

Adaptive cross-contextual word embedding for word polysemy with unsupervised topic modeling

Shuangyin Li^a, Rong Pan^b, Haoyu Luo^{a,c}, Xiao Liu^d, Gansen Zhao^{a,c}

^a School of Computer Science, South China Normal University, Guangzhou, Guangdong, China

^b School of Data and Computer Science, Sun Yat-sen University, Guangzhou, Guangdong, China

^c Key Lab on Cloud Security and Assessment technology of Guangzhou, Guangdong, China

^d School of Information Technology, Deakin University, Geelong, Australia

ARTICLE INFO

Article history:

Received 14 March 2020

Received in revised form 31 January 2021

Accepted 2 February 2021

Available online 19 February 2021

Keywords:

Word polysemy

Representation learning

Adaptive word embeddings

Tailored word embedding

Topic modeling

Semantic learning

ABSTRACT

Because of its efficiency, word embedding has been widely used in many natural language processing and text modeling tasks. It aims to represent each word by a vector so such that the geometry between these vectors can capture the semantic correlations between words. An ambiguous word can often have diverse meanings in different contexts, a quality which is called polysemy. The bulk of studies aimed to generate only one single embedding for each word, whereas a few studies have made a small number of embeddings to present different meanings of each word. However, it is hard to determine the exact number of senses for each word, as meanings depend on contexts. To address this problem, this paper proposes a novel adaptive cross-contextual word embedding (ACWE) method for capturing the word polysemy in different contexts based on topic modeling, in which the word polysemy is defined over a latent interpretable semantic space. The proposed ACWE consists of two main parts, in the first of which an unsupervised cross-contextual probabilistic word embedding model is designed to obtain the global word embeddings, and each word is represented by an embedding in the unified latent semantic space. Based on the global word embeddings, an adaptive cross-contextual word embedding process is then devised in the second part to learn the local embeddings for each polysemous word in different contexts. In fact, a word embedding is adaptively adjusted and updated with respect to different contexts to generate different word embeddings tailored to the corresponding contexts. The proposed ACWE is validated on two datasets collected from Wikipedia and IMDb on different tasks including word similarity, polysemy induction, semantic interpretability, and text classification. Experimental results indicate that ACWE does not only outperform the established word embedding methods, which consider word polysemy on six popular benchmark datasets, but it also yields competitive performance compared with state-of-the-art deep learning-based approaches without considering polysemy. Moreover, the proposed ACWE significantly improves the performances of text classification both in precision and F1, and the visualizations of the semantics of words demonstrate the feasibility and advantage of the proposed ACWE model on polysemy.

© 2021 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

Representing words as dense or sparse embeddings makes it possible to improve many language understanding tasks and provides the foundation for word recognition. These word embeddings can be employed to measure word similarities by computing distances between the corresponding embeddings, which are widely used in many applications such as information retrieval, text classification, and neural language processing (NLP) tasks [1]. Recently, many models have been proposed to learn

effective word embeddings, such as word2vector (Skip-Gram and CBOW) [2], GloVe [3], non-negative sparse embedding (NNSSE) [4] and ELMo [5]. In the literature, Bert [6] and transformer-based methods [7], e.g., XLNET [8] and RoBERTa [9], have achieved great successes on many NLP tasks, which also provide ways to learn word representation. To satisfy different tasks, these methods benefit from large-scale corpora to learn high-quality and unique word embeddings.

However, due to homonymy and polysemy, it is obvious that modeling an individual polysemous word with a single embedding is insufficient. Many words have different senses in various contexts, where each sense captures one semantic in a special context. Recent studies have analyzed how to develop multiple embeddings for a polysemous word of various senses. To this end,

E-mail addresses: shuangyinli@scnu.edu.cn (S. Li), panr@sysu.edu.cn (R. Pan), luohy@whu.edu.cn (H. Luo), xiao.liu@deakin.edu.au (X. Liu), gzhao@m.scnu.edu.cn (G. Zhao).

some studies [10,11] have designed cluster-based models that would conduct unsupervised word sense induction by clustering word contexts based on nonparametric clustering models [12]. The main drawback of those models is the difficulty of determining the number of senses for a polysemous word. To handle this limitation and to construct multiple senses for a word, another type of methods [13,14], e.g., WordNet and Wikipedia, have used extra knowledge bases. In these methods, the number of meanings is determined according to pre-defined knowledge bases. However, such methods fail to handle new or tailored meanings of words that appear in new contexts with the meanings not elaborated in the knowledge bases. Thus, it is essential to find a complete solution to capture the polysemy of words based on their contexts.

