Entity Linking Leveraging Automatically Generated Annotation

Wei Zhang[†] **Jian Su**[‡] **Chew Lim Tan**[†] **Wen Ting Wang**[‡]

[†]School of Computing
National University of Singapore
{z-wei, tancl}
@comp.nus.edu.sg

[‡] Institute for Infocomm Research
{sujian, wwang}
@i2r.a-star.edu.sg

Abstract

Entity linking refers entity mentions in a document to their representations in a knowledge base (KB). In this paper, we propose to use additional information sources from Wikipedia to find more name variations for entity linking task. In addition, as manually creating a training corpus for entity linking is labor-intensive and costly, we present a novel method to automatically generate a large scale corpus annotation for ambiguous mentions leveraging on their unambiguous synonyms in the document collection. Then, a binary classifier is trained to filter out KB entities that are not similar to current mentions. This classifier not only can effectively reduce the ambiguities to the existing entities in KB, but also be very useful to highlight the new entities to KB for the further population. Furthermore, we also leverage on the Wikipedia documents to provide additional information which is not available in our generated corpus through a domain adaption approach which provides further performance improvements. The experiment results show that our proposed method outperforms the state-of-the-art approaches.

1 Introduction

The named entity (NE) ambiguation has raised serious problems in many areas, including web people search, knowledge base population (KBP), and information extraction, because an entity (such as *Abbott Laboratories*, a diversified pharmaceuticals health care company) can be referred to by multiple mentions (e.g. "*ABT*" and "*Abbott*"), and a mention (e.g. "*Abbott*") can be shared by different entities (e.g. *Abbott Texas*: a city in United States; *Bud Abbott*, an American actor; and *Abbott Laboratories*, a diversified pharmaceutical health care company).

Both Web People Search (WePS) task (Artiles et al. 2007) and Global Entity Detection & Recognition task (GEDR) in Automatic Content Extraction 2008 (ACE08) disambiguate entity mentions by clustering documents with these mentions. Each cluster then represents a unique entity. Recently entity linking has been proposed in this field. However, it is quite different from the previous tasks.

Given a knowledge base, a document collection, entity linking task as defined by KBP-09[1] (McNamee and Dang, 2009) is to determine for each name string and the document it appears, which knowledge base entity is being referred to, or if the entity is a new entity which is not present in the reference KB.

Compared with GEDR and WePS, entity linking has a given entity list (i.e. the reference KB) to which we disambiguate the entity mentions. Moreover, in document collection, there are new entities which are not present in KB and can be used for further population. In fact, new entities with or without the names in KB cover more than half of testing instances.

[1] http://apl.jhu.edu/~paulmac/kbp.html

Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), pages 1290–1298,
Beijing, August 2010

Entity linking has been explored by several researchers. Without any training data available, most of the previous work ranks the similarity between ambiguous mention and candidate entities through Vector Space Model (VSM). Since they always choose the entity with the highest rank as the answer, the ranking approaches hardly detect a situation where there may be a new entity that is not present in KB. It is also difficult to combine bag of words (BOW) with other features. For example, to capture the "category" information, the method of Cucerzan (2007) involves a complicated optimization issue and the approach has to be simplified for feasible computation, which compromises the accuracy. Besides unsupervised methods, some supervised approaches (Agirre et al. 2009, Li et al. 2009 and McNamee et al. 2009) also have been proposed recently for entity linking. However, the supervised approaches for this problem require large amount of training instances. But manually creating a corpus is labor-intensive and costly.

In this paper, we explore how to solve the entity linking problem. We present a novel method that can automatically generate a large scale corpus for ambiguous mentions leveraging on their unambiguous synonyms in the document collection. A binary classifier based on Support Vector Machine (SVM) is trained to filter out some candidate entities that are not similar to ambiguous mentions. This classifier can effectively reduce the ambiguities to the existing entities in KB, and it is very useful to highlight the new entities to KB for the further population. We also leverage on the Wikipedia documents to provide additional information which is not available in our generated corpus through a domain adaption approach which provides further performance improvements. Besides, more information sources for finding more variations also contribute to the overall 22.9% accuracy improvements on KBP-09 test data over baseline.

