TWE-WSD: An effective topical word embedding based word sense disambiguation

Lianyin Jia1,2 | Jilin Tang1 | Mengjuan Li3 | Jinguo You1 | Jiaman Ding1 | Yinong Chen4

1Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming, China
2Yunnan Key Laboratory of Artificial Intelligence, Kunming University of Science and Technology, Kunming, China
3Library, Yunnan Normal University, Kunming, China
4School of Computing, Informatics, and Decision Systems, Arizona State University, Tempe, Arizona, USA

Abstract
Word embedding has been widely used in word sense disambiguation (WSD) and many other tasks in recent years for it can well represent the semantics of words. However, the existing word embedding methods mostly represent each word as a single vector, without considering the homonymy and polysemy of the word; thus, their performances are limited. In order to address this problem, an effective topical word embedding (TWE)-based WSD method, named TWE-WSD, is proposed, which integrates Latent Dirichlet Allocation (LDA) and word embedding. Instead of generating a single word vector (WV) for each word, TWE-WSD generates a topical WV for each word under each topic. Effective integrating strategies are designed to obtain high quality contextual vectors. Extensive experiments on SemEval-2013 and SemEval-2015 for English all-words tasks showed that TWE-WSD outperforms other state-of-the-art WSD methods, especially on nouns.

1 | INTRODUCTION
Natural language has inherent ambiguity. Many words generally have multiple senses, for example, the word ‘cricket’ can be either a sport or an insect. For a human, it is rather simple to recognise the right sense under a certain context, but for a computer, word sense disambiguation (WSD) is a rather difficult task as it is considered as an AI-complete problem [1]. WSD has become one of the central challenges in natural language processing field and is widely used in machine translation [2,3], information retrieval [4], information extraction [5], name disambiguation [6] etc. In machine translation, an optimally machine translation system usually incorporate some kind of WSD component, a WSD system can help a machine translation system to choose the right translation for an ambiguous word. In information retrieval, the sense ambiguity may cause relevant documents precluded and irrelevant documents retrieved. A WSD system can help information retrieval system to improve the retrieval performance.

Most existing disambiguation studies are based on a distribution hypothesis [7], assuming that similar senses should have similar contexts. Some typical representatives are Lesk [8] and its derived studies [9,10], which try to exploit the overlaps between the context of ambiguous word and its sense definitions. However, the effects of these studies are limited, as they cannot capture semantic information hidden under the context.

Word embedding is a good solution to this problem. It captures both syntactic and semantic information and compresses them into a low dimensional real-valued continuous word vector (WV). The generated WVs have rich semantic information that is helpful to WSD and many other natural language processing tasks.

The quality of WVs has a great impact on WSD performance. For the past a few years, many word embedding
techniques have been proposed. Latent semantic analysis (LSA) [11] constructed the document-word co-occurrence matrix first, and then, it executed dimension reduction using singular value decomposition (SVD) to generate the corresponding WV s. Word2Vec, proposed by Mikolov et al. [12], uses a neural network-based supervised process to generate WV s. The generated WV s meet the requirements of simple algebraic operations, which is simple to use and can significantly improve the quality of WV s. Glove, proposed by Pennington et al. [13], made an explicit representation by a global word co-occurrence statistics and local window context. Based on these techniques, the quality of the generated WV s is significantly improved.

Considering the semantic information captured by word embedding, many studies have been conducted on integrating it into WSD. Taghipour et al. [14] generated and adapted WV s by using window approach network [15], and they integrated these WV s into it makes sense (IMS) [16], a well-established WSD platform, to improve the disambiguation performance. Rothe et al. [17] proposed AutoExtend, which improved the performance by integrating words, phrases and syntax into word embedding, and extending word embedding to synset embedding. Ignacio et al. [18] designed efficient integrating strategies and verified that integrating word embedding into IMS can improve the performance of WSD.

Although the aforementioned methods can improve WSD performance to certain extent, they have a major weakness: they generate a single context-independent WV for each word. We know words are generally polysemantic, which means word senses in different contexts are usually different. Therefore, it is hard for a single WV to capture all senses in different contexts.

