Named Entity Disambiguation Based on Bidirectional Semantic Path

Zimao Li[1,2], Yue Zhang[1,2(✉)], Fan Yin[1,2], and Mengyan Nie[1,2]

[1] College of Computer Science, South-Central University for Nationalities, Wuhan 430074, China
1114279203@qq.com

[2] Hubei Provincial Engineering Research Center for Intelligent Management of Manufacturing Enterprises, Wuhan 430074, China

Abstract. In the existing entity disambiguation algorithms, the full connection method is adopted to measure the association between candidate entities, ignoring the ambiguity of entity mention, and the shortest path feature used for feature extraction does not consider the influence of different path starting and ending points. Therefore, this paper proposes a progressive named entity disambiguation (NED) algorithm PNED, which uses the bidirectional semantic association feature and topic relevance feature extracted from the disambiguated entity to calculate the association relation between entities. The algorithm fully considers the difficulty of different entity mention disambiguation, and gradually completes the NED task from easy to difficult, which improves the implementation efficiency of NED. Experimental results show the effectiveness of the algorithm.

Keywords: Entity disambiguation · Bidirectional semantic path · Progressive disambiguation

1 Introduction

Named entity disambiguation has been extensively studied and has achieved many outstanding results, but the related research still needs to be further improved. Although many entity disambiguation methods have been reported successively, the asymmetry of the shortest path between entities has not been taken into account when using the shortest path feature to measure the degree of association between candidate entities. In fact, different starting point may lead to two shortest paths with different lengths [1]. As shown in Fig. 1, the shortest path length from "Buffalo Bills" to "UCF Knights football" is 2, while the length from "UCF Knights football" to "Buffalo Bills" is 1.

To mine the path association between two entities more accurately, we propose a new method to redefine the shortest path length between two entities. Besides, existing entity disambiguation algorithms use a fully connected

Z. Li—Major Projects of Technological Innovation in Hubei Province (No. 2019ABA101).

Q. Liang et al. (Eds.): Artificial Intelligence in China, LNEE 854, pp. 440–447, 2022.
https://doi.org/10.1007/978-981-16-9423-3_55

method to measure the association between candidate entities, ignoring the degree of discrimination of entities, which increases the complexity of algorithm execution. Therefore, we use the bidirectional semantic association feature and topic relevance feature to calculate the association relation between the disambiguated entity and the remaining entity vertex, which disambiguate the entities gradually.

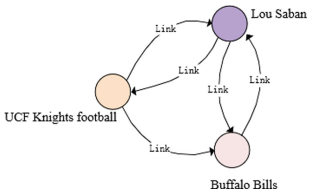

Fig. 1. The shortest path on knowledge graph

2 Related Work

The main idea of NED algorithm is using the extracted features to measure the similarity between the entity mention and the candidate entity. In 2011, Hoffert et al. [2] took entity mention and candidate entity as the vertex of graph structure, and used the greedy algorithm to find the smallest strong connected graph to link entities. In the same year, Han et al. [3] proposed an integrated reasoning algorithm in the graph structure to capture the correlation between vertex. In 2014, Alhelbowy et al. [4] proposed two integrated entity disambiguation algorithms using graph structure. The first algorithm is to sort all vertex by PageRank algorithm [4–7], and select candidate vertex according to PR value and local confidence. The second algorithm is to use the clustering algorithm to find the most important sub-cluster, and expand it until all entity mentions are disambiguated. In 2016, Stefan et al. [6] introduced a new "pseudo" theme vertice in the entity correlation graph to enhance the consistency between entities and the theme vertice. In 2017, Pappu et al. [8] used forward-backward algorithm to only consider the adjacent relation between entities, and achieved the fastest disambiguation result. In 2019, Liu et al. [9] introduced an extension of BERT (KBERT) in his work, in which KG triples are injected into the sentences as domain knowledge. In 2020, Mulang et al. [10] use the context extracted from the knowledge graph to provide enough information for the model, and improves the performance of NED.

In summary, there is a wide variety of approaches in the literature for NED, which provides more possibilities to break the shackles of traditional search engines.

3 Progressive Named Entity Disambiguation (PNED)

3.1 Problem Description and Formalization

In practical application, the ambiguity degree of each entity mention in the text is not always the same. When disambiguating the entities, entity with least ambiguity is firstly disambiguated, and then disambiguating the remaining entities gradually. In particular, the disambiguated entities can provide additional information to assist in the disambiguation of the remaining entities.

This paper proposes a progressive named entity disambiguation algorithm PNED, which combined with the idea of the minimum spanning tree. The entity mentions contained in text D is marked as $M = \{m_1, \ldots, m_n\}$, and its candidate set in knowledge base W is $C = \left\{\left\{e_1^1, e_1^2, \ldots\right\}, ..., \left\{e_n^1, e_n^2, \ldots\right\}\right\}, C_i \in C$. The task of PNED is to find an entity e_i in each candidate subset C_i, so that the sum of edge weights between tree vertex is minimized, which is marked as L.

