Topic Modeling for Entity Linking using Keyphrase

A.M. Naderi, H. Rodríguez, J. Turmo
{anaderi,horacio,turmo}@lsi.upc.edu

TALP Research Center, UPC, Spain.

Abstract. This paper proposes an Entity Linking system that applies a topic modeling ranking. We apply a novel approach in order to provide new relevant elements to the model. These elements are keyphrases related to the queries and gathered from a huge Wikipedia-based knowledge resource.

1 Introduction

Recently, the needs of world knowledge for Artificial Intelligence applications are highly increasing. As a part of world knowledge, Knowledge Bases (KB) are appropriate for both human and machine readability, involved to keep and categorize entities and their relations. The KB profits by improving the ability of obtaining more amount of discriminative information in a shorter range of time than discovering through all unstructured resources. But the high cost of manual elicitation to create KB forces toward automatic acquisition from text. This requires two main abilities. 1) extracting relevant information of mentioned entities including attributes and relations between them (Slot Filling), and 2) linking these entities with entries in the ontology (Entity Linking–EL). This paper focuses on the latter.

EL is the task of linking an entity mention occurring in a document (henceforth, background document) to a unique entry within a reference KB, e.g. when seeing the text "American politician Chuck Hagel", if the involved KB is Wikipedia (WP), the entity mention "Chuck Hagel" should be linked to the WP entry `http://en.wikipedia.org/wiki/Chuck_Hagel`. Assigning the correct reference of entities such as persons, organizations, and locations is highly challenging since one entity can be referred to by several mentions (synonymy), as well as same mention may be used to depict distinct entities (ambiguity). For instance, "George Yardley" might refer to either the Scottish former footballer, or the American basketball player (ambiguity), who is also known by nicknames such as "Yardbird" or shortly "Bird" (synonymy). The ambiguity can be more challenging, e.g. in the sentence "they have Big Country in this NBA match.", the surface form "Big Country" is referring to "Bryant Reeves", the NBA professional basketball player. In Discussion Fora (DF) such as blogs, etc. the texts might contain grammatical irregularities which make the EL even harder, e.g. consider the sentence "James Hatfield is working with Kirk Hammett". The surface form "James Hatfield" can be referred to the American author, but the correct grammatical form of "Hatfield" is "Hetfield" referring to the main songwriter and co-founder of heavy metal band Metallica. These synonymy and ambiguity challenges make it difficult for natural language processors to realize the correct reference of entity mentions

in the text. In addition, as further challenges faced to the EL, an entity can be mentioned in a text by its partial names (rather than its full name), acronyms or other types of name variation.

This paper proposes an Entity Linking system that applies a topic modeling ranking to face to the ambiguity problem. We apply a novel approach in order to provide elements of the model by taking advantage of keyphrases gathered from a huge WP-based knowledge resource.

2 Literature Review

The recent works on EL in its contemporary history are inspired from the older history of Word Sense Disambiguation (WSD) where this challenge firstly arised. Many studies achieved on WSD are quite relevant to EL. Disambiguation methods in the state of the art can be classified into supervised methods, unsupervised methods and knowledge-based methods [17].

Supervised Disambiguation (SD). The first category applies machine-learning techniques for inferring a classifier from training (manually annotated) data sets to classify new examples. Researcher proposed different methods for SD. A *Decision List* [22] is a SD method containing a set of rules (if-then-else) to classify the samples. In continue, [10] used learning decision lists for *Attribute Efficient Learning*. [13] introduced another SD method *Decision Tree* that has a tree-like structure of decisions and their possible consequences. *C4.5* [19], a common algorithm of learning decision trees was outperformed by other supervised methods [16]. [9] studied on the *Naive Bayes* classifier. This classifier is a supervised method based on the Bayes' theorem and is a member of simple probabilistic classifiers. The model is based on the computing the conditional probability of each class membership depending on a set of features. [16] demonstrated good performance of this classifier compared with other supervised methods. [14] introduced *Neural Networks* that is a computational model inspired by central nervous system of organisms. The model is presented as a system of interconnected neurons. Although [24] showed an appropriate performance by this model but the experiment was achieved in a small size of data. However, the dependency to large amount of training data is a major drawback [17]. Recently, different combination of supervised approaches are proposed. The combination methods are highly interesting since they can cover the weakness of each stand-alone SD methods [17].

