An Entity Linking Algorithm Derived from Graph Convolutional Network and Contextualized Semantic Relevance

Bingjing Jia [1] , **Chenglong Wang** [1] , **Haiyan Zhao** [1] **and Lei Shi** [2],*

[1] School of Computer Science, Anhui Science and Technology University, Bengbu 233000, China
[2] State Key Laboratory of Media Convergence and Communication, Communication University of China, Beijing 100024, China
* Correspondence: leiky_shi@cuc.edu.cn

Abstract: In the era of big data, a large amount of unstructured text data springs up every day. Entity linking involves relating the mentions found in the texts to the corresponding entities, which stand for objective things in the real world, in a knowledge base. This task can help computers understand semantics in the texts correctly. Although there have been numerous approaches employed in research such as this, some challenges are still unresolved. Most current approaches utilize neural models to learn important features of the entity and mention context. However, the topic coherence among the referred entities is frequently ignored, which leads to a clear preference for popular entities but poor accuracy for less popular ones. Moreover, the graph-based models face much noise information and high computational complexity. To solve the problems above, the paper puts forward an entity linking algorithm derived from the asymmetric graph convolutional network and the contextualized semantic relevance, which can make full use of the neighboring node information as well as deal with unnecessary noise in the graph. The semantic vector of the candidate entity is obtained by continuously iterating and aggregating the information from neighboring nodes. The contextualized relevance model is a symmetrical structure that is designed to realize the deep semantic measurement between the mentions and the entities. The experimental results show that the proposed algorithm can fully explore the topology information of the graph and dramatically improve the effect of entity linking compared with the baselines.

Keywords: symmetry and asymmetry; collaborative computing; entity linking; graph convolutional network; distilled BERT; contextualized semantic relevance

1. Introduction

It is common that tremendous volumes of textual data spring up on the Web each day. The diverse sources of the data and its large-scale and non-standard presentations pose great challenges for people to collaborate for useful knowledge. Therefore, people urgently hope that computers can understand natural language and process data intelligently to provide knowledge services. In order to describe various entities or concepts existing in the real world, people have begun collaborating to construct knowledge bases (KB), which contain multiple types of the entities, such as organizations, persons, locations, games, movies, positions, etc. Here, Wikipedia [1] and Freebase [2] have drawn wide attention. A Wikipedia page represents an entity, which contains a set of carefully defined relations and attributes. Entity linking (EL) connects the mentions in a text with their corresponding entities in a KB for a better understanding of the text corpus. Figure 1 shows an illustration for the EL task. The plain text "Singapore premiums for Australian kilo bars were quoted unchanged at between 25 45 cents an ounce over spot loco London prices with South Korean origin premiums also steady at 10 20 cents an ounce" has four mentions, which are "Sigapore", "Australian", "London", and "South Korean". In summary, the mention is

ambiguous, which leads to link it to more than one entity in KB. To deal with the problems, the methods used in the previous research mainly focused on designing hand-crafted features to measure the similarity between the mention and the entity in different aspects, including prior popularity [3], context similarity [4], type similarity [5], etc.
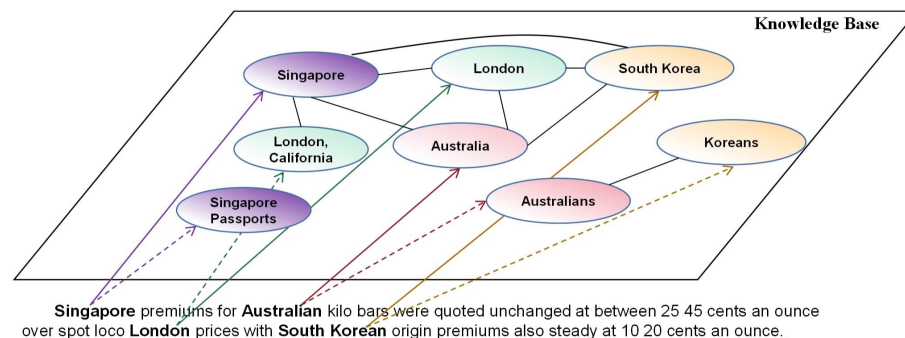

Figure 1. The entity linking task. The mentions in the text are in boldface. The nodes linked by arrowed lines are the candidate entities, where solid lines denote target entities.

Generally, the mentions in the same document are semantically related to each other, and the results of the linking are interactive and dependent. Recently, researchers have tried to construct an entity graph based on KB, and some strategies are used for joint reasoning. Therefore, the EL problem can be cast as an entity ranking problem, and the entity with the highest score is predicted as the correct match. The PageRank and Random Walk algorithm [6–8] usually tend to capture topic consistency between the mentions in the text. With ABACO [9], a sub-graph is extracted from a KB and pruned based on nodes' degree of centrality. Specifically, the algorithm fully considers the Wikipedia page of an entity to compute its semantic similarity with the document topics. The authors in [10] propose to leverage an asymmetric graph convolutional network for entity embeddings, which can integrate global semantic information and latent relation between the entities. KGEL [11] utilizes a knowledge graph to improve the correlation information between the entities. Although traditional graph-based models have achieved significant improvements, they suffer from high computation costs with the increase in the number of candidate entities. In addition, the semantic features implied in context are often ignored due to the data sparsity issue. As the context is composed of words, a large number of studies utilize word embeddings to learn context features [12,13]. For these methods, words are represented by low dimensional vectors in a continuous space, and features for mentions as well as candidate entities are automatically learned from data. They cannot model the topic coherence among various mentions in the text. This paper puts forward the entity linking algorithm derived from asymmetric graph convolutional network and the contextualized semantic relevance (GCNCS), which can utilize neighboring node information and alleviate the problem of excessive noise.

GCNCS differs from the state of the art in how it captures the topic coherence among the various mentions in the entire document. It depends on the construction of the entity graph. For this purpose, the model adopts a distilled version of bidirectional encoder representation from transformers (DistilBERT) to create an asymmetric graph whose nodes are entities and edges. The asymmetric graph is a crucial feature for measuring the distance between the candidate entities. Therefore, our method explores more possibilities between the candidate entities and avoids isolated nodes. The asymmetric graph convolutional network aggregates neighborhoods' features across related entities in a graph by the flexible encoding of entity graphs, resulting in entity embeddings. At the same time, GCNCS takes the contextual information and the prior probability into account. These features collaborate to find the correct target. In a word, the key contributions in this paper are listed as follows:

- A novel strategy for building entity graphs that drastically explores the semantic space consistency among the candidate entities is presented. This not only reduces the

time needed to build an entity graph, but also enhances the coherence of the finally built graph.

- The asymmetric graph convolutional network is used to learn entity embeddings, which improves discriminative signals of the entities by fully exploring the asymmetric structural features of the entity graphs. In addition, the final EL features combine the contextual information and the prior probability as well.
- Experiments with benchmark datasets demonstrate the superior performance of our approach compared with the state-of-the-art EL methods. Our experimental studies also illustrate the influences of the key features.

Here is the main structure of the paper. Section 2 introduces and compares recent EL approaches. The defining of the EL problem starts in Section 3. Section 4 describes the architecture of GCNCS and details the key modules. Section 5 evaluates the proposed model and some baselines, and also measures the roles of various configurations and features. Section 6 summarizes the results obtained through the different approaches and points out the conclusions and the future work.

