Proceedings of Machine Learning Research vol 245:1–8, 2024 2024 International Conference on Machine Learning and Intelligent Computing

A Word Sense Disambiguation Method Based on Multiple Sense Graph

Wenjun Liu
School of Computer Science and Engineering, University of Electronic Science and Technology of China (UESTC), Chengdu, 611731, China
School of Computer and Software Engineering, Xihua University, Chengdu, 610039, China

Hailan Wang
School of Computer and Software Engineering, Xihua University, Chengdu, 610039, China

Xiping Wang
School of Computer and Software Engineering, Xihua University, Chengdu, 610039, China

Mengshu Hou* MSHOU@UESTC.EDU.CN
School of Computer Science and Engineering, University of Electronic Science and Technology of China (UESTC), Chengdu, 611731, China
School of Big Data and Artificial Intelligence, Chengdu Technological University, Chengdu, 611730, China

Editors: Nianyin Zeng and Ram Bilas Pachori

Abstract

Word Sense Disambiguation is a process to determine the best meaning of an ambiguous word according to its contextual semantic information. Many methods of Word Sense Disambiguation cannot deal with polysemous words well because they only consider the meaning of the adjacent words before and after ambiguous words, and cannot consider the meaning of all words in the sentence globally. In order to solve the above problems, this paper proposes a word sense disambiguation method based on Multiple Sense Graph. This method applies the BERT model to generate word sense vectors, and globally considers the feature relationship between the ambiguous word and all words in the context. In addition, this method applies the PageRank algorithm to score the importance of each sense vector of the word, and the scoring results are sorted to obtain the best sense of the ambiguous word. The experimental results indicate that the proposed BERT-PageRank method improves the evaluation index compared with the other two semantic disambiguation methods. In summary, the proposed method improves the accuracy of word sense disambiguation to obtain the best word sense.

Keywords: Word Sense Disambiguation, BERT Model, PageRank Algorithm, Word Sense Disambiguation Graph

1. Introduction

Word Sense Disambiguation (WSD) is always a core and difficult problem in natural language processing. The sense of a word has different meanings in different contexts, and Word Sense Disambiguation refers to the process of determining the meaning of the object word according to the context (Kouris et al., 2022). In different actual situations, a word may have multiple senses. If Word Sense Disambiguation is not carried out on ambiguous words, it will affect the correct understanding of the text (Toddenroth, 2022). Word Sense Disambiguation has been widely used in many important applications such as search engines and text understanding systems, so it has important theoretical value and practical significance (Pu et al., 2023). However, there are still some

problems with Word Sense Disambiguation methods: (1) Traditional word embedding methods use a unique vector to represent any word, which cannot handle polysemous words well. (2) The above Word Sense Disambiguation methods only consider the sense of the adjacent words of the ambiguous word, and cannot consider the sense of all the words in the sentence globally.

In order to solve the above problems, this paper proposes a word sense disambiguation method based on Multiple Sense Graph. This method applies BERT model to generate word sense vectors to solve the problem of polysemy, and globally considers the feature relationship between the ambiguous word and all words in the context. In addition, this method applies the PageRank algorithm to score the importance of each sense vector of the word, and the scoring results are sorted to obtain the best sense of the ambiguous word. Therefore, the proposed method can be named as the BERT-PageRank method. The experimental results indicate that the BERT-PageRank method improves the evaluation index compared with the other two semantic disambiguation methods. In summary, the proposed method improves the accuracy of word sense disambiguation to obtain the best word sense.

2. Related Works

Word Sense Disambiguation is the process of determining the meaning of an object based on its context. In different actual scenarios, a word may have multiple senses (Yang and Zheng, 2023). If the Word Sense Disambiguation is not carried out, the correct understanding of the text will be affected. The common methods of Word Sense Disambiguation can be divided into three categories: knowledge base based on Word Sense Disambiguation, supervised Word Sense Disambiguation, unsupervised Word Sense Disambiguation.