Unsupervised Disambiguation (UD). The underlying hypothesis of UD is that, each word is correlated with its neighboring context. Co-located words generate a cluster tending to a same sense or topic. No labeled training data set or any machine-readable resources (e.g. dictionary, ontology, thesauri, etc.) are applied for this approach [17]. *Context Clustering* [23] is a UD method by which each occurrence of a target word in a corpus is indicated as a context vector. The vectors are then gathered in clusters, each indicating a sense of target word. A drawback of this method is that, a large amount of un-labeled training data is required. [12] studied on *Word Clustering* a UD method based on clustering the words which are semantically similar. Later on, [18] proposed a word clustering approach called *clustering by committee* (CBC). [25] described another UD method *Co-occurrence Graphs* assuming that co-occurrence words and their

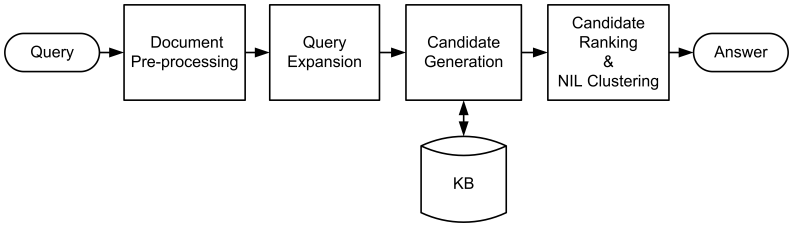


Fig. 1: General architecture of the EL systems.

relations generate a co-occurrence graph. In this graph, the vertices are co-occurrences and the edges are the relations between co-occurrences.

Knowledge-based Disambiguation (KD). The goal of this approach is to apply knowledge resources (such as dictionaries, thesauri, ontologies, collocations, etc.) for disambiguation [11][2][5][3][15]. Although, these methods have lower performance compared with supervised techniques, but they have a wider coverage [17].

Recently, some collective efforts are done to research in this field in form of challenging competitions. The advantage of such competitions is that, the performance of systems are more comparable since all participants assess their systems in a same testbed including same resources and training and evaluation corpus. To this end, *Knowledge Base Population* (KBP) EL track at *Text Analysis Conference* (TAC) [1] is the most important challenging competition being subject of significant study since 2009. The task is annually organized by which many teams present their proposed systems.

3 Methodology and Contribution

The method proposed in this paper follows the typical architecture in the state of the art (Figure 1). Briefly, given a query, consisting of an entity mention and a background document, the system preprocesses the background document (Document Pre-processing step). Then, the background document is expanded integrating more related and discriminative information corresponding to each query in order to facilitate finding the correct reference of each query mention in the KB (Query Expansion step). Subsequently, those KB nodes which can be potential candidates to be the correct entity are selected (Candidate Generation step). Finally, the candidates are ranked in a top-down hierarchy and the candidate having the highest order is selected. Furthermore, all queries belonging to the same Not-In-KB (NIL) entity are clustered together assigning the same NIL id (Candidate Ranking and NIL clustering step). The final task (Candidate Ranking and NIL Clustering) is the most challenging and highly crucial among steps above. In order to rank candidates, we apply topic modeling. As a contribution, we take advantage of keyphrases to enrich the background document in the Query Expansion step in order to improve the performance of the system in ranking candidates.

Details of each step are provided next.

[1] The TAC is organized and sponsored by the U.S. National Institute of Standards and Technology (NIST) and the U.S. Department of Defense.

