

Improving Entity Linking in Chinese Domain by Sense Embedding Based on Graph Clustering

Zhao-Bo Zhang (张照博), *Member, CCF*, Zhi-Man Zhong (钟芷漫), *Member, CCF*
Ping-Peng Yuan (袁平鹏), *Senior Member, CCF, Member, ACM, IEEE*, and
Hai Jin* (金海), *Fellow, CCF, IEEE, Life Member, ACM*

*National Engineering Research Center for Big Data Technology and System, Huazhong University of Science and Technology
Wuhan 430074, China*

*Service Computing Technology and System Laboratory, Huazhong University of Science and Technology
Wuhan 430074, China*

Cluster and Grid Computing Laboratory, Huazhong University of Science and Technology, Wuhan 430074, China

School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China

E-mail: zhang_zb@hust.edu.cn; zhongzm@hust.edu.cn; ppyuan@hust.edu.cn; hjin@hust.edu.cn

Received September 16, 2022; accepted January 10, 2023.

Abstract Entity linking refers to linking a string in a text to corresponding entities in a knowledge base through candidate entity generation and candidate entity ranking. It is of great significance to some NLP (natural language processing) tasks, such as question answering. Unlike English entity linking, Chinese entity linking requires more consideration due to the lack of spacing and capitalization in text sequences and the ambiguity of characters and words, which is more evident in certain scenarios. In Chinese domains, such as industry, the generated candidate entities are usually composed of long strings and are heavily nested. In addition, the meanings of the words that make up industrial entities are sometimes ambiguous. Their semantic space is a subspace of the general word embedding space, and thus each entity word needs to get its exact meanings. Therefore, we propose two schemes to achieve better Chinese entity linking. First, we implement an n -gram based candidate entity generation method to increase the recall rate and reduce the nesting noise. Then, we enhance the corresponding candidate entity ranking mechanism by introducing sense embedding. Considering the contradiction between the ambiguity of word vectors and the single sense of the industrial domain, we design a sense embedding model based on graph clustering, which adopts an unsupervised approach for word sense induction and learns sense representation in conjunction with context. We test the embedding quality of our approach on classical datasets and demonstrate its disambiguation ability in general scenarios. We confirm that our method can better learn candidate entities' fundamental laws in the industrial domain and achieve better performance on entity linking through experiments.

Keywords natural language processing (NLP), domain entity linking, computational linguistics, word sense disambiguation, knowledge graph

1 Introduction

Entity linking aims to match mentions from texts to their referent entities, and is also known as entity matching^[1]. This task has a robust auxiliary effect on question answering^[2, 3], text summarization^[4], and

knowledge base completions^[5, 6]. For instance, Li *et al.* proposed an end-to-end entity linking model for questions with a biencoder to jointly perform candidate entity generation and entity ranking in one pass^[2]. Chen *et al.* addressed entity ambiguity by entity types^[3]. At present, English entity linking is relative-

Regular Paper

Special Section in Honor of Professor Kai Hwang's 80th Birthday

The work was supported by the National Natural Science Foundation of China under Grant Nos. 61932004 and 62072205.

*Corresponding Author

©Institute of Computing Technology, Chinese Academy of Sciences 2023

ly mature. However, due to the specific differences between Chinese and English, such as segmentation, capitalization, and polysemy, the studies of English entity linking are not all suitable for Chinese entity linking.

Recently, several studies on Chinese entity linking (EL) have been published, but most of them are based on the general domain, such as several studies on the Chinese Microblog texts or medical domains^[7-10]. However, the inherent complexity of Chinese requires a specific design to overcome the challenges of diversity and ambiguity caused by the direct migration of English methods. Furthermore, in Chinese domains, such as industry, there are also entity nesting and semantic monotony caused by the combination of Chinese multi-granularity and domain characteristics. Therefore, it is necessary to carry out specific improvements of the methods to solve these problems.

A linked string in the text is called a mention. The first step of entity linking is to retrieve a set of plausible candidate entities from the knowledge base for each mention. Next, we need to determine the most appropriate boundaries for the mentions of their corresponding entities. In the industrial domain, entity names are often long and contain many parts with separate meanings. Meanwhile, most of the domain's knowledge graphs are characterized by a high degree of manualism and a small number of entities, resulting in confusing names and inconsistent standards. The contradiction between long mentions and insufficient knowledge inclusions forms a serious entity nesting problem. For example, “Dfvf3000 发电机抗压实验仪表 (Dfvf3000 generator compression test instrument)” in Fig.1 can find “Dfvf3000”, “发电机 (generator)”, “抗压实验 (compression test)”, “仪表 (instrument)”, and other combined entities, but they should be a whole. To figure out the appropriate boundaries, we propose an n -gram based candidate entity generation method with an inverted index. By generating the n -gram variable-length sequence of mentions, we can obtain as many knowledge base entities corresponding to mentions as possible and guarantee accuracy through the subsequent robust ranking schemes.

The second step is to rank the aforementioned candidate entities and obtain the probability of each candidate to select the most probable entity as the linked entity. First, we must choose suitable features for calculating the candidates' hit likelihoods. We comprehensively select three features from the structure and semantics. Structurally, words are their in-

ternal basic semantic units, and their overlap is a feature. Furthermore, Chinese characters' overlap can be used as an auxiliary feature because of the multi-granularity and no spaces. Semantically, the semantic similarity of words from entities is a vital feature. Second, we feed the three features into the XGBoost classifier^[11] for training, and we can get appropriate feature weights to ensure a better assignment.

The semantic similarity calculated by embeddings is the most important among the three features. The traditional schemes in the general domain usually adopt the word vectors as the representations of the entities and the candidates and then calculate their similarity^[8]. However, the word embedding only provides one vector for a polysemous word, regardless of the different senses in different domains and relations between words. This is because word embeddings are based on a large-scale corpus, mostly coming from social media such as news and encyclopedias, rather than a low-resource industrial corpus. For instance, “仪表 (instrument/appearance)” can express a person's appearance in Chinese daily conversations, but it only means a tool for displaying parameters when used in the industrial domain because of the precise requirements of a professional expression. Therefore, the sense of the word is single and unambiguous in the industrial domain, unlike in the general domain with many possibilities.

Traditional word embeddings cannot meet the accuracy requirements for calculating word similarity in the industrial domain due to their inherent ambiguities. Furthermore, customizing word embedding for industrial domains is impractical because of the low-resource corpus and inappropriate word segmentation. The n -gram scheme also considers the inability of mentions to be found in existing word embedding vocabulary.

To better represent words and improve the performance of entity linking in the Chinese industrial domain, we introduce sense embedding to calculate semantic similarity. The sense embedding takes into account contextual and interpretable meanings in the individual scenario and is more suitable for computing associations between entities in the domain. We present a novel sense embedding model, which can dynamically summarize the senses of words by graph clustering and learn the embedding representation of the senses. Specifically, we design two new graph clustering algorithms in sense induction and apply them to generate sense embedding. By introducing sense