The remainder of the paper is organized as follows. Section 2 reviews related work for entity linking. In Section 3 we detail our algorithm including name variation and entity disambiguation. Section 4 describes the experimental setup and results. Finally, Section 5 concludes the paper.

2 Related Work

The crucial component of entity linking is the disambiguation process. Raphael et al. (2007) report a disambiguation algorithm for geography. The algorithm ranks the candidates based on the manually assigned popularity scores in KB. The class with higher popularity will be assigned higher score. It causes that the rank of entities would never change, such as Lancaster (California) would always have a higher rank than Lancaster (UK) for any mentions. However, as the popularity scores for the classes change over time, it is difficult to accurately assign dynamic popularity scores. Cucerzan (2007) proposes a disambiguation approach based on vector space model for linking ambiguous mention in a document with one entity in Wikipedia. The approach ranks the candidates and chooses the entity with maximum agreement between the contextual information extracted from Wikipedia and the context of a document, as well as the agreement among the category tags associated with the candidate entities. Nguyen and Cao (2008) refer the mentions in a document to KIM (Popov et al. 2004) KB. KIM KB is populated with over 40,000 named entities. They represent a mention and candidates as vectors of their contextual noun phrase and co-occurring NEs, and then the similarity is determined by the common terms of the vectors and their associated weights. For linking mentions in news articles with a Wikipedia-derived KB (KBP-09 data set), Varma et al. (2009) rank the entity candidates using a search engine. Han and Zhao (2009) rank the candidates based on BOW and Wikipedia semantic knowledge similarity.

All the related work above rank the candidates based on the similarity between ambiguous mention and candidate entities. However, the ranking approach hardly detects the new entity which is not present in KB.

Some supervised approaches also have been proposed. Li et al. (2009) and McNamee et al. (2009) train their models on a small manually created data set containing only 1,615 examples. But entity linking requires large training data. Agirre et al. (2009) use Wikipedia to construct their training data by utilizing Inter-Wikipedia links and the surrounding snippets of text. However, their training data is created from a different domain which does not work well in the targeted news article domain.

3 Approach

In this section we describe our two-stage approach for entity linking: name variation and entity disambiguation. The first stage finds variations for every entity in the KB and generates an entity candidate set for a given query. The second stage is entity disambiguation, which links an entity mention with the real world entity it refers to.

3.1 Name Variation

The aim for Name Variation is to build a Knowledge Repository of entities that contains vast amount of world knowledge of entities like name variations, acronyms, confusable names, spelling variations, nick names etc. We use Wikipedia to build our knowledge repository since Wikipedia is the largest encyclopedia in the world and surpasses other knowledge bases in its coverage of concepts and up-to-date content. We obtain useful information from Wikipedia by the tool named Java Wikipedia Library[2] (Zesch et al. 2008), which allows to access all information contained in Wikipedia.

Cucerzan (2007) extracts the name variations of an entity by leveraging four knowledge sources in Wikipedia: "entity pages", "disambiguation pages" "redirect pages" and "anchor text".

Entity page in Wikipedia is uniquely identified by its title – a sequence of words, with the first word always capitalized. The title of Entity Page represents an unambiguous name variation for the entity. A redirect page in Wikipedia is an aid to navigation. When a page in Wikipedia is redirected, it means that those set of pages are referring to the same entity. They often indicate synonym terms, but also can be abbreviations, more scientific or more common terms, frequent misspellings or alternative spellings etc. Disambiguation pages are created only for ambiguous mentions which denote two or more entities in Wikipedia, typically followed by the word "*disambiguation*" and containing a list of references to pages for entities that share the same name. This is more useful in extracting the abbreviations of entities, other possible names for an entity etc. Besides, both outlinks and inlinks in Wikipedia are associated with anchor texts that represent name variations for the entities.

Using these four sources above, we extracted name variations for every entity in KB to form the Knowledge Repository as Cucerzan's (2007) method. For example, the variation set for entity *E0272065* in KB is {*Abbott Laboratories, Abbott Nutrition, Abbott* …}. Finally, we can generate the entity candidate set for a given query using the Knowledge Repository. For example, for the query containing "*Abbott*", the entity candidate set retrieved is {*E0272065, E0064214* …}.