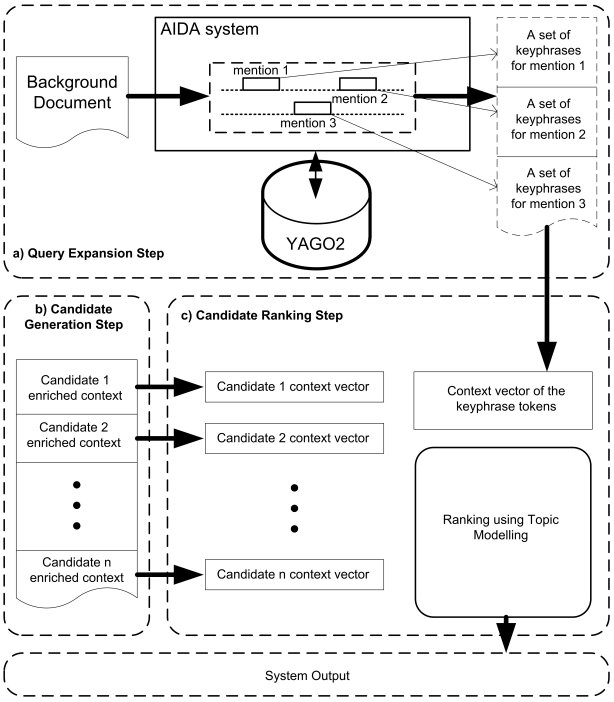


Fig. 2: Detailed architecture of applying keyphrases for ranking candidates.

3.1 Document Pre-processing

Initially, the background document must be converted to a standard structure to be used by other components. To this objective, the system pre-processes the document in the following way.

Document Partitioning and Text Cleaning. This component separates the textual and non-textual parts of the document. Then, the further steps are only applied over the textual part.

In addition, each document might contain several HTML tags and noise (e.g. in Web documents) which are removed by the system.

Sentence Breaking and Text Normalization. This module operates on the context of documents as following:

- *Sentence Breaking.* The documents are splitted by discovering sentence boundaries.
- *Capitalization.* Initial letters of words occurring in titles and all letters of acronyms are capitalized.

dictionary	entity_counts
entity_ids	entity_inlinks
entity_keyphrases	entity_keywords
entity_lsh_signatures_2000	entity_rank
keyphrase_counts	keyword_counts
meta	word_expansion
word_ids	

Table 1: List of tables in YAGO2.

– *Soft Mention (SM).* Entity mentions represented with abbreviations are expanded, e.g. "Tech. Univ. of Texas", is replaced with "Technical University of Texas". To this end, a dictionary-based mapping are applied.

3.2 Query Expansion

In most queries, query name might be ambiguous, or background document contains poor and sparse information about the query. In these cases, query expansion can reduce the ambiguity of query name and enrich the content of documents through finding name variants of the query name, integrating more discriminative information, and tagging meta data to the content of documents.

For doing so, we apply the following techniques:

Query Classification. Query type recognition helps to filter out those KB entities with type different to the query type. Our system classifies queries into 3 entity types: PER (e.g. "George Washington"), ORG (e.g. "Microsoft") and GPE (GeoPolitical Entity, e.g. "Heidelberg city"). We proceed under the assumption that a longer mention (e.g. "George Washington") tends to be less ambiguous than a shorter one (e.g. "Washington") and, thus, the type of the longest query mention tends to be the correct one. The query classification is performed in three steps. First, we use the Illinois Named Entity Recognizer and Classifier (NERC) [20] to tag the types of named entity mentions occurring in the background document. Second, we find the set of mentions in the background document referring to the query. More concretely, we take mention m_1 defined by the query offsets within the background document (e.g. "Bush") and find the set of mentions that include m_1 (e.g. "Bush", "G. Bush", "George W. Bush"). Finally, we select the longest mention from the resulting set of mentions and take its type as the query type.

