I2R-NUS-MSRA at TAC 2011: Entity Linking

Wei Zhang[1], **Jian Su**[2], **Bin Chen**[2], **Wenting Wang**[2],
Zhiqiang Toh[2], **Yanchuan Sim**[2], **Yunbo Cao**[3], **Chin Yew Lin**[3] **and Chew Lim Tan**[1]
[1]School of Computing, National University of Singapore
[2]Institute for Infocomm Research
[3]Microsoft Research Asia
[1]{z-wei,tancl}@comp.nus.edu.sg
[2]{sujian,bchen,wwang,ztoh,ycsim}@i2r.a-star.edu.sg
[3]{yunbo.cao,cyl}@microsoft.com

Abstract

In this paper, we report the joint participation of I2R-NUS team and MSRA team in entity linking task for Knowledge Base Population at Text Analysis Conference 2011. I2R-NUS team submitted two results with the full system and the partial system for diagnosis purpose. Both results incorporate the new technologies: acronym expansion, instance selection and topic modeling proposed in our recent papers. In clustering step, three clustering algorithms: spectral graph partitioning (SGP), hierarchical agglomerative clustering (HAC) and latent Dirichlet allocation (LDA) are combined for the full system. The full system achieves a competitive F-score 0.831[1]. The partial system uses only Wikipedia Source to generate candidates for KB linking and only LDA for clustering , which leads to 0.813 F-score. Although due to the time constrain, the combined result of I2R-NUS full system with MSRA KB linking result was not submitted, it shows 0.828 F-score afterwards.

1 Introduction

The aim of Knowledge Base Population (KBP) track at Text Analysis Conference (TAC) 2011 is to automatically discover information about named entities and to expand a Knowledge Base (KB) with these information. It requires linking the entities mentioned in the documents with the corresponding KB entries and extracting related information about these entities from the documents. Thus, the KBP task has been divided into two sub tasks: entity linking and slot filling. We participate in the first sub task.

Entity linking sub task is a follow-up to the KBP entity linking evaluation at TAC 2010. The entity linking task 2010 requires either linking entity mentions in the documents with entries in the KB or highlighting these mention as non-KB (*NIL*) entries. In order to create new KB entry, the entity linking system 2011 is further required to group *NIL* mentions referring the same entities together.

In this paper, we describe the joint participation of I2R-NUS team and MSRA team in Entity Linking task for Knowledge Base Population at TAC 2011. I2R-NUS team submitted two results with the full system and the partial system. The full system achieves a competitive F-score 0.831[1]. The partial system results 0.813 F-score. Although due to the time constrain, the combined result of I2R-NUS full system with MSRA KB linking result was not submitted, it shows 0.828 F-score afterwards. I2R-NUS approach will be elaborated in this paper. We'll also describe the combination with MSRA KB linking results and the corresponding performance. MSRA's approach for entity linking and the whole system performance can be found in another paper lead by MSRA.

In I2R-NUS approach for both the full system and the partial system, entity linking is done through three step: 1) Expanding query to reduce the ambiguities of the mention 2) linking the entities to KB entries or *NIL* and 3) clustering *NIL* queries. In KB linking step, the new technologies for entity linking: acronym expansion, instance selection and topic modeling proposed in our recent papers

[1]Highest: 0.846, Median: 0.716 among 44 submitted runs from 21 participating teams

(Zhang et al., 2011a; Zhang et al., 2011b) are incorporated into our system. In clustering step, we analyze three clustering algorithms: spectral graph partitioning (SGP), hierarchical agglomerative clustering (HAC) and latent Dirichlet allocation (LDA) and develop a supervised learner for combining these approaches. According to our experiment, LDA performs best for clustering *NIL* queries. The full system combining the above three clustering algorithms achieves better F-score than the individual algorithms including the partial system submission which only uses LDA for clustering. For the combination of the I2R-NUS full system and MSRA system after the submissions, as MSRA system doesn't cluster *NIL* queries, a binary classifier is trained to combine the two system results at the KB linking step [2].