To address this problem, Huang et al. [19] proposed a multi-prototype system, which uses K-means clustering to decompose single word embedding into multiple prototypes. Each prototype represents an independent word sense. Multi-prototype system can solve the problem to certain extent, but it still faces a few challenges. For example: (1) the ability to capture the semantic information of word is insufficient; (2) it divides the context of a word into non-overlapping clusters, which in fact omits the correlation of different senses.

Motivated by the weakness of word embeddings and the insufficiency of multi-prototype system, the following two objectives are aimed:

1. Design effective topical word embedding that incorporates topical model into word embedding to better capture semantic information for ambiguous words under different topics.
2. Design topical word embedding based contextual vector generating strategy and further implement an effective all-word WSD system on all-word WSD tasks.

To achieve these two goals, a topical word embedding (TWE) based WSD (TWE-WSD) system is proposed. Instead of generating a single WV for each word, TWE-WSD generates multiple topical word vectors (TWV) for each word by integrating Latent Dirichlet Allocation (LDA) and word embedding. Effective integrating strategies are designed to obtain high quality contextual vectors (CV). These CVs exploit some key information (contextual topic distribution, contextual word embedding and the distance to the ambiguous word) to better capture the semantic information underlying the contexts.

In addition, TWE-WSD is implemented on the latest SupWSD framework [20], which has a faster WSD speed and higher WSD accuracy compared with IMS based methods. TWE-WSD is implemented and compared it with several other state-of-the-art WSD methods on all-word WSD tasks. Extensive experiments on SemEval-2013 and SemEval-2015 for English all-words tasks showed that TWE-WSD outperforms the existing methods, especially in nouns.

The following contributions are made:

1. A novel topical word embedding based WSD system named TWE-WSD is devised, which combines LDA and word embedding together to improve WSD performance. To the best of knowledge, we are the first to combine these two techniques together to implement supervised WSD.
2. Effective strategies are designed to generate high quality contextual vectors (CV), which can capture explicit semantic information underlying the contexts effectively.
3. Comprehensive experiments show that TWE-WSD achieves substantial improvements over the state-of-the-art techniques.

The rest of the paper is organised as follows. Related work is categorised and discussed in Section 2. TWE-WSD is presented in detail in Section 3. Experiments and result analyses are described in Section 4. Finally, conclusions and future work are presented in Section 5.

2 | RELATED WORKS

2.1 | Supervised word sense disambiguation

WSD can be divided roughly into two categories: supervised WSD and knowledge-based WSD.

Supervised WSD works on manually sense-tagged data and then uses supervised machine learning methods [such as Support Vector Machine (SVM)] to establish a classification model for each word [21–23]. Earlier supervised WSD methods focused on combining different features with classification methods. Ng et al. [21] combined semantic features extracted from different knowledge sources and used example-based learning to train the classification model. Lee et al. [22] evaluated WSD performance of combining four different semantic features and four different supervised learning methods to find the best combination and found that using all the four features and SVM could achieve a higher accuracy. IMS [16] implemented a comprehensive framework by combining part-of-speech (POS) tags, surrounding words and local collocation and linear SVM. IMS is flexible because different features and classifiers can be easily inserted and
verified. Due to this, it has been extensively used as a fundamental experimental platform.

With the development of word embedding techniques in recent years, there is a growing trend to integrate word embedding into WSD. Taghipour et al. [14] integrated WVs proposed by Collart et al. [15] and Iacobacci et al. [18] integrated WVs generated by Word2Vec into IMS system, respectively. They turned out integrating these WV is beneficial to WSD. In considering the difficulties of code expansion and optimisation for big datasets in IMS, Papandrea et al. [24] implemented a new platform, SupWSD. SupWSD has many virtues, for example taking much less parameters than IMS, supporting most widely used preprocessing tools, and supporting multi-language. All these approaches make it flexible for a programmatic use and easy for expanding.

Recently, there is an increasingly trend to introduce neural models into WSD [25,26], and the most prominent representative is Context2Vec [25]. Context2Vec uses a bidirectional LSTM recurrent neural network to learn a generic context embedding function from large corpora, and this function is rather flexible and can be used in a large number of tasks, such as sentence completion, lexical substitution and WSD.