Background Document Enrichment. This task includes two subsequent steps applied to the background document: a) *mention disambiguation*, and b) *keyphrase exploitation* for each mention. As explained in Section 3.4, we apply *Vector Space Model* (VSM) for ranking candidates. As VSM components are extracted from the background document of each query, we need as most disambiguated entities as possible. For doing so, AIDA system [8] is applied. AIDA is useful for entity detection and disambiguation. Given an unstructured text, it maps entity mentions onto entities registered in YAGO2 [7], a

entity	id
Bill_Gates	134536

(a)

entity	keyphrase	keyphrase_tokens	keyphrase_token_weights	source	count	weight
134536	18098	{18098}	0.0001	linkAnchor	56	0.009

(b)

word	id
IBM	18098

(c)

Table 2: The entity id for the entity "Bill_Gates" (Table 2a), the information of a sample keyphrase for this entity (Table 2b), and the associated keyphrase name (with the length of 1 token) with the keyphrase id 18098 (Table 2c).

huge semantic KB derived from WP, WordNet [4], and Geonames[2]. YAGO2 contains more than 10 million entities and around 120 million facts about these entities. Using AIDA, the system disambiguates as much as possible mentions in the content of background document. Table 1 shows the list of tables in YAGO2 containing the structured information about entities. Each entity in YAGO2 contains several types of information, including weighted keyphrases. A keyphrase which can be used to disambiguate entities, is contextual information extracted by YAGO authors from link anchor, in-link, title and WP category sources of the relevant entity page. For instance, the keyphrase information related to a named entity mention e.g. "Bill Gates" occurring in the background document is respectively obtained from YAGO2 by following SQL commands:

i. `select * from entity_ids where entity='Bill_Gates';`

ii. `select * from entity_keyphrases where entity=134536;`

iii. `select * from word_ids where id=18098;`

The Tables 2a, 2b, and 2c show the commands output respectively. We appended a new component to AIDA system to automatically gather the necessary keyphrases of each entity mention from YAGO2 (performing the SQL commands mentioned above) in order to use these keyphrases in EL task (Figure 2a).

Given that there are several thousands of weighted keyphrases for each entity in YAGO2, a keyphrase weight threshold (set to 0.002) was manually determined for filtering out the less reliable (all keyphrases with weight less than 0.002 in YAGO2) and

[2] `www.geonames.org`

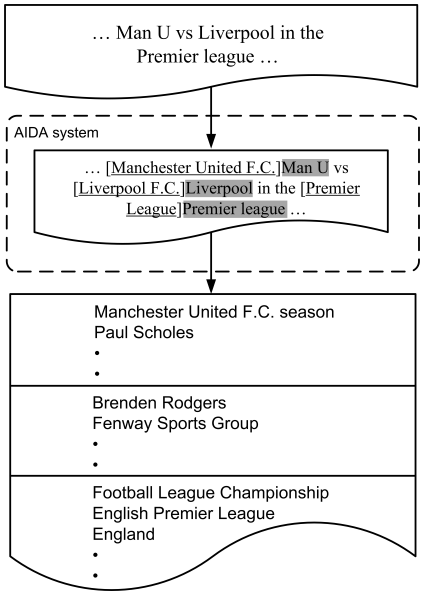


Fig. 3: Sample of generating keyphrases by the system.

```
<entity wiki_title="Parker,_Florida"
type="GPE" id="E0000012" name="Parker,
Florida">

<facts class="Infobox Settlement">
<fact name="official_name">
Parker, Florida
</fact>

<fact name="subdivision_name">
<link entity_id="E0679687">United States</
link>
</fact>

<fact name="subdivision_name1">
<link entity_id="E0373950">Florida</link>
</fact>
......
</facts>

<wiki_text>
<![CDATA[Parker, Florida
Parker is a city in Bay County, Florida,
United States [...] 4.6% of those age 65 or
over.
]]>
</wiki_text>

</entity>
```

Fig. 4: Sample KB candidate entity page containing a set of facts and its informative context.

getting a smaller and more focused set of keyphrases. In general, our system extracts $\sim$300 keyphrases for each entity mention in the background document. Figure 3 shows an example of using AIDA to exploit keyphrases related to following mentions "Man U," "Liverpool," and "Premier league" occurring in the background document.