The word sense disambiguation based on knowledge base uses a variety of knowledge resources as the knowledge base, and judges the ambiguous words through the context environment (Wang et al., 2020; Ayetiran et al., 2021). Popular knowledge bases include WordNet, CLKB, HowNet, etc. The classical Lesk algorithm assumes that the meaning of an ambiguous word is related to the sentence it is in, so the best word sense is selected by calculating the sentence and each sense of the ambiguous word.

The supervised word sense disambiguation is a machine learning model built from manually labeled data that matches all the senses of words in the knowledge base (Filimonov et al., 2022; Lu, 2019). The commonly used supervised models for Word Sense Disambiguation are using classifiers in decision trees and decision tables, or using naive Bayes model. Supervised Word Sense Disambiguation can match the best word sense of the ambiguous word in the knowledge base without considering the complex context.

The unsupervised word sense disambiguation is realized by obtaining information from knowledge base without manual annotation (Moradi et al., 2019; Pesaranghader et al., 2019). Common unsupervised methods include graph - based, semantic clustering, cross - semantic disambiguation and multilingual parallel knowledge base. The biggest difference between unsupervised methods and supervised methods is that only a small amount or no manual annotation of corpus is required.

3. Proposed Method

This paper proposes a word sense disambiguation method based on Multiple Sense Graph. Figure 1 shows the flowchart of the word sense disambiguation method based on multiple sense embedding

and sense sequence graph. In Figure 1, this Word Sense Disambiguation method can be divided into four different modules: word sense annotation retrieval, vector generation, word sense graph construction, and correct word sense identification.

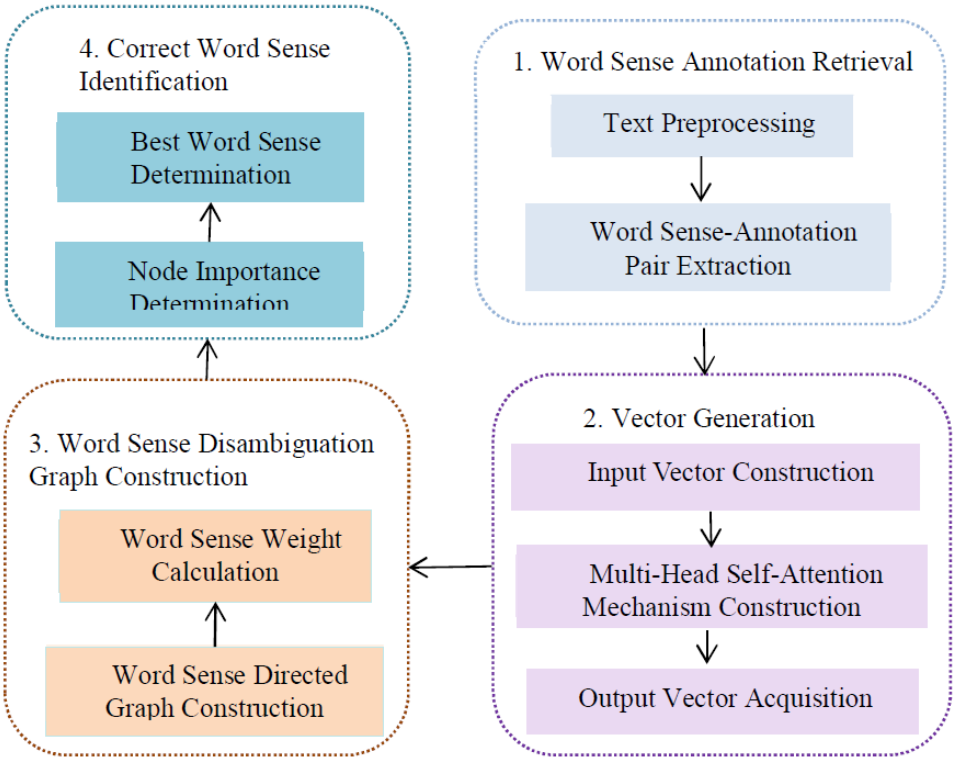

Figure 1: The flowchart of a word sense disambiguation method based on multiple sense embedding and sense sequence graph.

