Improvement of Graph based Named Entity Disambiguation

Xiao Yang[1, a], Su-Juan Qin[2, b]

[1]State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, 100876, China

[a]yx10542@163.com, [b]qsunjuan@bupt.edu.cn

Keywords: named entity disambiguation, graph ranking, LSI, PageRank.

Abstract. Named entity disambiguation (NED) is the task that mapping the named entities appearing in the text to their correct corresponding in the knowledge bases (KB). This paper presented a disambiguation approach based mainly on graph ranking by combining the PageRank and LSI(Latent Semantic Index) model. Experiments on 27,819 named entities showed the effectiveness of using, and the accuracy is higher than the state-of-art methods.

Introduction

As the foundation of nature language processing (NLP), named entity disambiguation plays vital role in the information retrieval, machine translation, online recommendation system and other NLP applications. In the task of information retrieval (IR), named entity disambiguation can help to distinguish different entities appeared with the same text. By identifying specific entities, it can extract useful information for specific entity in the huge amount of documents exists in Internet, and expand the contents in the knowledge base like Wikipedia.

The entities we see every day are usually ambiguous. For example, in some circumstances, “Jordan” may refer to the famous basketball player “Michael Jordan”, and in other circumstances, it may refer to a professor who majors in Machine Learning. The ambiguous entities exists in the documents are always influence the performance of application based on NLP, so that lots of researchers made efforts to solve it. A new solution [1][2] was proposed recently, which is based on graph and can achieve good performance. The basic idea of this method is that the mentions appeared in the same text is correlated, so we can use graph to measure this relatedness. But a problem is that the semantic relatedness between entities is not fully utilized. The previous researches [3] are mainly on hyperlinks between Wikipedia pages which is difficult to capture the semantic relatedness between entities. Besides, the effect of the method [4] could be easily influenced by the different training dataset.

In this paper, we proposed an improved named entity disambiguation algorithm based on graph method. The foundation of our improvement is also constructing a solution graph. In the graph, the candidates of entities consists of the nodes of solution graph, edges in our graph represent relations between candidates. The key problem of graph based named entity disambiguation is the quality of solution graph, we use LSI [5] to calculate the semantic relatedness between candidate entities and construct a high quality solution graph, then combined with PageRank and prior value, we proposed new method to select best candidate for every entity. We use the public dataset to test our improvement, and compare with the existing methods. The result shows our improvement can increase the accuracy rate.

Graph Model

The input is a document containing pre-tagged named entities textual mentions. We extracted the necessary elements to construct a graph model $G(V,E)$. The set of named entities is $M=(m_1,...,m_k)$, for any $m_i \in M$, there is a set of candidates $C_i=(c_1^i,...c_n^i)$, So we can build the vertex of the graph model, defined as $V=\{(m_i,c_j^i)\,|\,m_i \in M, c_j^i \in C_i\}$.

NE Candidate Generations. Candidate entity generation is to find a set of candidate entities for

each entity appeared in the background text. The extended form of entity reference is matched with the resources based on the resources acquired by Wikipedia. If the entity matches the entity in extended form, then the entity in extended form is used as a candidate entity.

Weight of Node. Each node in the graph consists of an entity and a candidate (m,c). The weight of node defined as $prior(m,c)$, it represents the importance of a candidate entity and obtained by following methods.

Levenshtein Distance is the string similarity between entity and its candidate.

$$prior(m,c) = \text{Levenshtein}(m,c) \tag{1}$$

Freebase provides an API [6] to get an entity's popularity score. This score is a function of the entity's inbound and outbound link counts in Freebase and Wikipedia.

$$prior(m,c) = \text{pop}(c) \tag{2}$$

Weight of Edge. The edges in the graph represent the relationship between any two entities, and the semantic relatedness is used to represent the relationship. The prior method [7] use the Ref value as the edge of the graph, Ref is based on hyperlink relations between candidates Wikipedia documents.

$$Ref(c_i,c_j) = \begin{cases} 1 & \text{if } e_i \text{ has a hyperlink point to } e_j \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

In order to capture the semantic relatedness between entities, we use LSI model to calculate the semantic relatedness between candidates.

$$LSICoh = \text{LSI}\,Sim(c_i,c_j) \tag{4}$$

When the value of LSICoh meets the threshold value ε, we insert the edge. Because the relationship between the semantic relationships is bidirectional, so we build a graph in the end is an undirected graph.

Graph based named entity disambiguation

The main idea of the algorithm is the named entities that appear in the same text are related to each other. We use this relationship to help every named entity to find its' own optimal candidate entities.

Computation of Semantic Relatedness. For any two named entity candidates c_i, c_j, we extract their Wikipedia introduction texts, remove the stop words and stem process. Then we get word items-document matrix. After the SVD (singular value decomposition) process, we can calculate the text semantic similarity between two candidates $\text{LSI}sim(c_i,c_j)$, and then we set the threshold ε. Only when the semantic relatedness between the two candidate entities is larger than ε, that is $\text{LSI}\,sim(c_i,c_j) > \varepsilon$, there is an edge between two candidate entities in the graph. The aim is to reduce the number of edges in the graph and reduce the computational complexity. At the same time, the weight of each edge of the weighted undirected graph is the semantic relatedness between two candidate entities.