2. Related Work

Entities are things in the objective world, which are unambiguous and the basic elements of the KB [14,15]. However, the mentions in a text may be ambiguous, which means a mention could denote several different entities in different contexts and an entity may be described by different mentions, which can prevent us from understanding the text properly. Since EL is mainly aiming to address the multiple meanings of the mention and diverse mentions with the same meaning, EL maps the mentions in a document into proper entities in a given KB. In the last several years many EL approaches have emerged, mainly including the collective entity linking method and the individual method. The first deals with all mentions simultaneously, considering the global interdependence between the mentions in the same document. The second covers mentions independently, ignoring the influence among the mentions. Table 1 summarizes the popular entity linking models. Due to limitations of space, "na" and "ctx" denote the mention's surface form and the context, respectively. Correspondingly, "tl" refers to the entity title, "ds" refers to the entity description, "enl" refers to the semantic relatedness of the entities, "pr" refers to the prior popularity, and "cg" refers to the entity category. The symbol ✓ indicates that the corresponding model uses the feature, while ✗ means the feature is not used.

Table 1. Comparative analysis of popular entity linking models.

Category	Model	Input	Knowledge Base	Mention		Entity				
				na	ctx	tl	ds	enl	pr	cg
Individual entity linking	DBpedia Spotlight [16]	document	DBpedia Wikipedia	✓	✓	✓	✓	✗	✓	✗
	CNNContex [17]	document	Wikipedia	✓	✓	✓	✗	✗	✗	✓
	MemNet(C+L) [18]	document	Wikipedia	✓	✓	✓	✓	✗	✗	✗
	MPME [19]	document	Wikipedia	✓	✓	✓	✓	✗	✓	✗
Collective entity linking	AIDA [20]	document	Yago Wikipedia	✗	✓	✗	✓	✓	✓	✗
	Babelfy [21]	document	BabelNet	✗	✗	✗	✗	✓	✗	✗
	L2R-WNED [6]	document	Wikipedia	✗	✓	✗	✓	✓	✗	✗
	Deep-ed [22]	document	Yago Wikipedia	✗	✓	✓	✗	✗	✓	✗
	NCEL [23]	document	Wikipedia	✓	✓	✓	✓	✓	✓	✗
	SeqGAT [24]	document	Wikipedia	✓	✓	✓	✓	✓	✓	✗

2.1. Individual Entity Linking

Early studies show that individual entity linking mainly depends on the similarity between the mention's context and the entity description. Bunescu et al. [25] were the first to select the entity category and description from Wikipedia. Moreover, they ranked the entities based on the support vector machine, picking the one with the highest probability.

Zheng et al. [26] tried to use learning to rank models in EL task, including Ranking Perceptron [27] and ListNet [28]. Their experiments showed that the learning to rank models brought about better results than classification models, and ListNet obtained the highest accuracy. However, those models mostly exploited hand-engineered features to compute context similarities, ignoring the semantic relations among words. DBpedia Spotlight [16] not only annotated text documents automatically, but also linked mentions to DBpedia based on various features such as the prior popularity, the context similarity, and the topic coherence.

Deep learning can learn multiple levels of distributed representations from the given documents and KBs, which pushes EL models. He et al. [5] utilized stacked denoising auto-encoders to learn the input document representation, but ignoring the entity description made the effect not obvious. Francis [29] used different kinds of topic information from multiple granularities. Vector representations were produced with CNNs, and then combined with sum pooling. CNNContex [17] leveraged the semantic representations of the mention, the context, and the entity, and embedded the positions of the context words to capture the distance between the context word and the mention. Memory networks [30] can convert the input into the internal features, and select the key features by jointly using reasoning components and long-term memory components. MemNet [18] could seek important information based on an attention mechanism and two external memories. Therefore, the proposed method had the ability to interact with the memory multiple times, and learn complex functions with multiple levels of composition. TypeCoAtt [4] was a type-aware co-attention model, which read entity type information to improve the co-attention mechanism. The model designed by Zhang et al. [31] for incomplete knowledge base contained three parts, which were encoder, co-attention, and decoder. Recently, mapping unstructured text and a structured knowledge base into the same semantic space has drawn increasing attention. Yamada et al. [32] jointly learned the embedding of words and entities by extending the skip-gram model. Particularly, the generated embeddings were combined with traditional EL features, and the GBRT was employed to learn to rank. Subsequently, MPME [19] considered that the same semantic space is affected by the ambiguity of mentions, and it was necessary to learn various sense embeddings for each mention. EAT [33] took into account anchor texts for representing words and entities in a unique space, which skipped the extra alignment strategy.

2.2. Collective Entity Linking

Graph structure denotes the complex relationship between things in the objective world, which provides new ideas for EL. Han et al. [34] proposed a collective inference algorithm based on the referent graph which contained the mentions and the entities. GLOW [35] treated the EL task as an optimization problem on the basis of local and global variants and analyzed the advantages and the disadvantages of them. AIDA [20] tackled EL as NP-hard optimization problems by enriching the weighted graph with more local features. Therefore, the best joint mention–entity mapping was approximated by computing a dense subgraph. Alhelbawy et al. [36] ranked nodes based on the PageRank algorithm and selected the final result with initial confidence. The accuracy of 85.79% on the AIDA dataset indicated the effectiveness of this algorithm. Guo et al. [6,37] considered that most approaches were not suitable for less popular entities. The constructed graph was based on semantic relatedness, which could overcome the feature sparsity issue. Babelfy [21] used BabelNet to create semantic signatures, and exploited random walks with restarts to weight the network's edges. PPRSim [38] combined local and global features based on Personalized PageRank to filter out noise brought about by incorrect entities.

Specifically, Zwicklbauer et al. [39] not only integrated semantic entity and entity-context embeddings into a graph, but also introduced a topic node to represent all unambiguous and already-disambiguated mentions. Nguyen et al. [12] introduced a model which benefitted the EL task by jointly learning the local context similarities and the topic-related features. Globerson et al. [40] observed that a non-salient entity only depended

on a small subset of mentions. The proposed multi-focal attention could capture some coherence information for local and pairwise scores. Ganea et al. [22] used differentiable inference on the combination of entity embeddings, an adaptive local score, and contextual attention. Yang et al. [41] could perform a better local search from past and future global information based on Beam Search with a gold path. Gradient tree boosting was first employed to optimize entity assignments for all the mentions. Because of noise, data sparsity, and incomplete knowledge base, Phan et al. [42,43] considered that not all mentions were densely related to each other. Pair-linking strategies were presented to iteratively select the pair with the highest confidence at each step of decision making. Le et al. [44–46] modeled multiple relations between mentions, and treated them as latent variables. What is more, unlabeled texts and the multi-instance matching method were used to create initial noisy labels. Then, the noise detection component could identify and exclude the wrong entities. Yamada et al. [47] utilized the new masked entity prediction to randomly mask entities when learning contextualized embeddings of words and entities based on BERT.

Recently, graph neural networks have made great progress [48,49]. The low dimensional vectors generated by graph representation learning contain rich semantic information of node and edge [50]. This also provides a promising research direction for future EL methods [51,52]. NCEL [23] took various features and entity graphs into the asymmetric graph convolutional network to explore structural information among entities, which improved the computation efficiency by performing on a subgraph of adjacent mentions. RLEL [53] considered the global linking as a sequence decision problem, and the reinforcement learning optimized the results based on the linked mentions. Following the models above, SeqGAT [24] introduced BERT to learn local features, which were the input of the graph attention network to capture the topic coherence of the mentions. GNED [54] applied graph convolutional network on the entity–word graph to generate enhanced entity embeddings, which were fed to a CRF for collective EL. Deng et al. [10] generated entity embeddings based on graph convolutional network for combining global semantic information and latent relation between the entities. In addition, multi-hop attention was used to improve the representation of the mention context. DGCN [55] utilized a dynamic graph convolutional network to capture various features, which benefited the quality of EL. CoGCN [56] leveraged contexts to enhance the entity representation. Xue et al. [57] introduced random-walk layers to model the semantic interdependence between the entities from external knowledge.

