



Collective Entity Disambiguation Based on Deep Semantic Neighbors and Heterogeneous Entity Correlation

Zihan He, Jiang Zhong^(✉), Chen Wang, and Cong Hu

Chongqing University, Chongqing 400044, People's Republic of China
zhongjiang@cqu.edu.cn

Abstract. Entity Disambiguation (ED) aims to associate entity mentions recognized in text corpus with the corresponding unambiguous entry in knowledge base (KB). A large number of models were proposed based on the topical coherence assumption. Recently, several works have proposed a new assumption: topical coherence only needs to hold among neighboring mentions, which proved to be effective. However, due to the complexity of the text, there are still some challenges in how to accurately obtain the local coherence of the mention set. Therefore, we introduce the self-attention mechanism in our work to capture the long-distance dependencies between mentions and quantify the degree of topical coherence. Based on the internal semantic correlation, we find the semantic neighbors for every mention. Besides, we introduce the idea of “simple to complex” to the construction of entity correlation graph, which achieves a self-reinforcing effect of low-ambiguity mention towards high-ambiguity mention during collective disambiguation. Finally, we apply the graph attention network to integrate the local and global features extracted from key information and entity correlation graph. We validate our graph neural collective entity disambiguation (GNCED) method on six public datasets and the results demonstrate a better performance improvement compared with state-of-the-art baselines.

Keywords: Entity disambiguation · Local topical coherence · Long-distance dependencies · Entity correlation graph

1 Introduction

As the key technology of multiple natural language processing tasks, such as knowledge graph construction, information extraction, and so on, entity disambiguation (ED) has gained increasing attention. Formally, it aims to associate

Supported by National Key Research and Development Program of China Grant 2017YFB1402400, in part by the Key Research Program of Chongqing Science and Technology Bureau No. cstc2019jscx-fxyd0142, in part by the Key Research Program of Chongqing Science and Technology Bureau No. cstc2019jscx-mbdxX0012.

© Springer Nature Switzerland AG 2020
X. Zhu et al. (Eds.): NLPCC 2020, LNAI 12431, pp. 193–205, 2020.
https://doi.org/10.1007/978-3-030-60457-8_16

entity mentions recognized in unstructured text with the corresponding unambiguous entry in a structured knowledge base (KB) (e.g., Wikipedia). However, this task is challenging due to the inherent ambiguity between surface form mentions. A unified form of mention in different context may refer to different entities, and different surface form mentions may refer to the same entity in some cases. For example, the mention “Titanic” can refer to a movie, a ship, or a shipwreck in different contexts.

To solve the problem, current ED methods have been divided into local disambiguation models and global disambiguation models. The former focus on the local information around the mention and related candidate entity. The latter additionally consider the correlation between entity mentions in the same document. Generally, based on the assumption that the mentions in the same document shall be on the same topic, large numbers of global models have been proposed. In particular, the work [1, 18] claimed that topical coherence only need to hold among mention neighbors, which we called “local topical coherence” in this paper. They calculated sequence distance and syntactic distance respectively to determine the mention neighbors, which may lead to inconsistent mention sets due to insufficient mining of deep semantic associations between entities. In fact, our paper will be developed based on the same assumption.

To solve the above problems, our paper tries to calculate the semantic distance between mention pairs and select a set of mention neighbors with the closest semantic distance for each mention. Then, we introduce the self-attention mechanism into our model to model the text deeply and better capture the internal relevance of entity mentions.

Besides, we introduce the simple to complex (S2C) idea to the construction of entity correlation graph. We fully exploit the key information brought by the low-ambiguity mentions and the supplementary information obtained from the external KB to promote the disambiguation of the high-ambiguity mentions, to achieve the self-reinforcing of the collective process. In particular, we build a heterogeneous entity correlation graph based on the correlation information between mentions, and further aggregate the feature data.