The word senses are usually adaptable and are adjusted based on different contexts. Intuitively, encountering a word in a particular context, a reader would judge its meaning according to its context and the original senses that the reader has already learned. When the context changes, the reader repeats this procedure. Therefore, a practical and flexible solution is that the number of different senses for a polysemous word should not be limited and adjusted. Instead, the embeddings of a polysemous word should be identified and updated based on contexts. To achieve this, it is first necessary to learn global embeddings for polysemous words in a multi-semantic space, and then the global embeddings in the multi-semantic space should be adjusted and updated adaptively to generate local embeddings in specific contexts. This is similar to the human learning process by which people learn words first and then their specific meanings in different documents.

Hence, with this inspiration, this paper proposes an Adaptive Cross-contextual Word Embedding (ACWE) model based on topic modeling. The ACWE can capture and represent the polysemous words in an interpretable latent semantic space. In the model each semantics denotes a sense cluster that is defined as a distribution over the vocabulary. A polysemous word defines a probability distribution over all the latent semantics which represent the global word embedding. The latent semantic space is defined by sentence-level learning through a similar approach to topic models. In addition, attention signals are considered while learning the semantic distribution of a word with its contextual words in a sentence. Given different contexts, this paper proposes an adaptive context-based word embedding process to tailor a word embedding to generate different local embeddings. In particular, the proposed adaptive context-based word embedding process infers a targeted polysemous word by using its global embedding and neighboring words in a sentence to obtain a newly adaptive embedding for the targeted word where neighboring words are treated as the context of the targeted word in the sentence.

This work conducts experiments to validate the proposed ACWE with two public available datasets, Wikipedia and IMDb. To evaluate the effectiveness of the proposed model, experiments are conducted on the tasks of word similarity, text classification, word polysemy and word interpretability. With respect to the tasks of word similarity, the experimental results show that the ACWE significantly outperforms the existing methods on six benchmark datasets. Meanwhile, the proposed model is tested on Wikipedia and IMDb to demonstrate the capacity of text classification, and the proposed ACWE method yields state-of-the-art performance, and the results are significantly better than those obtained by the other word embedding methods on precision and F1. Moreover, by following the methodology of polysemy induction [15], the experiments on the tasks of text classification show that the proposed ACWE is effective on word polysemy. The visualizations of word embeddings on word polysemy and

interpretability corroborate that the proposed method is capable of capturing multi-semantics of the polysemous words, which is crucial for the tasks of homonymy and polysemy.

The main contributions of this work can be summarized as follows.

1. To tackle the issue of word polysemy, this paper proposes a novel adaptive cross-contextual word embedding (ACWE) method based on topic modeling, which is able to learn an unlimited number of tailored word embeddings for a targeted polysemous word in different contexts.
2. An adaptive context-based word embedding process is proposed, by which the proposed ACWE is able to adaptively generate local and tailored word embeddings in different contexts for a polysemous word.
3. The proposed ACWE embeds the words into a nonnegative semantic space, which leads up a fruitful perspective for word representation learning, where each word embedding is highly interpretable since each semantic is defined by a distribution over explicable vocabulary.
4. An online algorithm is also proposed that allows the ACWE to be employed in different scenarios of the stream documents to make it efficient and easy to use. It can help the proposed ACWE to be trained on the large-scale corpus, such as Wikipedia.

The remainder of the paper is organized as follows. Section 2 surveys the related research on word embedding and polysemy. Section 3 proposes an adaptive cross-contextual embedding process to tackle the issue of word polysemy. Section 4 elaborates the model inference and the online learning algorithm. Section 5 presents the model analysis and comparisons. Section 6 reports the experiments on word similarity, polysemy, text classification tasks, and the case studies on word embedding interpretability. Section 7 concludes the paper.

2. Related works

In the literature, most existing works focus on learning word embeddings. For example, Bengio et al. [16] extended the traditional n -gram language models with a neural network. Tomas et al. [2] presented a computationally efficient log-linear neural language model to obtain word embeddings, named word2vector (Skip-Gram and CBOW). Pennington et al. [3] presented GloVe to obtain embedding for words by aggregating global word-word co-occurrence statistics. Murphy et al. [4] proposed non-negative sparse embedding (NNSE), which is a variant of matrix factorization to embed words into a nonnegative semantic space. The most important limitation of this method is that it does not consider the word polysemy. Several studies, such as Sparse Coding [17] and Sparse CBOW [18], have tried to embed words into a sparse space. Meanwhile, many efforts have been made to learn word representation through different technology [19,20]. Recently, several researchers have proposed the use of neural network-based techniques for word embedding called Bidirectional Encoder Representations from Transformers (BERT) [6]. These techniques process words in relation to all the other words in a sentence, rather than one-by-one. BERT achieved great success in many real applications with several variants, such as XLNET [8], RoBERTa [9], and SensEmBERT [21]. These transformer-based methods are the context-aware representation methods, where the word embeddings are learned through modeling the contexts of each word. Cove [22] utilized a neural machine translation encoder to compute contextualized representations. Context2vec [23] used a bidirectional LSTM to encode the context around a pivot word.