2.2 Knowledge based word sense disambiguation

Supervised methods require sense-labelled training data. However, in most cases, labelled data is sparse and difficult to obtain. Unlike supervised methods, knowledge based methods only need the structure or content of external knowledge source, for example WordNet, BabelNet, but no sense annotated training data. They generate its own annotated corpus and work on an assumption that similar senses occur in similar contexts.

There are two popular types of knowledge-based methods: overlap-based method and graph-based method.

The most famous overlap-based method is Lesk [8], which computes the overlaps between the context of ambiguous word and its sense definitions. Lesk suffers from similar sentences might have very few overlaps. To overcome this problem, many variants have been proposed to replace overlap with distributed similarity, thus improving the performance [10].

Different with overlap-based method, graph-based method first builds semantic graphs from lexical resources and then uses different techniques (Personalised Page Rank [27], Markov Random Field [28], Entity Linking [29,30], Game theory [31]) to design algorithms to exploit the structural properties of semantic graphs.

2.3 Latent Dirichlet allocation-based word sense disambiguation

LDA proposed by Blei et al. [32] works on Bayesian topic model. It describes the hierarchical relations among words, topics and documents. In LDA, a topic is an implicit variable, each document can be described by a distribution of topics and each topic can be described by a distribution of words.

LDA is widely used in WSD and many LDA based WSD methods. As far as we know, LDAWN [33] proposed by Boyd-Graber et al., is the first LDA-based WSD method which uses LDA and WordNet hierarchy together in WSD. LDAWN works on the assumption that senses with similar paths in the WordNet are more likely have the same topic. Cai et al. [23] used extracted topic distribution as features and then put these features into a supervised WSD system to improve WSD performance. Li et al. [34] introduced LDA to decompose the conditional probability of sense phrases and proposed three models with different degrees of resource availability to improve WSD performance. In a recent study, Chaplot et al. [35] designed WSD-TM, a topical model-based WSD, which can scale linearly with the number of words in the context, which decreases the training time significantly. The topical model used was a LDA variant in which the topic proportions for a document were replaced by synset proportions.

For more description of WSD-related approaches, readers are asked to read recent surveys [36,37] for more details.

3 TOPICAL WORD EMBEDDING-BASED WORD SENSE DISAMBIGUATION

3.1 General framework

As mentioned earlier, it is hard for traditional word embedding to deal polysemey easily because it uses a single WV to describe an ambiguous word, thus deteriorating the WSD performance.

It is well known that sense and topic are well related. A polysemous word usually expresses a certain sense in a certain topic and a certain topic is more likely to contain words with a certain sense. For example, word 'cricket' more likely to expresses the sense of sport in the sports topic, whereas it usually expresses the sense of insect in the animal topic.

Based on the analysis above, topical model is a natural choice to improve the performance of WSD. topical model is integrated into word embedding and a novel TWE-based WSD method (TWE-WSD) is proposed, which can exploit the benefits of both topical model and word embedding to further improve the performance of WSD. The basic framework of TWE-WSD is shown in Figure 1.

TWE-WSD consists of the following four basic steps:

(i) Topic model training;
(ii) Topical WV generating;
(iii) Contextual vectors generating;
(iv) Supervised disambiguation.

Note that the first two steps are executed on a large un-tagged topical training corpus to generate TWVs, and the latter two steps are executed on TWVs generated from the first two steps and a sense-tagged disambiguation dataset.
3.2 | Topical word vectors generation

After the topic model trained, all documents in the topical training corpus are divided into \( k \) topical document subsets according to topic model built in Section 3.2. Note that assigning each document to multiple topics is time-consuming. In order to speed up the training process, each document is assigned to the subset corresponding to its maximum probability topic. Then, each subset is trained independently using word embedding to obtain TWVs for each topic. The TWV of word \( w \) under topic \( t \) is represented by \( V^t_w \).