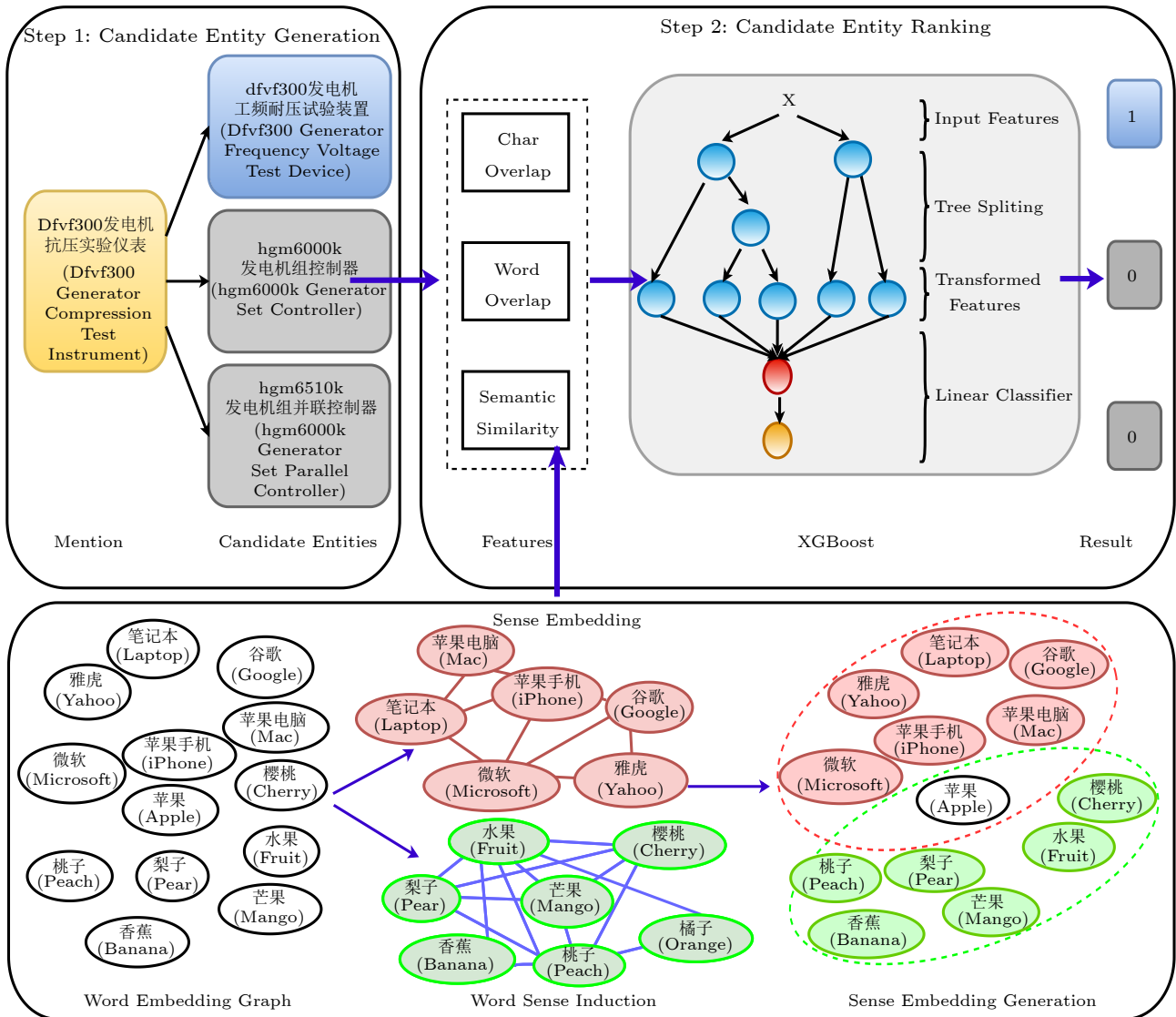


Fig.1. Entity linking model. Target candidate entity is marked in blue with label 1 and the others are in gray with label 0.

embedding, we can provide more accurate and unambiguous word representations in mentions and entities, thereby achieving better entity linking performance.

In summary, we devise a new scheme, namely Sense Enhanced Chinese Entity Linking (SECEL), for the Chinese EL in the industrial domain. First, according to the nesting phenomenon of Chinese industrial entities, we propose an n -gram based candidate entity generation method; then, with word senses determined in the industrial domain represented by sense embedding, which avoids ambiguity, we achieve an accurate and unambiguous candidate entity ranking.

We validate the effectiveness of our sense embedding model on two widely-recognized datasets. More-

over, for Chinese entity linking in industry, we currently find no relevant datasets. Therefore, to test the performance of our scheme, we generate two Chinese industrial datasets in the low-resource industrial domain, integrated with our semi-automated collection of industrial knowledge graphs from projects. Our contributions are as follows.

- We obtain an in-depth discussion on the Chinese entity linking problem in the industrial domain and achieve significant improvements.
- We design a dynamic clustering word sense embedding model with two customized algorithms and apply it to generate semantic similarity features for ranking the candidate entities.
- We customize Chinese knowledge graph and entity linking datasets according to the characteristics

of the industrial domain.

The rest sections are organized as follows. [Section 2](#) provides an overview study of entity linking and sense embedding models. [Section 3](#) introduces the sense embedding model based on graph clustering. [Section 4](#) describes the proposed scheme on Chinese entity linking, and we also describe how to achieve the candidate ranking progress with sense embedding. Experimental description and evaluation results are discussed in [Section 5](#). [Section 6](#) concludes our work and takes a brief introduction to the future work.

2 Related Work

2.1 Entity Linking

Moro *et al.* unified entity linking and word sense disambiguation (WSD) to a graph-based approach^[12] and coupled the loose candidate meaning identification and the densest subgraph heuristic method to choose high-coherence semantic interpretations. Khosrovian *et al.* utilized the Python package, GENSIM^[13], to represent entities and words, mapped them into the same continuous embedding space, and employed LSTM (long short-term memory)^[14] and an attention mechanism for entity disambiguation^[15, 16]. Lei *et al.* proposed a solution based on a knowledge graph for entity linking and question entity discovery, and implemented a scheme based on TF-IDF weighting and word embedding to measure the lexical and semantic similarity between two strings^[17]. Inan and Dikenelli designed a two-fold neural mode that extracts easy mention-entity pairs, and used domain information as a filter to improve performance^[18]. Logeswaran *et al.* presented the zero-shot entity linking task to realize the robust transfer from the general domain to the highly specialized domains with only description text^[19]. Chen *et al.* proposed a lightweight neural method for biomedical domain entity linking with a fraction of the parameters of a BERT model and much fewer computing resources^[20]. Li *et al.* bridged keywords and sequential information captured by a self-attention mechanism by implementing a highway network framework^[6].

2.2 Representation of Sense Embedding

2.2.1 Knowledge-Based Models

These models usually use external resources such as HowNet^[21] and WordNet^[22] to define the sense inventory. Pilevvar and Collier proposed DeConf^[23]

which extracts words that can effectively represent the semantics of each synset from the WordNet’s semantic network through graph-based operations. Some models use ontology as an external resource^[24, 25]. Scarlini *et al.* proposed SENSEMBERT, which couples the expression ability of language modeling with semantic network knowledge to obtain multilingual high-quality semantic representation^[26]. Eyal *et al.* designed a method for word sense induction based on pre-trained masked language models to tag the corpus with the sense based on the sense lists derived from the corpus^[27].

2.2.2 Unsupervised Models

These models usually cluster the context or the semantic network to obtain sense inventory. Nee-lakantan *et al.* proposed a model based on context clustering^[28], which clusters the context vectors and predicts the meaning of a word as the cluster closest to its context vector. A new sense will be generated when the similarity between the existing cluster center of the word and the context vector is less than a threshold. The context vectors and sense vectors are continuously updated in the iteration. Pelevina *et al.* proposed a model based on semantic network clustering^[29]. This model constructs an ego network based on word semantics and uses the Chinese Whispers algorithm to cluster nodes in the ego network to achieve word sense induction. Finally, the average of all word vectors in each cluster is taken as the sense vector. DIVE is also a model based on semantic network clustering^[30], but it uses the spectral clustering algorithm. Han and Shirai took collocations into account to determine a sense of a given word, and defined a dependency collocation, which is a syntactic dependency relation between a target word and another word in a sentence^[31].

3 Learning Sense Embedding

We aim to verify that sense embedding can enhance accuracy when ranking entities, and thus a more convenient unsupervised approach based on graph clustering is designed to generate representations. Here, sense embedding can be summarized as two steps: word sense induction and sense embedding generation. Specifically, sense induction clusters the adjacent words of the target word to obtain sense clusters; sense embedding generation integrates the aforementioned clusters to specific vector representations.

3.1 Word Sense Induction

The word embeddings learn the semantic information of different senses and synthesize them into a single vector. This will cause adjacent words with different senses to cluster around a target word. For instance, we take the classic ambiguous word “苹果 (apple)” to select the similar words according to the distance between the word vectors. There are not only words such as “桃子 (peach)” that belong to the “水果 (fruit)” sense, but also words such as “苹果电脑 (Mac)” that belong to the “公司产品 (company and product)” sense. The two-dimensional distribution of the corresponding word vectors is shown in Fig.2.