Some challenges still remain in the previous-introduced models. On the one hand, individual methods mainly depend on the similarity between the mention context and entity description, which ignores the relations between entities. On the other hand, most collective methods rely on the graph structure, which uses the PageRank algorithm to rank entities. However, this may lead to the problem of not making full use of neighboring node information and excessive noise in the graph. In addition, graph representation learning can leverage the structural information of nodes to optimize node vectors, and asymmetric graph convolutional network iteratively updates node representations by aggregating neighboring nodes and edges. This provides a new way to solve the problems above. Therefore, GCNCS is proposed based on the asymmetric graph convolutional network and the contextualized semantic relevance.

3. Preliminaries

3.1. Entity Linking

The EL task can be formalized by the five-tuple $T = <d, M, K, E, f>$, where $d \in D$ represents a document and K denotes background KB. d contains zero or more mentions. Assume the set of the identified mentions is $M = \{m_1, m_2, m_3, \cdots, m_i\}$. $E = \{E_1, E_2, E_3, \cdots, E_i\}$ denotes all the possible entities referred to by M. The goal is to find the correct entity for a mention based on relevant features. Otherwise, it will return NIL. This can be described by formula: $f : M \times K \to E \cup NIL$. For this study, the premise is that the mentions have been recognized in advance. For each mention $m_i \in M$, an entity

set $E_i = \{e_{i1}, e_{i2}, e_{i3}, \cdots, e_{ij}\}$ can be generated according to the proposed method in [58], where j denotes the number of possible entities of m_i. To get a better linking strategy f, the key is to construct an entity graph. Assume $G = (v, \varepsilon)$ is the corresponding weighted entity graph of d, where $v = \{(m_i, e_{i,j}) | \forall m_i \in M, \forall e_{i,j} \in E_i)\}$ stands for the set of the nodes and ε represents edges between the nodes. To enhance the semantic information in the graph, undirected edges are drawn between two entities of the different mentions. Let $relationship(v_p, v_q)$ denote the relatedness between $v_p = (m_i, e_{i,j})$ and $v_q = (m_t, e_{t,l})$. Therefore, $relationship(v_p, v_q) > 0$ indicates v_p and v_q are relevant, and the weight of the edge is $relationship(v_p, v_q)$. The more accurate the edge weight is, the better the EL effect is.

3.2. Entity Graph

Graphs can intuitively describe the complex relationships between the entities, which are divided into four categories such as unweighted undirected graphs, unweighted directed graphs, weighted undirected graphs, and weighted directed graphs. Figure 2a indicates that the two nodes of an edge can be swapped, but not in Figure 2b. The weighted edges in Figure 2c,d capture the relatedness of two nodes. The bigger the weight is, the more closely related the nodes are. This section explores the graph construction from the different views.
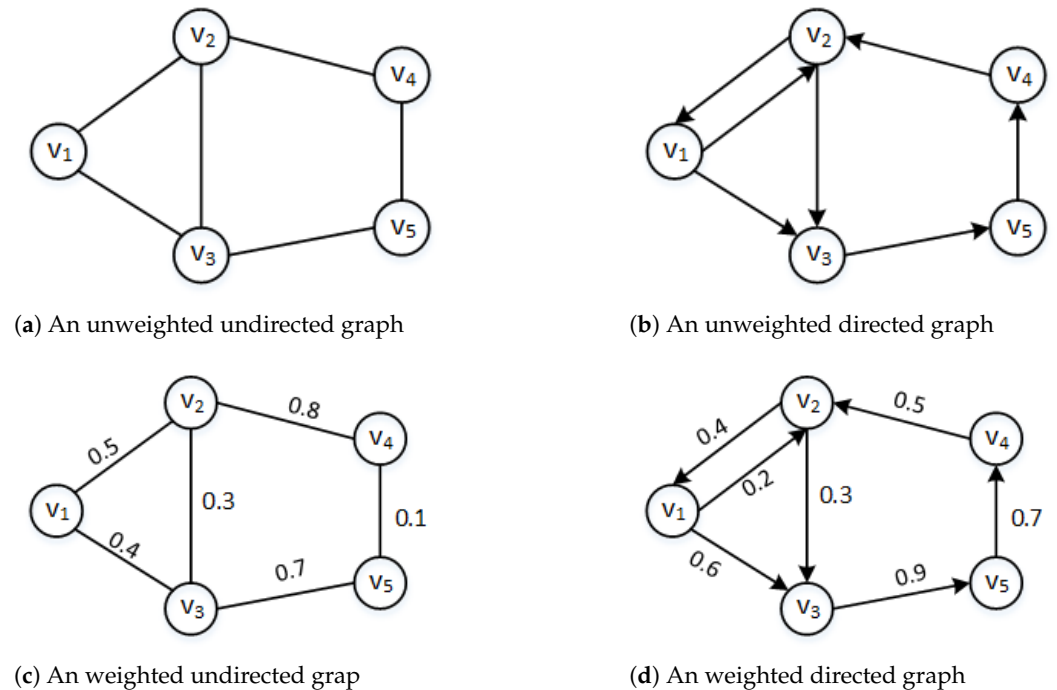

(**a**) An unweighted undirected graph   (**b**) An unweighted directed graph

(**c**) An weighted undirected grap   (**d**) An weighted directed graph

Figure 2. Example of graphs.

3.2.1. Normalized Google Distance-Based Entity Graph

Normalized Google distance (NGD) measures relatedness between the entities on the basis of Google's search results. Suppose the terms are w_1 and w_2. $f(w_1)$ and $f(w_2)$ represent the search results for the individual term, while $f(w_1, w_2)$ indicates the search results for the pair of terms. Then the relatedness between the terms can be measured using the formula:

$$NGD(w_1, w_2) = \frac{max\{log f(w_1), log f(w_2)\} - log f(w_1, w_2)}{log T - min\{log f(w_1), log f(w_2)\}} \tag{1}$$

where T denotes total search engine indexes. The combination of NGD and Wikipedia can make the relatedness between entities more accurate and complete. For $e_{i,j}$ and $e_{t,l}$, the relatedness is computed through the following formula.

