EDL: A Framework for Entity Disambiguation and Linking to Knowledgebases

Behnam Ojaghi Kahjogh
Computer Engineering Dept.
TOBB University of
Economics and Technology
Ankara, Turkey
Email: bojaghikahjogh@etu.edu.tr

Jeyhun Karimov
Computer Engineering Dept.
TOBB University of
Economics and Technology
Ankara, Turkey
Email: jkarimov@etu.edu.tr

Erdogan Dogdu
Computer Engineering Dept.
TOBB University of
Economics and Technology
Ankara, Turkey
Email: edogdu.edu.tr

***Abstract*—Extraction and integration of entities from textual data and linking them to knowledgebases (for further information or processing) is useful for many applications in natural language processing. However, a major problem in this process is disambiguation, named entities might refer to different things. In this work, we propose a novel method to disambiguate named entities. Our method is a combination of search engine results and knowledgebase repository mining results. We obtained 84% correct disambiguation rates with the data sets we used.**

I. Introduction

Entity linking and disambiguation is a major research area especially with the recent developments in the Web. Semantic web and linked data[1] are two common terms that relates to the waste amount of structured data on the Web. Linked data resources are utilized for entity linking and disambiguation quite extensively. Entities in free text are recognized and linked to linked data resources in this context.

One of the major problems in entity linking is the disambiguation, in which a named entity mention can be about a number of things in the linked data knowledgebase and this requires identification of the right entity from the knowledgebase. Here we propose a new method for disambiguation and show that it has a comparable success.

II. Literature Review

A lot of research has focused on named entity recognition and disambiguation in recent years [1], [2]. The traditional methods disambiguate named entities based on the bag of words (BOW) model. Bagga and Baldwin [3] represented a name as a vector of its contextual words, then the similarity between two named entities are determined by the co-occurring words, and finally two names are predicted to be the same entity if their similarity is above a threshold. Disambiguation by Cucerzan [4] is through linking them to Wikipedia entities and comparing their term vector representations. Mann and Yarowsky [5] further improved the model by extracting biographic facts. Bunescu and Pasca [6] disambiguated the names in Wikipedia by linking them to the most similar Wikipedia entities using the similarity computed using a disambiguation SVM kernel [7].

There is an urgent need to integrate the extracted facts with an existing knowledge base. They are largely based on the co-occurrence statistics of terms between the text around the entity mention and the document associated with the entity. In [8] LINDEN, a novel framework to link named entities in text with a knowledgebase unifying Wikipedia and WordNet is proposed, by leveraging the rich semantic knowledge embedded in the Wikipedia and the taxonomy of the knowledgebase [9]. In information extraction, entity linking is a new method that has drawn a lot of attention from NLP researchers recently. This method has varied applications ranging from linking patient health records to maintaining personal credit files, prevention of identity crimes, and supporting law enforcement.

III. Proposed Model

Our proposed model consists of 3 steps: (1) querying knowledgebases for the particular named entity candidates in text, (2) querying search engines for the named entity candidates, (3) comparing the results from the previous two steps and output the best matching results.

To process and compute the results, we used the class structure presented in Figure 1. Here, BOW-Map<String,Object> structure is used to collect the results from knowledgebase (P1Result) which is done in first part and search engine (P2Result) which is done in second part, and BOW-Map structure is presented in Figure 2. Below we explain the three steps.

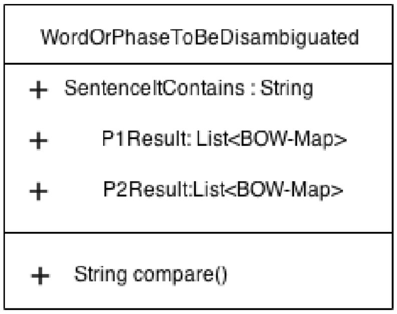
WordOrPhaseToBeDisambiguated
+ SentenceItContains : String
+ P1Result: List<BOW-Map>
+ P2Result:List<BOW-Map>
+ String compare()

Fig. 1: Class structure for keeping intermediate results

[1] www.linkeddata.org

978-1-5090-0662-5/16 $31.00 © 2016 IEEE
DOI 10.1109/ICSC.2016.77

A. Collecting Data from Knowledgebase

In this step, we get a sentence or free text, and a constant number c_w as input. c_w is the size of a sliding window for keyword lookup. The sentence is parsed and split into word and the words in sliding window space are queried in DBPedia[2], a popular and large linked data data set. We specifically search $abstract$ data in the data set to match the search keyword(s). While querying DBPedia, we put a constraint that the word or phrase must have disambiguation. There is no doubt that in most cases we will have more than one query result from DBPedia. After getting query results, if not empty, we create an object for the phrase as shown in Figure 1 and assign the necessary fields. We assign $SentenceItContains$ and $P1Result$ fields. The last one is constructed using the collection of $abstract$ fields from DBPedia results in a form of bag-of-words as shown in Figure 2.