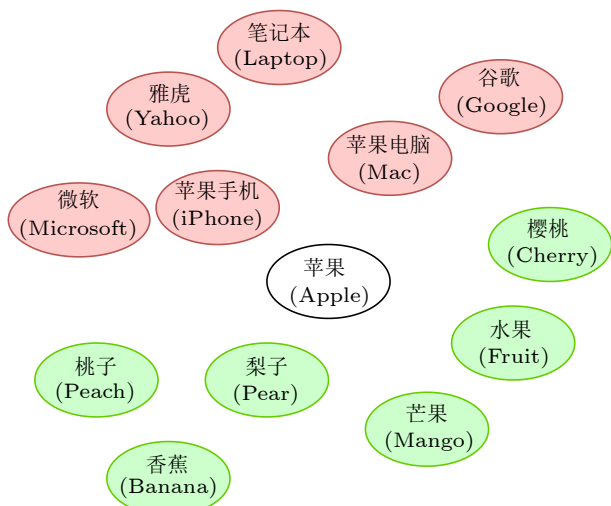


Fig.2. Two-dimensional distribution.

Word sense induction refers to detecting and listing the possible senses of a word automatically. First, we model the sense distribution of a word as a graph, meaningfully expressing the semantic relationship between the senses of the word. Then, we implement the graph clustering algorithms to make the words representing a similar sense form an independent sense cluster.

Our clustering is similar to performing community search^[32] on a word graph rather than on a social network. The nodes are selected according to the criteria of the ego network^[29], but the weights of the edges are cosine similarities calculated from the word vectors. The required graph consists of a target word, the nodes the target word connects with semantic strengths above a threshold, and the edges between these nodes.

Based on the constructed graph, we realize two graph clustering algorithms that can determine how

many clusters to generate automatically and improve the original algorithm to make it more suitable for the scene of word sense induction. The Semantic-Clique Percolation Method (SCPM) can divide the clusters by identifying the k -cliques on the graph and merge these clusters by percolation operation^[33]. The Weak Flow Filtered Markov Clustering (WFF-MCL) algorithm randomly walks in the Markov network to strengthen the strong flow and weaken the weak flow until the weak flow is eliminated. The clusters will be generated by the part connected by the strong flow^[34].

3.1.1 Semantic-Clique Percolation Method

The k -Clique Percolation method^[33] can find the k -clique, which refers to a complete subgraph with k nodes according to the topological relationship. In order to adapt this algorithm to weighted graphs, we consider not only topological relations but also incorporate semantic strength (similarity), and we define the semantic-clique based on the k -clique. The semantic-clique refers to a complete subgraph that the semantic similarity between two nodes is greater than a threshold θ . On this basis, we present SCPM, which identifies the semantic-clique to form the sense clusters.

We first identify neighboring maximal complete subgraphs (MCS, the red dotted triangle in Fig.3) as semantic-cliques in the aforementioned graph, and then combine these semantic-cliques to form naturally overlapping word clusters. Second, we introduce the fixed parameter k , the size to determine the minimum number of nodes for MCSs. k is empirically set to 3 or 4. The smaller the parameter k is, the more sparse the senses of the word are and the larger the number of sense clusters is.

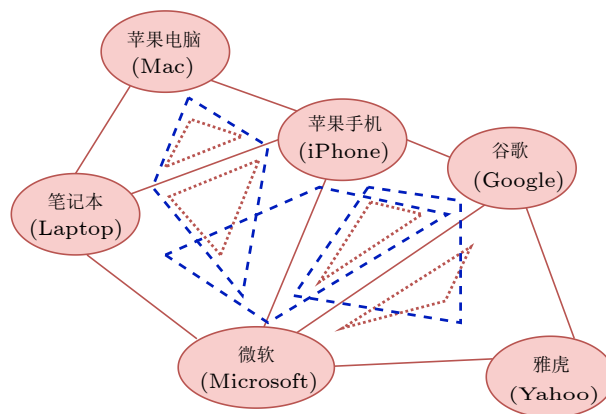


Fig.3. Illustration of a sense cluster.

Here, we define that two semantic-cliques with size k are adjacent if they share $k - 1$ nodes, i.e., if they differ only in a single node (the two triangles included by the blue quadrilateral in Fig.3). Sense clusters are the unions of all adjacent semantic cliques. The specific process to generate sense clusters is provided in Algorithm 1.

Algorithm 1. Semantic-Clique Percolation

Require:

a set of all MCSs C ,
size of the threshold clique k

Ensure:

all semantic-clique percolation clusters P

```

1:  $P \leftarrow \emptyset$                                 ▷ Initialize the output set
2:  $B \leftarrow \text{Zeros}(|C|, |C|)$                 ▷ Initialize a matrix
3: for  $i \leftarrow 0$  to  $|C|$  do
4:   for  $j \leftarrow 0$  to  $|C|$  do
5:      $B[i][j] \leftarrow |C_i \cap C_j|$ 
6:     if  $(i = j) \cup (B[i][j] < k)$  then
7:        $B[i][j] \leftarrow 0$                     ▷ off-diagonal element  $< k$ 
8:     end if
9:     if  $(i \neq j) \cup (B[i][j] < k - 1)$  then
10:       $B[i][j] \leftarrow 0$                     ▷ diagonal element  $< k - 1$ 
11:    end if
12:  end for
13: end for
14:  $P \leftarrow \text{DFS}(B)$ 

```

3.1.2 Weak Flow Filtered Markov Clustering

The Markov clustering algorithm utilizes random walks in the Markov network to strengthen the strong flow and weaken the weak flow until the weak flow is eliminated. Then the clusters will be generated from the connected strong flow^[34]. This algorithm repeats the expansion and inflation operations on the adjacency transition probability matrix M of a graph until convergence. The expansion operation multiplies the power of e by M , and the inflation operation multiplies the power of r by each entry of M . Empirically, e is set to 2, and r can be adjusted. The higher the value of r is, the stronger the penalty for the weak flow is, and the greater the number of the generated clusters is.

We propose WFF-MCL (Weak Flow Filtered Markov Clustering) to accelerate the convergence progress. Since the original algorithm iterates many times to make the matrix's weak terms all converge to 0, the accelerating key is to eliminate the weak flow as soon as possible. We introduce a filter operation before the expansion and inflation operation to judge the relevance between nodes. When the seman-

tic similarity between nodes is less than the filter factor ω (a statistical value), the random walk probability between the two nodes is directly set to 0. Thus, it is equivalent to removing some weak flow in advance, which can reduce the calculation cost. All operations and details are formed in Algorithm 2.

Algorithm 2. Weak Flow Filtered Markov Clustering

Require:

the non-full connected graph G ,
expansion factor e ,
inflation factor r

Ensure:

all Markov clusters P

```

1:  $M \leftarrow \text{Adj}(G)$                                 ▷ Initialize adjacency matrix
2: for  $i \leftarrow 0$  to  $|G.nodes|$  do
3:    $M[i][i] \leftarrow 1$ 
4:    $M[:,i] \leftarrow \text{Normalization}(M[:,i])$ 
5:   for  $j \leftarrow 0$  to  $|G.nodes|$  do
6:     if  $M[j][i] < \omega$  then
7:        $M[j][i] \leftarrow 0$                     ▷ weak flow filter
8:     end if
9:   end for
10: end for
11: while not  $\text{Convergence}(M)$  do
12:   for  $i \leftarrow 0$  to  $|G.nodes| - 1$  do
13:      $M[:,i] \leftarrow \text{Normalization}(M[:,i])$ 
14:   end for
15:    $M \leftarrow (M)^e$                                 ▷ Expansion
16:   for  $i \leftarrow 0$  to  $|G.nodes|$  do
17:     for  $j \leftarrow 0$  to  $|G.nodes|$  do
18:        $M[i][j] \leftarrow (M[i][j])^r$             ▷ Inflation
19:     end for
20:   end for
21: end while
22:  $P \leftarrow \text{DFS}(M)$ 

```

3.2 Sense Embedding Generation

We have achieved word sense induction through graph clustering so that each sense corresponds to a sense cluster. Thus, sense embedding can be generated from semantically similar words in the clusters. However, the clusters are inevitably laced with many ambiguous words that are entrained with noise. For more precise semantic representation, we filter the words within PageRank^[35] to obtain the sense biasing words.