Graph Ranking. The goal of graph ranking is to measure the importance of a candidate entity in network. Our model based on PageRank [8]. The idea of PageRank is based on the random walk model. First given a graph and a starting point, and then jump to a new vertex in the probability of ε, or take the current node as a starting point, move to the next vertex connected to the current node with the probability of $1\text{-}\varepsilon$. Repeat the process, until the whole system stabilized. The PageRank value of each node represents the probability that the vertex is accessed.

Disambiguation Process. We use the LSI model to measure the semantic relatedness between candidate entities, which is the edge of the graph model. Then we calculate the PageRank value of the graph. We define the score of a candidate:

$$Score(c) = PageRank(c) * \text{Prior}(m,c) \tag{5}$$

So the best candidate is the highest scoring candidate:

$$d(m) = \arg\max_{c \in C_m} score(c) \quad (6)$$

Experiments and Results

We use a publicly available data set AIDA [9] to verify the effectiveness of our improved algorithm; AIDA data set contains 1393 text and 34965 artificial tagging named entities. We made three comparison experiments to verify the effectiveness of our model.

We use two evaluation metrics: (1) Micro accuracy is the fraction of correctly disambiguated entities; (2) Macro accuracy is the proportion of textual mentions, correctly disambiguated per entity, averaged over all entities.

Table 1: Results using weighed edges

	Edit Distance		Entity Popularity	
Edge Weight	Micro	Macro	Micro	Macro
Ref	78.67	77.04	80.21	79.37
LSICoh	82.60	81.33	85.89	86.20

Experiment I**.** In the first experiment, our goal is to measure the impact of every feature. Table 1 shows the accuracy when using different combinations of node weights and edge weights, the result shows that the combination of entity popularity+LSICoh acquires the best result as the Micro and Macro accuracy of 85.89% and 86.20%, except that, we can find that entity popularity is more suitable for node weights. When using LSICoh as the edge weights, we can achieve better performance in accuracy and effectiveness of our improved method could be verified.

Table 2: Performance of our model compared with state-of-the-art models on AIDA dataset

Models	Cucerzan	Kulkarni	Hoffart	Shirakawa	Alhelbawy	Ours
Micro	51.03	72.87	81.82	82.29	87.59	85.89
Macro	43.74	76.74	81.91	83.02	84.19	86.20

Experiment II**.** In the second experiments, we use Hoffart [10] results as they reimplemented two other systems and also ran them over the AIDA dataset. We also compare with Alhelbawy [11] and Shirakawa [12] who carried out their experiments using the same dataset. The result shows that our model outperforms existing models.

Conclusions

In this paper we proposed an improved algorithm for named entity disambiguation based on graph model. Our main improvement is combining the semantic relatedness with graph model to filter the candidate entities. Our proposed features are very simple and easy to extract, and work well when employed in PageRank. The results show that our models can achieve better performance than existing models. In the future, we plan to explore other strategies and constraints for noise reduction in the document graph.

References

[1] Xianpei Han, Le Sun, and Jun Zhao. 2011. Collective entity linking in web text: a graph-based method. In Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval, pages 765–774. ACM

[2] Xiao Cheng and Dan Roth. 2013. Relational inference for wikification. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 1787–1796, Seattle, WA

[3] Pershina, Maria, Y. He, and R. Grishman."Personalized Page Rank for Named Entity Disambiguation." NAACL 2015.

[4] Ravi Sinha and Rada Mihalcea. 2007. Unsupervised graph-basedword sense disambiguation using measures of word semantic similarity. In Proceedings of the International Conference on Semantic Computing, pages 363–369

[5] Hofmann, Thomas. "Probabilistic latent semantic indexing." International ACM SIGIR Conference on Research and Development in Information Retrieval ACM, 2004:56-73.

[6] Information on https://developers.google.com/freebase/v1/search

[7] Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 2011. Local and global algorithms for disambiguation to wikipedia. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pages 1375–1384, Portland, OR.

[8] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November. Previous number = SIDL-WP-1999-0120.

[9] Information on http://www.mpi-inf.mpg.de/yago-naga/aida/

[10] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Furstenau, Manfred Pinkal, Marc Spaniol, ¨Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pages 782–792.

[11] Ayman Alhelbawy and Robert Gaizauskas. 2014. Graph Ranking for Collective Named Entity Disambiguation. In Proceedings of the 52nd Annual Meeting of the
Association for Computational Linguistics (Volume 2:Short Papers), pages 75–80

[12] Masumi Shirakawa, Haixun Wang, Yangqiu Song, Zhongyuan Wang, Kotaro Nakayama, Takahiro Hara, and Shojiro Nishio. 2011. Entity disambiguation based on a. Technical report, Technical report, Technical Report MSR-TR-2011-125, Microsoft Research.