Therefore, the main contributions of our ED method can be summarized as:

- (1) We propose a semantic-information based mention neighbors selection method to capture the semantic relevance between mentions and find top-k closest semantic distance mention neighbors for each mention to disambiguate.
- (2) We construct a new collective disambiguation entity correlation graph and introduce the idea of simple to complex to dig the disambiguation effect of the low-ambiguity mentions on the high-ambiguity mentions.
- (3) We evaluate our method on several public datasets. The experimental results compared with existing state-of-the-art ED baselines verify the efficiency and effectiveness of our model.

2 Related Work

Entity Disambiguation. Entity disambiguation in nature language processing tasks, has gained increasing attention in recent years. Many research work has been proposed based on two main disambiguation models: local models and global models. Early local ED models mainly extracted string features between candidate entities and the local context of current mention to find the optimal solution for each mention [1, 3, 13]. Since the increasing popularity of deep learning, recent ED approaches had fully used neural network, such as CNN/LSTM-encoders [4, 8], to learn the representation of context information and model the local features. By contrast, a large number of collective disambiguation models have been proposed based on the hypothesis: all mentions in a document shall be on the same topic. However, the maximization of coherence between all entity disambiguation decisions in the document is NP-hard. [11] had firstly tried to solve it by turning it into a binary integer linear program and relaxing it to a linear program (LP). [9] proposed a graph-pruned method to compute the dense sub-graph that approximated the best joint mention-entity mapping. [7, 12, 15, 19] applied the Page Rank, Random Walk, Loop Belief Propagation algorithm respectively to quantify the topical coherence for finding the optimal linking assignment. Recently, [1, 10, 18] applied graph neural network into the calculation of global coherence, such as GCN/GAT.

Self-attention. The self-attention mechanism was firstly proposed in the task of machine translation [16], which caused a great of focus. Self-attention mechanism can associate any two words in a sequence to capture the long distance dependency between them. And, it had been cited by a large number of studies and generalized to many NLP tasks [2, 17, 21]. In our paper, we apply the self-attention mechanism to capture the dependencies between distant mentions to hold the topical coherence assumption.

3 Graph Neural Collective Entity Disambiguation

3.1 Overview of Framework

As with most entity disambiguation work, we take a document collection as input where all the candidate entity mentions have been identified. Formally, we define the collective disambiguation task as follows: given a set of mentions $M(D)$ in a document D and the candidate entities generated, $C(m_i) = \{e_{i1}, e_{i2}, \dots, e_{ij}\}$, the goal of our model is to find an optimal linking assignment. As the Fig. 1 shown, our model mainly includes the mainly two modules: feature extraction module and graph neural disambiguation module. The details are as follows:

Embedding of Word, Mention and Entity: In the first step, we need to get the embedding vector to avoid manual features and better encode the semantics of words and entities. Following the work of [6], we train the embedding of each

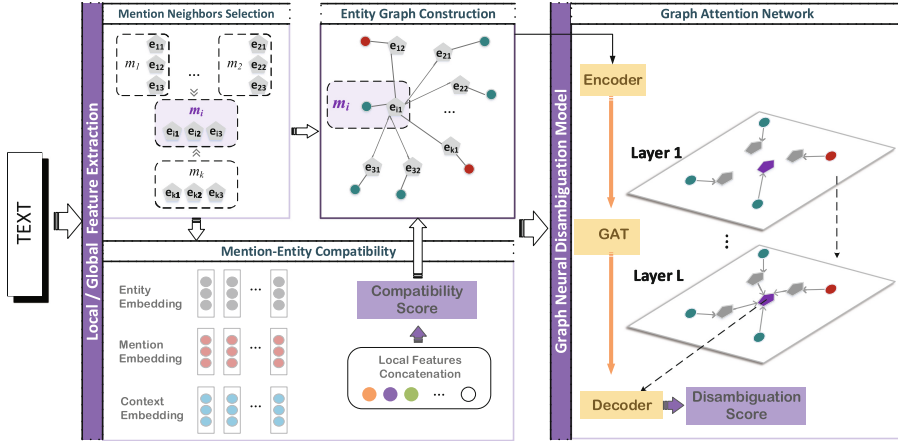


Fig. 1. Overview of framework.

word and related entity at the same time (the mention embedding can calculate from related word embedding).