3.2 Candidate Entity Representation Model

The representation model is presented as graph structure [11], which is recorded as $G(V, E)$. The candidate entities of mentions in text D are formalized as V of the graph, and the relation between entities is expressed as E. We use Stanford NER tool to get entity mention set $M = \{m_1, \ldots, m_n\}$, and obtain the candidate entity set through the Search API provided by Google Knowledge Graph.

Definition 1 Initial Confidence of Candidate Entities. The initial confidence of a candidate entity is the possibility of being linked without considering the context information. In our paper, the Search API of Google Knowledge Graph is used to return the entity's result score as the initial confidence as shown in Formula 1. We use the minimum edit distance to filter the candidate entities, and the top 5 candidate entities are selected to generate the final candidate entity set.

$$CM(v_a) = \frac{ResultScore(v_a)}{\sum_{v_a \in V} ResultScore(v_a)} \tag{1}$$

Definition 2 Bidirectional Path Associations Between Candidate Entities. It is the shortest path length of the hyperlink between entries of candidate entities in Wikipedia:

$$RelPath(v_a, v_b) = \frac{1}{1 + ShortPath(v_a, v_b)} \tag{2}$$

$ShortPath(v_a, v_b)$ is the shortest path length between the candidate entity vertice v_a and v_b, which can be obtained by Formula 3:

$$ShortPath(v_a, v_b) = \frac{FShortPath(v_a, v_b) + BShortPath(v_a, v_b)}{2} \tag{3}$$

$FShortPath(v_a, v_b)$ represents the shortest path length from the vertice v_a to v_b in Wikipedia; while $BShortPath(v_a, v_b)$ represents the length from the vertice v_b to v_a.

Definition 3 Bidirectional Semantic Associations Between Candidate Entities. Using semantic information to weight the shortest path can better express the correlation between two vertex. The calculation formula is shown in Formula 4.

$$RelBsa(v_a, v_b) = (1 - \alpha)SimText(v_a, v_b) + \alpha RelPath(v_a, v_b) \tag{4}$$

$SimText(v_a, v_b)$ is the semantic similarity between the text description of candidate entities, which is calculated by cosine similarity [12]. For the elements in the candidate entity set corresponding to the same entity mention, the $RelBsa$ value is 0.

Definition 4 Topic Relevance. Entity mentions in a text usually have a common topic. Combining the topic relevance features of candidate entities into the disambiguation process can further enrich the features and help to improve the accuracy of algorithm. $Topic(e_i^k)$ represents the topic vector of the kth candidate entity e_k of entity mention m_i. The topic relevance is calculated in Formula 5:

$$RelTopic(e_i^k, e_j) = \frac{Topic(e_i^k) \cdot Topic(e_j)}{\left\|Topic(e_i^k)\right\| \left\|Topic(e_j)\right\|}, e_i^k \in C_i, e_j \in E_d \tag{5}$$

Definition 5 Entity Relevance. The relevance of entities is related to the disambiguation of the remaining entities, which is determined by the bidirectional semantic association of entities and the topic relevance between candidate entities. The entity association degree of disambiguated entities and remaining entities is shown in Formula 6:

$$Rel(e_i^k, e_j) = \beta RelBsa(e_i^k, e_j) + (1 - \beta)RelTopic(e_i^k, e_j), e_i^k \in C_i, e_j \in E_d \tag{6}$$

3.3 Determination of Initial Vertice

To adapt to the task of NED, in this paper, the initial vertice of the minimum spanning tree is not randomly selected as the traditional Prim algorithm, but the entity with the least ambiguity.

Definition 7 Easy Entity Mention (EEM). The entity mention with the smallest amount of information has more certainty, the least ambiguity, and the lowest disambiguation difficulty, which is determined as Easy Entity Mention (EEM). Its real semantics and attributes can help to resolve the ambiguity of other entity mentions. The EEM should satisfy the following two conditions: (1) If the number of candidate entities corresponding to an entity mention is 1, the entity mention is EEM. (2) If the number of candidate entities corresponding to all entity mentions in the entity mention set is more than 1, the entity mention with the smallest amount of information is EEM.

To determine the EEM, the topic vector of the candidate entity is used to represent the information quantity of entity mention. In this paper, we train LDA topic model and construct the entity mention topic matrix through gensim tool. Then, Singular Value Decomposition (SVD) is used to obtain the singular values of the entity mention topic matrix.