The words in the cluster are initialized with the same weight, and then any target word's weight is voted by other words. Each surrounding word evenly distributes its weight to the target word, and the target word w sums all the weights passed and obtains a new weight, which can be iteratively calculated according to (1):

$$weight(w) = \frac{1-d}{|S|} + d \sum_{i=1}^{|S|} \frac{weight(w_i)}{L(w_i)}, \quad (1)$$

where $L(w_i)$ is the outgoing degree of the i -th word w_i in the sense cluster S , and d is the damping parameter to adjust the transfer probability (empirically 0.85).

After iterations, we select the top K in the weight ascending ranking as the sense biasing words. Table 1 shows top 5 sense biasing words extracted for two senses of the word “苹果 (apple)”.

Table 1. Top 5 Sense Biasing Words of “苹果 (Apple)”

Sense Biasing Word	
Sense 1	水果 (fruit), 芒果 (mango), 桃子 (peach), 樱桃 (cherry), 梨子 (pear)
Sense 2	苹果手机 (iPhone), 微软 (Microsoft), 苹果电脑 (Mac), 笔记本 (laptop), 谷歌 (Google)

The generation of sense vector \mathbf{v}_{s_i} of the sense s_i requires the simultaneous consideration of two principles: 1) as close as possible to the word vector \mathbf{v}_t ; 2) as close as possible to the above sense biasing words. We set the objective function according to the Euclidean distance limitation in these two aspects, as shown in (2):

$$\arg \min_i \alpha d(\mathbf{v}_{s_i}, \mathbf{v}_t) + \sum_{k=1}^k \beta_k d(\mathbf{v}_{s_i}, \mathbf{v}_k), \quad (2)$$

where α represents the weight of \mathbf{v}_{s_i} close to \mathbf{v}_t , $d(\mathbf{v}_{s_i}, \mathbf{v}_t)$ represents the Euclidean distance between \mathbf{v}_{s_i} and \mathbf{v}_t , and β_k represents the weight of \mathbf{v}_{s_i} close to the word vector \mathbf{v}_k of the k -th sense biasing word. With an optimization proposed by Bengio^[36] for (2), the sense embedding \mathbf{v}_{s_i} can be obtained as shown in (3):

$$\mathbf{v}_{s_i} = \frac{\alpha \mathbf{v}_t + \sum_{k=1}^k \beta_k \mathbf{v}_k}{\alpha + \sum_{k=1}^k \beta_k}, \quad (3)$$

where we set $\alpha = 1$, and we specify two generation function strategies for the value selection of β_k by taking the average weight or the weight obtained by the aforementioned PageRank algorithm, as shown in (4) and (5) respectively:

$$\mathbf{v}_{s_i} = \left(\mathbf{v}_t + \frac{\sum_{k=1}^k \mathbf{v}_k}{m} \right) / 2, \quad (4)$$

$$\mathbf{v}_{s_i} = \left(\mathbf{v}_t + \sum_{k=1}^k weight(w_k) \mathbf{v}_k / \sum_{k=1}^k weight(w_k) \right) / 2, \quad (5)$$

where m is the number of the sense biasing words, and $weight(w_k)$ represents the the weight of the k -th sense biasing word obtained by PageRank.

4 Sense Enhanced Chinese Entity Linking

In this section, we introduce the n -gram based candidate entity generation scheme and the sense enhanced candidate entity ranking.

4.1 N-Gram Based Candidate Entity Generation

We find that many entities in the industrial domain are long and have the superposition of nouns. To ensure the recall rate in the low-resource industrial domain, we design an n -gram based candidate entity generation method. The n -gram is a basic feature that considers all combinations without missing potential entities^[37]. This method not only expands the range of candidates but also effectively alleviates the out-of-vocabulary problem of long strings in word embedding lookup. It takes n consecutive characters in a sliding window, which guarantees the granularity of extracted mentions from characters to long entities.

This method is mainly divided into two steps. First, we establish an inverted index lookup table of the n -gram sequences and their corresponding entities containing n -gram sequences in the knowledge base. Then, we extract the n -gram sequences of the mentions and match them with the inverted indexes to obtain a candidate entity set, as shown in Fig.4 and Algorithm 3.

To construct the inverted index lookup table, we exhaust the n -gram sequences of entities in the knowledge base, where n ranges from 2 to the entity's length. This results in many entities having some common n -gram sequences. For example, the 2-gram sequence “苹果 (Apple)” is included in “苹果手机 (Apple iPhone)” and “苹果无线耳机 (Apple AirPods)”. Following the intuition that candidate entities with more minor differences in length from mentions usually have higher credibility, we group the entities with common n -gram sequences by length and fine-tune the algorithm so that entities with higher length overlap are retrieved first. As shown in the first step in Fig.4(a), we group entities containing str_1 sequences by length p_1 to p_m .


Second, we slide the windows to extract the n -gram of the mentions to match the inverted index

Step 1: Construct Inverted Index

n -Gram (Entities)	entity_id
str_1	p_1 : e1, e2, e3 ⋮ p_m : e10, e13
⋮	⋮
str_n	p_1 : e5 ⋮ p_m : e6, e3

(a)

Mention:
(m1, m2, m3, m4, m5)



Step 2: Match Dictionary

n	n -Gram(Mentions)	Matched	Candidates
5	(m1, m2, m3, m4, m5)	\emptyset	\emptyset
4	(m1, m2, m3, m4)	str_1	e1, e2, e3
	(m2, m3, m4, m5)	...	e4, e5
3	(m1, m2, m3)	...	e6, ..., e10
	(m2, m3, m4)	\emptyset	\emptyset
	(m3, m4, m5)	...	e11, ..., e13
2	(m1, m2)	...	e14, ..., e18
	(m2, m3)	\emptyset	\emptyset
	(m3, m4)	\emptyset	\emptyset
	(m4, m5)	...	e19, ..., e23

(b)

Fig.4. n -gram based candidate entity generation. e_i : entity i , $1 \leq i \leq 23$; m_j : mention j , $1 \leq j \leq 5$.

and obtain corresponding candidate entities. For efficiency, the search length of the mentions is from large to small, because the longer the n -gram is, the greater the overlap with the target entity is, which means the successful probability is higher, as shown in Fig.4(b).

Algorithm 3. N -Gram Based Candidate Entity Generation
Require:

constructed inverted index I ,
 mention character sequence M ,
 length difference threshold θ_1 ,
 threshold for the number of candidates θ_2

Ensure:

candidate entity set O

```

1:  $O \leftarrow \emptyset$  ▷ Initialize the output set
2:  $N \leftarrow |M|$  ▷ The length of the mention
3: for  $i \leftarrow N$  to 1 do
4:   for  $j \leftarrow 0$  to  $N - i + 1$  do
5:      $C \leftarrow I[M[j : j + i]] \triangleright \{\text{len:ents}\}$  for  $M[i : j]$ 
6:     for  $p, E$  in  $C$  do
7:       if  $\text{abs}(N - p) > \theta_1$  then
8:         Break
9:       end if
10:      for  $e$  in  $E$  do
11:        if  $|O| < \theta_2$  and  $e \notin O$  then
12:          Add  $e$  into  $O$ 
13:        end if
14:      end for
15:    end for
16:  end for
17: end for
18: return  $O$ 

```

We also set two thresholds to avoid the proliferation of candidates. The first threshold is the difference between the length of the mention and the candidates. We filter out pairs with a large length difference because of the proportional relationship between characters' overlap and credibility. The second threshold is the number of overall candidates to avoid short mentions matching too many candidates.

We use characters instead of words to generate n -grams because of the non-interval characteristic of Chinese. Compared with the model based on words^[10], our method can generate more mention-entity pairs for a high recall rate, and the inverted index can improve the matching efficiency.

