ChinaRxiv 中X

AI translation · View original & related papers at
chinarxiv.org/items/chinaxiv-202205.00137

Postprint: Text Semantic Disambiguation via SCDV and Anisotropic BERT Tuning

Authors: Li Baozhen, Gu Xiulian

Date: 2022-05-18T16:08:24+00:00

Abstract

Text representation needs to address the ambiguity of textual terms and accurately define their semantic features within specific contextual environments. To tackle the problems of word polysemy and contextual characteristics, we propose an SCDVAB model for text semantic disambiguation. The main innovations are as follows: based on partitioned averaging techniques, transforming scene corpora into document embeddings and introducing anisotropy to improve the sparse composite document vector (SCDV) algorithm for soft clustering, thereby enhancing BERT’ s contextualized representation capability; and using anisotropy-adjusted BERT word embeddings as document embeddings for static word vectors to improve text semantic disambiguation performance. Extensive experiments further demonstrate that the SCDVAB model significantly outperforms traditional text disambiguation algorithms and can effectively enhance the comprehensive performance of text semantic disambiguation.

Full Text

Preamble

Vol. 39 No. 10
Application Research of Computers
Accepted Paper

Text Semantic Disambiguation Based on SCDV and Anisotropy-Adjusted BERT

Li Baozhen, Gu Xiulian
(College of Information Engineering, Nanjing Audit University, Nanjing 211815, China)

Abstract: Text representation requires solving the ambiguity problem of textual words and accurately defining their semantic characteristics within specific contextual environments. Addressing the challenges of polysemy and contextual features, this paper proposes a semantic disambiguation model called SCDVAB. The main innovations are: (a) employing partition averaging techniques to transform scene corpora into document embeddings while introducing anisotropy to improve the sparse composite document vector (SCDV) algorithm for soft clustering, thereby enhancing BERT' s contextualized representation capability; and (b) using anisotropy-adjusted BERT word embeddings as document embeddings for static word vectors to improve text semantic disambiguation ability. Extensive experiments demonstrate that the SCDVAB model significantly outperforms traditional text disambiguation algorithms and effectively improves the comprehensive performance of text semantic disambiguation.

Keywords: semantic disambiguation; anisotropy; BERT; sparse composite document vector; text representation

0 Introduction

Text semantics heavily depends on the words comprising the text. The same word may carry different meanings in different contextual environments, creating ambiguity interference. How to improve text representation accuracy through disambiguation has been a persistent focus in both theory and practice. A series of studies on text representation have shown that weighted averaging of word vectors for sentence representation often outperforms more complex neural models.

SCDV (Sparse Composite Document Vectors) combines word embedding models that can define word scene semantics with latent topic models capable of handling different word senses, thereby enhancing word expressive power. Using soft clustering techniques on embeddings can effectively learn topic feature spaces, while sparse operations on document vectors reduce time and space complexity for vector-based tasks and effectively handle distributed paragraph vectors for text representation.

A significant problem with static word embeddings is that all senses of a polysemous word share a single fixed static vector, making it difficult to effectively resolve polysemy. Replacing static word embeddings with context-based word embeddings can improve word disambiguation effectiveness. For example, deep neural language models like BERT can replace static embeddings with contextualized word embeddings. Through pre-trained BERT models, polysemous words can be placed in semantic spaces with different meanings, outputting different word vectors to solve polysemy issues that static embeddings cannot effectively address, achieving interpretable word sense disambiguation based on contextualized embeddings.

Furthermore, contextualized word representations in BERT exhibit anisotropic characteristics—they are not uniformly distributed in all directions but occupy a narrow cone in vector space [1]. Anisotropy refers to the property where all or part of a text word' s meanings change with variations in semantic space dimension directions, showing differences across different semantic space dimensions. For example, the word "apple" has more prominent projections in fruit-related feature dimensions when the context is fruit, and more prominent projections in electronics-related feature dimensions when the context is electronic products. Less than 5% of the variation in a word' s contextualized representation can be explained by its static embedding [2], providing a necessary rationale for adjusting anisotropy to reduce its impact on contextualized word representation.