Candidate Generation: As the essential procedure, the candidate generation step affect the accuracy of entity disambiguation and the recall rate directly. Generally, we generate candidate entities for each entity mention in document based on the mapping dictionary built by [1, 9, 14], noted as $C(m_i) = \{e_{i1}, e_{i2}, \dots, e_{ij}\}$, where each entity corresponds to a specific entity entry in the knowledge base (Wikipedia in our paper).

Feature Extraction: Disambiguation is the key step in the entity disambiguation task. In this part, we consider extract two types of evidence to support the final decision: local features and global features. The features include there parts: string compatibility between the string of mention and candidate entities; contextual similarity between the text surrounding the mention and the candidate entity; entity relatedness for all mentions in the document. Following the work of [20], we construct the string compatibility features using the edit distance, noted as Sim_{str} . To make full use of the context and external information, we extract word level and sentence level contextual similarity evidence. On the basis of above features, we come to extract the global features. In particular, considering the local topical coherence, we propose a selection strategy based on semantic information to select most relevant mention neighbors for each mention. Then, we build the entity semantic correlation graph $G = (V, E)$ for each mention to characterize the relatedness between entities with the introduction of the idea of simple to complex (S2C) and dig deep into the contextual information and external KB, which achieves a self-reinforcing effect. The details will be explained in Sect. 3.2–3.4.

Neural Network Disambiguation Model: After the process of feature extraction, we can get a set of local similarity representation, and entity correlation graph G for each mention. Considering the special representation of

structured graph, Graph Attention Network (GAT) will be used in our paper to better aggregate feature data and ensure the validity of feature information transmission. The detailed implementation of the model will be explained in Sect. 3.5.

3.2 Word and Sentence Level Contextual Compatibility

To extract local features, we first get the surrounding context of a mention and the textual representation (from external KB) of the given candidate entity. For mention m_i , we can get a c -word context $C(m_i) = \{w_1, w_2, \dots, w_{C_1}\}$, where C_1 is the context window size. For every candidate entity, we can get the complete description page from the knowledge base. To obtain more accurate keywords and reduce information processing complexity, we focus on the first two paragraph of the description page as the textual representation and extract the top C_2 terms with the highest TF-IDF score for given candidate entity e , noted as $C(e) = \{w_1, w_2, \dots, w_{C_2}\}$. To represent the local context information mentioned and the description information of the candidate entity more accurately, we design our model in word and sentence level.

Firstly, based on pre-trained word embedding, we can directly obtain the context representation at the word level [1]. The word level contextual compatibility $Sim(m, e)$ is defined as follows:

$$Sim(m_i, e)_{word} = \frac{D_m \cdot D_e}{\|D_m\| \|D_e\|} \quad (1)$$

where D_m and D_e are the weighted average of context vectors corresponding to the mention's and entity's textual representations.

Secondly, we try to use the Bi-LSTM model to encode sentence-level evidence. Differently, the evidence at sentence level takes the positional relation between words into consideration, which is more conducive to retaining the deep meaning of language. Feeding the sentence containing the mention m and the entity description information (contains several sentences) into the model respectively, we can obtain the final hidden state $\langle h_m, h_e \rangle$ as the sentence level vectors of the mention and entity. Then, the sentence level similarity is defined as follows:

$$Sim(m_i, e)_{sen} = \frac{H_m \cdot H_e}{\|H_m\| \|H_e\|} \quad (2)$$

3.3 Semantic Information Based Mention Neighbors Selection

In this section, we introduce our mention neighbors selection strategy based on the assumption of local topical coherence. The whole process is shown in Fig. 2.

