IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 1, Issue 2, April-May, 2013
ISSN: 2320 - 8791
www.ijreat.org

A Novel Approach to Compute Semantic Relationship between Words

K.Morarjee[1], Santosh Kumar Eruventi[2], D.Baswaraj[3], A. Vasanthi[4]
[1]Assistant Prof, CSE Dept., CMRIT, Hyderabad, India
[2]M.Tech (CSE) Student, CSE Dept., CMRIT, Hyderabad, India
[3,4] Associate Prof, CSE Dept., CMRIT, Hyderabad, India

ABSTRACT

A web search engine returns many pages, when user searches information regarding a particular person. Some of these pages may be for other peoples with the same name. How can we disambiguate these peoples with the same name? This paper presents an unsupervised algorithm which produces unique phrases to disambiguate different people with the same name (i.e. namesakes). Our algorithm takes in a personal name and outputs multiple sets of phrases which uniquely identify the different namesakes on the web. These phrases could then be added to the query to narrow down the search to a specific namesake. We evaluated the algorithm on a collection of documents retrieved from the Web. Experimental results show a significant improvement over the existing methods proposed for this task.

Index Terms—Web mining, information extraction, web text analysis.

I. INTRODUCTION

Measuring the semantic similarity between words accurately is an important problem in various applications like web mining, information retrieval, and natural language processing. Web mining applications such as, community extraction, relation detection, and entity disambiguation; require the ability to accurately measure the semantic similarity between concepts or entities. In information retrieval, one of the main problems is to retrieve a set of documents that is semantically related to a given user query. Efficient estimation of semantic similarity between words is critical for various natural language processing tasks such as word sense disambiguation (WSD), textual entailment, and automatic text summarization.

Semantically related words of a particular word are listed in manually created general-purpose lexical ontologies' such as WordNet. In WordNet, asynset contains a set of synonymous words for a particular sense of a word. However, semantic similarity between entities changes over time and across domains. For example, apple is frequently associated with computers on the web. However, this sense of apple is not listed in most general-purpose thesauri or dictionaries. A user, who searches for apple on the web, might be interested in this sense of apple and not apple as a fruit. New words are constantly being created as well as new senses are assigned to existing words. Manually maintaining on topologies to capture these new words and senses is costly if not impossible.

We propose an automatic method to estimate the semantic similarity between words or entities using web search engines. Because of the vastly numerous documents and the high growth rate of the web, it is time consuming to analyze each document separately. Web search engines provide an efficient interface to this vast information. Page counts and snippets are two useful information sources provided by most web search engines. Page count of a query is an estimate of the number of pages that contain the query words. In general, page count may not necessarily be equal to the word frequency because the queried word might appear many times on one page. Page count for the query P AND Q can be considered as a global measure of co-occurrence of words P and Q. For example, the page count of the query “apple” AND “computer” in Google’s 288,000,000, whereas the same for “banana” AND “computer” is only 3,590,000. The more than 80 times more numerous page counts for “apple” AND “computer” indicate that apple is more semantically similar to computer than is banana.

However, identification of disambiguation-enabling knowledge types is only one side of the story, and to build a practical WSD system knowledge also needs to be efficiently acquired at a large scale. In general, knowledge used in a practical WSD system need satisfy the following criteria:

1) Disambiguation-enabling. Obviously useful WSD knowledge should be capable of disambiguating senses. Identification of such knowledge is still a very active research topic, and new knowledge is constantly being proposed and examined.

2) Comprehensive and automatically acquirable. The disambiguation knowledge need cover a large number

of words and their various usages. Such a requirement is not easily satisfied since a natural language usually contains thousands of words, and some words can have dozens of senses. For example, the Oxford English Dictionary has approximately 301,100 entries, and the average polysemy of WordNet inventory is 6.18. Obviously, such a large-scale knowledge acquisition can only be achieved with automatic techniques.

3) Dynamic and up to date. A natural language is not a static phenomenon. New usage of existing words emerges, which creates new senses. New words are created, and some words may “die” over time. It is estimated that every year around 2,500 new words appear in English. Such dynamics requires constant and timely maintenance and updating of WSD knowledge base, which makes any manual interference (e.g., sense annotation and supervised learning) even more impractical.

II. RELATEDWORK

Generally WSD techniques can be divided into four categories,

1) Dictionary and knowledge based methods uses Lexical Knowledge Bases (LKB), such as dictionaries to extract knowledge from word definitions and relations among words/senses. Recently, several graph-based WSD methods were proposed. In these approaches, first a graph is built with senses as nodes and relations among words/senses (e.g., synonymy, antonym) as edges, and the relations are usually acquired from a LKB (e.g., Word Net). Then a ranking algorithm is conducted over the graph, and senses ranked the highest are assigned to the corresponding words. Different relations and ranking algorithms were experimented with these methods, such as Tex Rank algorithm, personalized Page Rank algorithm, a two-stage searching algorithm and centrality algorithms.