Addressing these issues, this paper proposes a simple yet effective unsupervised representation method called the SCDVAB (SCDV+Anisotropy+BERT) model. The main innovations are: (a) converting scene corpora into document embeddings through soft-clustering sparse composite document vector (SCDV) partition averaging techniques; (b) adjusting anisotropy based on self-similarity, intra-similarity, and maximum explainable variance within the SCDV pipeline to improve BERT' s contextualized representation capability; and (c) using anisotropy-adjusted BERT word embeddings as document embeddings for static word vectors to enhance text semantic disambiguation ability. Experimental results demonstrate that the SCDVAB model outperforms existing technologies in accuracy and improves performance on related tasks such as concept matching and semantic text similarity.

1 Related Work

For short text and document representation tasks, word embeddings must be extended to entire paragraphs and documents. In 2014, Le and Mikolov proposed two distributed text representation models: Distributed Memory Paragraph Vector (PV-DM) and Distributed Bag-of-Words Paragraph Vector (PV-DBoW), treating each sentence as a shared global latent vector [3]. These models train word and paragraph vectors to predict context while sharing word embeddings across paragraphs. However, words may have different semantics in different contexts. Vectors for texts containing the same word with different meanings must account for this distinction to accurately represent text semantics. Additionally, although paragraph vectors can contain multiple topics and word senses, they exist in the same space as word vectors and assume all words contribute equally in weight and quality, ignoring word importance and uniqueness across different texts.

Ling mapped word embeddings to latent topic spaces to capture different meanings of word occurrences [4]. However, representing complex documents in the same space as words reduces expressive power. In 2015, Mukerjee et al. proposed idf-weighted averaging of word vectors to form document vectors [5], but assumed all words in a text belong to the same semantic topic. In 2016, Gupta proposed a method using word embeddings and tf-idf values to form composite document vectors called Bag-of-Words Vector (BoWV) [6]. The core idea behind BoWV is that semantically different words belong to different topics, but the model' s word vector averaging setup has certain limitations.

In 2017, Mekala et al. formed Sparse Composite Document Vectors (SCDV) through soft sparse clustering of precomputed word vectors using tf-idf weighting [7]. As a document feature vector formation technique, SCDV overcame some limitations of widely used distributed paragraph vector representations. However, this method largely ignored text word ambiguity issues and contextual semantic feature problems. In 2020, Gupta et al. extended SCDV to SCDV-MS by obtaining multi-sense embeddings on word vectors, emphasizing how polysemous embeddings resolve clustering disambiguation and improve embedding performance, further enhancing SCDV [8]. This demonstrated that disambiguating polysemous words based on context enables better document representation. Gupta also showed that sparsity constraints in clustering are beneficial, and that further improving SCDV' s document representation capability requires enhanced text word disambiguation ability.

To address these limitations, this paper combines pre-trained BERT contextual embeddings as more robust semantic disambiguation-aware word embeddings with SCDV soft clustering and adjusts anisotropy to improve comprehensive performance in text semantic disambiguation, thereby enabling more effective text representation.

2 Model Architecture

The proposed SCDVAB model framework consists of four main modules: (1) corpus contextualization, (2) anisotropy adjustment, (3) word-cluster vector formation, and (4) document representation formation. First, the corpus contextualization module disambiguates each occurrence of a word in the corpus documents, processing every unique word in the corpus. Second, anisotropy adjustment reduces the impact on text word contextualization through the BERT model. Third, the word-cluster vector formation module clusters the contextualized word embeddings obtained in the previous step into k partitions through sparse probability distribution weighting to obtain word-cluster vectors, processing each disambiguated word in the corpus. Finally, the document representation module generates the sparse composite document vector. The specific process is shown in Algorithm 1.