Alternate Name Generation. Generating Alternate Names (AN) of each query can effectively reduce the ambiguities of the mention, under the assumption that two name variants in the same document can refer to the same entity. We follow the techniques below to generate AN:

- *Acronym Expansion.* Acronyms form a major part of ORG queries and can be highly ambiguous, e.g. "ABC" is referred to around 100 entities. The purpose of acronym expansion is to reduce its ambiguity. The system seeks inside the background document to gather all subsequent tokens with the first capital orderly matched to the letters of the acronym. Also, the expansions are acquired before or inside the parentheses, e.g. "Congolese National Police (PNC)", or "PNC (Congolese National Police)".
- *Gazetteer-based AN Generation.* Sometimes, query names are abbreviations. In these occasions, auxiliary gazetteers are beneficial to map the pairs of ⟨*abbreviation, expansion*⟩ such as the US states, (e.g. the pair ⟨*CA, California*⟩ or ⟨*MD, Maryland*⟩), and country abbreviations, (e.g. the pairs ⟨*UK, United Kingdom*⟩, ⟨*US, United States*⟩ or ⟨*UAE, United Arab Emirates*⟩).

– *Google API.* In more challenging cases, some query names contains grammatical irregularities or a partial form of the entity name. Using Google API, more complete forms of the query name are probed across the Web. For doing so, the component captures title of first (top ranked) result of the Google search engine as possibly better form of the query name. For instance, in the case of query name "Man U", using the method above, the complete form "Manchester United F.C." is obtained.

3.3 Candidate Generation

Given a particular query, q, a set of candidates, C, is found by retrieving those entries from the KB whose names are similar enough (Figure 2b), using Dice measure, to one of the alternate names of q found by the query expansion. In our experiments we used a similarity threshold of 0.9, 0.8 and 1 for PER, ORG and GPE respectively. By comparing the candidate entity type extracted from the corresponding KB page and query type obtained by NERC, we filter out those candidates having different types to attain more discriminative candidates.

In general, each KB entity page contains three main parts, a) information of the entity (title, type, id, and name), b) facts about the entity, and c) an informative context about the entity (Figure 4). As shown in the figure, these facts might in turn include the id of other relevant entities. The system enriches the context part of each KB candidate by extracting the fact ids to get their corresponding KB pages in order to merge their relevant informative contexts with the current one. By applying this technique, the context of each candidate could be more informative. The mentioned figure shows the KB page corresponding to "Parker, Florida." The module collects the `wiki_text` information of its related entities "United States" and "Florida" to enrich the `wiki_text` of "Parker, Florida."

3.4 Candidate Ranking and NIL Clustering

In EL, query expansion techniques are alike across systems, and KB node candidate generation methods normally achieve more than 95% recall. Therefore, the most crucial step is ranking the KB candidates and selecting the best node.

– *Topic Modeling.* This module sorts the retrieved candidates according to the likelihood of being the correct referent. We employs VSM [21], in which a vectorial representation of the processed background document is compared with the vectorial representation of the candidates' `wiki_text`. The vector space domain consists of the whole set of words within the keyphrases found in the enriched background document and the rank consists of their tf/idf computed against the set of candidates' `wiki_text`. All common words (using stop-list) and all words that appear only once are removed. We use cosine similarity. In addition, in order to reduce dimensionality we apply *Latent Semantic Indexing* (LSI) (Figure 2c). The system selects the candidate having the highest degree of similarity as a correct reference of the query name.
– *Term Clustering.* For those queries referring to entities which are not in the KB (NIL queries), the system should cluster them in groups, each referring to a same