2) Supervised methods include a training phase and a testing phase. In the training phase, a sense-annotated training corpus is required, from which syntactic and semantic features are extracted to build a classifier using machine learning techniques, such as Support Vector Machine. In the following testing phase, the classifier picks the best sense for a word based on its surrounding words. Currently supervised methods achieved the best disambiguation quality (about 80% in precision and recall for coarse-grained WSD in the most recent WSD evaluation conference SemEval 2007). Nevertheless, since training corpora are manually annotated and expensive, supervised methods are often brittle due to data scarcity and it is impractical to manually annotate huge number of words existing in a natural language.

3) To overcome the knowledge acquisition bottleneck faced by supervised methods, semi-supervised methods make use of a small annotated corpus as seed data in a bootstrapping process. A word-aligned bilingual corpus can also serve as seed data.

4) Unsupervised methods acquire knowledge from unannotated raw text, and disambiguate senses using similarity measures. Unsupervised methods over-come the problem of knowledge acquisition bottleneck, but none of existing methods can outperform the most frequent sense baseline, which makes them not useful at all in practice. For example, the best unsupervised systems only achieved about 70% in precision and 50% in recall in the SemEval 2007 Workshop. One recent study utilized automatically acquired dependency knowledge and achieved 73% in precision and recall, which are still below the most-frequent-sense baseline (78.89% in precision and recall in the SemEval 2007 Task 07).

Additionally there exist some “meta-disambiguation” methods that ensemble multiple disambiguation algorithms following the ideas of bagging or boosting in supervised learning. The multiple sources were utilized to achieve optimal WSD performance [15]. Our approach is different in that our focus is identification and ensemble of new disambiguation-enabling and efficiently acquirable knowledge sources. In this paper we propose a new fully automatic WSD method by integrating three types of knowledge: dependency relations, glosses, and the most-frequent-sense (MFS) information. In next section we will discuss how to acquire and represent the knowledge.

III. SOLUTIONS TO THE KEY PROBLEMS OF THE BILINGUAL CORPUS ACQUIRING SYSTEM

When processing with CJFD, we were confronted with lots of problems, such as transaction, page information obtaining, and so on. The following presents the solutions to some problems we met.

1) Page Information Obtaining

Lots of information is contained in data Scroller field, the system has to get their texts. CJFD use Asynchronous JavaScript and XML (AJAX) to page the query results. We simulate the turn page function by sending turn page information to the remote Website, so that we can get all the page information sequentially.

According to this, we successfully get all the URLs of journals and articles.

2) Translation Pairs Extraction

After downloading the Web pages of papers, the next task is to extract the parallel pairs in them. Again, we utilize the structured HTML markup and the sequence of the pairs in the Web pages. Moreover, we get the category navigation information from this layer of pages.

3) Transaction

Because of the enormous data from CJFD, BTCD could not visit all the pages in a relatively short period. During the run phase, we will meet many errors, such as remote server maintenance, power failure, network problem and so on. Therefore, it is necessary to add a transaction to keep BTCD safe and reliable.

To solve these problems, we need to set backup points for BTCD, and journals URLs are perfect one. Because BTCD spend 2-3 hours on each journal on average, we can limit the delay less than 3 hours when errors occur. For example, if BTCD could not visit CJFD, the system would stop and wait for one hour before restart. From the process record, system finds that it run to “Journal of Computer”, then BTCD deletes all the pairs related to this journal and restart from it. As the corpus grows, the normal delete SQL statement can hardly work effectively, thus it will cause some errors. Therefore we optimized this statement, the new SQL statement is “delete from corpus where document from= ‘Journals%’ order by document ID desc limit 10000”. This statement aims to delete pairs from the bottom of data and limit the deleting count.

IV. SYSTEM IMPLEMENTATION

The system architecture is quite simple. To implement it, we developed a small system according to CJFD without using a general purpose web crawler.

1) System Flow

To enhance the efficiency of the system, BTCD just obtains useful pages and URLs instead of using professional web crawler to download all the pages in Website. Moreover, if we use web crawler to download pages, the system would remove duplicates by content. For example, the URL of one paper is http://dlib.cnki.net/kns50/detail.aspx?QueryID=39&CurRec=1; it is not a URL which links to this paper, but a search statement which is sent to remote Website with session. The following is our basic processing steps:

Step.1 Get the index URLs of all the journals in CJFD a total of 9056 in our case.

Step.2 Access these URLs and analyze the page contents one by one, then find all the sub-URLs belonging to this journal, such as URLs of Year 2008, 2009.