2.1 Corpus Contextualization

The first step in the SCDVAB representation is corpus contextualization, which aims to disambiguate word occurrences in corpus documents through individual interpretation. For example, the word “水分” (moisture) in “植物是靠它的根从土壤中吸收水分” (Plants absorb moisture from soil through their roots) and “他说的话有很大的水分” (There is much exaggeration in his words) has different meanings based on its usage context. Given a word W and its contextual forms $w_1, w_2, \ldots, w_n$ across all corpus texts, we obtain its contextualized embedding representation b_i for each w_i using pre-trained BERT. The word disambiguation problem is treated as a local clustering problem of contextualized word vectors [10]. We cluster the contextualized word embeddings b_i obtained through the pre-trained BERT model, using k-means clustering to group the semantic disambiguation word vectors w_i into k partitions of word W in corpus C, where k represents all possible interpretations of word W across all corpus texts. In contextual semantic space, cosine distance can reflect directional differences, so we use cosine distance between text words as the clustering metric.

Algorithm 1: SCDVAB (SCDV+Anisotropy+BERT) Algorithm
Input: Document D
Output: Document vector V_D

```
For each $w_i$, compute contextualized embedding representation using BERT model;
Compute idf values: idf($W_i$);
Cluster $b_i$ based on K-means model to form $K$ clusters;
Let $C_1^w, C_2^w, \ldots, C_k^w$ serve as center nodes for the $K$ clusters;
Compute conditional dependency probability based on word $W_i$ and cluster $C_k$;
For each word in vocabulary $V$
  For each $k$
    For $n \in (1..N)$ do
      Initialize text vector $V_{D_n}$;
      For word $W_i$ in $D_n$
```

2.2 Word-Cluster Vector Formation

Let $C_1^w, C_2^w, \ldots, C_k^w$ be the k cluster centroids obtained after k-means clustering of word W. These k centroids represent the polysemous meanings of word W. After clustering each occurrence of word W in the corpus, we perform contextualized word sense disambiguation by computing cosine similarity between the BERT representation and centroid embeddings (i.e., b_i and C_j^w) to find the nearest cluster centroid C_k^w, using that sense as the contextual disambiguation word embedding for that occurrence of word W. We designate the nearest cluster centroid to embedding b_i as the contextualized disambiguation word embedding for that occurrence of word W. Repeating this process for all occurrences of word W yields the final contextualized disambiguation word embeddings. Each contextualized embedding of word W serves as a disambiguated word vector.

2.3 Anisotropy Adjustment

The anisotropy adjustment process uses three different metrics to measure a word’ s contextual representation: self-similarity, intra-similarity, and maximum explainable variance [11, 12]. For self-similarity and intra-similarity, the baseline is the average cosine similarity between uniformly randomly sampled word representations from different contexts. For maximum explainable variance (MEV), we compute the proportion of variance explained by the first principal component in uniformly randomly sampled word representations and subtract this proportion from the original MEV. We use the last layer of BERT for word embeddings [13].

Self-similarity refers to the average cosine similarity between contextualized representations across n unique contexts. Let f_s^i be a function mapping w_i to its representation in layer l of model M. The more contextualized a word W is, the lower its self-similarity. A sentence’ s intra-similarity is the average cosine similarity between its word representations and the sentence vector (the average of these word vectors).

Maximum explainable variance is the proportion of variance in contextualized representations for a given layer that can be explained by the first principal component, indicating the extent to which static embeddings can substitute for a word’ s contextual representation. Let W be the event matrix of w_i and σ_i be the singular values of the matrix.