The remainder of the paper is organized as follows. Section 2 describe the query expansion process. In Section 3, we elaborate our algorithm for KB linking step in detail. Section 4 describes the three clustering approaches and the combined systems. The experimental results are discussed in Section 5. Finally, Section 6 concludes this paper.

2 Query Expansion

Expanding the query from its context can effectively reduce the ambiguities of the mention, under the assumption that two name variants in the same document refer to the same entity. For example, *Roth* in Wikipedia refers to seventy-six entries, but its expansion *John D. Roth* only refers to two entries. *TSE* in Wikipedia refers to thirty-three entries, but with its full name *Tokyo Stock Exchange*, which is unambiguous, we can directly link it with the correct entry without disambiguation. Thus query expansion is performed as the first step for entity linking.

For a given query, the system expand it using the following approach:

• For the capitalized query, we use the acronym expansion approach described in our paper (Zhang et al., 2011a) to expand it from its context. First, we find all possible candidate expansions by text markers such as *"long(short)", "short(long)"* and first letter matching (e.g. for the acronym *"ACM"*, all the word sequences beginning with *"A"* such as *"Association for Computing Machinery has granted the ..."* are considered.). Then, we rely on a SVM classifier for selecting the correct candidate expansion, where the candidate with highest confidence score is selected.

• If query is wholly contained in a string of named entity in the associated document, this named entity is selected as the expansion. In the case *"Roth"*, *"John D. Roth"* is retrieved.

In our system, the expansion instead of the original query would be used in the following processing.

3 KB Linking

To link the mentions with the entries in KB or *NIL* for non-KB entries, we perform the following three steps:

3.1 Name Variation Resolution

Name variation resolution finds variants for each entry in KB. In our approach, we extract the name variants by leveraging on the knowledge sources in Wikipedia: "titles of entity pages", "disambiguation pages " "redirect pages" and "anchor texts".

3.2 Candidate Generation

Candidate Generation finds all the possible KB candidates for the given query using the following approach:

• **Wikipedia Source**: The query matches a name variant of the KB entry obtained in Section 3.1. Then, this KB entry is selected as a candidate.

• **String Match**: The query are wholly contained in the title of the KB entry (e.g. *Cambridge* and *Cambridge, Massachusetts*) or they exactly match.

3.3 Candidate Ranking

First, using a learning to rank method, we rank all the retrieved KB candidates to identify the most likely candidate. In this learning to rank method, each name mention and the associated candidates are formed by a list of feature vectors. During linking, the score for each candidate entry is given by the ranker. The learning algorithm we used is ranking SVM (Herbrich et al., 2000). Next, the preferred KB candidate is presented to a binary classifier (Vapnik, 1995) to determine if it is believed as the target entry for a name mention. From here, we

[2] MSRA team submitted a result using I2R-NUS LDA clustering

Name	Description
Surface	
Surface Match	True if the query matches the title of the candidate
Substring Match 1	True if the title of the candidate begins with the the query (e.g. *"Cambridge, Massachusetts"* and *"Cambridge"*)
Substring Match 2	True if the title of the candidate ends with the the query (e.g. *"Venice-Simplon Orient Express"* and *"Orient Express"*)
Word Match	Number of the same words between the title of the candidate and the query
Word Miss	Number of the different words between the title of the candidate and the query
Edit Distance	Levenshtein distance between query and the title of the candidate
Source	
Wikipedia Source	True for each Wikipedia source (i.e. "entity pages", "disambiguation pages", "redirect pages" and "anchor texts") which generates the candidate (Section 3.2)
String Match	For the candidate not generated from Wikipedia source, true if it is generated from full match. otherwise, false. (Section 3.2)
Contextual	
Bag of Words	The cosine similarity (tf.idf weighting) between the document and text of the candidate.
Similarity Rank	The inverted cosine similarity rank of the candidate in the candidate set.
Co-occurring NEs	Number of the same named entities appearing in the document and the text of the candidate.
Semantic	
NE type	True if NE type (i.e. Person, GPE, Organization) of the query and the candidate is consistent.
Topic Similarity	Similarity between the document and the text of candidate in a topical space (Zhang et al., 2011b)

Table 1: Feature Set for Classifier and Ranker.

can decide whether the mention and top candidate are linked. If not, the mention has no corresponding entry in KB (*NIL*).