We use the self-attention mechanism [16] to obtain the relevant features of the text from multiple angles. The self-attention mechanism is to do the attention inside the sequence and find the connection of the inner part of the sequence. We apply self attention mechanism to the entire document to catch the key

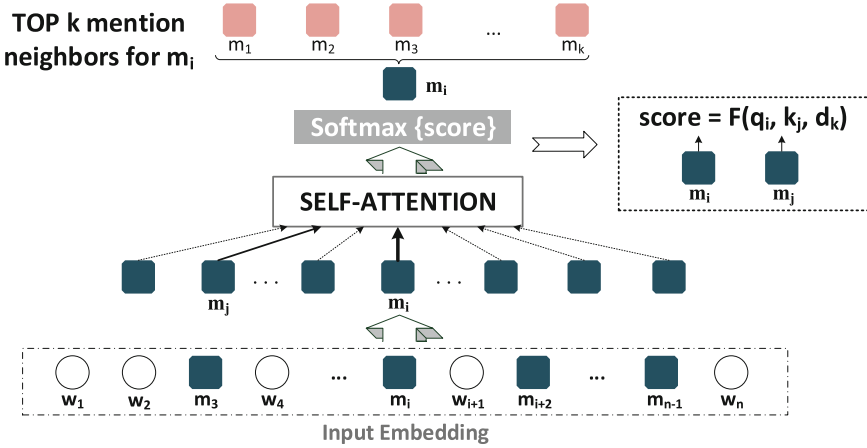


Fig. 2. Mention neighbors selection

semantic information among entity mentions. Considering that there are many words other than the mentions in the document and the needs of the problem, we only calculate the attention value with other mentions and the context words for every m_i , which is used to measure the semantic correlation between each mention pairs, which we called semantic distance α_{sd} .

To calculate the α_{sd} , we construct a basic multi-layer self-attention module to model mentions in the entire document. We use $\{X_1, X_2, \dots, X_n\}$ to represent the entire document, including all mentions X_{m_i} and their context words X_w . For the calculation of each self-attention layer, the embedding of mention m_i will be updated by encoding the context information and the associated information between mention pairs. The calculation process is as follows:

$$X'_{m_i} = \sum_{j,q} w_{ij} X_{m_j} + w_{iq} X_{w_q}; \quad w = \frac{Q \cdot K^T}{\sqrt{d_K}} \quad (3)$$

In the last layer of self-attention, we directly output the normalized attention value between mention pairs.

$$[\alpha_{sd}]_{ij} = X'_{m_i}{}^T X'_{m_j} \quad (4)$$

$$[\alpha_{sd}]'_{ij} = \frac{\exp[\alpha_{sd}]_{ij}}{\sum_j \exp[\alpha_{sd}]_{ij}} \quad (5)$$

After the above calculation, the semantic correlation between any two mentions in the document D can be represented as $[\alpha_{sd}]'_{ij}$. The larger the semantic correlation value, the closer the semantic distance between mention pairs. For mention m_i , we select mentions with the top-k minimum semantic distance as neighbors of the current mention m_i , $N(m_i) = \{m_1, m_2, \dots, m_k\}$.

3.4 Construction of S2C Entity Correlation Graph

The entity correlation graph is the key module of feature extraction as the structure of carrying and transmitting local and global information. To model the global semantic relationships, we construct a heterogeneous entity semantic graph for each mention m_i based on its neighbor mentions $N(m_i)$.