To adjust anisotropy effects, we use three anisotropy baselines, each corresponding to a contextual metric. For self-similarity and intra-similarity, the baseline is the average cosine similarity between uniformly randomly sampled word representations from different contexts. The more anisotropic the word representations in a given layer, the closer this baseline is to 1. For maximum explainable variance, the baseline is the proportion of variance explained by the first principal component in uniformly randomly sampled representations. The more anisotropic the representations in a given layer, the closer this baseline is to 1. We subtract each metric’ s respective baseline value to obtain anisotropy-adjusted contextual metrics. Both original metrics and baselines are estimated using 1K uniformly randomly sampled word representations.

Where O is the set of all word occurrences. Contextualized representations are typically more anisotropic in higher layers [14]. Contextual anisotropy also manifests differently across models. Higher BERT layers show lower average self-similarity; conversely, higher layers produce more specific contextualized representations [15]. Representations of the same word in different contexts still show greater cosine similarity than representations of two different words, with this self-similarity being much lower in upper layers. Upper layers of contextualized models produce more specific contextual representations, similar to how upper layers of LSTMs generate more task-specific representations.

2.4 Document Representation

For each word W_i obtained from pre-trained BERT, we compute its word vector w_i and idf value idf(W_i), where V is the vocabulary size. By introducing soft clustering, we ensure each word belongs to each cluster category with a certain probability $P(C_k|W_i)$.

We compute the probability of a given topic word and given word W_i using Bayes’ rule. For each word W_i in vocabulary V and each cluster C_k, we create k different d-dimensional word-cluster vectors WCV_{ik} by weighting the word’ s probability in the k-th cluster.

We concatenate all V word-cluster vectors WCV_{ik} into a $K \times d$-dimensional embedding and weight it using the inverse document frequency (idf) of W_i to form a contextualized word-topic vector $W_{tV}i$:

$$W_{tV}i = \bigoplus_{k=1}^{K} \mathrm{idf}(W_i) \times W_i \times P(C_k|W_i)$$

where $\bigoplus$ denotes concatenation. We initialize document vector V_{D_n} for document D_n by summing the word-topic vectors of all words appearing in D_n:

$$V_{D_n} = \sum_{i=1}^{j} W_{tV}i$$

Finally, for document D_n, we normalize the vector. Most values in V_{D_n} are very close to zero [16]. We make the document vector sparse by zeroing out attribute values with absolute values close to the threshold, generating the sparse composite document vector:

$$\mathrm{SCDV}_{D_n} = \mathrm{sparse}(V_{D_n})$$

3 Experiments and Analysis

To evaluate the comprehensive performance of the SCDVAB algorithm, we compared its embedding accuracy against other state-of-the-art contextual embedding techniques and conducted experiments on concept matching and semantic textual similarity tasks.

3.1 Experimental Environment

The experimental environment is detailed in Table 1.

Table 1. Experimental Environment

Component	Specification
CPU	Intel® Core™ i7-10710U
Programming Language	Python 3.7.11
OS	Windows 10
IDE	PyCharm
Deep Learning Framework	TensorFlow 2.4.1

3.2 Datasets and Baselines

To analyze contextualized word representations, sentences were input into pre-trained models. Experiments comparing accuracy were conducted on four widely used public classification datasets: (1) Amazon dataset with 4 categories and 8,000 texts; (2) Classic dataset with 4 categories and 7,095 texts; (3) 20NG dataset, a newsgroup text dataset with 20 categories and equal samples per category, totaling 18,846 texts; and (4) Twitter dataset with 3 categories and 3,115 texts.

Baseline methods included doc2vecC, idf-weighted word2vec, BERT, SCDV+word2vec, SCDV+BERT (weighted average), and SCDV+BERT. Notably, SCDV+BERT (weighted average) was set as a baseline to analyze whether word vectors based on word sense disambiguation can more effectively capture multiple word meanings. The SCDV+BERT baseline was established to analyze the impact of reduced anisotropy. We used $k = 6$ with anisotropy adjustment. Baseline results were taken from the experimental section of Gupta et al., 2020 [17].