4.2 Candidate Entity Ranking

4.2.1 Features for Ranking

In the Chinese entity linking task, the mention and its candidate entities usually contain some common characters or words. Therefore, we select two features to represent the commonalities of structural composition: the char overlap and the word overlap.

Meanwhile, the mention and the target entity have similar semantics. Ma *et al.*^[10] considered using the editing distance of Pinyin as a semantic consideration, but this is not enough. For example, “仪表 (instrument/appearance)” and “装置 (device)” have no phonological correlation, but the sense “仪表 (instrument)” of “仪表” is very close to that of sense “装置 (device)” of “装置”. In most traditional schemes, se-

semantic similarity is computed by word vectors. However, word vectors are static and context-independent, which leads to the fact that the semantic similarity with entities cannot be well computed for ambiguous words appearing in the mentions. On the contrary, sense embedding that incorporates sense-level information can automatically adjust the representation of words according to the context. They can express the senses of the words within the mentions more transparently and play a certain advantage in this task. Therefore, the third feature is set to the semantic similarity calculated by sense vectors.

4.2.2 Word Sense Disambiguation

We can obtain the appropriate sense for each word involved in the computation for a more accurate semantic similarity. The aforementioned sense embedding provides an unsupervised solution for accurately representing the senses of words in mentions. Also, we adopt a hybrid local and global scheme to evaluate the confidence of the sense in a given context, as follows:

$$\arg \max_i global(s_i) \times local(s_i), \quad (6)$$

where s_i represents the i -th sense of the target word, and $global(s_i)$ and $local(s_i)$ represent the global confidence and the local confidence of the i -th sense of the word, respectively.

Local Confidence. The local confidence refers to the probability that each sense of the word may appear in the given context. The correct sense and the context words absolutely have similar semantics. For instance, the sense of “仪表 (instrument/appearance)” in Fig.1 is clearly directed to the “instrument” rather than the “appearance” in the context of “发电机 (generators)”, “抗压实验 (compression test)”. Since the cosine similarity between vectors can measure the semantic similarity, we can calculate the cosine similarity between the context vector and the sense vector as the local confidence of a sense, as shown in (7):

$$local(s_i) = d(\mathbf{v}_{\text{context}}, \mathbf{v}_{s_i}), \quad (7)$$

where $\mathbf{v}_{\text{context}}$ is the context vector, and \mathbf{v}_{s_i} represents the sense vector of the i -th sense of the word.

We use the context words filtered by keywords and the part-of-speech to generate $\mathbf{v}_{\text{context}}$ instead of directly assigning the average vector of all contextual words. For the part-of-speech, we only keep verbs, nouns, and adjectives due to words such as prepositions and adverbs that do not provide semantics.

Keyword filtering is to strengthen the words with discriminative power. We measure the discriminative ability of a contextual word by its highest and lowest similarity with senses, as (8):

$$s(w_j) = \max_m d(\mathbf{v}_{s_m}, \mathbf{v}_j) - \min_i d(\mathbf{v}_{s_n}, \mathbf{v}_j), \quad (8)$$

where s_i is the i -th sense of the target word and s_m (s_n) represents the one that matches the maximum (minimum) similarity with word \mathbf{v}_j after traversing all senses. \mathbf{v}_j represents the word vector corresponding to the j -th word in context words after the part-of-speech filtering. Then, we sort the context words according to the discriminating ability and take the top n words as the keywords. n is obtained by dividing the number of the retained context words by 2, which allows the long text to retain more keywords and the short text to retain fewer keywords.

Global Confidence. Global confidence refers to the probability that each sense of the word in all contexts may occur. The higher the popularity of a sense is, the more the words there are in the corresponding sense cluster. The global confidence of a sense can be calculated by the ratio of the number of the items in the corresponding sense cluster to the sum of the numbers of items in all the sense clusters of the target word, as shown in (9):

$$global(s_i) = \frac{|S_i|}{\sum_1^k |S_i|}, \quad (9)$$

where $|S_i|$ represents the number of the items in the sense cluster of the i -th sense of the word, and k is the number of the sense clusters of the target word. The global confidence can exclude sparse clusters and enhance the robustness of the model.

4.2.3 XGBoost Classifier

We choose XGBoost to train the classifier rather than other advanced models because it focuses on illuminating the importance of features without changing their intrinsic properties. We aim to emphasize that sense embedding can enhance entity linking over word embedding, which is rarely demonstrated on the neural network model. The neural network model iteratively optimizes a large number of parameters, shortening the gap between the word vector and the sense vector, and making their effects converge.

Here, we combine all features into the input vector \mathbf{x}_i and their corresponding results y_i to form a training dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\} (\mathbf{x}_i \in \mathbb{R}^m, y_i \in \mathbb{R})$, which

is inputted into the XGBoost model for training. Here, m is the number of features. A tree ensemble model (shown in Fig.1) uses K additive functions to predict the categories of candidates.

$$\hat{y}_i = \phi(\mathbf{x}_i) = \sum_{k=1}^K f_k(\mathbf{x}_i), \quad f_k \in \mathcal{F},$$

where $\mathcal{F} = \{f(\mathbf{x}) = w_{q(\mathbf{x})}\} (q: \mathbb{R}^m \rightarrow T, w \in \mathbb{R}^T)$ is the regression tree space (also known as CART). Here, q is the structure of each tree that maps the input vector to their corresponding leaf. T is the number of leaves in the tree and w_i represents the weight of the i -th leaf. To learn an accurate model, we need to minimize the following regularized objective in (10):

$$\mathcal{L}(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k), \quad (10)$$

where $\Omega(f) = \gamma T + \frac{1}{2} \lambda \|w\|^2$, l is a differential loss function to measure the difference between the predicted \hat{y}_i and the target label y_i , and Ω is a penalty term to prevent over-fitting. XGBoost can efficiently capture potential classification features and achieve high accuracy performance.

4.2.4 Workflow

First, we construct the inverted index table and retrieve the candidate entities based on the mentions. Second, we segment the mentions and candidate entities to calculate the character and word overlap. Next, each Chinese word vector is extracted from the word embeddings trained with the zhWiki of Chinese Wikipedia Dump corpus^[38], and sense vectors are obtained based on word vectors by the above mentioned method and disambiguated within the context from the mentions. For each disambiguated word in the mentions, the semantic similarities with the candidate entities are calculated respectively and averaged. We adopt two similarity calculation strategies to get scores between the word in the mentions and candidate entities. The first one is the greedy strategy, which takes the highest similarity score between the word in the mentions and each word of the candidate entities. The other is the average strategy, that is, the average similarity between the word and each word of the candidate entity.

Finally, the above three features are normalized and inputted into the XGBoost classifier for training to learn each feature’s weight and predict whether the candidates are true later.

5 Experiments

We divide the experiments into two categories to enhance persuasiveness. First, we separately evaluate the expressive ability of optimized sense embeddings and their disambiguation performance in downstream tasks. Due to the broad applicability of our embedding approach and the lack of industrial corpus, we test on two more general English datasets. We evaluate the semantic expressiveness on the Stanford Context Word Similarity (SCWS) dataset^[39], and apply the Word Sense Disambiguation (WSD) task to verify the performance in a downstream task and highlight the importance of word sense embedding^[40]. Then, we test our model on the entity linking task in the Chinese industrial domain.

5.1 Evaluation on SCWS

This task tests the quality of the constructed sense vectors, that is, whether the features of word senses are learned accurately.

Dataset. For some semantic similarity tasks, such as WordSim-353, the similarity scores of the word pairs are isolated from the context, which does not allow for sense differentiation. Therefore, Huang created SCWS^[39], which has 2 003 word pairs with corresponding contexts and the similarity scores judged by human. This dataset can test the quality of the sense vectors.

Evaluation Metrics. The word pairs in the context are first matched with the sense vectors by (6), and then the cosine similarity between the sense vector pairs is calculated. Finally, the Pearson correlation coefficient is used to measure the correlation between the calculated similarity scores and the human judged scores.