3.1. Word Sense Annotation Retrieval

The word sense annotation retrieval is to preprocess the sentence, identify the ambiguous words in the sentence, and obtain all the sense-annotation pairs of these ambiguous words. The word sense annotation retrieval is divided into two parts including text preprocessing and word sense-annotation pair extraction. The following sections describe the two modules in more detail. The text preprocessing is to normalize the input sentence and extract effective sentence and effective information. The word sense-annotation pair extraction is to filter the ambiguous words in the sentence through the WordNet knowledge base and obtain all the sense-annotation pairs of these ambiguous words.

3.2. Vector Generation

3.2.1. INPUT VECTOR CONSTRUCTION

The input vector construction is to convert all the sense-annotation pairs of an ambiguous word and its sentence into the input vector of the BERT model in the form of encoding. The input vector construction is divided into four parts including tokenizer tag, embedding matrix construction, feature matrix acquisition, and input vector generation. The tokenizer tag is to add classification tags [CLS] to the head and segmentation tags [SEP] to the tail of the input sentence. The tokenized word is called token, and a hyperparameter is set to specify the length of the sequence, the long one is truncated, and the short one is filled with tags. The embedding matrix construction is to generate symbol embedding matrix, position embedding matrix and segment embedding matrix of the input sentence, which are all formed by one-hot encoding and learning matrix inner product. The feature matrix acquisition is the addition of the three matrices generated in the embedding matrix construction to generate the feature matrix of the sentence, which is the input of the Encoder. The input vector generation is formed by the feature matrix of the sentence, and each row of the feature matrix is the input vector of the corresponding token.

3.2.2. MULTI-HEAD SELF-ATTENTION MECHANISM CONSTRUCTION

The multi-head self-attention mechanism construction is the most important sub-layer in the Encoder. The Encoder uses the multi-head self-attention mechanism to bi-directionally encode information that can consider the context to enhance the correlation between words. The multi-head self-attention mechanism construction formulas are as follows:

$$\begin{aligned}
&Q_{P\times H} = E_{P\times H}\cdot W^{Q}_{H\times H}, K_{P\times H} = E_{P\times H}\cdot W^{K}_{H\times H}, V_{P\times H} = E_{P\times H}\cdot W^{V}_{H\times H}\\
&d_k = \frac{H}{h}\\
&E^{head_i}_{P\times d_k} = soft\max(\frac{Q_{P\times H}W^{Q_i}_{H\times d_k}\cdot K_{P\times H}W^{K_i}_{H\times d_k}{}^{T}}{\sqrt{d_k}})\cdot V_{P\times H}W^{V_i}_{H\times d_k}, i\in[1,h]\\
&E^{Multihead_i}_{P\times H} = Concat(E_{head_1},\cdots,E_{head_h})\cdot W^{O}_{H\times H}
\end{aligned} \tag{1}$$

where Q is the Query matrix, $W^{Q}_{H\times H}$ is a learning matrix, K is the Key matrix, $W^{K}_{H\times H}$ is a learning matrix, V is the Value matrix , $W^{V}_{H\times H}$ is a learning matrix, d_k is the dimension after the multi-head mechanism, h is the number of heads, $head_i$ is the i-th head, $E^{head_i}_{P\times d_k}$ is the self-attention score matrix for the i-th head, $W^{Q}_{H\times d_k}$ is a learning matrix, $W^{K_i}_{H\times d_k}$ is a learning matrix, $W^{V_i}_{H\times d_k}$ is a learning matrix, $E^{Multihead_i}_{P\times H}$ is the feature matrix of the augmented semantics in the i-th layer encoder, $W^{O}_{H\times H}$ is a learning matrix.