Suppose an entity mention m_i has l candidate entities, and the topic matrix $MTM_{i,l\times c}$ of m_i is constructed from the topic vectors of l candidate entities, where c represents the number of topics in LDA topic model. The singular value reflects the information quantity of matrix $MTM_{i,l\times c}$. Formula 7 is used to approximately calculate the singular degree (SD). σ is the diagonal element of the diagonal matrix after the matrix $MTM_{i,l\times c}$ singular value decomposition.

$$SD = \frac{\sigma_1}{\sum_{h=1}^{c} \sigma_h} \tag{7}$$

The amount of information $H(m_i)$ of entity mention decreases with the increase of singular value concentration degree SD of entity mention topic matrix, as shown in Formula 8:

$$H(m_i) \propto \frac{1}{SD} \tag{8}$$

Definition 8 Initial Vertice of Minimum Spanning Tree. The candidate entity that the EEM refers to is regarded as the initial vertice of the minimum spanning tree, which is also added to the disambiguated entity set E_d.

If the number of candidate entities corresponding to the EEM is 1, then the candidate entity is the real linked entity of the EEM. If the number is more than 1, the candidate entities are filtered according to the initial confidence and the text relevance between. The real direction of the EEM is determined as shown in Formula 9:

$$E_d = \left\{ e_i^k | argmax(SimCon(e_i^k) + CM(e_i^k)), e_i^k \in EEM \right\} \tag{9}$$

3.4 Progressive Named Entity Disambiguation (PNED)

Combined with the idea of getting the minimum spanning tree by Prim algorithm, this paper proposes a progressive NED algorithm. The execution process is as follows: firstly, the EEM is selected. After that, the EEM is disambiguated according to the initial confidence of candidate entities and the text relevance. Then, by calculating the bidirectional semantic association and topic relevance, the remaining entities are gradually disambiguated.

Suppose that a text contains five entity mentions, and each entity refers to 2–3 candidate entities. The schematic diagram of PNED candidate entity representation model construction process is shown in Fig. 2.

4 Experiment and Analysis

4.1 Dataset

Wikipedia corpus covers a wide range of information as a comprehensive knowledge base. The knowledge base of NED algorithm in this paper is constructed by

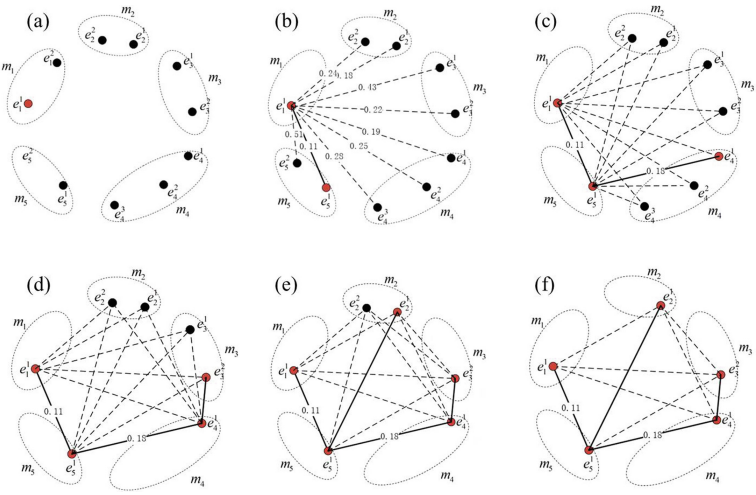

Fig. 2. Schematic diagram of construction process of PNED candidate entity representation model

English Wikipedia, which contains more than 5 million entities and more than 30 million hyperlinks. We use AIDA dataset to test the NED effect of PNED algorithm. There are 1393 documents in AIDA dataset, including 34956 entity mentions, 27820 linkable entity mentions, and 7136 non-linkable (NIL) entity mentions.

4.2 Comparison Algorithm

In order to verify the effectiveness of PNED algorithm in the NED task, it is compared with the current mainstream algorithm on AIDA dataset. The algorithms involved in the comparison include the algorithm proposed by Alhelbawy et al. [4], the algorithm proposed by Hoffert et al. [2], GCEM algorithm proposed by Liu [12] in 2018, and BSANED, our NED method based on bidirectional semantic association but without progressive disambiguation.

4.3 Experimental Parameter Setting

In PNED algorithm, parameter α is used to weigh the weight of edge E in candidate entity representing model. The TestA test set in AIDA dataset is used to determine the value of α, which is set to 0.4, and the maximum length of the shortest path, which is set to 5.

4.4 Analysis of Experimental Results

In order to verify the effectiveness of the BSANED algorithm proposed in this paper, it is compared with four algorithms which also use AIDA dataset.