From our observation, for some queries the retrieved candidate set is empty. If the entity for the query is a new entity, not present in KB, empty candidate set is correct. Otherwise, we fail to identify the mention in the query as a variation, commonly because the mention is a misspelling or infrequently used name. So we propose to use two more sources "Did You Mean" and "Wikipedia Search Engine" when Cucerzan (2007) algorithm returns empty candidate set. Our experiment results show that both proposed knowledge sources are effective for entity linking. This contributes to a performance improvement on the final entity linking accuracy.

Did You Mean: The "*did you* mean" feature of Wikipedia can provide one suggestion for misspellings of entities. This feature can help to correct the misspellings. For example, "Abbot Nutrition" can be corrected to "Abbott Nutrition".

Wikipedia Search Engine: This key word based search engine can return a list of relevant entity pages of Wikipedia. This feature is more useful in extracting infrequently used name.

Algorithm 1 below presents the approach to generate the entity candidate set over the created Knowledge Repository. $\boldsymbol{Ref_E(s)}$ is the entity set indexed by mention $\boldsymbol{s}$ retrieved from Knowledge Repository. In Step 8, we use the longest common subsequence algorithm to measure the similarity between strings $\boldsymbol{s}$ and the title of the entity page with highest rank. More details about longest common subsequence algorithm can be found in Cormen et al. (2001).

[2] http://www.ukp.tu-darmstadt.de/software/JWPL

Algorithm 1 Candidate Set Generation

```
Input: mention s;
if $Ref_E(s)$ is empty
    s' ← Wikipedia"did you mean"Suggestion
    If s' is not NULL
        s ← s'
    else
        EntityPageList ← WikipediaSearchEngine(s)
        EntityPage←FirstPage of EntityPageList
        Sim=Similarity(s,EntityPage.title)
        if Sim > Threshold
            s ← EntityPage.title
        end if
    end if
end if
Output: $Ref_E(s)$;
```

3.2 Entity Disambiguation

The disambiguation component is to link the mention in query with the entity it refers to in candidate set. If the entity to which the mention refers is a new entity which is not present in KB, *nil* will be returned. In this Section, we will describe the method for automatic data creation, domain adaptation from Wikipedia data, and our supervised learning approach as well.

3.2.1 Automatic Data Creation

The basic idea is to take a document with an unambiguous reference to an entity E1 and replacing it with a phrase which may refer to E1, E2 or others.

Observation: Some full names for the entities in the world are unambiguous. This phenomenon also appears in the given document collection of entity linking. The mention "*Abbott Laboratories*" appearing at multiple locations in the document collection refers to the same entity "*a pharmaceuticals health care company*" in KB.

From this observation, our method takes into account the mentions in the Knowledge Repository associated with only one entity and we treat these mentions as unambiguous name. Let us take *Abbott Laboratories-{E0272065}* in the Knowledge Repository as an example. We first use an index and search tool to find the documents with unambiguous mentions. Such as, the mention "*Abbott Laboratories" occurs in* document *LDC2009T13* and *LDC2007T07* in the document collection. The chosen text indexing and searching tool is the well-known Apache Lucene information retrieval open-source library[3].

Next, to validate the consistency of NE type between entities in KB and in document, we run the retrieved documents through a Named Entity Recognizer, to tag the named entities in the documents. Then we link the document to the entity in KB if the document contains a named entity whose name exactly matches with the unambiguous mention and type (i.e. Person, Organization and Geo-Political Entity) exactly matches with the type of entity in KB. In this example, after Named Entity Recognition, "Abbott Laboratories" in document *LDC2009T13* is tagged as an Organization which is consistent with the entity type of *E0272065* in KB. We link the *"Abbott Laboratories"* occurring in *LDC2009T13* with entity *E0272065.*

Finally, we replace the mention in the selected documents with the ambiguous synonyms. For example, we replace the mention *"Abbott Laboratories"* in document *LDC2009T13* with *"Abbott"* where *Abbott-{E0064214, E0272065...}* is an entry in Knowledge Repository. *"Abbott"* is ambiguous, because it is referring not only to *E0272065*, but also to *E0064214* in Knowledge Repository. Then, we can get two instances for the created data set as Figure 1, where one is positive and the other is negative.