Note that after the division of the topical training corpus, the number of documents in each subset may be skewed, that is some subsets may contain very few documents, which will degenerate the quality of generated TWVs. To solve this problem and speed up the training process, those subsets whose number of documents below a certain threshold are ignored, for example 10. The TWVs for these subsets are considered as zero vectors.

3.3 | Topic model training

Topic model is the basis of TWE-WSD. Here, LDA is used, the most common topic model currently in use, to train the topic model. Based on LDA, given a topic number parameter \( k \), the topical training corpus is trained, and \( k \) topics can be recognised. Besides, the document-topic and the topic-word distributions can be computed for future use.

Taking into account the slow convergence of LDA, a fast online LDA proposed by Hoffman et al. [38] is applied to train the topic model. The online LDA bases on online stochastic optimisation with a natural gradient step and supports stream processing. As a result, it can obtain high quality topic models in a shorter training time compared with traditional LDAs.

3.4 | Contextual vector generating

Context is important to WSD. Thus, instead of using TWVs of the ambiguous word as training features directly, CVs of the ambiguous word is generated that will make our model fully exploit the context. To do this, the following three steps are executed to generate high-quality CVs.

3.4.1 | Contextual topic distribution

Intuitively, it is too naïve to directly assign a context \( C \) to a specific topic \( t \). Thus, before generating CVs for \( C \), the first
3.4.2 Contextual word vector generating

After obtaining the topic distribution, it is important to obtain WV $V_{w_j}^\theta$, a WV of word $w_j$ under topic distribution $\theta$. To achieve this goal, a weighted average strategy to compute $V_{w_j}^\theta$ according to Equation (2) is designed:

$$V_{w_j}^\theta = \frac{\sum_{i=1}^{p} p(t_i|C) V_{w_j}^{t_i}}{\sum_{i=1}^{p} p(t_i|C)}$$

where $p(t_i|w_j)$ denotes the conditional probability of that the $j$-th word $w_j$ in $C$ is assigned to topic $t_i$, $|C|$ denotes the number of words in context $C$.

3.4.3 Contextual vector generating

As we use CVs as final features to be trained in supervised classifiers, a good strategy for generating CVs can efficiently promote WSD performance. Intuitively, the closer a word to the ambiguous word is, the greater contributions to recognise the ambiguous word it will be. Therefore, an exponential decay strategy is designed to integrate the contextual WVs to generate the final CV $V_C$. This strategy puts more emphasis on words close to ambiguous word by exponentially setting the weight of word $w_j$ according to its distance to the ambiguous word as shown in Equation (3).

$$V_C = \sum_{j=1}^{|C|} V_{w_j}^\theta \alpha^d$$

where $d$ represents the distance between word $w_j$ and the ambiguous word in $C$, and $\alpha = 0.1^{2/|C|}$ represents the decay factor of $C$. Here, $0.1$ means a surrounding word with distance $d$ to the ambiguous word contributes 10 times than the one with distance $d+1$.

3.5 Supervised word sense disambiguation

For the final disambiguation step, the following features are extracted: POS tags, surrounding words, local collocations, syntactic relations and pre-trained CVs, then LibLinear is used, a built-in classifier in SupWSD, to train these features. Readers are asked to read paper [20] for a detailed description of training process.

4 EXPERIMENTS

4.1 Experimental environment

The performance of TWE-WSD is evaluated on all-words task. All experiments were conducted on a PC with a 4-core Intel i7-7700 CPU running at v3.6 GHz and 16 GB of memory. Built-in Word2Vec and online LDA are used in Gensim [39] to train TVWs and LDA model, respectively.

4.2 Dataset

4.2.1 Topical training corpus

The English Wikipedia$^1$ is used as our topical training corpus, which includes about 4.5 million documents and three billion words.

4.2.2 Disambiguation dataset

Disambiguation dataset is used for generating topic distribution and CVs. The following training and testing datasets are widely used in WSD:

1. Training dataset

SemCor [40] is used as the training dataset, which includes 352 documents and 360,000 words, of which 226,036 words are sense-tagged.