Table 1. Experimental results of PNED and other algorithms on AIDA (%)

Algorithms	P	R	F1
Hoffart	81.82	81.91	81.86
Alhelbawy	87.59	84.19	85.85
GCEM	84.25	81.36	82.78
BSANED	**90.75**	**87.84**	**89.27**
PNED	**87.76**	**85.82**	**86.78**

Table 2. Average execution time of each text on AIDA

Algorithms	Time
BSANED	**0.51** s
PNED	**0.37** s

As shown in Table 1, compared with the PageRank algorithm proposed by Alhelbowy et al. [4], the algorithm using the most dense sub-graph proposed by Hofhart et al. [2], and the GCEM algorithm proposed by Liu [12], the PNED performs better, which shows that the PNED is effective for NED. However, compared with BSANED, PNED has lower accuracy, recall rate and F1 value. This is mainly because in PNED, only one entity mention is disambiguated in the first disambiguation process. If the disambiguation can be referred to multiple EEMs in the first step, the topic of the text can be captured more accurately. In progressive named entity disambiguation, the PNED makes use of the topic relevance and bidirectional semantic association features of the entities. If the topic vector of the disambiguated entity deviates from the topic vector of the text, it is easy to cause topic drift, which will affect the accuracy of subsequent entity disambiguation.

In Table 2, we compared PNED with BSANED by the average execution time of each text. PNED reduces the time by about 37.8%. This is because BSANED calculates the association among all candidate entities in the same text in batch while PNED reduces the number of vertices in the graph structure by removing invalid candidate entities, which can improve the efficiency of the whole disambiguation process.

5 Summary

In this paper, a progressive named entity disambiguation algorithm is proposed according to the different degrees of ambiguity of the entity mention. The minimum spanning tree is constructed in the graph structure to achieve the purpose of gradually disambiguating the entity mentions from easy to difficult. The task of progressive entity disambiguation is completed by combining the topic relevance feature obtained from EEM and bidirectional semantic relevance feature. The disambiguation of EEM can bring more clear and reliable context information, which can further enrich the feature. Moreover, by removing the remaining invalid candidate entities, the number of vertice in the graph structure and the calculation of the association relation between entities are reduced, which can help to improve the efficiency of the whole disambiguation. Experimental results

on AIDA dataset show that the proposed algorithm can improve the implementation efficiency of NED algorithm.

Acknowledgments. This work was supported by the Major Projects of Technological Innovation in Hubei Province (No. 2019ABA101), Industry-University-Research Innovation Fund-"Moss Digital Intelligence Integration" Collaborative Innovation Project (2020QT08), Science and Technology Development Center of Ministry of Education, and the Research Team of Key Technologies of Smart Agriculture and Intelligent Information Processing and Optimization.

References

1. Yang, G.: Research on named entity disambiguation based on graph method. Harbin Industrial University (2015). (in Chinese)
2. Hoffart, J., Yosef, M.A., Bordino, I., et al.: Robust disambiguation of named entities in text. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pp. 782–792. ACL, Edinburgh (2011)
3. Han, X., Zhao, J.: Structural semantic relatedness: a knowledge-based method to named entity disambiguation. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pp. 50–59. ACL, Uppsala (2010)
4. Alhelbawy, A., Gaizauskas, R.: Collective named entity disambiguation using graph ranking and clique partitioning approaches. In: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pp. 1544–1555. ACL, Dublin (2014)
5. Alhelbawy, A., Gaizauskas, R.: Graph ranking for collective named entity disambiguation. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pp. 75–80. ACL, Baltimore (2014)
6. Zwicklbauer, S., Seifert, C., Granitzer, M.: Robust and collective entity disambiguation through semantic embeddings. In: International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 425–434. ACM, Pisa (2016)
7. Haveliwala, T.H.: Topic sensitive PageRank: a context-sensitive ranking algorithm for web search. IEEE Trans. Knowl. Data Eng. **15**(4), 784–796 (2003)
8. Pappu, A., Blanco, R., Mehdad, Y., Stent, A., Thadani, K.: Lightweight multilingual entity extraction and linking. In: International Conference on Web Search and Data Mining Conference Series: Web Search and Data MiningLink, pp. 365–374. ACM, Cambridge (2017)
9. Liu, W., et al.: K-BERT: enabling language representation with knowledge graph. In: The 10th AAAI Symposium on Educational Advances in Artificial Intelligence, pp. 2901–2908. AAAI, New York (2019)
10. Mulang, I.O., Singh, K., Prabhu, C., Nadgeri, A., Hoffart, J., Lehmann, J.: Evaluating the impact of knowledge graph context on entity disambiguation models. In: Proceedings of the 29th ACM International Conference on Information and Knowledge Management, pp. 2157–2160. ACM, New York (2020)
11. Milne, D., Witten, I.H.: Learning to link with Wikipedia. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management, pp. 509–518. ACM, Napa Valley (2008)
12. Liu, B.: Entity Linking Based on Graph and deep learning. Central China Normal University (2018). (in Chinese)