Not-In-KB entity. To this objective, a term clustering method is applied to cluster such queries. Each initial NIL query forms a cluster assigning a NIL id. The module compares each new NIL query with each existing cluster (initial NIL query) using a dice coefficient similarity between all ANs (including query name) of both queries. If the similarity is higher than the predefined NIL threshold, the new NIL query is associated to this cluster, otherwise it forms a new NIL cluster. In our experiments we used 0.8 as NIL threshold.

4 Evaluation Framework

We have participated in the framework of the TAC-KBP 2012 and TAC-KBP 2013 mono-lingual EL evaluation tracks[3].

Given a list of queries, each consisting of a name string, a background document, and a pair of character offsets indicating the start and end position of the name string in the document, the system is required to provide the identifier of the KB entry to which the name refers if existing, or a NIL ID if there is no such KB entry. The EL system is required to cluster together queries referring to the same non-KB (NIL) entities and provide a unique ID for each cluster. The reference KB used in this track includes hundreds of thousands of entities based on articles from an October 2008 dump of English WP, which includes 818,741 nodes. The evaluation query sets in 2012 and 2013 experiments contain 2229 and 2190 queries respectively. Entities generally occur in multiple queries using different name variants and/or different background documents. Some entities share confusable names, especially challenging in the case of acronyms.

5 Results and Analysis

Previously, we participated in TAC-KBP 2012 and TAC-KBP 2013 EL evaluation tracks. The system presented in this paper is an improved version of the system with which we participated in the TAC-KBP 2013 track. Using the TAC-KBP 2012 and TAC-KBP 2013 evaluation queries, we present our results splitted into four parts: official results of TAC-KBP 2012 and TAC-KBP 2013 named by '2012' [6] and '2013' [1], and results of the improved system named by '2012*' and '2013*'.

Twenty five teams participated and submitted 98 runs to the TAC-KBP English EL evaluation in 2012, and 26 teams submitted a total of 111 runs to the TAC-KBP in 2013. Tables 3a and 3b illustrate the results obtained by our systems (both base-line and improved systems) over the TAC-KBP 2012 and TAC-KBP 2013 EL evaluation frameworks using B-cubed+ metric (including Precision, Recall, and F1). The tables split the results by those query answers existing in reference KB (in-KB) and those not in the KB (NIL). Evaluation corpus in TAC-KBP 2012 includes two kinds of genres, News Wires (NW) and Web Documents (WB). In TAC-KBP 2013, a new genre, Discussion Fora (DF) was associated to the evaluation corpus. DF is highly challenging since it contains many grammatical irregularities extracted from fora, blogs, etc. The tables also indicate the results by three different query types, PER, ORG, and GPE.

[3] `http://www.nist.gov/tac/`

System	B^3+ F1							
	All (2226)	in-KB (1177)	NIL (1049)	NW (1471)	WB (755)	PER (918)	ORG (706)	GPE (602)
2012	0.421	0.311	0.545	0.460	0.344	0.599	0.382	0.194
2012*	**0.611**	**0.524**	**0.710**	**0.665**	**0.507**	**0.771**	**0.560**	**0.426**
Median	0.536	0.496	0.594	0.574	0.492	0.646	0.486	0.447
Highest	0.730	0.687	0.847	0.782	0.646	0.840	0.717	0.694