Baselines. We compare our proposed sense embedding model with some baselines, including GloVe^[41], GenSense^[24], and ELMo^[42]. The word embedding model, GloVe, aggregates the local context and global word-word co-occurrence overall statistics from a corpus. The knowledge-based sense embedding model, GenSense, takes three components: semantic relatedness, the relation strength, and the semantic strength to retrofit a generalized model. The context-based dynamic word embedding model, ELMo, pre-trains a language model based on a bidirectional LSTM network to learn context information, which takes advantage on the OOV vocabulary- and character-level representation^[43]. The language model infers the

meaning of each word when a sentence is received as input and generates a dynamic word vector corresponding to each word based on the context.

Discussion of Results. Table 2 shows the performance of the sense embeddings generated by different graph clustering algorithms, clustering factors, and generating functions we mentioned above. WFF-MCL outperforms SCPM in terms of best performance (the best result is in bold). WFF-MCL can achieve the best performance with 3 as the inflation factor and the averaging strategy. In addition, the quality of the sense embeddings constructed by generation function 1 is better than that of generation function 2. Here, the generation functions 1 and 2 correspond to the averaging strategy in (4) and the weighting strategy in (5), respectively. There is a gap of about 1% in the correlation, which shows that the averaging strategy for the sense biasing words is better than the weighting strategy when generating sense vectors.

Table 2. Results of Our Model on SCWS

Algorithm	Factor	Generation	Correlation
SCPM	$k = 3$	1	0.576
		2	0.567
	$k = 4$	1	0.575
		2	0.567
WFF-MCL	$r = 2$	1	0.578
		2	0.567
	$r = 3$	1	0.580
		2	0.569
	$r = 4$	1	0.578
		2	0.568

Note: The highest score is in bold.

Table 3 indicates that our sense embedding model outperforms all baselines. The experimental results validate the improvement of the sense embedding model over the word embedding model GloVe and demonstrate the advantages over other context-aware representation models. Our graph clustering algorithm clearly has more excellent semantic segmentation capabilities than the global co-occurrence but no sense discrimination model, GloVe. Besides, our model essentially includes the features of GenSense, allowing for better semantic induction due to the modified clustering algorithm. For ELMo, the neural network introduces noise when generating sense vectors from all contextual words, which will inevitably lead to semantic deviation. However, our model restricts and filters contextual words when generating sense vectors, which can effectively mitigate the above problems.

Table 3. Results on SCWS

Model	Correlation
GloVe ^[41]	0.533
GenSense-syn ^[24]	0.548
GenSense-ant ^[24]	0.529
GenSense-all ^[24]	0.542
ELMo ^[42]	0.562
SECEL	0.580

5.2 Evaluation on TWSI

This task tests the performance of the constructed sense vectors on a WSD task in the general domain. It indicates that our model works well in the downstream task.

Dataset. The TWSI (Turk Bootstrap Word Sense Inventory) dataset was created by Biemann through crowdsourcing^[40]. Each entry in TWSI contains the target word, the senses, their representative words, and the true sense label. The goal of this task is to identify the correct sense labels of each noun in different contexts, which can be used to test the performance of the sense vectors on the WSD task.

Evaluation Metrics. We need to map the predicted sense tags to all the senses in the standard sense catalog of TWSI. For each clustered sense, we calculate the similarity between the clustering sense embedding from (4) and the representative word embeddings of each sense in the standard sense catalog. When the highest similarity is greater than 0, we determine that the mapping is successful; otherwise it fails. We consider the prediction to be correct if the sense label corresponding to the highest similarity is the same as the true label in TWSI. Based on the number of samples successfully mapped and predicted, we can calculate the precision, recall, and *F1*.

Baselines. We set Random Sense, which randomly chooses senses, as the basic baseline. In addition, we select three sense embedding models that use different clustering algorithms as the comparative models. MSSG is based on the k -means algorithm to achieve semantic clustering^[28]; CWMS is based on the Chinese Whispers algorithm to realize clustering^[29]; and DIVE adopts the spectral clustering algorithm as the core for word sense induction^[30]. Furthermore, we introduce a model (namely, Dependencies) that enhances word sense reduction by labeling senses with hypernyms and images^[31].

Discussion of Results. Table 4 shows that all the sense embedding models achieve significant improve-

Table 4. Results on TWSI (%)

Model	Precision	Recall	F1
Random Sense	53.6	53.4	53.5
MSSG ^[28]	66.2	65.8	66.0
CWMS ^[29]	68.6	68.1	68.4
DIVE ^[30]	67.6	67.2	67.4
Dependencies ^[31]	68.9	68.1	68.5
SCPM	68.9	68.4	68.6
WWF-MCL	70.1	70.0	70.0

ments over Random Sense, demonstrating that sense embedding can play an important role in disambiguation. In the baselines, Dependencies outperforms the other baseline models, and it performs 15% better than Random Sense. Compared with all baseline models, SCPM demonstrates competitive results with the combined global and local confidence in (6). Furthermore, WWF-MCL achieves the best results and has a 1.5% advantage over Dependencies, indicating the effectiveness of our sense embedding for the downstream disambiguation task in the general domain.

5.3 Entity Linking on IEL

The experiment above verified that the sense embedding we designed could effectively realize improvement in both internally semantic capture and externally downstream tasks. Here, we apply our best sense embedding (in Table 2) to entity linking in the Chinese industrial domain. This task tests the impact of using sense embedding on entity linking, and our aim is to verify that sense vectors can significantly improve the performance.

Dataset Construction. Since there is no publicly available Chinese entity linking dataset in the industrial domain, we need to construct a dataset to meet the evaluation requirement. We propose Industrial Entity Linking (IEL), a dataset derived from the knowledge graph of industrial domain scenarios. It is realized by four steps of: 1) selecting some representative entities in the industrial domain knowledge base; 2) transforming expressions of the selected entities, including full names, abbreviations, synonyms, typos, etc., to generate the mentions; 3) providing the context for each mention; and 4) generating the candidate entities and tags for each mention, and combining the corresponding context to generate a sample.

To ensure the diversity of the selected entities, we consider four attributes of the entities: 1) character length; 2) word length; 3) whether the entity is a combination of Chinese and English; and 4) whether

the entity contains ambiguous words. Using a threshold or a ratio for features to extract entities, we randomly select the entities corresponding to each feature value to generate a set of entities. We then merge the entity sets of the four features and remove the duplicates. This ensures coverage of the entities corresponding to each value of each feature. IEL-1 generated by the thresholds has 390 entities and 12 525 samples, while IEL-2 generated by the ratio of 20% has 359 entities and 13 067 samples.

Evaluation Metrics. Entity linking can be considered as a classification task, i.e., selecting the candidate with the highest probability among the candidates as a result. We can evaluate the precision, recall, and F1 of this task to measure performance. The three metrics can be calculated in two ways: a micro approach, assigning the same weight to each sample; and a macro approach, taking into account the differences in each category and giving each category the same weight. In IEL, we generate only one target entity, while the number of candidate entities is variable, leading to an imbalance between different categories. To reduce this imbalance, we choose the macro approach to calculate the metrics.

Baselines. We use the word vector scheme for computing semantic similarity as a baseline. In addition, we test different strategies mentioned in the workflow of Subsection 4.2.4 to find the best way to calculate the similarity between mentions and candidates.

Discussion of Results. Table 5 shows that in either the dataset IEL-1 or IEL-2, whether the average or greedy strategy is used to calculate the similarity feature, the sense vectors compared with the word vectors have a significant improvement of about 4% in precision, recall, and F1. This indicates that a scheme employing sense vectors can indeed lead to improvements over the traditional word vector scheme in the Chinese industrial entity linking task.