2. Testing dataset

Two all-words WSD tasks are used to evaluate TWE-WSD: SemEval-13 task 12 (SE13) [41]: it consists of 13 documents from different fields, in which there are 1644 sense-tagged nouns.

SemEval-15 task 13 (SE15) [42]: it consists of four documents from biomedicine, mathematics and sociology, in which there are 1022 sense-tagged nouns, verbs, adjectives and adverbs.

\(^{1}\text{http://download.wikipedia.org}\)
4.3 Results and discussions

In training the topic model, the number of iterations are set to 1000, and the other parameters are left as default in Gensim. In generating TWV, Skip-gram model is used with negative sampling 10 and sub-sampling of frequent words $10^{-5}$. The length of the WV is set to 400 as default value recommended in Reference [18].

4.3.1 The effects of $k$

In order to investigate the effects of the number of topics $k$ on WSD performance, $p = 5$ is set and the whole document is used as the topical context, and then, $F1$ scores are computed [43] for $k = 50, 80, 100, 130$ respectively. The results are shown in Table 1.

In Table 1, ALL represents the comprehensive scores of SE13 and SE15 as shown in Equation (4).

$$F1_{ALL} = \frac{N_{SE13}F1_{SE13} + N_{SE15}F1_{SE15}}{N_{SE13} + N_{SE15}}$$

where $N_{SE13}$ and $N_{SE15}$ represent the number of documents in SE13 and SE15, respectively.

As can be seen in Table 1, all the $F1$ scores increase with the increase of $k$. When $k = 100$, $F1$ scores tend to be stable. Therefore, in the experiments below, when not pointed out explicitly, $k$ is set to 100.

4.3.2 The effects of different context types

When computing topic distribution, different context types will have different impact on the topic distribution, and thus affecting the WSD performance. Two types of contexts are considered: sentence (use the sentence where the ambiguous word is located as a context) and document (use the whole document where the ambiguous word is located as a context). The $F1$ scores of these two context types are shown in Table 2.

As can be seen in Table 2, a better performance can be achieved when using document as context. The reason is that with the increase of context size, more words can be used to recognise the topic of the context. Therefore, to obtain a better disambiguation performance, when not pointed out explicitly, document is used as default context.

4.3.3 The effects of $p$

To investigate the effects of parameter $p$, $p = 1, 5, 10, 20$ is set, and then, the corresponding $F1$ scores are computed, respectively, as shown in Table 3.

As can be seen in Table 3, $p = 1$ means the highest probability topic assigned to each context, and in this case, the

\begin{table}[h]
\centering
\caption{F1 scores of different $k$}
\begin{tabular}{|c|c|c|c|}
\hline
$k$ & SE13 & SE15 & ALL \\
\hline
50 & 0.694 & 0.705 & 0.698 \\
80 & 0.691 & 0.707 & 0.697 \\
100 & 0.705 & 0.714 & 0.708 \\
130 & 0.708 & 0.713 & 0.709 \\
\hline
\end{tabular}
\end{table}

\begin{table}[h]
\centering
\caption{F1 scores for sentence-context and document-context}
\begin{tabular}{|c|c|c|c|}
\hline
Context Types & SE13 & SE15 & ALL \\
\hline
Sentence & 0.655 & 0.688 & 0.668 \\
Document & 0.705 & 0.714 & 0.708 \\
\hline
\end{tabular}
\end{table}

\begin{table}[h]
\centering
\caption{F1 scores of different $p$}
\begin{tabular}{|c|c|c|c|}
\hline
$p$ & SE13 & SE15 & ALL \\
\hline
1 & 0.682 & 0.659 & 0.673 \\
5 & 0.705 & 0.714 & 0.708 \\
10 & 0.704 & 0.716 & 0.708 \\
20 & 0.703 & 0.716 & 0.708 \\
\hline
\end{tabular}
\end{table}

$F1$ scores are the lowest. When $p \geq 5$, the $F1$ scores tend to be stable. This is consistent with our previous analysis in Section 3.4.2. As a result, no need to compute all the probabilities in $\theta_p$. $p = 5$ is used as our default value thereafter.