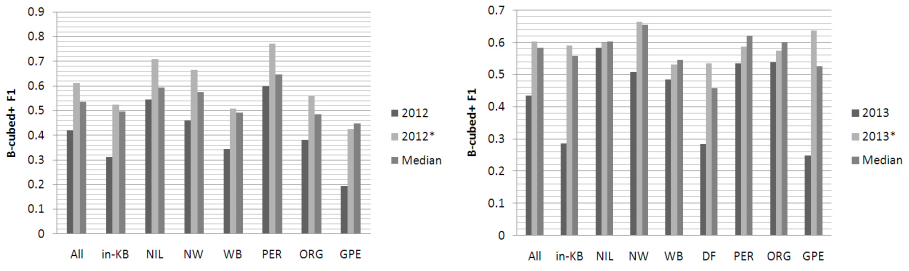
(a)

System	B^3+ F1								
	All (2190)	in-KB (1090)	NIL (1100)	NW (1134)	WB (343)	DF (713)	PER (686)	ORG (701)	GPE (803)
2013	0.435	0.285	0.584	0.508	0.485	0.284	0.535	0.538	0.248
2013*	**0.602**	**0.591**	**0.599**	**0.663**	**0.532**	**0.535**	**0.586**	**0.575**	**0.636**
Median	0.584	0.558	0.603	0.655	0.546	0.458	0.620	0.599	0.526
Highest	0.721	0.724	0.720	0.801	0.673	0.633	0.758	0.737	0.720

(b)

Table 3: The results comparison between the systems over TAC-KBP2012 (Table 3a) and TAC-KBP2013 (Table 3b) mono-lingual EL evaluation framework.

In both experiments, the proposed system achieved a significant improvement compared with our base-line results shown in the mentioned tables. In 2012 experiment, as shown in Table 3a, 2012* achieved improvement in all portions over the median of the results achieved by all the participants, except in the case of GPE query types that we attained a little decrement less than median. In addition, in 2013 experiment, as shown in Table 3b, 2013* also improved the performance compared with the previous results of our participation in TAC-KBP 2013. The system obtained a notable result in the case of GPE and grabbed scores a little less than median for PER and ORG. The reason that the results for the same query type varies in 2012 and 2013 is because the nature of queries are different in these years, e.g. most GPE queries in 2012 are focused on the U.S. states but most GPE queries in 2013 are about countries. The EL task for the evaluation queries in 2013 was increasingly more strict than those presented in 2012, including more ambiguous and partial query names along with a lot of grammatical irregularities. In both experiments, the scores obtained for the NW genre is higher than WB since the NW documents contain more structured text. In addition, in 2013, the scores obtained for DF genre are the lowest compared against NW and WB genres. The reason is that, DF genre includes many typos and grammatical errors. The difference between our overall result and the median in 2012 experiment is higher than its difference in 2013. Since the teams participated in TAC-KBP 2012 and TAC-KBP 2013 are different, thus comparing between the medians in 2012 and 2013 is not possible. In total, thanks to our proposed system we gained the overall scores more than median in both 2012 and 2013 experiments (Figures 5a and 5b).



(a) (b)

Fig. 5: B^3+ F1† comparison between our systems and the median of the results achieved by all the participants in the 2012 (Figure 5a) and 2013 (Figure 5b) experiments.
†: n.b. the B^3+ metric measures Precision, Recall, and F1 of systems focusing on ability of them to cluster queries.

6 Conclusions and Future Work

The improved version of the system presented here ran over the TAC-KBP2012 and TAC-KBP2013 EL evaluation framework. The results were compared with those obtained by all participants. Most participants have achieved much research in the field of EL having many contributions in this regard. Thus, the results are considered a good framework for comparing the performance of the systems. We achieved the comparison using B-cubed+ F1 metric. The measure depicted a significant improvement compared with our previous results and higher than the median of participant results.

As a future work, we carry on to improve the system results through profound analyzing of semantics of the keyphrases in order to increase the accuracy of the task. The deep analysis of the keyphrases is beneficial when the queries are highly ambiguous to realize their correct references.

7 Acknowledgments

This work has been produced with the support of the SKATER project (TIN2012-38584-C06-01).

