sentences simplification in NLP area. (Vickrey D. *et al.*2008) is one of most related works, but the difference is that we are focusing on a somewhat different task. For example, in (Vickrey D. et al. 2008), the authors simplify the sentence "I was not given a chance to eat." as "I ate". For mapping purposes, we cannot afford simplify the sentence in a way that largely distorts the semantics.

In the area of WordNet, some researchers have proposed the notion of semantic distance. In **Introduction**, we have shown the difference. Here, we introduce a project eXtended WordNet[5], which produces parse trees of the all glosses in WordNet. But Their parse trees contains many errors. For implementation purpose, we use Stanford Parser.

Another group of related work focuses on ontology mapping. Since 2004, OAEI has been organizing evaluation campaigns, and providing opportunities for evaluating ontology mapping systems. Although many methods and tools were developed, and some of them also use WordNet as domain knowledge. To our knowledge, most of them only use synonyms, hypernyms and hyponym relations of the words in WordNet and the actual meanings provided in natural English is often ignored. This served as one of the motivating factors for our research.

Conclusion and Future Work

In this paper, we proposed an ontology mapping technique where we have used the meanings of the concepts as given in Wordnet (in English) for semantic mapping by constructing their parse trees first and then simplifying them for computing similarity measures. Our experimental results show that our method performs well in Recall and F-Measure as compared to other approaches.

In the future, there are a number of improvements that could be made. (1) Since the existing parsers do not parse the sentences in WordNet always correctly. We plan to produce a tool which will fix the errors in parse trees using semi-automatic means. (2) Currently, we only compare each pair of layers of the parse trees independently. How to integrate all the layers as one comprehensive entity, is what we will plan to do in the near future. (3)In the rules set we proposed, we only considered the glosses from WordNet, and these rules may not work for any general sentences outside WordNet. We plan to improve the rule set to handle more general forms of sentences as well.

References

Budanitsky A. and Hirst G.. Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures, Proceedings of the NAACL 2001

Dan Klein and Christopher D. Manning. Fast Exact Inference with a Factored Model for Natural Language Parsing. NIPS 2002

Euzenat, J., Shvaiko, P.: Ontology Matching. Springer 2007

Isaac, A., Meij, L., Schlobach, S., and Wang, S.An empirical study of instance-based ontology matching. ISWC 2007

Klyne, G., and Carroll, J.J. (eds.). Resource Description Framework (RDF): Concepts and Abstract Syntax. http://www.w3.org/TR/rdf-concepts/

Lin D. An information-theoretic definition of similarity. In Proceedings of the 15th ICML 998

Nagy, Mi., Maria V., Stolarski, P., and Motta E. DSSim results for OAEI 2008, The Third International Workshop on Ontology Matching, 2008

Patel-Schneider, P.F., Hayes, P., and Horrocks, I. (eds.). OWL Web Ontology Language Semantics and Abstract Syntax. http://www.w3.org/TR/owl-semantics/

Udrea O. Getoor L. and Miller R.J. Leveraging Data and Structure in Ontology Integration, SIGMOD 2008

Vickrey D. and Koller D. Sentence Simplification for Semantic Role Labeling, ACL 2008

Yves R. Jean M., Mansur R. ASMOV: results for OAEI 2008. The Third International Workshop on Ontology Matching, 2008

[5] http://xwn.hlt.utdallas.edu/index.html