4.3.4 Comparison with other word sense disambiguation systems

In order to compare performances of different methods, TWE-WSD was compared with seven state-of-the-art methods: Lesk [8], Most Frequent Sense (MFS), IMS [16], SupWSD [24], SupWSD + emb [24], Context2vec [25] and WSD-TM [35]. Here, MFS is a heuristic baseline, which simply labels each ambiguous word with its MFS in the training corpus. SupWSD + emb is SupWSD with appending word embedding into its training features. The results are showed in Table 4.

It can be seen from Table 4 that TWE-WSD has comparable performance with the best method on SE15 (Context2Vec), it outperforms all other state-of-the-art methods significantly on SE13, and thus having an overall performance on ALL, TWE-WSD has 5.54% improvement of $F1$ score on SE13 than SupWSD + emb (the best method on SE13), which can be viewed as a significant improvement because it is the only one among these methods that gets a $F1$ score higher than 0.7 on SE13. The $F1$ score of TWE-WSD is 0.021 higher than SupWSD + emb on ALL which shows the benefits of integrating topical model into TWE-WSD (TWE-WSD can be seen as integrating topical model into SupWSD + emb). The $F1$ score of SupWSD + emb is 0.013 higher than SupWSD, which shows
that integrating embedding into WSD can also improve the performance of WSD. Compared with WSD-TM, another topical model based WSD algorithm, the F1 score of TWE-WSD is 5.82% higher on All. The partial reason may lie in that TWE-WSD is supervised and can exploit word embedding, thus its performance can be significantly improved.

Why TWE-WSD outperforms SupWSD + emb and on which POS tags it work best are further analysed. Experiments on each POS are carried out separately, and the result is shown in Table 5.

As shown in Table 5, TWE-WSD works much better on nouns than any other POS tags. This is probably because nouns are more capable to distinguish topics than the other POS tags. As SE13 has nouns only, while SE15 covers nouns, verbs, adjectives and adverbs, so TWE-WSD works much better on SE13 than SE15. This finding gives us valuable choices for disambiguating datasets with different POS tags.

### 5 | CONCLUSION AND FUTURE WORK

TWE-WSD, a novel TWE-based WSD system is proposed. TWE-WSD is effective by leveraging topic model and word embedding. TWVs instead of WVs are generated to better capture different senses in different contexts. Effective integrating strategies are designed to generate high quality CV. Experimental results show that TWE-WSD achieves substantial improvements over the state-of-the-art techniques. Based on these results, the following major findings are discussed:

1. Integrating word embedding into WSD has a positive effect on improving the performance of WSD.
2. Integrating topical model into WSD can further improve the performance of WSD.
3. Topical model works better on nouns than any other POS tags.

For future work, following three aspects can be focused:

(1) As topical model mainly works well on nouns; some strategies, for example verb subcategorisation [44], can be used to improve the accuracy of TWE-WSD on other POS tags other than nouns. (2) Besides, as neural model-based methods such as Context2Vec works well on SE15 and are complementary to TWE-WSD, so integrating our topical model with neural models may be another promising way to improve the performance on other POS tags. (3) Moreover, hybrid approaches [45,46] can also be introduced to further improve WSD performance.

### ACKNOWLEDGEMENTS

This work was partially supported by the National Natural Science Foundation of China (61562054), the Fund of China Scholarship Council (201908530036) and the Talents Introduction Project of Guangxi University for Nationalities (2014MDQD020).