To prepare the training data for the ranker and classifier, an instance selection strategy (Zhang et al., 2011b) is used to select a subset for effective disambiguation from a large-scale data set auto-generated from the paper (Zhang et al. , 2010). The instance selection is an iterative process of selecting a representative, informative and diverse batch of instances at each iteration. The batch sizes at each iteration change according to the variance of classifier's confidence or accuracy between batches in sequence.

The features adopted for both learning to rank and classification include 13 feature groups divided to 4 categories. A summary of the features is listed in Table 1. The features of *bag of words* and *co-occurring named entities* to model the context suffer from sparseness issue. Thus, the new feature *topic similarity* proposed by our paper (Zhang et al., 2011b) is incorporated to our system. We model the contexts as the probability distributions over Wikipedia categories, which allows the context similarity being measured in a semantic space instead of just being a comparison of the literal terms. In our approach, Wikipedia serves as a document collection with multiple topical labels, where we learn the posterior distribution over words for each topical label (i.e. Wikipedia category). Then, from the observed words in the document and KB text, we can estimate the distribution of the contexts over the Wikipedia categories.

4 Clustering

In this section, the queries tagged as *NIL* in KB linking step are clustered based on the identity of the entity, such that queries referring the same entity are converged into the same cluster.

From our observation, lots of *NIL* query pairs can be separated from each other by NE type comparison (e.g. the person "Canton" and the GPE "Canton") or their names (e.g. "John Smith" and "George Bush"). Thus, we perform query set partitioning using the following approach:

(1) The queries are divided to three subsets - *Person, GPE and Organization* according to the NE types of the queries.

(2) The queries in each subset are further clustered based on their names.

• Queries with the same name are clustered together.

• The name of a query is wholly contained in or contain the name of the other query. Then, the two queries go to the same cluster.

• The query pair has a strong string similarity score with each other. In our system, Levenshtein distance is used to measure the string similarity.

• The first letters of each word in a query match another query (e.g., **A***ssociation for* **C***omputing* **M***achinery* and ***ACM***).

Using the above strategies, the *NIL* queries are divided to some high recall subsets. Then, we rely on the clustering algorithms to further cluster the queries based on the entities for each subset. We explore the following three algorithms for this purpose.

4.1 Spectral Graph Partitioning

First, we present the query-pair to the classifier mentioned in Section 3.3. The confidence score given by the classifier can be used as the similarity between this query-pair. Then, we further use spectral graph partitioning as described in (Ng et al., 2002) to form the globally optimized entity clusters. Spectral graph partitioning (a.k.a. Spectral clustering) has made its success in a number of fields such as image segmentation in (Shi and Malik, 2000) and gene expression clustering in (Shamir and Sharan, 2002).

Given a set of data points *A*, the similarity matrix may be defined as a matrix *S* where $S_{i,j}$ represents a measure of the similarity between points *i,j* in *A*. Spectral clustering makes use of the spectrum of the similarity matrix of the data to perform dimensionality reduction for clustering in fewer dimensions. Spectral clustering uses information obtained from the eigenvalues and eigenvectors of the Laplacians or the similarity matrices. Since spectral clustering is a matured technique widely used in 1990s, we will not include the theory behind spectral clustering. Instead, readers can refer to (Luxburg, 2007) for detailed explanation.

Compared to the traditional clustering algorithms such as k-means or minimum-cut, spectral clustering has many fundamental advantages. Results obtained by spectral clustering often outperform the traditional approaches, spectral clustering is very simple to implement and can be solved efficiently by standard linear algebra methods. More attractively, according to (Luxburg, 2007) , spectral clustering does not intrinsically suffer from local optima problem.