(Abbott, LDC2009T13) E0272065 +

(Abbott, LDC2009T13) E0064214 -

…

+ refer to - not refer to

Figure 1: An instance of the data set

However, from our studies, we realize some limitations on our training data. For example, as shown in Figure 1, the negative instance for *E0272065* and the positive instance for

[3] http://lucene.apache.org

E0064214 are not in our created data set. However, those instances exist in the current document collection. We do not retrieve them since there is no unambiguous mention for *E0064214* in the document collection.

To reduce the effect of this problem, we propose to use the Wikipedia data as well, since Wikipedia data has training examples for all the entities in KB. Articles in Wikipedia often contain mentions of entities that already have a corresponding article, and at least the first occurrence of the mentions of an entity in a Wikipedia article must be linked to its corresponding Wikipedia article, if such an article exists. Therefore, if the mention is ambiguous, the hyperlink is disambiguating it. Next, we will describe how to incorporate Wikipedia data.

Incorporating Wikipedia Data. The document collection for entity linking is commonly from other domains, but not Wikipedia. To benefit from Wikipedia data, we introduce a domain adaption approach (Daumé III, 2007) which is suitable for this work since we have enough "target" domain data. The approach is to augment the feature vectors of the instances. Denote by X the input space, and by Y the output space, in this case, X is the space of the real vectors $\varphi(\mathrm{x})$ for the instances in data set and $Y= \{+1,-1\}$ is the label. D^s is the Wikipedia domain dataset and D^t is our automatically created data set. Suppose for simplicity that $X=R^F$ for some $F > 0$ (R^F is the space of F-dimensions). The augmented input space will be defined by $\breve{X} = R^{3F}$. Then, define mappings $\varphi^s, \varphi^t : X \rightarrow \breve{X}$ for mapping the Wikipedia and our created data set respectively. These are defined as follows:

$$\varphi^{s}(\mathrm{x}) =< \varphi(\mathrm{x}), \varphi(\mathrm{x}), 0 >$$

$$\varphi^{t}(\mathrm{x}) =< \varphi(\mathrm{x}), 0, \varphi(\mathrm{x}) >$$

Where $\boldsymbol{0}=<0,0,...,0> \in R^F$ is the zero vector. We use the simple linear kernel in our experiments. However, the following kernelized version can help us to gain some insight into the method. K denotes the dot product of two vectors. $K(x,x')=< \varphi(x), \varphi(x')>$. When the domain is the same, we get: $\breve{K}(x,x') =< \varphi(x), \varphi(x') > +< \varphi(x), \varphi(x') = 2K(x,x')$. When they are from different domains, we get: $\breve{K}(x,x') =< \varphi(x), \varphi(x') >= K(x,x')$. Putting this together, we have:

$$\breve{K} = \begin{cases} 2K(x,x') & \text{same domain} \\ K(x,x') & \text{diff. domain} \end{cases}$$

This is an intuitively pleasing result. Loosely speaking, this means that data points from our created data set have twice as much influence as Wikipedia points when making predictions about test data from document collection.

3.2.2 The Disambiguation Framework

To disambiguate a mention in document collection, the ranking method is to rank the entities in candidate set based on the similarity score. In our work, we transform the ranking problem into a classification problem: deciding whether a mention refers to an entity on an SVM classifier. If there are 2 or more than 2 candidate entities that are assigned *positive* label by the binary classifier, we will use the baseline system (explained in Section 4.2) to rank the candidates and the entity with the highest rank will be chosen.

In the learning framework, the training or testing instance is formed by (*query*, *entity*) pair. For Wikipedia data, (*query*, *entity*) is *positive* if there is a hyperlink from the article containing the mention in query to the entity, otherwise (*query*, *entity*) is *negative*. Our automatically created data has been assigned labels in Section 3.2.1. Based on the training instances, a binary classifier is generated by using particular learning algorithm. During disambiguation, (*query*, *entity*) is presented to the classifier which then returns a class label.

Each (*query*, *entity*) pair is represented by the feature vector using different features and similarity metrics. We chose the following three classes of features as they represent a wide range of information - lexical features, word-category pair, NE type - that have been proved to be effective in previous works and tasks. We now discuss the three categories of features used in our framework in details.