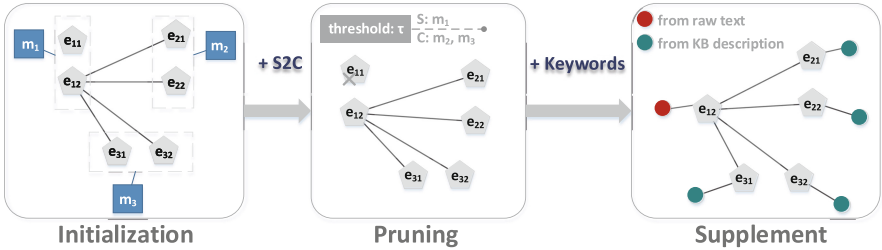


Fig. 3. Illustration of entity graph construction

As shown in Fig. 3, the process is divided into three steps: (1) **Initialization of the entity graph**: Take the candidate entities of mention m_i and its neighbor mentions as the initial nodes of the graph, and build graph G_1 , and establish edges between the candidate entities mentioned by different mentions. (2) **Pruning of the entity graph**: Introduce the idea of S2C. First, we will divide the entire mention set into simple and complex parts according to the threshold setting τ . In this setting, we make full use of local features to preferentially link (Simple) mentions with low ambiguity. Once the final entity referred to by Simple mention is identified, the redundant candidate entity nodes that mention has and the corresponding edges connected to these nodes are removed from the initial diagram G_2 . (3) **Supplement of the entity graph**: Introduce evidence nodes other than entity nodes. To maintain the influence of text context, we introduce two kinds of evidence nodes into entity graph G_2 : one is the top S_1 surrounding words of the simple mention selected from the document; another is the top S_2 key words for entity selected from the description page. We connect these evidence nodes with corresponding entity nodes to form new edges. Then, the construction of the entity correlation graph G is completed.

For every entity node, we initialize the representation with the concentration of pre-trained entity embedding and obtained local features, including Sim_{str} , Sim_{word} , Sim_{sen} . For every keyword node, we initialize the representation with the concentration of pre-trained word embedding and weights between keywords and corresponding entities. The initial representation have been expressed as f .

3.5 Disambiguation Model on Entity Correlation Graph

Our model adopts a Graph Attention Network to deal with the document-specific S2C entity semantic graph. In particular, the input of the neural model is the

sub-graph structure $G = \{V, E\}$, where contains all the entity and keyword nodes we need. All nodes in the graph G represented by the entity and word embedding are in the same space, so that the information between different nodes can be directly calculated. The overall goal of our model is to maximize the value in Eq. 6, where $Score(m, e_i)$ is a scoring function that our network model learns from multi-features for mention m and its candidate entities.

$$\Gamma(m) = \arg \max_{e_i \in \phi(m)} Score(m, e_i) \quad (6)$$

Encoder: In the first step, we use a multi-layer perception structure to encode the initial feature vector, where F is the matrix containing all the candidate entities and related word node representations f for a certain mention.

$$h^1 = \sigma(FW^1 + b^1) \quad (7)$$

Graph Attention Network: The graph attention network module aims to extract key features from the hidden state of the mention and its neighbor mentions. Then, we can derive the new representation for each mention as:

$$h^l = \sigma(Ah^{l-1}W^{l-1}) + h^{l-1} \quad (8)$$

where A is the symmetric normalized adjacent matrix of the input graph with self-connections. We normalize A such that all rows sum to one, avoiding the change in the scale of the feature vectors. To enable the model to retain information from the previous layer, we add residual connections between hidden layers.

Decoder: After going through multi-layer graph attention network, we will get the final hidden state of each mention in the document-specific entity graph, which aggregate semantics from their neighbor mentions in the entity semantic graph. Then, we can map the hidden state to the number of candidates as follows:

$$Score = W^L h^L + b^L \quad (9)$$

Training: To train the graph neural disambiguation model, we aim to minimize the following cross-entropy loss, where $P(\Delta)$ is a probability function calculated by $Score(m, e_i)$.

$$L_m = - \sum_{j=1}^n y_j \log(P(\hat{y} = e_j; f, \tilde{A}, w)) \quad (10)$$

4 Experiments

In this section, we compared with existing state-of-the-art methods on six standard datasets to verify the performance of our method.

4.1 Setup

Datasets: We conducted experiments on the following sets of publicly-available datasets used by previous studies: (1) AIDA-CoNLL: annotated by [9], this dataset consists of three parts: AIDA-train for training, AIDA-A for validation, and AIDA-B for testing; (2) MSNBC, AUIAINT, ACE2004: cleaned and updated by [7]; (3) WNED-CWEB, WNED-WIKI: two larger but less reliable datasets that are automatically extracted from ClueWeb and Wikipedia respectively [5, 7]. The composition scale of the above datasets can be seen in Table 1.