The actual spectral partitioning is carried out with three special settings. First, queries which are very close to each other are grouped together to form a number of small clusters before the final clustering step on all mentions. This is to provide a number of high confidence cluster seeds for final clustering. Second, after computed the Eigen-decomposed position vector for each query, the vector is normalized to a unity vector in the normalized vector space. This is to remove the differences in the length and distribution density of vectors. Last but not least, all the mentions are sorted according to their string length from longest to shortest. This is to cluster the queries according to their meaning representativeness by assuming the longer the query is, the more specific it is.

4.2 Hierarchical Agglomerative Clustering

For clustering, Wikipedia concepts are also extracted as features of a document, together with other conventional features such as bag-of-words and named entities, since Wikipedia can help find alternative names for an entity and disambiguate a number of entities. The document is converted into a feature vector based on these three types of features extracted from the text.

The Query Relevance Weighting Model (Long and Shi, 2010) is used to estimate the weight of each feature in the feature vector by its closest sentence distance to the query name appeared in the text, i.e. words or concepts that appear close to the query name in the text are more relevant than distant ones. We use the original query name and its Wikipedia redirected names to find exact string match in the document. In addition, we also include coreference chain information of the query name. For example, if one sentence does not contain the query name *"Michael Jordan"* or its alternative names but only talks about "*his second MVP award*" then it should still be considered as an appearance of the query name, words or concepts close to this sentence are still be considered as relevant. Then each feature in a vector is measured by its TFIDF score and accumulated weights under the distance weighting method.

After that the similarity score between two different documents containing the same query name is calculated through their feature vectors based on two similarity measures: cosine similarity and overlap similarity.

Finally, documents referred to the same entity are clustered using Hierarchical Agglomerative Clustering algorithm according to the pair-wise similarity score calculated in the previous step

4.3 Latent Dirichlet Allocation

Recently, topic modeling methods have found widespread applications in various NLP tasks such as summarization, selectional preferences and cross-document co-reference resolution.

As a popular topic model, Latent Dirichlet Allocation (LDA) (Blei et al., 2003) is a three-level hierarchical Bayesian model and used to represent hidden topics (where a topic is a probability distribution over words) underlying the documents. In this section, we cluster the *NIL* queries using LDA, where the learned topics represent the underlying entities of the ambiguous name.

Given a set of documents *D*, our task is to group the documents into *K* sets such that each subset corresponds to one entity. Our generative story is as follows:

for each entity e_k (k=1,2...,K) **do**
  Generate β_{e_k} according to *Dir*(η)
end for
for each document *i* in the set *D* **do**
  Choose $\theta_i \sim$*Dir*(α)
  for each word $w_{i,j}$ (j=1,2,...,N_i) **do**
    Choose an entity $z_{i,j} \sim$ *Multinomial*(θ_i)
    Choose a word $w_{i,j} \sim$ *Multinomial*($\beta_{z_{i,j}}$)
  end for
end for

The number of entities *K* is automatically choose based on the model that leads to the highest posterior probability (Griffiths and Steyvers, 2004).

4.4 System Combination

After developing the three clustering algorithms, we combine them into the I2R-NUS full system using a supervised learning method. The query pair is presented to a three-classes SVM classifier to decide which clustering system should be trusted. The features used in this combination are the scores given by the three systems for the query pair.

As MSRA system doesn't cluster NIL queries, we thus combine the KB linking result in I2R-NUS full system with MSRA's KB linking system. A binary SVM classifier is trained to select the trusted system for each query, where the features are the scores given by the two systems.

5 Experiments and Discussions

5.1 Experiment Setup

The training data of KBP 2011 for entity linking has 3,904 queries in Eval 09 set and 2,250 queries in Eval 10 set, across three named entity types: Person, Geo-Political Entity and Organization. However, according to our experiments, the *NIL* queries in the two data sets are not so ambiguous. After query set partitioning described in the beginning of Section 4, on average each subset has 10.7 queries and 1.34 entities in Eval 09 set, and 3.05 queries and 1.10 entities in Eval 10 set. Thus, we annotate additional data to increase the ambiguities of Eval 10 data set, such that the new Eval 10^+ data set has 2.4 entities for each ambiguous name on average.

The scoring metric used in KBP 2011 to evaluate entity linking system is $B\text{-}Cubed^+$.