Lexical features. For Bag of Words feature in Web People Search, Artiles et al. (2009) illustrated that noun phrase and n-grams longer than 2 were not effective in comparison with token-based features and using bi-grams gives the best results only reaching recall 0.7. Thus, we use token-based features. The similarity metric we choose is cosine (using standard tf.idf weighting). Furthermore, we also take into account the co-occurring NEs and represent it in the form of token-based features. Then, the single cosine similarity feature is based on Co-occurring NEs and Bag of Words.

Word Category Pair. Bunescu (2007) demonstrated that word-category pairs extracted from the document and Wikipedia article are a good signal for disambiguation. Thus we also consider word-category pairs as a feature class, i.e., all *(w,c)* where *w* is a word from Bag of Words of document and *c* is a category to which candidate entity belongs.

NE Type. This feature is a single binary feature to guarantee that the type of entity in document (i.e. Person, Geo-Political Entity and Organization) is consistent with the type of entity in KB.

4 Experiments and Discussions

4.1 Experimental Setup

In our study**,** we use KBP-09 knowledge base and document collection for entity linking. In the current setting of KBP-09 Data, the KB has been generated automatically from Wikipedia. The KB contains 818,741 different entities. The document collection is mainly composed of newswire text from different press agencies. The collection contains 1.3 million documents that span from 1994 to the end of 2008. The test data has 3904 queries across three named entity types: Person, Geo-Political Entity and Organization. Each query contains a document with an ambiguous mention.

Wikipedia data can be obtained easily from the website[4] for free research use. It is available in the form of database dumps that are released periodically. In order to leverage various information mentioned in Section 3.1 to derive name variations, make use of the links in Wikipedia to generate our training corpus and get word category information for the disambiguation, we further get Wikipedia data directly from the website. The version we used in our experiments was released on Sep. 02, 2009. The automatically created corpus (around 10K) was used as the training data, and 30K training instances associated with the entities in our corpus was derived from Wikipedia.

For pre-processing, we perform sentence boundary detection and Chunking derived from Stanford parser (Klein and Manning, 2003), Named Entity Recognition using a SVM based system trained and tested on ACE 2005 with 92.5(P) 84.3(R) 88.2(F), and coreference resolution using a SVM based coreference resolver trained and tested on ACE 2005 with 79.5%(P), 66.7%(R) and 72.5%(F).

We select SVM as the classifier used in this paper since SVM can represent the stat-of-the-art machine learning algorithm. In our implementation, we use the binary SVM^{Light} developed by Joachims (1999). The classifier is trained with default learning parameters.

We adopt the measure used in KBP-09 to evaluate the performance of entity linking. This measure is micro-averaged accuracy: the number of correct link divided by the total number of queries.

4.2 Baseline Systems

We build the baseline using the ranking approach which ranks the candidates based on similarity between mention and candidate entities. The entity with the highest rank is chosen. Bag of words and co-occurring NEs are represented in the form of token-based feature vectors. Then tf.idf is employed to calculate similarity between feature vectors.

To make the baseline system with token-based features state-of-the-art, we conduct a series of experiments. Table 1 lists the performances of our token-based ranking systems. In our experiment, local tokens are text segments generated by a text window centered on the mention. We set the window size to 55, which is the value that was observed to give optimum performance for the disambiguation problem (Gooi and Allan, 2004). Full tokens and NE are all the tokens and named entities co-occurring in the text respectively. We notice that tokens of the full text as well as the co-occurring named entity produce the best baseline performance, which we use for the further experiment.

[4] http://download.wikipedia.org

	Micro-averaged Accuracy
local tokens	60.0
local tokens + NE	60.6
full tokens + NE	61.9

Table 1: Results of the ranking methods

4.3 Experiment and Result

As discussed in Section 3.1, we exploit two more knowledge sources in Wikipedia: "did you mean" (DYM) and "Wikipedia search engine" (SE) for name variation step. We conduct some experiments to compare our name variation method using Algorithm 1 in Section 3.1 with the name variation method of Cucerzan (2007). Table 2 shows the comparison results of different name variation methods for entity linking. The experiments results show that, in entity linking task, our name variation method outperforms the method of Cucerzan (2007) for both entity disambiguation methods.