### REFERENCES

1. Navigli, R.: Word sense disambiguation: a survey. ACM Comput. Surv. 41(2), 1–69 (2009)
2. Tang, G., Sennrich, R., Nivre, J.: An analysis of attention mechanism: the case of word sense disambiguation in neural machine translation. In: Proceedings of the Third Conference on Machine Translation: Research Papers, pp. 26–35. Brussels (2018)
3. Neale, S., et al.: Word sense-aware machine translation: including senses as contextual features for improved translation models. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation, pp. 2777–2783 (2016)
4. Hassan, A.K.A., Hadi, M.J.: Sense-based information retrieval using fuzzy logic and swarm intelligence. Int. J. Multi, Ubiqu. Eng. 12(1), 363–376 (2017)
5. Ruan, T., Grosky, W.: Keyword extraction through contextual semantic analysis of documents. In: Proceedings of the 9th International Conference on Management of Digital Ecosystems, pp. 150–156 (2017)
6. Deng, C., Deng, H., Li, C.: A scholar disambiguation method based on heterogeneous relation-fusion and attribute enhancement. IEEE Access. 8, 28375–28384 (2020)
7. Curran, J.R., Moens, M.: Improvements in automatic thesaurus extraction. In: Association for Computational Linguistics, Philadelphia, PA, pp. 59–66 (2002)
8. Lesk, M.: Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In: Proceedings of the 5th Annual International Conference on Systems Documentation, pp. 24–26 (1986)
9. Banerjee, S., Pedersen, T.: An adapted Lesk algorithm for word sense disambiguation using Wordnet. In: International conference on intelligent text processing and computational linguistics, pp. 136–145 (2002)
10. Basile, P., Caputo, A., Semeraro, G.: An enhanced Lesk word sense disambiguation algorithm through a distributional semantic model.
11. Landauer, T.K., Dumais, S.T.: A solution to Plato’s problem: the latent semantic analysis theory of acquisition, induction, and representation of knowledge. Psychol. Rev. 104(2), 211 (1997)

12. Mikolov, T., et al.: Efficient estimation of word representations in vector space. In: Proceedings of Workshop at ICLR, arXiv:1301.3781v1 (2013)

13. Pennington, J., Socher, R., Manning, C., Glove: global vectors for word representation. In: Association for Computational Linguistics, Doha, Qatar, pp. 1532–1543 (2014)

14. T aghipour, K., Ng, H.T.: Semi-supervised word sense disambiguation using word embeddings in general and specific domains. In: Association for Computational Linguistics, Denver, CO, pp. 314–323 (2015)

15. Collobert, R., Weston, J.: A unified architecture for natural language processing: deep neural networks with multitask learning. In Proceedings of the 25th international conference on Machine learning, pp. 160–167 (2008)

16. Zhong, Z., Ng, H.T.: It makes sense: a wide-coverage word sense disambiguation system for free text. In: Association for Computational Linguistics, Uppsala, Sweden, pp. 78–83 (2010)

17. Rothe, S., Schütze, H.: Autoextend: extending word embeddings to embeddings for synsets and Lexemes. In: Association for Computational Linguistics, Beijing, China, pp. 1793–1803 (2015)

18. Iacobacci, I., Plechvar, M.T.,Navigli, R.: Embeddings for word sense disambiguation: an evaluation study. In: Association for Computational Linguistics, Berlin, Germany, pp. 897–907 (2016)

19. Huang, E., et al.: Improving word representations via global context and multiple word prototypes. In: Association for Computational Linguistics, Jeju Island, Korea, pp. 873–882 (2012)

20. Papandreou, S., Raganato, A., Bovi, C.D.: Supwsd: a flexible toolkit for supervised word sense disambiguation. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 103–108 (2017)

21. Ng, H.T., Lee, H.B.: Integrating multiple knowledge sources to disambiguate word sense: an exemplar-based approach. In: Proceedings of the 34th annual meeting on Association for Computational Linguistics, pp. 40–47 (1996)

22. Lee, Y.K., Ng, H.T.: An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. In: Association for Computational Linguistics, pp. 41–48 (2002)

23. Cai, J., Lee, W.S., Teh, Y.W.: Improving word sense disambiguation using topic features. In: Association for Computational Linguistics, Prague, Czech Republic, pp. 1015–1023 (2007)

24. Papandreou, S., Raganato, A., Delli Bovi, C., Supwsd: a flexible toolkit for supervised word sense disambiguation. In: Association for Computational Linguistics, Copenhagen, Denmark, pp. 103–108 (2017)

25. Melamud, O.,Goldberger, J.,Dagan, I.: Context2vec: learning generic context embedding with bidirectional lstm. In: Proceedings of the 20th SIGNLL conference on computational natural language learning, pp. 51–61 (2016)