Table 1. Statistics of datasets in this experiment.

Dataset	AIDA(B)	MSNBC	AQUAINT	ACE2004	WIKI	CWEB
Total documents	213	20	50	36	320	320
Total mentions	4486	656	699	248	6821	11154

We train the model on AIDA-train and validate on AIDA-A. For in-domain and out-domain testing, we test on AIDA-B and other datasets respectively.

Baselines: We compare our model with the following state-of-the-art methods:

- AIDA [9]: built a weighted graph of mentions and candidate entities and computed a dense sub-graph that maps the optimal assignment.
- Random-Walk [7]: proposed a graph-based disambiguation model, and applied iterative algorithm based on random-walk.
- DeepEL [6]: applied a deep learning architecture combining CRF for joint disambiguation and solved the global training using truncated fitting LBP.
- NCEL [1]: first introduced Graph Neural Network into the task of NED to integrate local and global features.
- MulRel [12]: designed a collective disambiguation model based on the latent relations of entities and obtained a set of optimal linking assignments by modeling the relations between entities.
- CoSimTC [18]: applied a dependency parse tree method to drive mention neighbors based on the topical coherence assumption.
- GNED [10]: proposed a heterogeneous entity-word graph and applies GCN on the graph to fully exploit the global semantic information.

Experimental Settings: Our experiments are carried out on the PyTorch framework. For fair comparison, we train and validate our model on AIDA-A, and test on other benchmark datasets (including AIDA-B). We use standard micro F1 score (aggregates over all mentions) as measurement. Following the work [6], we get the initial word embedding and entity embedding with size $d = 300$, $\gamma = 0.1$ and window size of 20 for the hyperlinks. Before training, we have removed the stop words. We use Adam with a initial learning rate of 0.01 for optimization. For the over fitting problem, we use the early stopping to avoid it. Then, we set epoch = 50 and batch size = 64 to train our model. Besides,

we set top 10 candidate entities for every mention and the context window size to 20 to extract the local features. For other hyper-parameters, we set different values according to the situation.

4.2 Experimental Results

Overall Results: In this section, we compare our model with precious state-of-the-art baselines on six public datasets. The results of the comparison are listed in Table 2. It can be seen that our proposed model outperformed the current SOTA baselines on more than half datasets. Our proposed method has achieved the highest micro F1 score on AIDA(B), AQUAINT, ACE2004, and WIKI. On average, we can see that our model has achieved a promising overall performance compared with state-of-the-art baselines. For in-domain testing, our proposed model reaches the performance of Micro F1 of 93.57%, which is a 0.5% improvement from the current highest score. For out-domain testing, our method has achieved relatively high-performance scores on three datasets of MSNBC, AQUAINT, and ACE2004, which the best is achieved on the AUQUAINT and ACE2004 datasets. However, the improvement of our model on WIKI and CWEB datasets is not obvious. We analyze the data and think that the reason for this result may have a lot to do with the noise problem of the data itself.

Table 2. The micro F1 scores on six public datasets.

Model	AIDA(B)	MSNBC	AQUAINT	ACE2004	WIKI	CWEB	AVG
AIDA [9]	–	79.00	56.00	80.00	58.60	63.00	67.32
Random-Walk [7]	89.00	92.00	87.00	88.00	77.00	84.50	86.25
DeepEL [6]	92.22	93.70	88.50	88.50	77.90	77.50	86.39
NCEL [1]	87.20	–	87.00	88.00	–	86.00	87.05
MulRel [12]	93.07	93.90	88.30	89.90	77.50	78.00	86.78
CoSimTC [18]	–	94.16	90.90	92.92	76.96	75.02	86.00
GNET [10]	92.40	95.50	91.60	90.14	77.50	78.50	87.61
GNCED (our model)	93.57	95.00	92.40	93.92	78.03	82.67	89.27

Table 3. The comparison of mention neighbors selection strategy.