Name Variation Approaches	Ranking Method	Our Disambiguation Method
Cucerzan (2007)	60.9	82.2
+DYM+SE	61.9	**83.8**

Table 2: Entity Linking Result for two name variation approaches. Column 1 used the baseline method for entity disambiguation step. Column 2 used our proposed entity disambiguation method.

Table 3 compares the performance of different methods for entity linking on the KBP-09 test data. Row 1 is the result for baseline system. Row 2 and Row 3 show the results training on Wikipedia data and our automatically data respectively. Row 4 is the result training on both Wikipedia and our created data using the domain adaptation method mentioned in Section 3.2.1. It shows that our method trained on the automatically generated data alone significantly outperforms baseline. Compared Row 3 with Row 2, our created data set serves better at training the classifier than Wikipedia data. This is due to the reason that Wikipedia is a different domain from newswire domain. By comparing Row 4 with Row 3, we find that by using the domain adaptation method in Section 3.2.1, our method for entity linking can be further improved by 1.5%. Likely, this is because of the limitation of the auto-generated corpus as discussed in Section 3.2.1. In another hand, Wikipedia can complement the missing information with the auto-generated corpus. So combining Wikipedia data with our generated data can achieve better result. Compared with baseline system using Cucerzan (2007) name variation method in Table 2, in total our proposed method achieves a significant 22.9% improvement.

	Micro-averaged Accuracy
Baseline	61.9
Wiki	79.9
Created Data	82.3
Wiki → Created Data	**83.8**

Table 3: Micro-averaged Accuracy for Entity Linking

To test the effectiveness of our method to deal with new entities not present in KB and existing entities in KB respectively, we conduct some experiments to compare with Baseline. Table 4 shows the performances of entity linking systems for existing entities (non-NIL) in KB and new entity (NIL) which is not present in KB. We can see that the binary classifier not only effectively reduces the ambiguities to the existing entities in KB, but also is very useful to highlight the new entities to KB for the further population. Note that, in baseline system, all the new entities are found by the empty candidate set of name variation process, while the disambiguation component has no contribution. However, our approach finds the new entities not only by the empty candidate set, but also leveraging on disambiguation component which also contributes to the performance improvement.

	non-NIL	NIL
Baseline	72.6	52.4
Wiki → Created Data	79.2	87.8

Table 4: Entity Linking on Existing and New Entities

Finally, we also compare our method with the top 5 systems in KBP-09. Among them, *Siel_093* (Varma et al. 2009) and *NLPR_KBP1* (Han and Zhao 2009) use similarity ranking approach; *Stanford_UBC2* (Agirre et al. 2009), *QUANTA1* (Li et al. 2009) and *hltcoe1* (McNamee et al. 2009) use supervised approach. From the results shown in Figure 2, we observe that our method outperforms all the top 5 systems and the baseline system of KBP-09. Specifically, our method achieves better result than both similarity ranking approaches. This is due to the limitations of the ranking approach which have been discussed in Section 2. We also observe that our method gets a 5% improvement over *Stanford_UBC2*. This is because they collect their training data from Wikipedia which is a different domain from document collection of entity linking, news articles in this case; while our automatic data generation method can create a data set from the same domain as the document collection. Our system also outperforms *QUANTA1* and *hltcoe1* because they train their model on a small manually created data set (1,615 examples), while our method can automatically generate a much larger data set.