Furthermore, greedy similarity calculations can extract the highest scores very well, thus allowing

Table 5. Results on IEL

Dataset	Strategy	Vector	Precision (%)	Recall (%)	F1 (%)
IEL-1	Average	Word	84.52	64.98	73.47
		Sense	88.71	68.35	77.21
	Greedy	Word	85.76	67.17	75.33
		Sense	89.67	69.04	78.01
IEL-2	Average	Word	77.36	75.49	76.41
		Sense	80.85	78.10	79.45
	Greedy	Word	80.96	76.53	78.68
		Sense	83.93	81.85	82.88

representative words to be better highlight features of mention, achieving superior results to average similarity schemes. Also, different datasets result in different datasets. IEL-1 selected according to the threshold is filtered for the four aforementioned features and has some filtering effect. IEL-2, on the other hand, which contains all feature values based on ratio extraction, is more representative, which suggests that IEL-2 must be more noisy. This inherent difference between these two datasets makes the model perform less well in IEL-2 than in IEL-1.

6 Conclusions

We provided a new idea to improve Chinese entity linking in the progress of candidate entity generation and ranking. Specifically, we used n -gram to improve the recall of generation and cluster-based sense embedding to improve the accuracy of semantic computation. Our graph clustering based sense disambiguation method achieves good results on both SCWS and TWSI datasets. Moreover, the overall entity linking model, SECEL, obtains a significant improvement over the traditional scheme on the IEL dataset we constructed.

Experimental results demonstrated the feasibility of our solution for Chinese domain entity linking and showed the advantages of sense embedding for semantic computation. These improvements proved that it makes sense to consider the polysemy of internal words for Chinese entity linking. Therefore, SECEL can be used to accurately express the semantics for entity linking in many domains as well. In the future, we plan to further develop the IEL dataset to make this dataset more representative, and also to improve the SECEL model to make it a high-quality open-sourced project.

References

- [1] Sun C C, Shen D R. Mixed hierarchical networks for deep entity matching. *Journal of Computer Science and Technology*, 2021, 36(4): 822–838. DOI: [10.1007/s11390-021-1321-0](https://doi.org/10.1007/s11390-021-1321-0).
- [2] Li B Z, Min S, Iyer S, Mehdad Y, Yin W T. Efficient one-pass end-to-end entity linking for questions. In *Proc. the 2020 Conference on Empirical Methods in Natural Language Processing*, Nov. 2020, pp.6433–6441. DOI: [10.18653/v1/2020.emnlp-main.522](https://doi.org/10.18653/v1/2020.emnlp-main.522).
- [3] Chen K, Shen G H, Huang Z Q, Wang H J. Improved entity linking for simple question answering over knowledge graph. *International Journal of Software Engineering and Knowledge Engineering*, 2021, 31(1): 55–80. DOI: [10.1142/S0218194021400039](https://doi.org/10.1142/S0218194021400039).
- [4] Amplayo R K, Lim S, Hwang S W. Entity commonsense representation for neural abstractive summarization. In *Proc. the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Jun. 2018, pp.697–707. DOI: [10.18653/v1/N18-1064](https://doi.org/10.18653/v1/N18-1064).
- [5] Shen W, Wang J Y, Han J W. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Trans. Knowledge and Data Engineering*, 2015, 27(2): 443–460. DOI: [10.1109/TKDE.2014.2327028](https://doi.org/10.1109/TKDE.2014.2327028).
- [6] Li M Y, Xing Y Q, Kong F, Zhou G D. Towards better entity linking. *Frontiers of Computer Science*, 2022, 16(2): 162308. DOI: [10.1007/s11704-020-0192-9](https://doi.org/10.1007/s11704-020-0192-9).
- [7] Fu J L, Qiu J, Guo Y L, Li L. Entity linking and name disambiguation using SVM in Chinese micro-blogs. In *Proc. the 11th International Conference on Natural Computation*, Aug. 2015, pp.468–472. DOI: [10.1109/ICNC.2015.7378034](https://doi.org/10.1109/ICNC.2015.7378034).
- [8] Huang D C, Wang J L. An approach on Chinese microblog entity linking combining Baidu encyclopaedia and word2vec. *Procedia Computer Science*, 2017, 111: 37–45. DOI: [10.1016/j.procs.2017.06.007](https://doi.org/10.1016/j.procs.2017.06.007).
- [9] Zeng W X, Tang J Y, Zhao X. Entity linking on Chinese microblogs via deep neural network. *IEEE Access*, 2018, 6: 25908–25920. DOI: [10.1109/ACCESS.2018.2833153](https://doi.org/10.1109/ACCESS.2018.2833153).
- [10] Ma C F, Sha Y, Tan J L, Guo L, Peng H L. Chinese social media entity linking based on effective context with topic semantics. In *Proc. the 43rd Annual Computer Software and Applications Conference*, Jul. 2019, pp.386–395. DOI: [10.1109/COMPSAC.2019.00063](https://doi.org/10.1109/COMPSAC.2019.00063).
- [11] Chen T Q, Guestrin C. XGBoost: A scalable tree boosting system. In *Proc. the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2016, pp.785–794. DOI: [10.1145/2939672.2939785](https://doi.org/10.1145/2939672.2939785).
- [12] Moro A, Raganato A, Navigli R. Entity linking meets word sense disambiguation: A unified approach. *Trans. Association for Computational Linguistics*, 2014, 2: 231–244. DOI: [10.1162/tacl_a_00179](https://doi.org/10.1162/tacl_a_00179).
- [13] Khosrovian K, Pfahl D, Garousi V. GENSIM 2.0: A customizable process simulation model for software process evaluation. In *Proc. the 2008 International Conference on Software Process*, May 2008, pp.294–306. DOI: [10.1007/978-3-540-79588-9_26](https://doi.org/10.1007/978-3-540-79588-9_26).
- [14] Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Computation*, 1997, 9(8): 1735–1780. DOI: [10.1162/neco.1997.9.8.1735](https://doi.org/10.1162/neco.1997.9.8.1735).
- [15] Phan M C, Sun A X, Tay Y, Han J L, Li C L. NeuPL: Attention-based semantic matching and pair-linking for entity disambiguation. In *Proc. the 2017 ACM Conference on Information and Knowledge Management*, Nov. 2017, pp.1667–1676. DOI: [10.1145/3132847.3132963](https://doi.org/10.1145/3132847.3132963).
- [16] Zeng W X, Zhao X, Tang J Y, Tan Z, Huang X Q. CLEEK: A Chinese long-text corpus for entity linking. In *Proc. the 12th Language Resources and Evaluation Conference*, May 2020, pp.2026–2035. DOI: [10.1145/3132847.3132963](https://doi.org/10.1145/3132847.3132963).
- [17] Lei K, Zhang B, Liu Y, Deng Y, Zhang D Y, Shen Y. A knowledge graph based solution for entity discovery and linking in open-domain questions. In *Proc. the 2nd International Conference on Smart Computing and Communi-*