Model	AIDA(B)	WIKI
Basic ED + all mentions	74.16	89.41
Basic ED + sequence distance	76.30	90.44
Basic ED + syntactic distance	76.55	90.80
Basic ED + self-attention	78.22	92.27

Impact of Mention Neighbors Selection Strategy: In this part, we designed experiments to verify the performance improvement brought by our

self-attention based mention neighbors selection strategy in the whole ED model. Specifically, we compared our selection strategy with the adjacency strategy [1] and the syntactic distance strategy [18] respectively. To facilitate observation and explanation, we implement experiments on two testing datasets, WIKI, and AIDA(B). The results are shown in Table 3. We can see that for the document-level disambiguation, our semantic-based mention neighbors selection strategy can effectively improve the performance of collective disambiguation by selecting a set of most semantically relevant subsets for each mention.

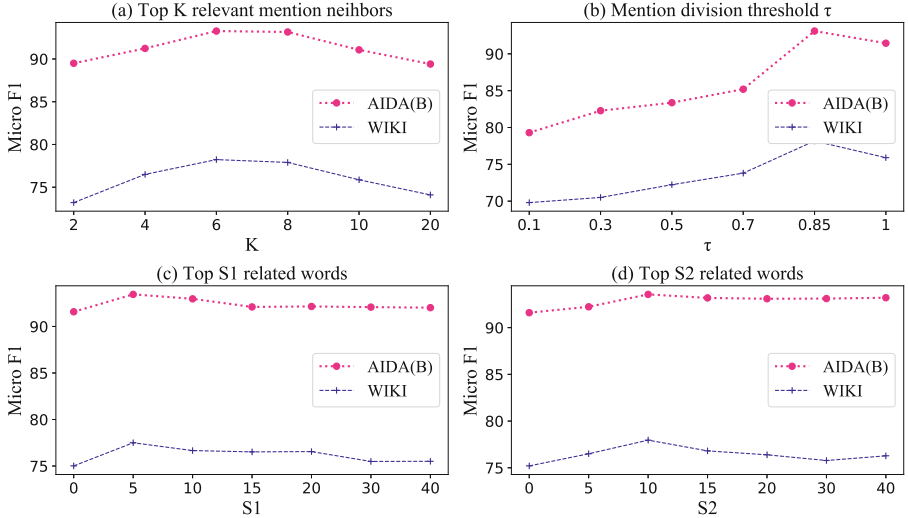


Fig. 4. The impact of hyper-parameters.

Impact of Hyper-Parameters: We analyzed the impact of three hyper-parameter settings in the model on the performance of the entire model. As in the last experiment, we completed this experiment on datasets, WIKI and AIDA(B). The parameters include the number K of top relevant mention neighbors for current mention m , the threshold parameter τ for mention division, the number S_1 of top related keywords for the entity of simple mentions, and the number S_2 of top related keywords for the entity of complex mentions. From Fig. 4, we can see that the parameters of K and τ have an obvious impact on the performance. Besides, the effects of parameters S_1, S_2 are big only when the values are between zero and non-zero but gradually become small as the values increase, which shows that the keywords selected from context and external KB improve the performance of our model. Generally, with the increasing of these parameters, the value of micro $F1$ will increase incrementally but decrease slightly after reaching a certain maximum value. After a large number of experiments, we found that the model performance can be the best when $K = 6, \tau = 0.85, S_1 = 5, S_2 = 10$.

5 Conclusion

In this paper, we propose a semantic based mention neighbors selection strategy for collective entity disambiguation. We use the self-attention mechanism to find the optimal mention neighbors among all mentions for the collective disambiguation. We also propose an entity graph construction method. We introduce the S2C idea to add more sufficient evidence information for the disambiguation process of high ambiguity mention and achieve a self-reinforcing effect in the disambiguation process. The results of experiments and module analysis have demonstrated the effectiveness of our proposed model.