$$rls(e_{i,j}, e_{t,l}) = 1 - \frac{log(max(|E_j|, |E_l|)) - log(|E_j \cap E_l|)}{log(|W|) - log(min(E_j, E_l))} \tag{2}$$

where E_j and E_l are the set of entities which can link to $e_{i,j}$ and $e_{t,l}$ correspondingly, and W is the set of all entities in Wikipedia. The edge weight $relationship(v_p, v_q)$ between v_p and v_q is replaced by $rls(e_{i,j}, e_{t,l})$. Moreover, the higher value reflects the stronger relatedness, and the maximum value is 1. For the example given in Figure 1, the edge weights are computed through NGD, and the corresponding entity graph is illustrated in Figure 3.
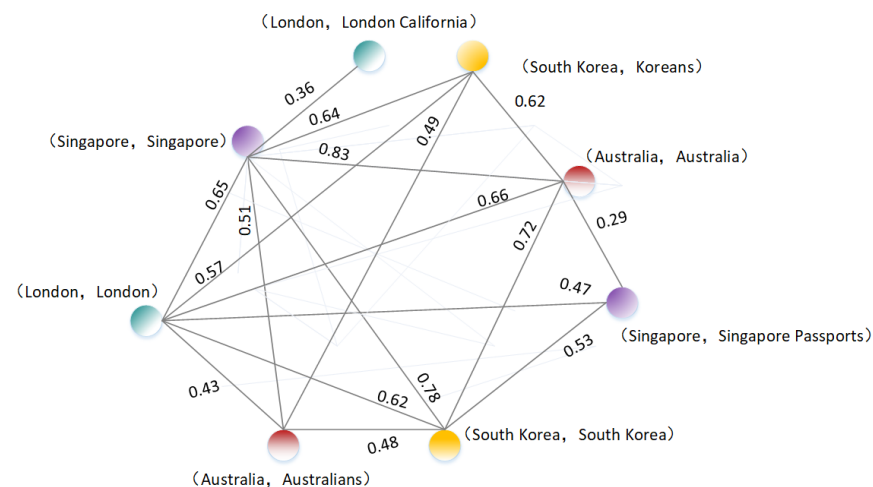

Figure 3. The entity graph based on NGD.

3.2.2. Link-Based Entity Graph

The rich structure in the background KB provides additional information. Take Wikipedia as an example, its nodes correspond to the entities and its edges connect the pairs of the entities that are semantically related. Based on this, we can construct graphs within the following constraints: (1) hyperlinks exist between the Wikipedia articles of two entities; (2) there is an article with hyperlinks to both entities within a window of 500 words. When both of the conditions above are satisfied, it can be assumed that there is a strong semantic relatedness between the entities, and the edge weight is set to 1. Therefore, the link-based entity graph corresponding to Figure 1 is shown in Figure 4.
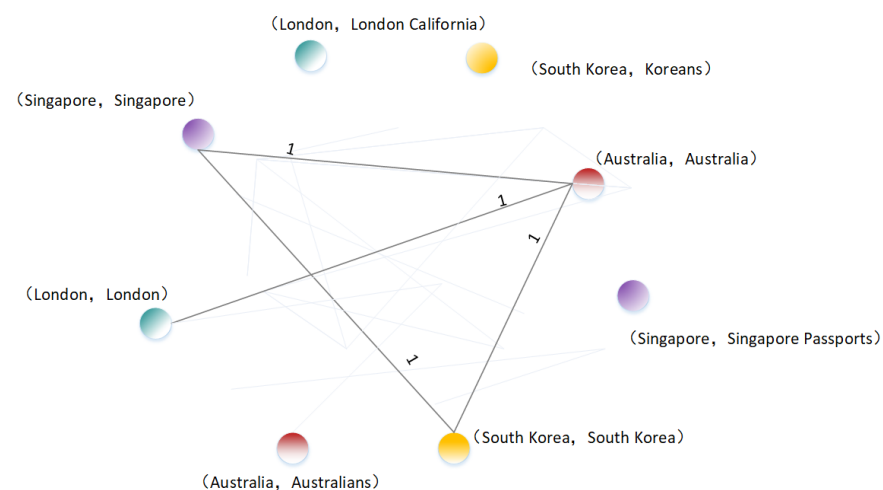

Figure 4. The link-based entity graph.

3.2.3. Embedding-Based Entity Graph

Both NGD and link-based methods depend on statistics, which leads to computational complexity. To reinforce semantic information of entities, it is essential to learn the entity embedding based on DistilBERT [59]. DistilBERT becomes a smaller, faster, and lighter model by reducing the number of layers, which can be finetuned with good performances. Compared to other variants of BERT, DistilBERT reduces the computation time. Different entity contexts can generate different entity embeddings. Therefore, entity embedding is closely related to context information, which can alleviate ambiguous problems. $v(e_{i,j})$ and $v(e_{t,l})$ denote the entity embeddings of $e_{i,j}$ and $e_{t,l}$, respectively. As shown in Formula (3), the semantic relatedness between the entities can be computed with cosine similarity. The edge weight between v_p and v_q is set to $sm(e_{i,j}, e_{t,l})$. Therefore, the embedding-based entity graph corresponding to Figure 1 is shown in Figure 5.

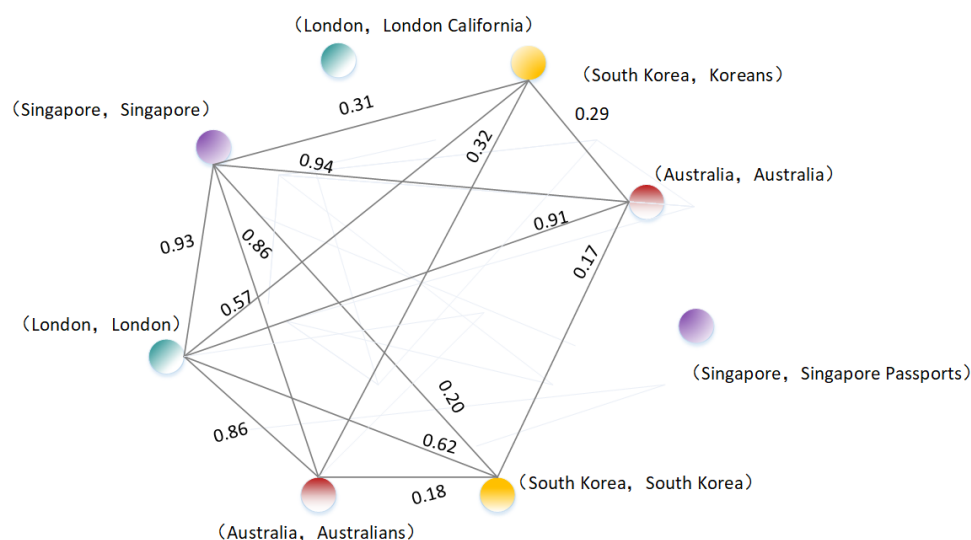$$sm(e_{i,j}, e_{t,l}) = cos(v(e_{i,j}), v(e_{t,l})) \tag{3}$$

Figure 5. Embedding-based entity graph.