- tion, Dec. 2017, pp.181–190. DOI: [10.1007/978-3-319-73830-7_19](https://doi.org/10.1007/978-3-319-73830-7_19).
- [18] Inan E, Dikenelli O. A sequence learning method for domain-specific entity linking. In *Proc. the 7th Named Entities Workshop*, Jul. 2018, pp.14–21. DOI: [10.18653/v1/W18-2403](https://doi.org/10.18653/v1/W18-2403).
- [19] Logeswaran L, Chang M W, Lee K, Toutanova K, Devlin J, Lee H. Zero-shot entity linking by reading entity descriptions. In *Proc. the 57th Annual Meeting of the Association for Computational Linguistics*, Jul. 2019, pp.3449–3460. DOI: [10.18653/v1/P19-1335](https://doi.org/10.18653/v1/P19-1335).
- [20] Chen L H, Varoquaux G, Suchanek F M. A lightweight neural model for biomedical entity linking. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021, 35(14): 12657–12665. DOI: [10.1609/aaai.v35i14.17499](https://doi.org/10.1609/aaai.v35i14.17499).
- [21] Dong Z D, Dong Q, Hao C L. HowNet and its computation of meaning. In *Proc. the 23rd International Conference on Computational Linguistics: Demonstrations*, Aug. 2010, pp.53–56. DOI: [10.5555/1944284.1944298](https://doi.org/10.5555/1944284.1944298).
- [22] Miller G A. WordNet: A lexical database for English. *Communications of the ACM*, 1995, 38(11): 39–41. DOI: [10.1145/219717.219748](https://doi.org/10.1145/219717.219748).
- [23] Pilehvar M T, Collier N. De-conflated semantic representations. In *Proc. the 2016 Conference on Empirical Methods in Natural Language Processing*, Nov. 2016, pp.1680–1690. DOI: [10.18653/v1/D16-1174](https://doi.org/10.18653/v1/D16-1174).
- [24] Lee Y Y, Yen T Y, Huang H H, Shiu Y T, Chen H H. GenSense: A generalized sense retrofitting model. In *Proc. the 27th International Conference on Computational Linguistics*, Aug. 2018, pp.1662–1671.
- [25] Ramprasad S, Maddox J. CoKE: Word sense induction using contextualized knowledge embeddings. In *Proc. the 2019 Spring Symposium on Combining Machine Learning with Knowledge Engineering*, Mar. 2019.
- [26] Scarlini B, Pasini T, Navigli R. SensEmBERT: Context-enhanced sense embeddings for multilingual word sense disambiguation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, 34(5): 8758–8765. DOI: [10.1609/aaai.v34i05.6402](https://doi.org/10.1609/aaai.v34i05.6402).
- [27] Eyal M, Sadde S, Taub-Tabib H, Goldberg Y. Large scale substitution-based word sense induction. In *Proc. the 60th Annual Meeting of the Association for Computational Linguistics*, May 2022, pp.4738–4752. DOI: [10.18653/v1/2022.acl-long.325](https://doi.org/10.18653/v1/2022.acl-long.325).
- [28] Neelakantan A, Shankar J, Passos A, McCallum A. Efficient non-parametric estimation of multiple embeddings per word in vector space. In *Proc. the 2014 Conference on Empirical Methods in Natural Language Processing*, Oct. 2014, pp.1059–1069. DOI: [10.3115/v1/D14-1113](https://doi.org/10.3115/v1/D14-1113).
- [29] Pelevina M, Arefiev N, Biemann C, Panchenko A. Making sense of word embeddings. In *Proc. the 1st Workshop on Representation Learning for NLP*, Aug. 2016, pp.174–183. DOI: [10.18653/v1/W16-1620](https://doi.org/10.18653/v1/W16-1620).
- [30] Chang H S, Agrawal A, Ganesh A, Desai A, Mathur V, Hough A, McCallum A. Efficient graph-based word sense induction by distributional inclusion vector embeddings. In *Proc. the 12th Workshop on Graph-Based Methods for Natural Language Processing*, Jun. 2018, pp.38–48. DOI: [10.18653/v1/W18-1706](https://doi.org/10.18653/v1/W18-1706).
- [31] Han S Z, Shirai K. Unsupervised word sense disambiguation based on word embedding and collocation. In *Proc. the 13th International Conference on Agents and Artificial Intelligence*, Feb. 2021, pp.1218–1225. DOI: [10.5220/0010380112181225](https://doi.org/10.5220/0010380112181225).
- [32] Chen H H, Jin H. Finding and evaluating the community structure in semantic peer-to-peer overlay networks. *Science China Information Sciences*, 2011, 54(7): 1340–1351. DOI: [10.1007/s11432-011-4296-6](https://doi.org/10.1007/s11432-011-4296-6).
- [33] Gao W, Wong K F, Xia Y Q, Xu R F. Clique percolation method for finding naturally cohesive and overlapping document clusters. In *Proc. the 21st International Conference on Computer Processing of Oriental Languages*, Dec. 2006, pp.97–108. DOI: [10.1007/11940098_10](https://doi.org/10.1007/11940098_10).
- [34] Gibbons T R, Mount S M, Cooper E D, Delwiche C F. Evaluation of BLAST-based edge-weighting metrics used for homology inference with the Markov Clustering algorithm. *BMC Bioinformatics*, 2015, 16: 218. DOI: [10.1186/s12859-015-0625-x](https://doi.org/10.1186/s12859-015-0625-x).
- [35] Brin S, Page L. Reprint of: The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 2012, 56(18): 3825–3833. DOI: [10.1016/j.comnet.2012.10.007](https://doi.org/10.1016/j.comnet.2012.10.007).
- [36] Yoshua B, Olivier D, Nicolas Le R. Label propagation and quadratic criterion. *Semi-Supervised Learning*, 2006: 192–216. DOI: [10.7551/mitpress/9780262033589.003.0011](https://doi.org/10.7551/mitpress/9780262033589.003.0011).
- [37] Serban O, Castellano G, Pauchet A, Rogozan A, Pecuchet J P. Fusion of smile, valence and N-Gram features for automatic affect detection. In *Proc. the 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*, Sept. 2013, pp.264–269. DOI: [10.1109/ACII.2013.50](https://doi.org/10.1109/ACII.2013.50).
- [38] Jin H, Zhang Z B, Yuan P P. Improving Chinese word representation using four corners features. *IEEE Trans. Big Data*, 2022, 8(4): 982–993. DOI: [10.1109/TBDATA.2021.3106582](https://doi.org/10.1109/TBDATA.2021.3106582).
- [39] Huang E H, Socher R, Manning C D, Ng A Y. Improving word representations via global context and multiple word prototypes. In *Proc. the 50th Annual Meeting of the Association for Computational Linguistics*, Jul. 2012, pp.873–882.
- [40] Biemann C. Turk bootstrap word sense inventory 2.0: A large-scale resource for lexical substitution. In *Proc. the 8th International Conference on Language Resources and Evaluation*, May 2012, pp.4038–4042.
- [41] Pennington J, Socher R, Manning C. GloVe: Global vectors for word representation. In *Proc. the 2014 Conference on Empirical Methods in Natural Language Processing*, Oct. 2014, pp.1532–1543. DOI: [10.3115/v1/D14-1162](https://doi.org/10.3115/v1/D14-1162).
- [42] Ilić S, Marrese-Taylor E, Balazs J A, Matsuo Y. Deep contextualized word representations for detecting sarcasm and irony. In *Proc. the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, Oct. 2018, pp.2–7. DOI: [10.18653/v1/w18-6202](https://doi.org/10.18653/v1/w18-6202).
- [43] Liu Y J, Che W X, Wang Y X, Zheng B, Qin B, Liu T. Deep contextualized word embeddings for universal dependency parsing. *ACM Trans. Asian and Low-Resource Language Information Processing*, 2020, 19(1): 9. DOI: [10.1145/3326497](https://doi.org/10.1145/3326497).



Zhao-Bo Zhang is currently pursuing his Ph.D. degree in computer science from Huazhong University of Science and Technology, Wuhan. His research interests include Chinese natural language processing and knowledge representation.



Zhi-Man Zhong received her M.S. degree in computer science from Huazhong University of Science and Technology, Wuhan. Her research focuses on question answering and word embedding.



Ping-Peng Yuan is a professor in the School of Computer Science and Technology at Huazhong University of Science and Technology, Wuhan. He received his Ph.D. degree in computer science from Zhejiang University, Hangzhou. His research interests include databases, knowledge representation and reasoning and natural language processing, with a focus on high-performance computing. During exploring his research, he implements systems and innovative applications in addition to investigating theoretical solutions and algorithmic design. Thus, he is the principle developer of multiple system prototypes, including TripleBit, PathGraph and SemreX.



Hai Jin is a chair professor of computer science and engineering at Huazhong University of Science and Technology, Wuhan. Jin received his Ph.D. degree in computer engineering from Huazhong University of Science and Technology, Wuhan, in 1994. In 1996, he was awarded a German Academic Exchange Service Fellowship to visit the Technical University of Chemnitz, Straße der Nationen. Jin worked at The University of Hong Kong, Hong Kong, between 1998 and 2000, and as a visiting scholar at the University of Southern California, Los Angeles, between 1999 and 2000. He was awarded Excellent Youth Award from the National Science Foundation of China in 2001. Jin is a CCF Fellow, IEEE Fellow, and a life member of ACM. He has co-authored 22 books and published over 900 research papers. His research interests include computer architecture, virtualization technology, distributed computing, big data processing, network storage, and network security.