References

1. Cao, Y., Hou, L., Li, J., Liu, Z.: Neural collective entity linking. In: Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, New Mexico, USA, August 2018, pp. 675–686. Association for Computational Linguistics (2018). <https://www.aclweb.org/anthology/C18-1057>
2. Chen, S., Hu, Q.V., Song, Y., He, Y., Wu, H., He, L.: Self-attention based network for medical query expansion, pp. 1–9 (2019)
3. Dredze, M., McNamee, P., Rao, D., Gerber, A., Finin, T.: Entity disambiguation for knowledge base population, pp. 277–285 (2010)
4. Francislandau, M., Durrett, G., Klein, D.: Capturing semantic similarity for entity linking with convolutional neural networks. arXiv: Computation and Language (2016)
5. Gabrilovich, E., Ringgaard, M., Subramanya, A.: FACCC1: Freebase annotation of cluweb corpora. Version 1, 2013 (2013)
6. Ganea, O.E., Hofmann, T.: Deep joint entity disambiguation with local neural attention. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, September 2017, pp. 2619–2629. Association for Computational Linguistics (2017). <https://doi.org/10.18653/v1/D17-1277>. <https://www.aclweb.org/anthology/D17-1277>
7. Guo, Z., Barbosa, D.: Robust named entity disambiguation with random walks. *Sprachwissenschaft* **9**(4), 459–479 (2017)
8. Gupta, N., Singh, S., Roth, D.: Entity linking via joint encoding of types, descriptions, and context, pp. 2681–2690 (2017)
9. Hoffart, J., et al.: Robust disambiguation of named entities in text. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 782–792. Association for Computational Linguistics (2011)
10. Hu, L., Ding, J., Shi, C., Shao, C., Li, S.: Graph neural entity disambiguation. *Knowl. Based Syst.* **195**, 105620 (2020)
11. Kulkarni, S.V., Singh, A.K., Ramakrishnan, G., Chakrabarti, S.: Collective annotation of wikipedia entities in web text, pp. 457–466 (2009)
12. Le, P., Titov, I.: Improving entity linking by modeling latent relations between mentions. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, July 2018 (Volume 1: Long Papers), pp. 1595–1604. Association for Computational Linguistics (2018). <https://doi.org/10.18653/v1/P18-1148>. <https://www.aclweb.org/anthology/P18-1148>
13. Liu, X., Li, Y., Wu, H., Zhou, M., Wei, F., Lu, Y.: Entity linking for tweets, vol. 1, pp. 1304–1311 (2013)

14. Spitzkovsky, V.I., Chang, A.X.: A cross-lingual dictionary for English wikipedia concepts. In: Conference on Language Resources and Evaluation (2012)
15. Usbeck, R., et al.: AGDISTIS - graph-based disambiguation of named entities using linked data. In: Mika, P., et al. (eds.) ISWC 2014. LNCS, vol. 8796, pp. 457–471. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11964-9_29
16. Vaswani, A., et al.: Attention is all you need, pp. 5998–6008 (2017)
17. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks, pp. 7794–7803 (2018)
18. Xin, K., Hua, W., Liu, Yu., Zhou, X.: Entity disambiguation based on parse tree neighbours on graph attention network. In: Cheng, R., Mamoulis, N., Sun, Y., Huang, X. (eds.) WISE 2020. LNCS, vol. 11881, pp. 523–537. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-34223-4_33
19. Xue, M., et al.: Neural collective entity linking based on recurrent random walk network learning. arXiv: Computation and Language (2019)
20. Yamada, I., Shindo, H., Takeda, H., Takefuji, Y.: Joint learning of the embedding of words and entities for named entity disambiguation. In: Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning, Berlin, Germany, August 2016, pp. 250–259. Association for Computational Linguistics (2016). <https://doi.org/10.18653/v1/K16-1025>. <https://www.aclweb.org/anthology/K16-1025>
21. Zuko Gregoric, A., Bachrach, Y., Minkovsky, P., Coope, S., Maksak, B.: Neural named entity recognition using a self-attention mechanism, pp. 652–656 (2017)