4. Proposed Model

EL is challenging due to the name variation and entity ambiguity. However, traditional methods require tedious feature engineering. To extract the semantic relatedness between entities, the graph representation learning generates low dimensional vectors for nodes, which also benefits effective calculation and inference. In order to fully leverage the relations between the entities extracted from the background KB, this paper proposes a model called GCNCS, which combines an asymmetric graph convolutional network and contextualized semantic relevance. Specifically, this model not only exploits the entity–entity topic coherence in a document, but also learns entity embeddings via propagating the semantic information from neighboring entities. In addition, prior popularity and context similarity are still considered. Figure 6 shows our overall structure, which mainly contains three modules: contextualized semantic relevance with bidirectional encoder representation from transformers (BERT), entity embedding with asymmetric graph convolutional network, and entity selector with multiple features. More concretely, the contextualized semantic relevance leverages BERT to encode mention context and entity description, obtaining local semantic relevance. In the entity embedding module, the entity graph represents the possible dependencies between the entities. Therefore, the entity embedding mainly relies on neighboring nodes. To get the best EL result, the entity selector module combines contextualized semantic relevance, prior popularity, and entity embedding.
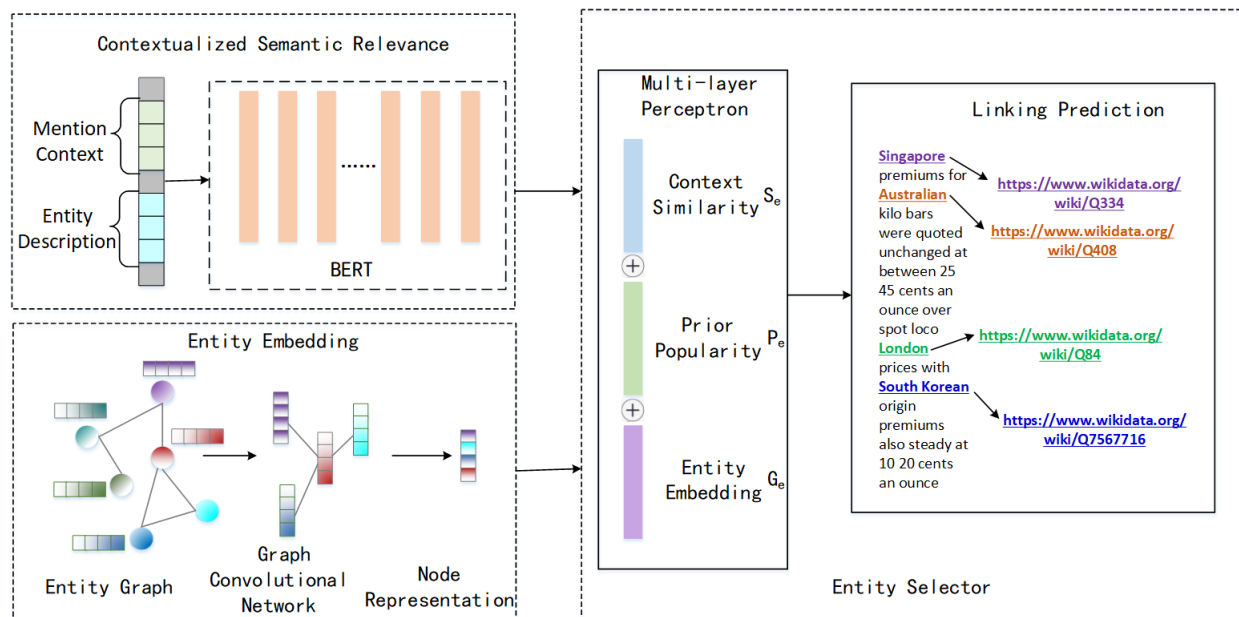
Figure 6. The overall structure of our GCNCS model.

4.1. Contextualized Semantic Relevance with BERT

Since a mention's surface form and entity title are generally short and the superficial characteristics vary relatively widely, EL cannot find sufficient evidence. However, context words may contain important information about a mention. For example, when "qualcomm" or "phone" appears in the surrounding context of mention "apple", it indicates that the referent entity is "Apple Inc." rather than "Apple". Moreover, the background KB provides rich textual data for the entities, which helps the mentions to find the target entity via contextualized semantic relevance.

BERT overcomes long-range dependencies problem caused by LSTM and captures information from multiple perspectives, including lexical, syntactic, and semantic through a large-scale pre-training corpus. The deep bidirectional representations for mention and entity are achieved by fusing the left with the right context. The BERT model is first initialized with the pre-trained parameters, and all of the parameters are fine-tuned according to the needs of the EL task. BERT can learn both single sentence embeddings and sentence pairs embeddings. Here, the mention context and the entity description are concatenated as the input to BERT. Then, the output of the last layer S_e is selected as the context similarity feature.

4.2. Entity Embedding with Asymmetric Graph Convolutional Network

The asymmetric graph convolutional network is applied to graph structure and can deal with data in a non-Euclidean space. Therefore, the topic coherence is achieved among the mentions within a document, and the node embeddings are iteratively updated. Note that the initialized state vectors of all nodes are node features or node attributes. Then, new node embeddings are generated by the combination of neighborhood aggregation and their own features. The main idea is the fusion of neighborhoods' embeddings into node embeddings. Let $G = (v, \varepsilon)$ be an undirected entity graph and let $v = \{v_1, \cdots, v_n\}$ be a set of nodes, where n represents the number of nodes. Note that a $n \times n$ adjacent matrix A can represent the aforementioned entity graph, where the edge weight a_{ij} is computed using a link-based method or embedding-based method. Moreover, the features of each node are initialized with entity embeddings learned by DistilBERT. The input set of node features is $\{h_1^0, \cdots, h_n^0\}$, and the target output set is $\{h_1^L, \cdots, h_n^L\}$ in the L-th layer. Suppose h_i^l represents the embeddings of node i in the l-th layer, which is gained by aggregating

the node embeddings from the (l-1)-th layer and its neighboring nodes. The formula is as follows:

$$h_i^l = \sigma(\sum_{j=1}^{n} a_{ij} W^{(l)} h_j^{(l-1)} + b^{(l)}) \tag{4}$$

where $W^{(l)}$ and $b^{(l)}$ denote weights and bias in the l-th layer, and σ is a non-linear activation. After numerious graph convolutions, the final node features can be acquired, which means the corresponding entity embeddings are represented by G_e.

4.3. Entity Selector with Multiple Features

The prior popularity P_e is a statistical feature of entities, which represents the probability of mapping a mention to an entity. The entity with the highest prior popularity feature is considered to be linked correctly. The contextualized semantic relevance S_e denotes the matching between the mentions and the entities at the semantic level, taking their background into account. The entity embedding G_e exploits link structure and dependencies between the entities, which is also an intuitive mapping of topic coherence among all mentions within a document. Therefore, in order to improve the effectiveness of EL, it is necessary to consider a combination of the prior popularity, the contextualized semantic relevance, and the entity embedding. This paper suggests an entity selector for mapping the mentions correctly. For each mention m and its entity e, the final score function is defined in Equation (5).

$$V_{m,e} = S_e \oplus P_e \oplus G_e \tag{5}$$

where $\oplus$ denotes vector concatenation. The size of each vector dimension is the same. The concatenated vectors are then fed into a multi-layer perceptron, and which is also used as the input of a softmax to predict the similarity between the mention m and the entity e. The details of this process are given in Equations (6) and (7).

$$f^l = Relu(W^l f^{l-1} + b^l) \tag{6}$$

$$sim(m, e) = softmax(f^l) \tag{7}$$

where f^l is the output of the last layer of MLP, and f^{l-1} is the output of the $(l-1)$th hidden layer. W^l and b^l are trainable parameters and the bias. The entity with the highest similarity score is regarded as the target mapping entity.