Handling words with multiple meanings, so-called polysemous words, has been an interest research topic in the literature. Reisinger et al. [10] introduced a method for constructing multiple sparse, high-dimensional vector representations of words by assigning a real-value vector to each meaning. Huang et al. [11] proposed a word embedding model by leveraging the global context information to learn multi-prototype embeddings. The method provided by them clusters embeddings of all the context words of a word in the corpus. Although many works [24–28] have been studied the multi-sense words on word similarity tasks, the probabilistic models [29–31], bilingual resources [32], or nonparametric models [12,25] have been explored for word polysemy tasks. Wu et al. [33] disambiguated sense embeddings from Wikipedia by clustering its documents. Chen et al. [13] used the WordNet dictionary to predefine word senses. Liu et al. [34] assumed that a single word embedding can be considered as a mixture of different word senses and then used context-sensitive word embedding to learn distributed representations of words based on the Skip-Gram. Sanjeev et al. [14] showed that each extracted word sense is accompanied by one of about thousands of “discourse atoms” that gives a succinct description of which other words co-occur with that word sense. Bahar et al. [35] suggested using word embeddings to predict combinations of multi-word expressions, taking into account both single and multi-prototype word embeddings. Terry et al. [36] proposed a novel approach called Most Suitable Sense Annotation, that disambiguates and annotates each word by its specific sense, considering the semantic effects of its context. Ben et al. [37] introduced a probabilistic FastText model for word embeddings that can capture multiple word senses, sub-word structure, and uncertainty information, where each word is represented by a Gaussian mixture density. Kazuki et al. [38] proposed a method to generate multiple word representations for each word based on dependency structure relations. Meanwhile, some researchers have focused on how to determine whether a word has different meanings [39–42].

The common feature of all the methods examined is that the word embeddings are fixed after the model training. Peters et al. [5] presented a deep contextualized word representation model (ELMo), which can fit the word representations by the contexts through a pre-trained bidirectional language model. However, the interpretability of the word representation still needs to be considered. Due to the ability to capture syntactic and semantic information from text, the topic model is a standard component of most state-of-the-art NLP architectures, including document modeling [43], sentence modeling [44], and word representation learning [45–49]. Li et al. [50] proposed a novel approach to learn the topics of the documents through the semantics of the sentences, which fully utilized the bi-directional sequential information of the sentences in a document. The benefit of topic modeling is that the semantics in corpora can be extracted through unsupervised learning [51,52] and the semantics are interpretability. This provides a way to learn semantics for different components, such as paragraphs and words. We thus take advantage of topic models to build the semantic space for the multiple senses of polysemous words and represent each polysemous word by a semantic embedding, a process which is different from the above works. Moreover, the semantic embeddings for polysemous words are adaptive in different contexts.

3. Adaptive cross-contextual word embedding (ACWE)

Word polysemy is common; however, it is not a common practice for word embedding to capture and represent the polysemy that lies in different contexts. This paper proposes an adaptive cross contextual word embedding (ACWE) method to tackle with this issue. To this end, this section describes how the ACWE employs cross-contextual information to generate word embeddings for word polysemy.

3.1. Overview

To capture the multi-senses of a polysemous word, the ACWE aims to embed words into a continuous semantic space. The latent semantic space can be extracted through unsupervised document modeling such as topic models [43,53]. Topic modeling is a technology for text modeling based on generative probabilistic models. For instance, the latent Dirichlet Allocation (LDA) [43] presents a three-level hierarchical Bayesian model in which each document is defined as a finite mixture over a set of latent topics. Topics are defined as the distributions over words in the dictionary, and each topic can be treated as a kind of semantics. A document is represented by a group of topic probabilities providing an explicit representation.