Step.3 Access the URL above sequentially, and filter the content and find all the URLs.

Step.4 Again Access the URL above sequentially, extracts URLs of all the papers. From this step, we should keep connection between the remote website; otherwise the URLs we got are useless.

Step.5 Connect the remote website with sessions and access the URLs obtained above. Then we will get the article pages.

2) System Modules

BTCD System is composed of journals URLs module, bilingual pages module, corpus module. Functions of these modules are shown in Table 1. Each module provides input data for the next module, and gets data from the previous module. The system has to download huge amount of web pages, we do not keep these pages if we filter them, so that we will save tons of hard memory space. BTCD is a java program, so it is a platform-independent system.

Module	Function
Journals URLs module	Use of query function of CJFD to get all URLs of Journals
Bilingual pages module	1) Connect to web pages without session. 2) Extracts URLs from HTML 3) Transactions
Corpus module	1) Filter noises in the HTML 2) Obtain Parallel texts and category information of them. 3) Extract structured parallel texts. 4) Save data to database

TABLE I: SYSTEM MODULES

3) Data Flow

Each system module contains several procedures. We package them respectively as Step1, Step2, and Step3. The data flow of BTCD is shown in Figure 2. Step1 just runs once, because it downloads all the URLs of journals and saves it as text which is provided to Step2.

Step3 gets data from Step2. After Step3 completing, it goes back to Step2 and re-obtains the bilingual pages for the next journal.

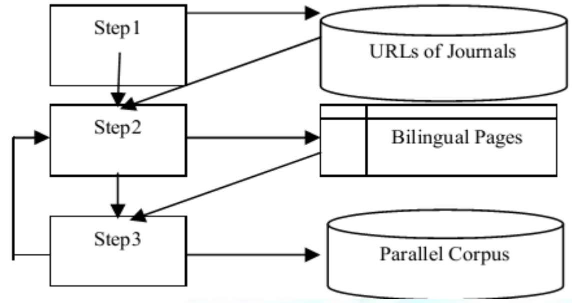

Figure 2. Data Flow

4) Database

According to the information of papers, we save bilingual titles, abstracts, names, and keywords. For transaction and classifying process, we also keep the journal information and category information. We store parallel texts into database instead of saving as plain text in order to reform these results for future usage.

V. EVALUATION

Research on WSD not only provides valuable insights into understanding of semantics, but also can improve performance of many important Natural Language Processing applications. Recently several workshops have been organized to evaluate WSD techniques in real world settings. In this section, we will discuss our experiment results with two large scale WSD evaluation corpora, Senseval-2 fine-grained English testing corpus and SemEval 2007 Task 7 coarse-grained testing corpus. Both evaluations require the disambiguation of all nouns, verbs, adjectives, and adverbs in the testing articles, which is usually referred as “all-words” task.

1) Experiment with Senseval-2 English testing corpus

Senseval-2, the Second International Workshop on Evaluating Word Sense Disambiguation Systems, evaluated WSD systems on two types of tasks (all word or lexical sample) in 12 languages. 21 research teams participated in English all-word task [14]. In Senseval-2 testing corpus, there are totally 3 documents, which include 2473 words that need to be disambiguated. Article 1 discusses churches in England and contains 684 words that need to be disambiguated, article 2 discusses a medical discovery about genes and cancers and contains 1032 words that need to be disambiguated, and article 3 discusses children education and contains 757 words that need to be disambiguated. Table I shows our system performance along with the ten best-performing systems participated in Senseval-2. Our WSD system achieves similar performance as the best supervised system, and also outperforms MFS baseline.

System	Precision	Recall	F1 score
SMUaw (supervised)	0.69	0.69	0.69
CNTS-Antwerp (supervised)	0.636	0.636	0.636
UHD system (unsupervised)	0.633	0.633	0.633
Sinequa-LIA-HMM (supervised)	0.618	0.618	0.618
MSF baseline	0.617	0.617	0.617
UNED-AW-U2 (unsupervised)	0.575	0.569	0.572
UNED-AW-U (unsupervised)	0.556	0.55	0.553
UCLA-gchao (supervised)	0.5	0.449	0.473
UCLA-gchao2 (supervised)	0.475	0.454	0.464
UCLA-gchao3 (supervised)	0.474	0.453	0.463
DIMAP (R) (unsupervised)	0.451	0.451	0.451
DIMAP (unsupervised)	0.416	0.451	0.433

TABLE II: COMPARISON WITH TOP-PERFORMING SYSTEMS INSENSEVAL-2.