References

1. Ageno, A., Comas, P.R., Naderi, A., Rodríguez, H., Turmo, J.: The talp participation at tac-kbp 2013. In: the Sixth Text Analysis Conference (TAC 2013), Gaithersburg, MD USA (2014)
2. Banerjee, S., Pedersen, T.: Extended gloss overlaps as a measure of semantic relatedness. IJCAI **3** (2003)
3. Bunke, H., Alberto Sanfeliu, e.: Syntactic and structural pattern recognition: theory and applications. World Scientific **7** (1990)

4. Fellbaum, C.: Wordnet: An electronic lexical database. MIT Press (1998)
5. Fu, K.S.: Syntactic pattern recognition and applications. Prentice-Hall (1982)
6. Gonzàlez, E., Rodríguez, H., Turmo, J., Comas, P.R., Naderi, A., Ageno, A., Sapena, E., Vila, M., Martí, M.A.: The talp participation at tac-kbp 2012. In: the Fifth Text Analysis Conference (TAC 2012), Gaithersburg, MD USA (2013)
7. Hoffart, J., Suchanek, F.M., Berberich, K., Weikum, G.: Yago2: a spatially and temporally enhanced knowledge base from wikipedia. Artificial Intelligence Journal (2013)
8. Hoffart, J., Yosef, M.A., Bordino, I., Furstenau, H., Pinkal, M., Spaniol, M., Taneva, B., Thater, S., Weikum, G.: Robust disambiguation of named entities in text. In: the EMNLP Conference, Scotland. (2011)
9. John, G.H., Langley, P.: Estimating continuous distributions in bayesian classifiers. In: the Eleventh conference on Uncertainty in artificial intelligence, Morgan Kaufmann Publishers Inc. (1995)
10. Klivans, A.R., Servedio., R.A.: Toward attribute efficient learning of decision lists and parities. The Journal of Machine Learning Research **7** (2006) 587–602
11. Lesk, M.: Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In: the 5th annual international conference on Systems documentation, ACM (1986)
12. Lin, D.: Automatic retrieval and clustering of similar words. In: the 17th international conference on Computational Linguistics. Volume 2., Association for Computational Linguistics (1998)
13. Magee, J.F.: Decision trees for decision making. Graduate School of Business Administration, Harvard University (1964)
14. McCulloch, W.S., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics **5**(4) (1943) 115–133
15. Mihalcea, R.: Co-training and self-training for word sense disambiguation. In: the Conference on Natural Language Learning. (2004)
16. Mooney, R.J.: Comparative experiments on disambiguating word senses: An illustration of the role of bias in machine learning. In: Conference on Empirical Methods in Natural Language Processing (EMNLP). (1996) 82–91
17. Navigli, R.: Word sense disambiguation: A survey. ACM Computing Surveys (CSUR) **41**(2) (1990)
18. Pantel, P., Lin, D.: Discovering word senses from text. In: the 8th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM (2002)
19. Quinlan, J.R.: C4. 5: programs for machine learning. Volume 1. Morgan Kaufmann (1993)
20. Ratinov, L., Roth, D.: Design challenges and misconceptions in named entity recognition. In: CoNLL. (2009)
21. Řehůřek, R., Sojka, P.: Software Framework for Topic Modelling with Large Corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, Valletta, Malta, ELRA (May 2010) 45–50 http://is.muni.cz/publication/884893/en.
22. Rivest, R.L.: Learning decision lists. Machine learning **2**(3) (1987) 229–246
23. Schutze, H.: Dimensions of meaning. In: Supercomputing '92: ACM/IEEE Conference on Supercomputing, IEEE Computer Society Press (1992) 787–796
24. Towell, G., Voorhees, E.M.: Disambiguating highly ambiguous words. Computational Linguistics **24**(1) (1998) 125–145
25. Widdows, D., Dorow, B.: A graph model for unsupervised lexical acquisition. In: the 19th international conference on Computational linguistics. Volume 1., Association for Computational Linguistics (2002)