The concept matching task associates concepts with related items. The concept matching dataset includes 537 item-concept pairs for 53 unique concepts from Next Generation Science Standards (NGSS) and 230 unique items from Science Buddies. Experiments compared cosine similarity with TF-IDF weighted vectors, SCDV+Word2Vec, and pre-trained BERT baselines. Baselines were taken from the 2020 experimental section of Zhang and Danescu-Niculescu-Mizil [18].

The sentence similarity task computes semantic similarity between two texts. Experimental input data came from 27 semantic textual similarity (STS) tasks spanning 2012-2016 [19]. The dataset includes 4 to 6 STS tasks per year, detailed in Table 2. These datasets were selected because they contain sentences with the same words appearing in different contexts, with multiple polysemous words in all datasets. Baselines were taken from Perone et al., 2018 [20], Devlin et al., 2019 [21], and Gupta et al., 2020 [17].

Table 2. STS Tasks

Year	Tasks
STS12	MSRpar, headlines, deft-forum, answer-forums
STS13	headlines, OnWN, deft-news, answers-students
STS14	plagiarism, SMT-eur, headlines, belief, postediting
STS15	ans-ans, ques-ques, SMT-news, images, Tweet-news, headline-images
STS16	Various tasks

The superiority of the SCDVAB model on concept matching tasks indirectly demonstrates its advantages in resolving text word ambiguity and accurately defining semantic features of words in specific contextual environments.

Table 5. Comparison of SCDVAB with Latest Embedding Technologies on Various STS Tasks

(Shows Pearson correlation coefficients multiplied by 100 across different models and tasks)

3.3 Experimental Settings

We used the BERT base pre-trained model to obtain word embeddings and K-means for contextual clustering of given words. For simplicity, experiments used a similarity threshold (τ) of 0.8 for all data, resulting in multiple polysemous representations per word. We analyzed the distribution of similarity degrees, excluding words appearing in fewer than 5 unique contexts. Training and test sets were split 80/20. For SCDV, word embedding dimensions were set to 200, $k = 6$ for anisotropy adjustment, and 5-fold cross-validation was used to tune SCDV’ s sparse threshold.

3.4 Experimental Results and Analysis

Table 3 shows the accuracy performance of SCDVAB versus other baseline models on the four datasets, with results averaged over 10 training runs. The results demonstrate that SCDVAB outperforms all other contextual text representation methods across all datasets.

Table 3. Accuracy Comparison Between SCDVAB and Baselines

Embedding Method	Amazon	Classic	Twitter
Doc2vecC	-	-	-
Word2vec (idf-weighted)	-	-	-
SCDV+word2vec	-	-	-
BERT (weighted average)+SCDV	-	-	-
BERT+SCDV	-	-	-
SCDVAB	-	-	-

Analysis of Table 3 reveals that contextualized BERT+SCDV performs better than weighted-average BERT+SCDV. While simple weighted averaging of word vectors often produces effective sentence representations, it is less effective for longer texts containing multiple sentences that may include words from numerous topics. Experimental results show that word vectors based on word sense disambiguation can capture multiple word meanings, proving the contribution of semantic disambiguation. Furthermore, SCDVAB achieves accuracy improvements of 0.85%, 1.72%, 1.2%, and 1.06% over BERT+SCDV across the four datasets, respectively, demonstrating the advantageous impact of anisotropy adjustment. SCDVAB’ s superior performance over BERT (weighted average)+SCDV indicates that SCDVAB’ s word vectors based on word sense disambiguation can effectively capture polysemous words, and anisotropy adjustment enhances contextualized representation capability, better aligning with corpus context.

Table 4. Comparison of Concept Matching Accuracy and F1 Values

Embedding Method	Accuracy	F1
TF-IDF	-	-
Word2vec+SCDV	-	-
BERT+SCDV	-	-
SCDVAB	-	-

Based on Table 4, SCDVAB achieves accuracy and F1 improvements of 4.2% and 4% over pre-trained BERT, and 5.3% and 4.6% over Word2Vec+SCDV, respectively. Compared to BERT+SCDV, SCDVAB shows improvements of 1.8% and 0.8% in accuracy and F1, proving the importance of considering anisotropy.