5.2 Clustering Algorithms Comparison

Table 2 compares the performances of different clustering algorithms on the *NIL* queries of the three data sets. We can see that LDA always achieves best performance under the three different data sets. By combining the three clustering systems, we can further improves entity linking F1 score to 0.852.

Algorithms	Eval 09	Eval 10	Eval 10^+
SGP	0.745	0.954	0.809
HAC	0.666	0.950	0.789
LDA	0.782	0.981	0.841
Combination	0.795	0.982	0.852

Table 2: Result for Three Clustering Algorithms

5.3 Submissions and Results

As we have seen above, our system is sophisticated and experiments are conducted in many settings. While some configuration choices can be explained theoretically, the optimal values of most parameters are to be determined empirically on the three training data sets. I2R-NUS submitted two results to KBP 2011, one with the full system combining 3 clustering approaches, one with the partial system using only Wikipedia Source to generate candidates for KB linking and only LDA for clustering. The results can be found in Table 3, which also includes the combined result of I2R-NUS full system with MSRA KB linking output after the submissions.

System	Acc.	Precision	Recall	F1
Full	0.863	0.815	0.849	**0.831**
Partial	0.844	0.797	0.829	0.813
+MSRA	0.858	0.812	0.846	0.828
Highest	-	-	-	0.846
Median	-	-	-	0.716

Table 3: Entity Linking Submission Scores

6 Conclusion

The KBP track at the TAC 2011 marks the third year of this information extraction evaluation. This year, English entity linking task is further required to cluster together *NIL* queries based on the identity of the entity. I2R-NUS team explores three clustering algorithms and incorporates acronym expansion, topic model and instance selection into the systems. The full system achieves a 0.831 F-score. The combination with MSRA system also show 0.828 F-score after the submissions.

Acknowledgments

This work is partially supported by Microsoft Research Asia eHealth Theme Program.

References

D. Blei, A. Y. Ng and M. I. Jordan. 2003. *Latent Dirichlet Allocation*. Journal of Machine Learning Research. 2003

T. L. Griffiths and M. Steyvers 2004. *Finding scientific topics*. Proceedings of the National Academy of Sciences of the United States of America, 2004.

R. Herbrich, T. Graepel and K. Obermayer. 2000. *Large Margin Rank Boundaries for Ordinal Regression*. Advances in Large Margin Classifiers. 115-132. 2000

C. Long and L. Shi 2010. *Web Person Name Disambiguation by Relevance Weighting of Extended Feature Sets*. Web People Search 3 Workshop. 2010

U. Luxburg. 2007. *A tutorial on spectral clustering*. Statistics and Computing, Volume 17 Issue 4, December 2007.

A. Ng, M. Jordan and Y. Weiss 2002. *On spectral clustering: analysis and an algorithm.*. In Advances in Neural Information Processing Systems 14 (pp. 849 856). MIT Press. 2002

R. Shamir and R. Sharan 2002. *Algorithmic approaches to clustering gene expression data*. Current Topics in Computational Molecular Biology, 2002.

J. Shi and J. Malik 2000. *Normalized cuts and image segmentation*. IEEE Transactions on Pattern Analysis and Machine Intelligence(PAMI).

V. Vapnik. 1995. *The Nature of Statistical Leaning Theory*. Springer-Verlag, New York. 1995

W. Zhang, J. Su, C. L. Tan and W. T. Wang. 2010. *Entity Linking Leveraging Automatically Generated Annotation*. 23rd International Conference on Computational Linguistics, August 23-27, 2010, Bejing, China

W. Zhang, Y. C. Sim, J. Su and C. L. Tan. 2011a. *Entity Linking with Effective Acronym Expansion, Instance Selection and Topic Modeling*. International Joint Conferences on Artificial Intelligence 2011. Jul 16-22, 2011. Barcelona, Spain.

W. Zhang, J. Su and C. L. Tan. 2011b *A Wikipedia-LDA Model for Entity Linking with Batch Size Changing Instance Selection* . International Joint Conference for Natural Language Processing, Chiang Mai, Thailand, November 8-13, 2011.