2) Experiment with SemEval 2007 Task 7 testing corpus

To further evaluate our approach, we evaluated our WSD system using SemEval-2007 Task 07 (Coarse-grained English All-words Task) test data [12]. The task organizers provide a coarse-grained sense inventory, trial data, and test data. Since our method does not need any training or special tuning, coarse-grained sense inventory was not used. The test data includes: a news article about “homeless”, a review of the book “Feeding Frenzy”, an article about some traveling experience in France, an article about computer programming, and a biography of the painter Masaccio. Two authors of [12] independently annotated part of the test set (710 word instances), and the pairwise agreement was 93.80%. This inter-annotator agreement is usually considered as an upper bound for WSD systems.

Senseval-2 and Semeval 2007 WSD test corpora provide evaluation for both coarse-grained and fine-grained senses, and cover diverse topics and a significant portion of commonly-used English words (A college graduate knows approximately 20,000 - 25,000 English words). Evaluation with these two testing corpora clearly shows the effectiveness of our approach and its potential application in many practical NLP systems.

VI. CONCLUSION

We present an automatic system which extracts parallel texts from CJFD. This system crawls all the useful pages on CJFD, benefits from the structured HTML, and filters all the noises easily. The method is quite simple, accurate and efficient. Finally, we extract parallel texts in the Web pages successfully. According to the features of CJFD, we can guarantee the size and quality of the generated bilingual corpus. The experimental results are very encouraging and we will build a Gigabyte level bilingual parallel corpus which is based on academic journals. In addition, we have successfully collected enormous bilingual terms which are valuable to lexical acquisition. In the future, we will focus on sentence alignment and download the latest articles from CJFD. In this way, the acquired corpus will keep updating.

REFERENCES

[1] S.N. Ye, Y.J. Lv, Y. Huang, and Q. Liu, “Automatic Parallel Sentences Extraction from Web”, Journal of Chinese Information Processing, China, 2006, pp. 67-73.

[2] L. Jiang, S. Yang, M. Zhou, “Mining Bilingual Data from the Web with Adaptively Learnt Patterns”, 47th Annual Meeting of the Association for Computational Linguistics (ACL 09) 2009.

[3] X.Y Ma and M.Y. Liberman, “Bits: A method for bilingual text search over the Web”, Proceedings of the Machine Translation Summit VII, 1999.

[4] J. Chen and J.Y. Nie, “Automatic construction of parallel English-Chinese corpus for cross-language information retrieval”, Proceedings of the International Conference on Chinese Language Computing, San Francisco, 2000, pp. 21-28.

[5] Matthias Blume, ‘Automatic entity disambiguation: Benefits to the, relation extraction, link analysis, and inference’, in *Proceedings of International Conference on Intelligence Analysis*, (2005).

[6] K.T. Frantzi and S. Ananiadou, ‘Extracting nested collocations’, *16th Conference on Computational Lingustics*, pp. 41–46, (1996)

[7] K.T. Frantzi and S. Ananiadou, ‘The c-value/nc-value domain independent method for multi-word term extraction’, *Journal of Natural Language Processing*, **6(3)**, 145–179, (1999)

[8] M. Hernandez and S. Stolfo, ‘The merge/purge problem for large databases’, *SIGMOD Conference*, pp. 127–138(1995)

[9] Dmitri V. Kalashnikov, Sharad Mehrotra, and Zhaoqi Chen, ‘Exploiting relationships for domain-independent data cleaning’, in *SIAM International Conference on Data Mining (SIAM SDM)*, Newport Beach, CA, USA, (April 21–23 2005).

[10] Ravi Kannan, Santosh Vempala, and Adrian Vetta, ‘On clustering: Good, bad and spectral’, *Computer Science*, (2000).

[11] Lillian Lee, ‘On the effectiveness of the skew divergence for statistical language analysis’, *Artificial Intelligence and Statistics*, 65–5, (2001).

[12] Xin Li, Paul Morie, and Dan Roth, ‘Semantic integration in text, from ambiguous names to identifiable entities’, *AI Magazine, American Association for Artificial Intelligence*, **Spring**, 45–58, (2005).

[13] Gideon S. Mann and David Yarowsky, ‘Unsupervised personal name disambiguation’, *Proceedings of CoNLL-2003*, pp. 33–40, (2003)

[14] Yutaka Matsuo, Junichiro Mori, and Mitsuru Hamasaki, ‘Polyphonet: An advanved social network extraction system from the web’, in *Proceedings of the World Wide Web Conference*, (to appear in 2006).

[15] A. McCallum and B. Wellner , ‘Toward conditional models of identity uncertainty with application to proper noun coreference’, in *IJCAI Workshop on Information Integration on the Web, 2003*, (2003).

[16] D. McCarthy, R. Koeling, J. Weeds, and J. Carroll, ‘Finding predominant word senses in untagged text’, in *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL’04)*, pp. 279– 286, (2004).