5. Experiment

5.1. Experimental Settings

5.1.1. Datasets

For the purposes of evaluation of GCNCS, datasets with different properties are used, including AIDA [20], MSNBC [60], AQUAINT [61], ACE04 [35], and CWEB [6]. Our model is trained on AIDA, which contains 1393 news documents. AIDA is much larger than most EL datasets, and the average number of mentions per document is 19.9. The other four datasets are used to test the generalization ability of each model. MSNBC collects 20 documents from 10 different domains, and the average number of mentions per document is 32.8. AQUAINT annotates 50 documents, and the average number of mentions per document is 14.5. ACE04 contains 35 documents, and the average number of mentions per document is 7.1. The three datasets above have many popular entities, and their EL results mainly rely on prior probability. There are still challenges for less popular entities. Therefore, Guo et al. [6] created a new dataset, CWEB, by mixing documents with different levels of difficulty. CWEB is a subset of ClueWeb, and it has 320 documents. Its

average number of mentions per document is 34.8. Figure 7 illustrates the prior popularity distribution over the entities in the test datasets.
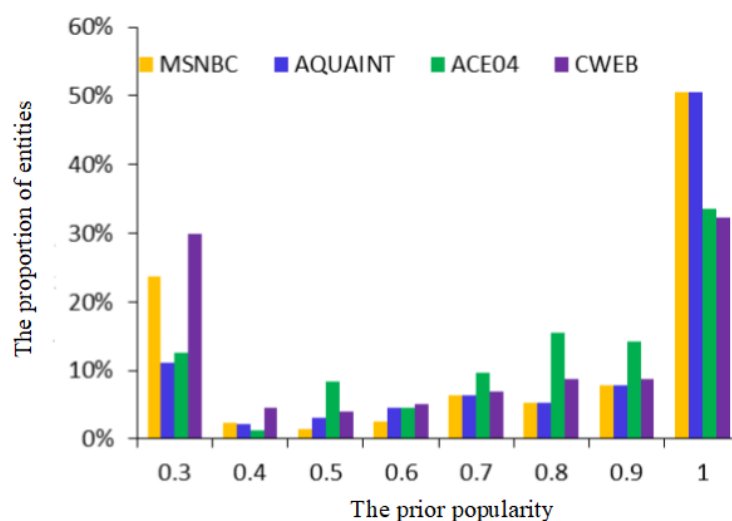

Figure 7. The prior popularity distribution over entities.

5.1.2. Parameters

A Wikipedia dump on 7 April 2016 is used as the target knowledge base. The entity embeddings are initialized by using $DISTILBERT_{BASE}$, which has 6 layers and 6 heads. To learn the contextualized semantic relevance, the pre-trained uncased $BERT_{BASE}$ is utilized, which has 12 heads and 768 hidden states. In addition, the entity graph is constructed based on embeddings. The number of asymmetric graph convolutional network layers is set to 3, the learning rate is 1e-3, and the MLP consists of 3 layers.

5.1.3. Complexity Analysis

The module of contextualized semantic relevance gets the context similarity by fine-tuning BERT. Therefore, the complexity of GCNCS is mainly determined by the entity embedding module. Traditional graph-based models have a high time complexity. A well-known approach, proposed by Hoffart et al. [20], has a complexity of $O(k^3 n^3)$, where k denotes the average number of mentions per document. GCNCS has a lower complexity of $O(T * k^2 n^2)$, where T is the number of asymmetric graph convolutional network layers.

5.2. Evaluation Metric

Following the previous research [4,62], GCNCS only considers non-NIL mentions which correspond to correct entities in KB. Therefore, micro-averaged accuracy is selected for evaluation, which only concerns the correctly linked mentions. The definition of the evaluation metric is given in Equation (8).

$$MicA = \frac{|correctly\ linked\ mentions|}{|total\ number\ of\ all\ mentions|} \tag{8}$$

5.3. Result and Discussion

This section details the results of GCNCS by comparing them with the state-of-the-art graph-based models and the prior popularity method. These models are briefly introduced as follows:

Prioris derived from Wikipedia entity hyperlinks, and can easily find the most likely entity to a mention, which relies on a direct ranking of the entities.

RI [63] leverages external knowledge from Wikipedia to extract relations between the mentions. Note that this method builds relation constraints and performs reasoning.

AIDA [20] integrates popularity priors, similarity measures, and coherence into a framework. Dense subgraphs are constructed to approximate the most promising linked entities. However, this greedy algorithm is time-consuming.

DoSeR [39] exploits the semantic embeddings by constructing a disambiguation graph twice. In addition, this method does not depend on any KB, and is more suitable for large datasets.

Table 2 presents the overall performance of various models on each dataset. Compared with Prior, GCNCS improves the MicA by 0.02 on MSNBC, 0.07 on AQUAINT, 0.1 on ACE04, and 0.04 on CWEB. GCNCS significantly outperforms all graph-based models on ACE04 and CWEB due to the short documents and the lower prior popularity feature. The results reveal the superiority of GCNCS for unpopular entities. It also means that our model not only captures deep semantic features between the mentions and the entities, but also generates the entity embeddings containing more neighbors and asymmetric graph structure information in the entity graph.

Table 2. The MicA comparison of various baselines with GCNCS.

Model	MSNBC	AQUAINT	ACE04	CWEB
Prior	0.89	0.83	0.84	0.70
AIDA [20]	0.79	0.56	0.80	0.58
RI [63]	0.90	0.90	0.86	0.68
DoSeR [39]	0.91	0.84	0.91	-
GCNCS	0.91	0.90	0.94	0.74

5.4. Impact of Different Modules

To explore the effects of contextualized semantic relevance and entity graph, we analyzed the variants of GCNCS in two aspects.

(1) Influence of contextualized semantic relevance

GCNCS mainly utilizes BERT to learn deep semantic features from the mention's context and the entity description. However, The bi-directional long short-term memory network (BiLSTM) is widely used in common NLP tasks, which can capture both previous and future contextual semantic information. To evaluate the performance of different computations for the contextualized semantic relevance, BERT is compared with BiLSTM in two ways. On the one hand, neglecting the entity embedding feature and the prior popularity feature, we just use BiLSTM and BERT to learn the contextualized semantic relevance. On the other hand, GCNBL replaces BERT in GCNCS for contextualized semantic relevance with BiLSTM. Comparative results in Table 3 illustrate that the models based on BERT obtain better performance on all the datasets. All in all, MicA has increased by 0.14 on average. In particular, GCNCS has achieved about 0.1 MicA improvement on average, indicating that BERT can obtain more valuable information from the mention's context and entity description. Apart from it, the results also show that even with a powerful language model, MicA is only around 0.6 if we only leverage the local semantic feature. Therefore, it is necessary to combine global features, such as the prior popularity feature and entity embedding feature.

Table 3. The MicA comparison of variants of GCNCS with different computations for the contextualized semantic relevance.

Model	MSNBC	AQUAINT	ACE04	CWEB
BiLSTM	0.56	0.44	0.62	0.48
BERT	0.74	0.57	0.77	0.58
GCNBL	0.78	0.77	0.82	0.71
GCNCS	0.91	0.90	0.94	0.74

(2) Influence of entity graph

The entity graph is the basis for graph-based EL models. Various graph construction methods reveal the strength of relatedness between entities from different perspectives.

The more accurate the weights of the edges between the entities are, the better the linking performance is. Since the Normalized Google distance-based entity graph utilizes the statistical features to compute edge weights, it does not contain semantic information. Therefore, we observe the impacts of the link-based entity graph and embedding-based entity graph on the linking result. In Table 4, GCNLJ applies asymmetric graph convolutional network to link-based graph for the linking inference. GCNEB uses the asymmetric graph convolutional network on an embedding-based graph to find the correct entity. GCNLR and GCNCS are based on GCNLJ and GCNEB, respectively, combining the prior popularity and entity embedding. The results demonstrate that the embedding-based graph can capture more semantic information between the entities. The overall effects are superior to those of the link-based graph.

Table 4. The MicA comparison of variants of GCNCS with different constructions for the entity graph.