Table 5 compares SCDVAB with various state-of-the-art embedding technologies. The experimental data represents Pearson correlation coefficients multiplied by 100. SCDVAB significantly outperforms other baseline models across datasets, proving the effectiveness of the improved model. The results show that algorithm models incorporating SCDV achieve better performance than other algorithms, primarily because SCDV extends representation capability from sentences to texts through soft sparse clustering of pre-trained word vectors, demonstrating SCDVAB' s superiority in leveraging SCDV. BERT+SCDV shows slight improvement over Word2vec+SCDV but remains inferior to the enhanced SCDVAB, as SCDVAB considers the impact of anisotropy adjustment on BERT word sense disambiguation.

To validate SCDVAB' s performance superiority, we present sample similarity visualizations and analysis from the MSRvid task in the STS12 dataset, with sample descriptions shown in Table 6 (standardized data).

Table 6. STS12 MSRvid Dataset Similarity Example Pairs

Sentence 1	Sentence 2	PSIF	BERT+SCDV	SCDVAB
Runners race around a track.	Runners compete in a track.	-	-	-
A man is riding a motorcycle.	A woman is riding a horse.	-	-	-
People are playing baseball.	The cricket player hit the ball.	-	-	-
An animated airplane is landing.	A plane is landing.	-	-	-

Table 6 shows that similarity scores from SCDVAB are closer to given similarities than other models, proving the improved model' s superiority in computing semantic similarity between two texts.

Table 7 presents experimental results on text similarity tasks from STS16 to further validate SCDVAB' s performance improvements.

Table 7. Experimental Results on Textual Similarity Tasks on STS16

Tasks	Skip thoughts	PSIF+PSL	BERT+SCDV	SCDVAB
headlines	-	-	-	-
plagiarism	-	-	-	-
Post editing	-	-	-	-
ans-ans	-	-	-	-
ques-ques	-	-	-	-
STS16	-	-	-	-

The results show that the improved model outperforms other algorithms on all datasets in STS16 tasks, proving SCDVAB' s superiority. PSIF+PSL outperforms skip-thoughts because P-SIF learns topic-specific vectors from text, considering text topic structure and utilizing partition averaging technology. Skip-thoughts, based on skip-gram, lacks consideration of semantic features in specific contexts. BERT+SCDV performs similarly to PSIF+PSL but slightly worse, possibly because unimproved BERT has text length limitations while PSIF+PSL is more targeted for long texts. SCDVAB' s improvement over BERT+SCDV further demonstrates the importance of considering anisotropy.

4 Conclusion

Considering the need to resolve word ambiguity in text representation and define semantic features of words in specific contextual environments, this paper proposes the SCDVAB algorithm model for text semantic disambiguation. By pre-training BERT contextualization and reducing anisotropy effects to enhance sparse text representation (SCDV), we provide a more efficient and accurate text representation method for contextual document representation. Using anisotropy-adjusted BERT semantic disambiguation word vectors with SCDV to convert them into text feature vectors can accurately represent semantic features of words in specific contexts, demonstrating strong practical significance. Experimental results show that SCDVAB outperforms other unsupervised methods and excels in comprehensive text semantic disambiguation performance. The model can effectively improve text representation-related tasks such as multi-topic long text representation, multi-scenario text concept disambiguation, and extractive reading comprehension.

References

[1] Mekala D, Zhang X, Shang J. META: Metadata-empowered weak supervision for text classification [C]// Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing. [S. I.]: Association for Computational Linguistics, 2020: 8351-8361.