26. Kägebäck, M.,Salomonsson, H.: Word sense disambiguation using a bidirectional lstm. In Proceedings of the 5th Workshop on Cognitive Aspects of the Lexicon, pp. 51–56 (2016)

27. Agirre, E., López de Lacalle, O., Soroa, A.: Random walks for knowledge-based word sense disambiguation. Comput. Ling. 40(1), 57–84 (2014)

28. Chaplot, D.S., Bhattacharyya, P., Paranjape, A.: Unsupervised word sense disambiguation using Markov random field and dependency parser, In: Proceedings of the AAAI Conference on Artificial Intelligence. 29(1), 2217–2223 (2015)

29. Moro, A., Raganato, A.,Navigli, R.: Entity linking meets word sense disambiguation: a unified approach. Trans. Assoc. Comput. Linguist. 2, 231–244 (2014)

30. Weissborn, D., et al.: Multi-Objective optimization for the joint disambiguation of nouns and named entities. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pp. 596–605 (2015)

31. Tripodi, R., Pelillo, M.: A game-theoretic approach to word sense disambiguation. Comput. Ling. 43(1), 31–70 (2017)

32. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. J. Mach. Learn. Res. 3, 993–1022 (2003)

33. Boyd-Graber, J.,Blei, D.: Putop: turning predominant senses into a topic model for word sense disambiguation. In: Association for Computational Linguistics, Prague, Czech Republic, pp. 277–281 (2007)

34. Li, L.,Roth, B.,Sporleder, C.: Topic models for word sense disambiguation and token-based idiom detection. In: Association for Computational Linguistics, Uppsala, Sweden, pp. 1138–1147 (2010)

35. Chaplot, D.S.,Salakhutdinov, R.: Knowledge-based word sense disambiguation using topic models. In: Proceedings of the AAAI Conference on Artificial Intelligence, 32(1), (2018)

36. Raganato, A.,Camaero-Collados, J.,Navigli, R.: Word sense disambiguation: a unified evaluation framework and empirical comparison. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Vol. 1, pp. 99–110 (2017)

37. Pal, A.R.,Saha, D.: Word sense disambiguation: a survey. arXiv preprint arXiv:1508.01346 (2015)

38. Hoffman, M.D.,Blei, D.M.,Bach, E.R.: Online learning for latent Dirichlet allocation. Adv. Neural Inf. Process. Syst. 23, 856–864 (2010)

39. Rehurek, R., Sojka, P.: Gensim—statistical semantics in python, statistical semantics; gensim; Python. LDA; SVD (2011)

40. Miller, G.A.,et al.: Using a semantic concordance for sense identification. In: Proceedings of the workshop on Human Language Technology, pp. 240–243 (1994)

41.Navigli, R.,Jurgens, D.,Vannella, D.: Semeval-2013 task 12: multilingual word sense disambiguation. In: Association for Computational Linguistics, Atlanta, GA, pp. 222–231 (2013)

42. Moro, A.,Navigli, R.: Semeval-2015 task 13: multilingual all-words sense disambiguation and entity linking. In: Association for Computational Linguistics, pp. 288–297 (2015)

43. Dongsook, O.,et al.: Word sense disambiguation based on word similarity calculation using word vector representation from a knowledge-based graph. In: Proceedings of the 27th International Conference on Computational Linguistics, pp. 2704–2714. Denver (2018)

44. Roberts, W.,Kondori, V.: Using verb subcategorization for word sense disambiguation. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation, pp. 829–832 (2012)

45. Garg, H.: A hybrid GSA-GA algorithm for constrained optimization problems. Inf. Sci. 478, 499–523 (2019)

46. Garg, H.: A hybrid PSO-GA algorithm for constrained optimization problems. Appl. Math. Comput. 274, 292–305 (2016)

How to cite this article: Jia L, Tang J, Li M, You J, Ding J, Chen Y. TWE-WSD: An effective topical word embedding based word sense disambiguation. CAI Trans. Intell. Technol. 2021;6:72–79. https://doi.org/10.1049/cit2.12006