Model	MSNBC	AQUAINT	ACE04	CWEB
GCNLJ	0.69	0.56	0.72	0.52
GCNEB	0.83	0.78	0.84	0.65
GCNLR	0.84	0.84	0.90	0.67
GCNCS	0.91	0.90	0.94	0.74

6. Conclusions

EL is beneficial for people to understand any text, due to the ambiguity of mentions. Existing models often ignore the coherence among entities and focus more on popular entities. To overcome the limitations mentioned above, this paper proposes GCNCS, which collaboratively combines the entity embeddings, prior popularity, and context similarity. Specifically, an entity graph is constructed to denote the dependencies among entities, indicating the topic coherence of mentions. Then, the asymmetric graph convolutional network produces entity embeddings by propagating the semantic information from neighboring entities. Moreover, the contextualized semantic relevance module has a symmetrical structure, which encodes mention context and entity description to achieve local semantic relevance based on BERT. Compared with the state of the art, GCNCS successfully finds the correct entity by employing local and global features. Experimental results demonstrate that GCNCS has superior performance in terms of MicA on four datasets.

Our study not only provides a new research perspective on EL, but is also the foundation of natural language understanding, which can create hyperlinks to Wikipedia for any kind of input text. Take search queries as an example. EL extends entity awareness in Web and news searches by understanding and expanding phrases that refer to semantic types.

Although some key issues have been addressed in this paper, there are still some limitations. GCNCS is on the basis of Wikipedia. It works well for news articles and other general-domain texts on the Web, but not for domain-specific texts such as research papers or novels. This is because Wikipedia can provide textual descriptions and relational information for the general-domain entities. The domain-specific entities only have a short descriptive sentence. In the future, we plan to explore EL tasks on multimodal data. The image information is helpful for finding correct entities. Entity recognition is the basis for EL. Combining the two tasks will enforce the effect of them.

References

1. Vrandečić, D.; Krötzsch, M. Wikidata: A free collaborative knowledgebase. *Commun. ACM* **2014**, *57*, 78–85.
2. Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; Taylor, J. Freebase: A collaboratively created graph database for structuring human knowledge. In Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, Vancouver, BC, Canada, 9–12 June 2008; pp. 1247–1250.
3. Guo, S.; Chang, M.W.; Kiciman, E. To link or not to link? A study on end-to-end tweet entity linking. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Atlanta, GA, USA, 9–14 June 2013; pp. 1020–1030.
4. Nie, F.; Cao, Y.; Wang, J.; Lin, C.Y.; Pan, R. Mention and entity description co-attention for entity disambiguation. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; pp. 5908–5915.
5. He, Z.; Liu, S.; Li, M.; Zhou, M.; Zhang, L.; Wang, H. Learning entity representation for entity disambiguation. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Sofia, Bulgaria, 4–9 August 2013; pp. 30–34.
6. Guo, Z.; Barbosa, D. Robust named entity disambiguation with random walks. *Semant. Web* **2018**, *9*, 459–479.
7. Xie, T.; Wu, B.; Jia, B.; Wang, B. Graph-ranking collective Chinese entity linking algorithm. *Front. Comput. Sci.* **2020**, *14*, 291–303.
8. Lamurias, A.; Ruas, P.; Couto, F.M. PPR-SSM: personalized PageRank and semantic similarity measures for entity linking. *BMC Bioinform.* **2019**, *20*, 1–12.
9. Rama-Maneiro, E.; Vidal, J.C.; Lama, M. Collective disambiguation in entity linking based on topic coherence in semantic graphs. *Knowl. Based Syst.* **2020**, *199*, 105967.
10. Deng, Z.; Li, Z.; Yang, Q.; Liu, Q.; Chen, Z. Improving Entity Linking with Graph Networks. In *International Conference on Web Information Systems Engineering*; Springer: Cham, Switzerland, 2020; pp. 343–354.
11. Li, Q.; Li, F.; Li, S.; Li, X.; Liu, K.; Liu, Q.; Dong, P. Improving Entity Linking by Introducing Knowledge Graph Structure Information. *Appl. Sci.* **2022**, *12*, 2702.
12. Nguyen, T.H.; Fauceglia, N.R.; Rodriguez-Muro, M.; Hassanzadeh, O.; Gliozzo, A.; Sadoghi, M. Joint Learning of Local and Global Features for Entity Linking via Neural Networks. In Proceedings of the COLING, Osaka, Japan, 13–16 December 2016; pp. 2310–2320.
13. Gupta, N.; Singh, S.; Roth, D. Entity Linking via Joint Encoding of Types, Descriptions, and Context. In Proceedings of the EMNLP, Copenhagen, Denmark, 7–11 September 2017; pp. 2681–2690.
14. Tang, Y.; Pedrycz, W. Oscillation-bound estimation of perturbations under Bandler-Kohout subproduct. *IEEE Trans. Cybern.* **2021**, *52*, 6269–6282.
15. Tang, Y.; Pedrycz, W.; Ren, F. Granular symmetric implicational method. *IEEE Trans. Emerg. Top. Comput. Intell.* **2021**, *6*, 710–723.
16. Mendes, P.N.; Jakob, M.; García-Silva, A.; Bizer, C. DBpedia spotlight: Shedding light on the web of documents. In Proceedings of the 7th International Conference on Semantic Systems, Graz, Austria , 7–9 September 2011; pp. 1–8.
17. Sun, Y.; Lin, L.; Tang, D.; Yang, N.; Ji, Z.; Wang, X. Modeling Mention, Context and Entity with Neural Networks for Entity Disambiguation. In Proceedings of the IJCAI, Buenos Aires, Argentina, 10–13 August 2015; Volume 15, pp. 1333–1339.
18. Sun, Y.; Ji, Z.; Lin, L.; Wang, X.; Tang, D. Entity disambiguation with memory network. *Neurocomputing* **2018**, *275*, 2367–2373.
19. Cao, Y.; Huang, L.; Ji, H.; Chen, X.; Li, J. Bridge text and knowledge by learning multi-prototype entity mention embedding. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, BC, Canada, 1–4 August 2017; pp. 1623–1633.
20. Hoffart, J.; Yosef, M.A.; Bordino, I.; Fürstenau, H.; Pinkal, M.; Spaniol, M.; Taneva, B.; Thater, S.; Weikum, G. Robust disambiguation of named entities in text. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, Edinburgh, UK, 27–29 July 2011; pp. 782–792.
21. Moro, A.; Raganato, A.; Navigli, R. Entity linking meets word sense disambiguation: A unified approach. *Trans. Assoc. Comput. Linguist.* **2014**, *2*, 231–244.
22. 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; Association for Computational Linguistics, Copenhagen, Denmark, 7–9 September 2017; pp. 2619–2629.
23. Cao, Y.; Hou, L.; Li, J.; Liu, Z. Neural Collective Entity Linking. In Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, NM, USA, 20–26 August 2018; pp. 675–686.
24. Fang, Z.; Cao, Y.; Li, R.; Zhang, Z.; Liu, Y.; Wang, S. High quality candidate generation and sequential graph attention network for entity linking. In Proceedings of the Web Conference, Taipei , Taiwan, 20–24 April 2020; pp. 640–650.

25. Bunescu, R.; Pasca, M. Using encyclopedic knowledge for named entity disambiguation. In Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics, Sydney, Australia, 17–21 July 2006; 117–126.

26. Zheng, Z.; Li, F.; Huang, M.; Zhu, X. Learning to link entities with knowledge base. In Proceedings of the Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Los Angeles, CA, USA, 2–4 June 2010; pp. 483–491.