3.2.3. OUTPUT VECTOR ACQUISITION

The output vector acquisition is to obtain the corresponding enhanced semantic vector after passing the sense-annotation pair and the sentence containing the ambiguous word through the BERT model for 12 times. The output vector is divided into three parts including encoding layer stacking, linear dimensionality reduction and function activation. The encoding layer stacking is to stack the sentence 12 times to obtain the final feature vector. Each layer of Encoder contains two sub-layers: a multi-head mechanism layer and a feed-forward neural network layer, and both of them are connected by residual connections. The linear dimensionality reduction is the process of linearly

converting the obtained feature matrix from high dimension to low dimension to achieve dimensionality reduction. The Function activation is to normalize the matrix after dimension reduction, and select the vector corresponding to [CLS] as the feature vector of the whole sentence to obtain the word sense vector and sentence vector of dynamic learning context information.

3.3. Word Sense Disambiguation Graph Construction

3.3.1. WORD SENSE DIRECTED GRAPH CONSTRUCTION

The word sense directed graph construction is to construct a unidirectional directed graph with all senses of a word as nodes and relations between senses as edges. The word sense directed graph construction is divided into two parts including word sense node construction and directed edge construction. The word sense node construction maps all the word sense-annotation pairs into word sense nodes, and divide them into different columns according to the order of the words corresponding to the nodes in the sentence. Each column is arranged in the order of getting the word sense-annotation pairs from the WordNet knowledge base. The directed edge construction is that the sense nodes within the same word are not connected, and each sense node between adjacent words is connected in the direction from left to right.

3.3.2. WORD SENSE WEIGHT CALCULATION

The word sense weight calculation is to calculate the similarity between each word sense and the context, and the result of normalization of the similarity value is used as the word sense node weight. The word sense weight calculation formulas are as follows:

$$\begin{aligned} &S\mathrm{im}(W_{ij}, S) = \overrightarrow{V_{wij}} \cdot \overrightarrow{V_S} = \sum_{k=1}^{n} V_{\mathrm{k}w_{ij}} V_{kS} \\ &i \in [1, \mathrm{P}], \mathrm{j} \in [1, H] \\ &NSim(w_{ij}, S) = \frac{S\mathrm{im}(w_{ij}, S)}{\sum_{j=1}^{\mathrm{p}} S\mathrm{im}(w_{ij}, S)} \end{aligned} \tag{2}$$

where $S\mathrm{im}(W_{ij}, S)$ is the similarity between each sense vector and the sentence vector, W_{ij} is the j-th sense-annotation pair of the i-th word, S is the sentence where the ambiguous word is located, $\overrightarrow{V_{w_{ij}}}$ is the word sense vector of W_{ij}, $\overrightarrow{V_S}$ is the sentence vector of S, $NSim(w_{ij}, S)$ is the normalized similarity value obtained by normalizing $S\mathrm{im}(w_{ij}, S)$.

3.4. Correct Word Sense Identification

3.4.1. NODE IMPORTANCE DETERMINATION

The node importance determination is determined by scoring the importance of each word sense node in the word sense disambiguation graph by the PageRank algorithm. The formulas to obtain the importance of each word sense node are as follows:

$$\begin{aligned} &Adj = [a_{ij}]_{K\times K}, i, j \in K, a_{ij} = a_{ji} = \begin{cases} 1, (n_{\mathrm{i}}, n_{\mathrm{j}}) \in E \\ 0, \text{ else} \end{cases} \qquad M = [m_{ij}]_{K\times K}, m_{\mathrm{ij}} = \frac{\mathrm{a}_{ij}}{\sum_{k=1}^{K} a_{\mathrm{kj}}} \\ &\overrightarrow{v} = (v_{11}, v_{12}, \cdots, v_{1k_1}, v_{21}, v_{22}, \cdots, v_{2k_2}, \cdots, v_{n1}, v_{n2}, \cdots, v_{nk_n}) \\ &k_1 + k_2 + \cdots + k_n = K, v_{ij} = NSim(w_{ij}, S) \\ &\overrightarrow{\mathrm{Pr}} = (r_{11}, r_{12}, \cdots, r_{1k_1}, r_{21}, r_{22}, \cdots, r_{2k_2}, \cdots, r_{n1}, r_{n2}, \cdots, r_{nk_n}) \quad r_{ij} = 1/K, \mathrm{c} = 0.85 \\ &\overrightarrow{\mathrm{Pr}'} = cM\overrightarrow{\mathrm{Pr}} + (1-c)\overrightarrow{v} \end{aligned} \tag{3}$$