For example, for the sentence "Jaguar is British multinational car manufacturer", we start querying first the term "Jaguar" in DBPedia. It is obvious that it is terms that corresponds to more than one entity in DBpedia. For the first entity we got from DBPedia, say it is a car, we put DBPedia link for the car entity as the key of BOW-Map structure and the entity's abstract value (text) as $value$ in the structure. The same is done for the remaining results from DBpedia and put into the list.

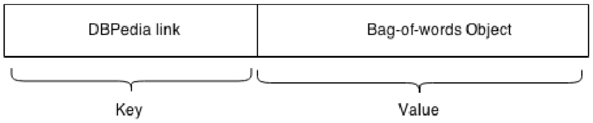

Fig. 2: Sample structure of hashmap used in P1 (BOW-Map)

B. Collecting Data from Search Engines

In this step we get an array of all objects constructed in the first step. First, for each object, we construct a new phrase by concatenating the nearest nouns to that word or phrase, and the verb of a sentence. Second, we query this new phrase in a search engine. We used Bing search engine for this step. After getting search results, we collect c_s number of search results with their title text shown in the search results page. Third, we construct the bag-of-words from these results as shown in Figure 2 and set the $P2Result$ to it.

C. Merging Results and Disambiguation

In the third step, we get all objects constructed in the previous two steps, and *compare* them. The compare() method compares $P1Result$ and $P2Result$ lists according to the maximum word similarity in the bag-of-words structure. This method returns the most similar bag-of-words result and its retrieved DBPedia link.

IV. Experiments

We used Meij twitter data set[3] to test our model. It contains 502 tweets. The accuracy result is 84% (Figure 3).

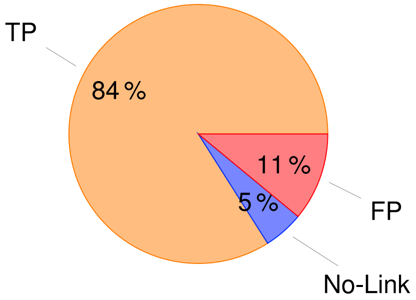

Fig. 3: The results for the proposed model.

As can be seen from Figure 3, the true positive rate is 84%. Some of the results do not a have DBpedia link, therefore they are categorized as "No-Link" (5%) and the remaining results are false positive.

V. Conslusion

In this paper, we proposed a solution to disambiguate named entities using a hybrid method based on knowledgebases and search engines. Our accuracy rate is 84% on Meij data set. We are working on collecting a larger data set and improving the method.

References

[1] S. Hakimov, S. A. Oto, and E. Dogdu, "Named entity recognition and disambiguation using linked data and graph-based centrality scoring," in *Proceedings of the 4th international workshop on semantic web information management*. ACM, 2012, p. 4.

[2] S. Hakimov, H. Tunc, M. Akimaliev, and E. Dogdu, "Semantic question answering system over linked data using relational patterns," in *Proceedings of the Joint EDBT/ICDT 2013 Workshops*. ACM, 2013, pp. 83–88.

[3] X. Han and J. Zhao, "Named entity disambiguation by leveraging wikipedia semantic knowledge," in *ACM CIKM*, 2009, pp. 215–224.

[4] S. Cucerzan, "Large-scale named entity disambiguation based on wikipedia data." in *EMNLP-CoNLL*, vol. 7, 2007, pp. 708–716.

[5] G. S. Mann and D. Yarowsky, "Unsupervised personal name disambiguation," in *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*. Association for Computational Linguistics, 2003, pp. 33–40.

[6] R. C. Bunescu and M. Pasca, "Using encyclopedic knowledge for named entity disambiguation." in *EACL*, vol. 6, 2006, pp. 9–16.

[7] A. Fader, S. Soderland, O. Etzioni, and T. Center, "Scaling wikipedia-based named entity disambiguation to arbitrary web text," in *Proceedings of the IJCAI Workshop on User-contributed Knowledge and Artificial Intelligence: An Evolving Synergy, Pasadena, CA, USA*, 2009, pp. 21–26.

[8] W. Shen, J. Wang, P. Luo, and M. Wang, "Linden: linking named entities with knowledge base via semantic knowledge," in *Proceedings of the 21st international conference on World Wide Web*. ACM, 2012, pp. 449–458.

[9] D. Rao, P. McNamee, and M. Dredze, "Entity linking: Finding extracted entities in a knowledge base," in *Multi-source, multilingual information extraction and summarization*. Springer, 2013, pp. 93–115.

[2] www.dpbedia.org

[3] http://edgar.meij.pro/data set-adding-semantics-microblog-posts