Inspired by topic models, the ACWE embeds polysemous words into such a latent semantic space as well as documents. The main advantage is that the multi-sense of one polysemous word can be represented by all semantics in the latent space. This method is completely different from existing polysemous word embedding models, as they assume that there is a fixed and limited number of senses for each word. Although not all the words in the dictionary are polysemous, it is still necessary to define the semantic distributions for every word. Under this assumption, we can capture the multiple senses of polysemous words and also obtain word embeddings for the non-polysemous words. Evidently, the senses of one word are related to its context. A sentence is defined as the context of the targeted word for the ease of presentation. By considering a document as a set of sentences, each sentence can be treated as a bag of words, where the words order can be neglected.

Fig. 1 shows the whole process of the proposed ACWE consisting of two main steps. The first step is to train a cross-contextual probabilistic word embedding model (see Section 3.2). The adaptive cross-contextual word embedding process is then implemented (see Section 3.3). In detail, it is first necessary to build an unsupervised cross-contextual probabilistic word embedding model, which benefits from sentences in large scale documents to learn the global embeddings for all words in the dictionary. The global embedding for each word is represented by the semantic distribution over the latent topic space obtained from topic modeling. Each dimension of the global embedding denotes an interpretable semantic. In the second step, an adaptive cross-contextual embedding process is performed to adaptively update the word embeddings with different contexts. This process is able to generate a tailored word embedding for a targeted word in a special context. With the proposed adaptive cross-contextual embedding process, the proposed ACWE can obtain unlimited word embeddings for polysemous words in different contexts.

The global word embedding for a polysemous word contains all the senses or semantic that appeared in the corpus, whereas the multiple senses of the polysemous word are represented by the probabilities of its semantic aspects. With special contexts, these probabilities will be adjusted to form the local or tailored word embeddings. Thus, the global word embeddings are first built, and the tailored word embeddings are then generated with the corresponding contexts.

3.2. Cross-contextual probabilistic word embedding

Let $C = \{\mathbf{d}^1, \mathbf{d}^2, \dots, \mathbf{d}^M\}$ denote a corpus which contains M documents, where $\mathbf{d}^i, i \in \{1, \dots, M\}$ denotes the i th document in the corpus. A document \mathbf{d}^i is defined as a set of sentences denoted by $(\mathbf{s}_1^i, \dots, \mathbf{s}_{S^i}^i)$, where S^i is the number of sentences in \mathbf{d}^i . Each sentence \mathbf{s}_j^i in \mathbf{d}^i is denoted by $\mathbf{s}_j^i = (w_{j,1}^i, \dots, w_{j,N^i}^i)$ with the

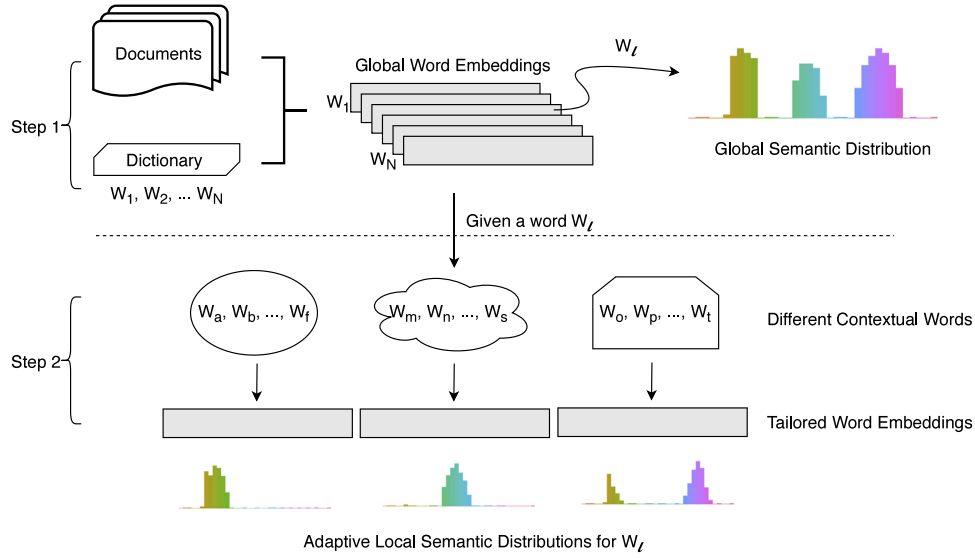


Fig. 1. Overview of the ACWE.

assumption of bag-of-words, where N_j^i is the number of words in sentence \mathbf{s}_j^i . \mathbf{v} denotes the dictionary in the corpus C , where the word is indexed by $\{1, \dots, V\}$.