where $Adj = [a_{ij}]_{K\times K}$ is the adjacency matrix of the digraph G, a_{ij} is the connection between the meaning nodes n_i and n_j, if there is a connection between nodes n_i and n_j, $M = [m_{ij}]_{K\times K}$ is a K×K transition probability matrix, m_{ij} is the probability that word sense nodes n_i and n_j are connected, $\overrightarrow{v}$ is the importance of the word sense node, v_{ij} is the importance of the j-th sense of the i-th word, k_i is the number of sense nodes of the i-th word, $\overrightarrow{\mathrm{Pr}}$ is a K×1 target ranking vector, r_{ij} is the final score of the j-th sense node of the i-th word, each component in $\overrightarrow{Pr}$ is initialized with 1/K, $\overrightarrow{Pr'}$ is the process ranking vector after $\overrightarrow{Pr}$ is updated by each iteration of the PageRank algorithm, c is the damping factor, the first term represents the probability of visiting each sense node according to the transition probability matrix M when convergence is stable, and the second term represents the probability of visiting each sense node completely randomly.

3.4.2. BEST WORD SENSE DETERMINATION

The best word sense determination is to select the word sense node with the largest sense importance score of each word, and the sense-annotation pair corresponding to the node is the best sense of the word in the current context. The best word sense determination formulas are as follows:

$$\begin{aligned} \overrightarrow{Pr_i} &= (r_{i1}, r_{i2}, \cdots, r_{ik_i}) \\ sense_{wi} &= \max(\overrightarrow{Pr_i}) \end{aligned} \tag{4}$$

where $\overrightarrow{Pr_i}$ is the target ranking vector for the i-th word in the target ranking vector $\overrightarrow{Pr}$, $r_{ij}(1 \le j \le k_i)$ is the final score of the j-th sense node of the i-th word, $sense_{wi}$ is the best sense of the i-th word, the component of $\overrightarrow{Pr_i}$ with the largest score is selected by the max function.

4. Experiment

The experimental system for word sense disambiguation is constructed that the BERT-PageRank method can improve the performance of word sense disambiguation. The experimental designs three different word sense disambiguation methods, and the performance of three word sense disambiguation methods is compared by the experimental results. The three word sense disambiguation methods are Word2vec, BERT and BERT-PageRank. In the experiments, this paper uses a total of five standard all-words WSD datasets including senseval3, senseval2007 and semeval2015. In this experiment, the performance of all word sense disambiguation methods is mainly evaluated by the metric F1 Score. The F1 Score is a statistical measure of the accuracy of a binary classification model, taking into account both Precision and Recall.

Figure 2 shows the comparison of F1 Scores of the three word sense disambiguation methods based on the word sense disambiguation results. In Figure 2, for all the WSD dataset test F1 Scores, the BERT-PageRank method has higher F1 Scores than the other two word sense disambiguation methods. The results of word sense disambiguation show that the BERT-PageRank method can improve the performance of word sense disambiguation.

