oWSD: A Tool for Word Sense Disambiguation in Its Ontology Context

Xia Wang
National University of Ireland, Galway
Digital Enterpise Research Institute
Galway, Ireland
xia.wang@deri.org

Vassilios Peristeras
National University of Ireland, Galway
Digital Enterpise Research Institute
Galway, Ireland
vassilios.peristeras@deri.org

ABSTRACT

Word sense disambiguation (abbr. WSD) is very important to the semantic web/web 2.0. However, there is still no easy-to-use tool available. As a remedy, here a simple and very efficient tool called $o\mathcal{WSD}$ is demonstrated. It disambiguates the senses of words in their ontological contexts, and obtains the right word senses from WordNet. It is very helpful to applications involving ontologies and natural language processing as well.

1. INTRODUCTION

Word sense disambiguation is an important issue to understand the semantic web/web 2.0, because the semantic web [5] can be regarded as addition of a machine-understandable and machine-tractable layer to complement the existing web of natural language hypertext, in order to support automatic communication between web-based applications. As the pages of the semantic web link to ontologies, concept ambiguities of heterogeneous ontologies easily lead to misunderstanding problems and errors.

Word sense ambiguity is a pervasive characteristic of human languages. The word "bank", for instance, has several senses and may refer to "a slope besides a body of water" or to a "bank building". Fortunately, it is also common that the specific sense intended is determined by the textual context of a word. For example, in a geographical ontology, "bank" only means "bank of water". Therefore, the problem of word sense disambiguation is defined as the task of automatically assigning the most appropriate meaning to a polysemous word within a given context [4].

2. $O\mathcal{WSD}$: A TOOL FOR WORD SENSE DISAMBIGUATION

The $o\mathcal{WSD}$ tool[1] is developed as an eclipse plug-in in Java. The main demonstrated features are as follows:

- *Import Ontology*: Currently, its input is limited to Web service modeling ontology (WSMO) [3] in WSML Syntax[2]. After importing an ontology, it parses this ontology and get all its ontological concepts. This Ontology is the text context of its concepts.

  As the example shown in Fig.1, it is a travel Ontology and it has 11 concepts which are listed in the left text filed.

- *Clean Ontological Concepts*: For heterogeneous ontologies are used in the Semantic web, concepts could be compound concepts or informal concepts with any delimiters, as words "trainTimeTable" and "terms" of the imported travel ontology. In this tool, the cleanness consists of splitting compound words, cleaning the delimiters, and replacing the abbreviations by normal words.

- *Load Dictionary*: So far, the dictionaries will be loaded. WordNet [1] is set as the default dictionary. For WordNet is a semantic lexicon for the English language, its database contains about 150,000 words organized in over 115,000 synsets for a total of 207,000 word-sense pairs. Moreover, there already have several mature Java APIs to WordNet. We use the JWord 2.0[3].

- *Execute WSD Algorithm*: Now, the concepts and the referred dictionary are both ready to execute WSD. Through reference to [2], the process of the WSD algorithm (defined in [6]) includes:

  (a) to analysis all the concepts in WordNet in order to get how many polysemy words, Single-words, exceptional words (which can not be dealt by WordNet) are. The principle is only the polysemy words will execute the WSD algorithm.

  (b) to generate all the wordwindows. If the size of wordwindow is assumed as 5, the first word of the wordwindow must be the target word and the rest are the context words. The principle is to use the single-words in each wordwindow as many as possible in order to improve efficiency.

  (c) to calculate the concept density by inspecting every subhierarchy of word sense branches retrieved from WordNet. Finally, the sense which has the highest concept density will be the result sense.

  For example, the sense #2 of word "delivery" in Fig.1 has the 0.6385 score, it will be the right sense.

- *Data Analysis*: We analysis this tool by the statistics results of concept cleaning method and the WSD algorithm. As shown in Fig.1, in the first table, after parseing and cleaning, the number concepts changes from 11 to 12 and there has one compound word "trainTimeTable" and one word "terms" has delimiter. In the second table, during the WSD algorithm, there are 11 polysemy words are 1 single-word. Finally, there are 10 polysemy words successfully eliminating their ambiguity. Therefore, the precision of our WSD algorithm is $1/11 = 0.909$.

[1]The demo is available at `http://wsao.deri.ie/`.