[2] Liu Huan, Zhang Zhixiong, Wang Yufei. Review on the main optimization and improvement methods of Bert model [J]. Data Analysis and Knowledge Discovery, 2021, 5 (1): 3-15.

[3] Gong H, Sakakini T, Bhat S, et al. Document similarity for texts of varying lengths via hidden topics [C]// Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, 2018: 2341-2351.

[4] Liu Shengjie, Xu Liang. Summary of research status of text representation based on word embedding technology [J]. Modern Computer, 2020, 673 (1): 40-43.

[5] Jiao Fenfen. Text clustering algorithm based on concept and semantic similarity [J]. Computer Engineering and Application, 2012, 48 (18): 136-141.

[6] Wang Ruiqin, Kong Fansheng. Research on unsupervised word sense disambiguation [J]. Journal of Software, 2009, 20 (8): 2138-2152.

[7] Mekala D, Gupta V, Paranjape B, et al. SCDV: Sparse Composite Document Vectors using clustering over distributional representations [C]// Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. Copenhagen: Association for Computational Linguistics, 2017: 659-669.

[8] Gupta V, Saw A, Nokhiz P, et al. Improving document classification with multi-sense embeddings. [C]/ Proceedings of the European Conference on Artificial Intelligence, 2020. (2020-11) [2022-2-20]. http://10.48550/arXiv.1911.07918.

[9] Ethayarajh K. How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings. [C]// Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing. Hong Kong: Association for Computational Linguistics, 2019: 55-65.

[10] Matthew P, Mark N, Mohit I, et al. Deep contextualized word representations [C]// Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics. [S. I.]: Human Language Technologies, 2018 (1): 2227-2237.

[11] Yosinski J, Clune J, Bengio Y, et al. How transferable are features in deep neural networks? [J]. Advances in Neural Information Processing Systems. 2014 (11): 3320-3328.

[12] Bhatia K, Jain H, Kar P, et al. Sparse local embeddings for extreme multi-label classification [J]. Advances in Neural Information Processing Systems, 2015 (1): 730-738.

[13] Yu Meng, Shen Jiaming, Zhang Chao, et al. Weakly-supervised hierarchical text classification [J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2019 (33): 6826-6833.

[14] Ye Xuemei, Mao Xuemin, Xia Jinchun, et al. Improvement of TF-IDF algorithm for text classification [J]. Computer Engineering and Application, 2019, 55 (2): 104-111.

[15] Dai Hongtao, Hou Kaihu, Zhou Zhou, et al. Word sense disambiguation method based on VCK vector model [J]. Software, 2020, 41 (2): 134-140.

[16] Wang Rui, Li Bicheng, Du Wenqian. Entity disambiguation method based on context word vector and topic model [J]. Chinese Journal of Information Technology, 2019, 33 (11): 46-56.

[17] Gupta V., Saw A., Nokhiz P., et al. P-SIF: Document embedding using partition averaging [J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34 (5): 7863-7870.

[18] Zhang J, Danescu-Niculescu C. Balancing objectives in counseling conversations: Advancing forwards or looking backwards [C]// Proceedings of the 58th Annual Meeting of the Association for Computational Linguists, 2020: 5276-5289.

[19] Kim, H K, Kim H, Cho S. Bag-of-concepts: Comprehending document

representation through clustering words in distributed representation [J]. Neurocomputing, 2017 (266): 336-352.

[20] Perone, C. S., Silveira, R., Paula, T. S. Evaluation of sentence embeddings in downstream and linguistic probing tasks [J]. 2018. arXiv preprint arXiv: 1806. 06259.

[21] Devlin, J., Chang, M-W., Lee, K., et al. Bert: Pre-training of deep bidirectional transformers for language understanding [C/OL]. NAACL, 2018. (2018-10-11) [2022-2-20]. https://arxiv.org/abs/1810.04805.

Note: Figure translations are in progress. See original paper for figures.

Source: ChinaXiv —Machine translation. Verify with original.