27. Shen, L.; Joshi, A.K. Ranking and reranking with perceptron. *Mach. Learn.* **2005**, *60*, 73–96.

28. Cao, Z.; Qin, T.; Liu, T.Y.; Tsai, M.F.; Li, H. Learning to rank: From pairwise approach to listwise approach. In Proceedings of the 24th International Conference on Machine Learning, Corvalis, OR, USA, 20–24 June 2007; pp. 129–136.

29. Francis-Landau, M.; Durrett, G.; Klein, D. Capturing semantic similarity for entity linking with convolutional neural networks. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Berlin, Germany, 7–12 August 2016; pp. 1256–1261.

30. Sukhbaatar, S.; Weston, J.; Fergus, R. End-to-end memory networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 1–9.

31. Zhang, S.; Lou, J.; Zhou, X.; Jia, W. Entity Linking Facing Incomplete Knowledge Base. In *International Conference on Web Information Systems Engineering*; Springer: Cham, Switzerland, 2018; pp. 325–334.

32. 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 26th International Conference On Computational Linguistics, Osaka, Japan, 11–16 December 2016; pp. 250–259.

33. Moreno, J.G.; Besançon, R.; Beaumont, R.; D'hondt, E.; Ligozat, A.L.; Rosset, S.; Tannier, X.; Grau, B. Combining word and entity embeddings for entity linking. In Proceedings of the European Semantic Web Conference, Vienna, Austria, 21–25 October 2017; Springer: Cham, Switzerland, 2–5 May 2017; pp. 337–352.

34. Han, X.; Sun, L.; Zhao, J. 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, Beijing, China, 24–28 July 2011; pp. 765–774.

35. Ratinov, L.; Roth, D.; Downey, D.; Anderson, M. Local and global algorithms for disambiguation to wikipedia. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, OR, USA, 19–24 June 2011; pp. 1375–1384.

36. Alhelbawy, A.; Gaizauskas, R. Graph ranking for collective named entity disambiguation. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Baltimore, MD, USA, 22–27 June 2014; pp. 75–80.

37. Guo, Z.; Barbosa, D. Robust entity linking via random walks. In Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, Shanghai, China, 3–7 November 2014; pp. 499–508.

38. Pershina, M.; He, Y.; Grishman, R. Personalized page rank for named entity disambiguation. In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, CO, USA, 31 May–5 June 2015; pp. 238–243.

39. Zwicklbauer, S.; Seifert, C.; Granitzer, M. Robust and collective entity disambiguation through semantic embeddings. In Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, Pisa, Italy, 17–21 July 2016; pp. 425–434.

40. Globerson, A.; Lazic, N.; Chakrabarti, S.; Subramanya, A.; elRingaard, M.; Pereira, F. Collective entity resolution with multi-focal attention. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Paper), Berlin, Germany, 7–12 August 2016; Association for Computational Linguistics: Berlin, Germany, 2016; pp. 621–631.

41. Yang, Y.; Irsoy, O.; Rahman, K.S. Collective entity disambiguation with structured gradient tree boosting. *arXiv* **2018**, arXiv:1802.10229.

42. Phan, M.C.; Sun, A.; Tay, Y.; Han, J.; Li, C. NeuPL: Attention-based semantic matching and pair-linking for entity disambiguation. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, Singapore, 6–10 November 2017; pp. 1667–1676.

43. Phan, M.C.; Sun, A.; Tay, Y.; Han, J.; Li, C. Pair-linking for collective entity disambiguation: Two could be better than all. *IEEE Trans. Knowl. Data Eng.* **2018**, *31*, 1383–1396.

44. 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, 15–20 July 2018; pp. 1595–1604.

45. Le, P.; Titov, I. Distant learning for entity linking with automatic noise detection. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; Association for Computational Linguistics: Florence, Italy, 2019; pp. 4081–4090.

46. Le, P.; Titov, I. Boosting entity linking performance by leveraging unlabeled documents. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 1935–1945.

47. Yamada, I.; Washio, K.; Shindo, H.; Matsumoto, Y. Global Entity Disambiguation with BERT. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Washington, DC, USA, 10–15 July 2022; pp. 3264–3271.

48. Ma, Q.; Fan, Z.; Wang, C.; Tan, H. Graph Mixed Random Network Based on PageRank. *Symmetry* **2022**, *14*, 1678.

49. Zhu, J.; Mao, G.; Jiang, C. DII-GCN: Dropedge Based Deep Graph Convolutional Networks. *Symmetry* **2022**, *14*, 798.

50. Guo, Q.; Xie, H.; Li, Y.; Ma, W.; Zhang, C. Social Bots Detection via Fusing BERT and Graph Convolutional Networks. *Symmetry* **2021**, *14*, 30.

51. Shen, W.; Li, Y.; Liu, Y.; Han, J.; Wang, J.; Yuan, X. Entity linking meets deep learning: Techniques and solutions. *IEEE Trans. Knowl. Data Eng.* **2021**, *Early Access*. https://doi.org/10.1109/TKDE.2021.3117715.

52. Sevgili, Ö.; Shelmanov, A.; Arkhipov, M.; Panchenko, A.; Biemann, C. Neural entity linking: A survey of models based on deep learning. *Semant. Web* **2022**, *13*, 527–570.

53. Fang, Z.; Cao, Y.; Li, Q.; Zhang, D.; Zhang, Z.; Liu, Y. Joint entity linking with deep reinforcement learning. In Proceedings of the The World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 438–447.

54. Hu, L.; Ding, J.; Shi, C.; Shao, C.; Li, S. Graph neural entity disambiguation. *Knowl.-Based Syst.* **2020**, *195*, 105620.

55. Wu, J.; Zhang, R.; Mao, Y.; Guo, H.; Soflaei, M.; Huai, J. Dynamic graph convolutional networks for entity linking. In Proceedings of the Web Conference, Taipei, Taiwan, 20–24 April 2020; pp. 1149–1159.

56. Jia, N.; Cheng, X.; Su, S.; Ding, L. CoGCN: Combining co-attention with graph convolutional network for entity linking with knowledge graphs. *Expert Syst.* **2021**, *38*, e12606.

57. Xue, M.; Cai, W.; Su, J.; Song, L.; Ge, Y.; Liu, Y.; Wang, B. Neural collective entity linking based on recurrent random walk network learning. *arXiv* **2019**, arXiv:1906.09320.

58. Jia, B.; Yang, H.; Wu, B.; Xing, Y. Collective entity disambiguation based on hierarchical semantic similarity. *Int. J. Data Warehous. Min. (IJDWM)* **2020**, *16*, 1–17.

59. Adel, H.; Dahou, A.; Mabrouk, A.; Abd Elaziz, M.; Kayed, M.; El-Henawy, I.M.; Alshathri, S.; Amin Ali, A. Improving crisis events detection using distilbert with hunger games search algorithm. *Mathematics* **2022**, *10*, 447.

60. Cucerzan, S. Large-scale named entity disambiguation based on Wikipedia data. In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), Prague, Czech Republic, 28–30 June 2007; pp. 708–716.

61. Milne, D.; Witten, I.H. Learning to link with wikipedia. In Proceedings of the 17th ACM Conference on Information and Knowledge Management, Napa Valley, CA, USA, 26–30 October 2008; pp. 509–518.

62. Jia, B.; Wu, Z.; Zhou, P.; Wu, B. Entity Linking Based on Sentence Representation. *Complexity* **2021**, *2021*, 8895742.

63. Cheng, X.; Roth, D. Relational inference for wikification. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Washington, DC, USA, 18–21 October 2013; pp. 1787–1796.