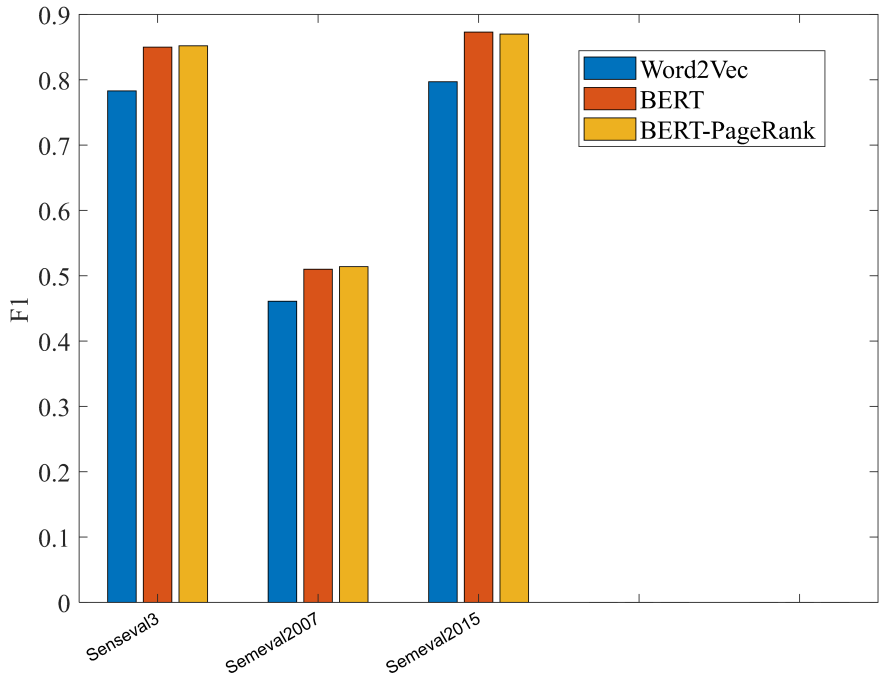

Figure 2: The comparison of the F1 Scores of the three word sense disambiguation methods based on the word sense disambiguation results

5. Conclusion

This paper proposes a word sense disambiguation method based on Multiple Sense Graph. Experimental results show that the proposed method can obtain higher F1 scores compared with other two methods. In a word, the proposed method can obtain more accurate best word sense, and further optimize the disambiguation results.

Acknowledgments

This work is supported by the Intelligent Policing Key Laboratory of Sichuan Province (Grant No. ZNJW2024KFQN012).

References

Eniafe Festus Ayetiran, Petr Sojka, and Vít Novotnỳ. Eds-membed: Multi-sense embeddings based on enhanced distributional semantic structures via a graph walk over word senses. *Knowledge-Based Systems*, 219:106902, 2021.

Maxim Filimonov, Daphné Chopard, and Irena Spasić. Simulation and annotation of global acronyms. *Bioinformatics*, 38(11):3136–3138, 2022.

Panagiotis Kouris, Georgios Alexandridis, and Andreas Stafylopatis. Abstractive text summarization: Enhancing sequence-to-sequence models using word sense disambiguation and semantic content generalization. *Computational Linguistics*, 47(4):813–859, 2022.

Wenpeng Lu. Word sense disambiguation based on dependency constraint knowledge. *Cluster Computing*, 22(Suppl 3):7549–7557, 2019.

Behzad Moradi, Ebrahim Ansari, and Zdeněk Žabokrtskỳ. Unsupervised word sense disambiguation using word embeddings. In *Proceedings of the 25th Conference of Open Innovations Association FRUCT*, pages 228–233, 2019.

Ahmad Pesaranghader, Stan Matwin, Marina Sokolova, and Ali Pesaranghader. deepbiowsd: effective deep neural word sense disambiguation of biomedical text data. *Journal of the American Medical Informatics Association*, 26(5):438–446, 2019.

Xiao Pu, Lin Yuan, Jiaxu Leng, Tao Wu, and Xinbo Gao. Lexical knowledge enhanced text matching via distilled word sense disambiguation. *Knowledge-Based Systems*, 263:110282, 2023.

Dennis Toddenroth. Evaluation of domain-specific word vectors for biomedical word sense disambiguation. *Studies in health technology and informatics*, 292:23–27, 2022.

Yinglin Wang, Ming Wang, and Hamido Fujita. Word sense disambiguation: A comprehensive knowledge exploitation framework. *Knowledge-Based Systems*, 190:105030, 2020.

Qihao Yang and Jiong Zheng. Step-wise discriminative learning on uncertain annotations for word sense disambiguation. *Journal of Engineering Research*, 11(2):100086, 2023.