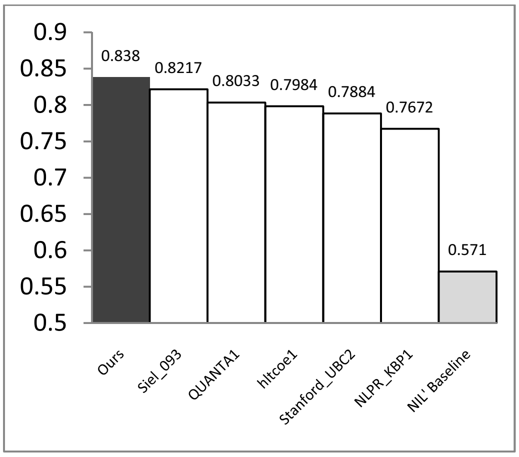

Figure 2: A comparison with KBP09 systems

5 Conclusion

The purpose of this paper is to explore how to leverage the automatically generated large scale annotation for entity linking. Traditionally, without any training data available, the solution is to rank the candidates based on similarity. However, it is difficult for the ranking approach to detect a new entity that is not present in KB, and it is also difficult to combine different features. In this paper, we create a large corpus for entity linking by an automatic method. A binary classifier is then trained to filter out KB entities that are not similar to current mentions. We further leverage on the Wikipedia documents to provide other information which is not available in our generated corpus through a domain adaption approach. Furthermore, new information sources for finding more variations also contribute to the overall 22.9% accuracy improvements on KBP-09 test data over baseline.

References

E. Agirre et al. Stanford-UBC at TAC-KBP. In *Proceedings of Test Analysis Conference 2009 (TAC 09).*

J. Artiles, J. Gonzalo, and S. Sekine. 2007. The semeval-2007 web evaluation: Establishing a benchmark for the web people search task. In *Proceeding of the Fourth International Work-shop on Semantic Evaluations (SemEval-2007).*

J. Artiles, E. Amigo and J. Gonzalo. 2009. The role of named entities in Web People Search. In *proceeding of the 47th Annual Meeting of the Association for Computational Linguistics.*

R. Bunescu. 2007. Learning for information extraction from named entity recognition and disambiguation to relation extraction. Ph.D thesis, University of Texas at Austin, 2007.

T. H. Cormen, et al. 2001. Introduction To Algorithms (Second Edition). *The MIT Press*, Page 350-355.

S. Cucerzan. 2007. Large-Scale Named Entity Disambiguation Based on Wikipedia Data. *Empirical Methods in Natural Language Processing*, June 28-30, 2007.

H. Daumé III. 2007. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics .*

C. H. Gooi and J. Allan. 2004. Cross-document coreference on a large scale corpus. In *proceedings of Human Language Technology Conference North American Association for Computational Linguistics Annual Meeting,* Boston, MA.

X. Han and J. Zhao. NLPR_KBP in TAC 2009 KBP Track: A Two-Stage Method to Entity Linking. In *Proceedings of Test Analysis Conference 2009 (TAC 09).*

T. Joachims. 1999. Making large-scale SVM learning practical. In *Advances in Kernel Methods - Support Vector Learning*. MIT Press.

D. Klein and C. D. Manning. 2003. Fast Exact Inference with a Factored Model for Natural Language Parsing*. In Advances in Neural Information Processing Systems 15 (NIPS 2002), Cambridge, MA: MIT Press, pp. 3-10.*

F. LI et al. THU QUANTA at TAC 2009 KBP and RTE Track. In *Proceedings of Test Analysis Conference 2009 (TAC 09).*

P. McNamee and H. T. Dang. 2009. Overview of the TAC 2009 Knowledge Base Population Track. In *Proceedings of Test Analysis Conference 2009 (TAC 09).*

P. McNamee et al. HLTCOE Approaches to Knowledge Base Population at TAC 2009. In *Proceedings of Test Analysis Conference 2009 (TAC 09).*

H. T. Nguyen and T. H. Cao. 2008. Named Entity Disambiguation on an Ontology Enriched by Wikipedia. *2008 IEEE International Conference on Research, Innovation and Vision for the Future in Computing & Communication Technologies.*

B. Popov et al. 2004. KIM - a Semantic Platform for Information Extraction and Retrieval. In *Journal of Natural Language Engineering*, Vol. 10, Issue 3-4, Sep 2004, pp. 375-392, Cambridge University Press.

V. Raphael, K. Joachim and M. Wolfgang, 2007. Towards ontology-based disambiguation of geographical identifiers. In *Proceeding of the 16th WWW workshop on I3: Identity, Identifiers, Identifications, 2007.*

V. Varma et al. 2009. IIIT Hyderabad at TAC 2009. In *Proceedings of Test Analysis Conference 2009 (TAC 09).*

T. Zesch, C. Muller and I. Gurevych. 2008. Extractiong Lexical Semantic Knowledge from Wikipedia and Wiktionary. In *Proceedings of the Conference on Language Resources and Evaluation (LREC)*, 2008.