[2]Web service modeling language (WSML) is at `http://www.wsmo.org/TR/d16/d16.1/v0.21/`

[3]http://www.seas.gwu.edu/ simhaweb/software/jword/

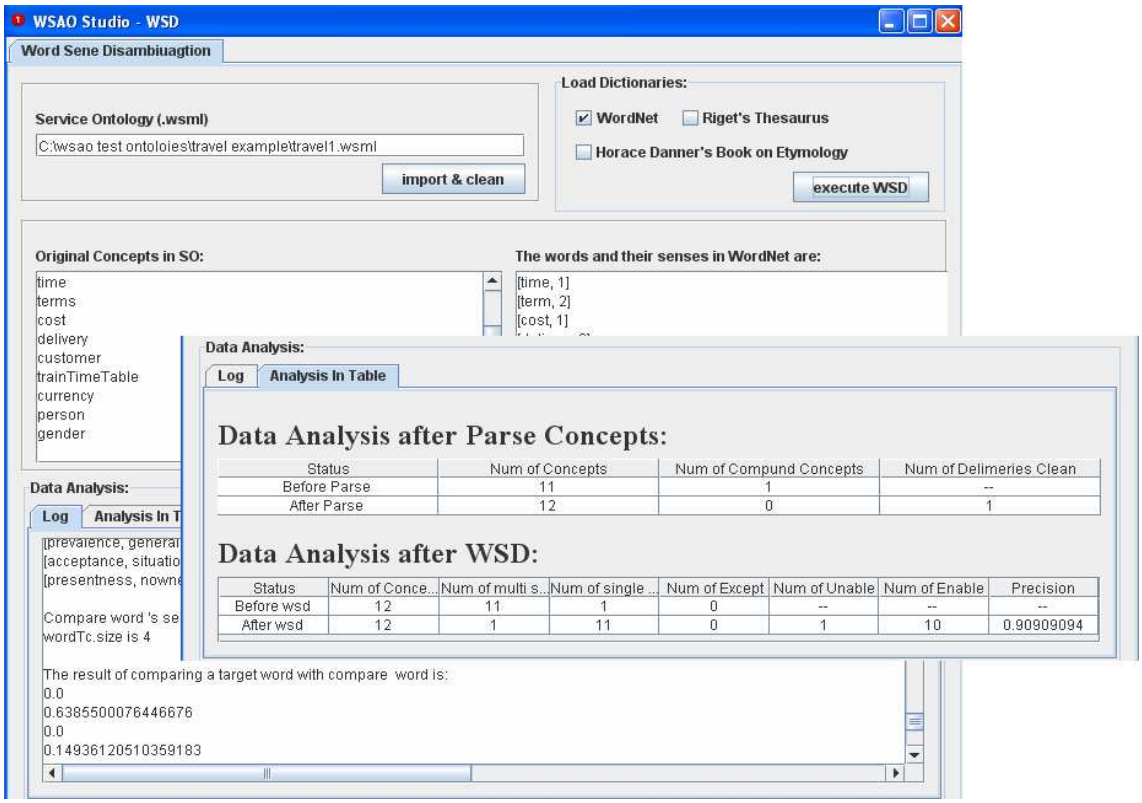

Figure 1: WSD Workbench

3. RELATED WORK

Although many approaches to WSD have been studied in the fields of natural language processing and ontologies, to the best of our knowledge there is no simple and running tool available. There is just a related, paper-based tool named GWSD for unsupervised graph-based word sense disambiguation, which was developed by Mihalcea[4]. It uses a representation of words as vertices and their senses as labels in graphs, and combines similarity metrics and graph centrality algorithms to select correct word senses. This approach can work very well within a weighted graph built, but it is unclear how to build such word sense dependency graphs.

In comparison, the advantages of $o\mathcal{WSD}$ are: (1) it is simple as it just imports an ontology and obtains concept senses from WordNet, (2) it is a running tool, (3) the WSD algorithm is efficient with specific quantifiable concept densities, and (4) the application ontology is the best choice as context of concepts.

4. CONCLUSION

The tool $o\mathcal{WSD}$ to disambiguate the senses of words in their ontological contexts was described. It uses WordNet as word dictionary, and selects correct word senses by calculating all concept densities. Its further development includes (1) extending the dictionaries by adding Riget's thesaurus, (2) extending the import file format to OWL or RDF, (3) adding a OWL-WSML translator as a small plug-in, and (4) evaluating its formally.

5. REFERENCES

[4]http://www.cs.unt.edu/ rada

[1] C. Fellbaum (Ed.), WordNet: An Electronic Lexical Database. Cambridge, MA: MIT Press 1998.

[2] E. Agirre and G. Rigau, Word Sense Disambiguation using Conceptual Density, COLING 1996: 16–22.

[3] H. Lausen, A. Polleres and D. Roman. Web Service Modeling Ontology (WSMO). Technical report, W3C Member Submission, 3 June 2005.

[4] R. Sinha and R. Mihalcea, Unsupervised Graph-based Word Sense Disambiguation Using Measures of Word Semantic Similarity, Proc. IEEE Intl. Conf. on Semantic Computing, Irvine, CA, 2007.

[5] T. Berners-Lee, J. Hendler and O. Lassila, The Semantic Web, *Scientific American*, 284(5):34–43, 2001.

[6] X. Wang, T. Vitvar, M. Hauswirth and D. Foxvog, Building Application Ontologies from Descriptions of Semantic Web, IEEE/WIC/ACM Intl. Conf. on Web Intelligence, 2007.