We define $\theta \in \mathbb{R}^{V \times K}$ as the semantic matrix of the words over latent semantics, where each row θ_n is the embedding of the word w_n , $n \in \{1, \dots, V\}$ and K is the number of the latent semantics. Note that all the words in the dictionary are considered to be polysemous, and each of them defines an individual semantic distribution. Let $\beta \in \mathbb{R}^{K \times V}$ denote the matrix of the latent semantics which defines the distributions over dictionary as in LDA [43]. ϑ_j^i is defined as the distribution of the sentence \mathbf{s}_j^i over latent semantics in document \mathbf{d}^i , and λ^i , a $1 \times K$ row vector, is defined as the semantic distribution of \mathbf{d}^i .

Here we introduce the attention mechanism into our model. The attentional mechanism used in deep neural networks [54] and topic models [55] is a popular approach to model important weights among the signals. We assume that the semantic distribution of sentence ϑ_j^i is determined by semantic distributions of the words in it with different attentional values, which means that ϑ_j^i is a weighted average of the semantic distributions of its words. Here we let ϵ_j^i denote the attentional vector of the words in the sentence \mathbf{s}_j^i .

It is intuitive that the semantics of a sentence is also affected by that of the host document to which the sentence belongs. Hence, the semantic distribution of the sentence is generated from those of both its own words and the host document. So we can obtain the semantic distribution of the sentence, ϑ_j^i , with Definition 1.

Definition 1. For $\theta \in \mathbb{R}^{V \times K}$, $\lambda^i \in \mathbb{R}^{1 \times K}$ and $\epsilon_j^i \in \mathbb{R}^{(N_j^i+1) \times 1}$, the semantic distribution of sentence \mathbf{s}_j^i , $\vartheta_j^i = (\epsilon_j^i)^T \times \left[\frac{\theta_{\mathbf{v}^{w_{ij}}}}{\lambda^i} \right]$, where $[\cdot]$ is an operation to stack two matrices into a bigger matrix.

In Definition 1, ϵ_j^i is an $(N_j^i + 1) \times 1$ attentional vector, $\mathbf{v}^{w_{ij}}$ denotes the word indices in the dictionary \mathbf{v} for the words in \mathbf{s}_j^i , and $\theta_{\mathbf{v}^{w_{ij}}}$ is an $N_j^i \times K$ submatrix of θ according to $\mathbf{v}^{w_{ij}}$. The element $\epsilon_{j,l}^i$, $l \in \{1, \dots, N_j^i\}$, in ϑ_j^i is the attention value of the word w_l in sentence \mathbf{s}_j^i . The element $\epsilon_{j,(N_j^i+1)}^i$ is the attention value of the host document.

Fig. 2 shows the graphical model of the cross-contextual probabilistic word embedding model for document \mathbf{d}^i . The generation

process of it for each document \mathbf{d}^i for $i \in \{1, \dots, M\}$ is defined as follows.

1. Draw $\lambda^i \sim \text{Dir}(\alpha)$;
2. For sentence \mathbf{s}_j^i , $j \in \{1, \dots, S^i\}$ in the document \mathbf{d}^i :
 - (a) Draw $z_{j,1}^i \sim \text{Mult}(\lambda^i)$, and draw $w_{j,1}^i \sim \text{Mult}(\beta_{z_{j,1}^i})$;
 - (b) Generate ϑ_j^i with Definition 1;
 - (c) For each word $w_{j,l}^i$, $l \in \{2, \dots, N_j^i\}$ in sentence \mathbf{s}_j^i :
 - i. Draw $z_{j,l}^i \sim \text{Mult}(\vartheta_j^i)$ and draw $w_{j,l}^i \sim \text{Mult}(\beta_{z_{j,l}^i})$;
 - ii. update ϑ_j^i with $w_{j,l}^i$;

In this process, $\text{Dir}(\cdot)$ denotes a Dirichlet distribution, and $\text{Mult}(\cdot)$ is a multinomial distribution. α is a parameter of a Dirichlet distribution. Each row in θ is defined as a distribution over latent semantics, which follows a Dirichlet distribution. Thus, the n th row in θ satisfies $\sum_{k=1}^K \theta_{nk} = 1$. It is noted that the generative process starts with generating the first word based on the semantic distribution of the host document and avoids the loop between generating the words and the sentence semantics.

Note that the dimension of attention vector ϵ_j^i depends on the number of words in the respective sentence. Thus, we let each attention vector follow a Dirichlet distribution, and the hyperparameter of the Dirichlet distribution is from a global vector $\pi \in \mathbb{R}^{1 \times (V+1)}$ corresponding to the word index in the host sentence. The last element π_{V+1} is a hyperparameter for the attentional value of the host document. In sentence \mathbf{s}_j^i , ϵ_j^i satisfies $\sum_{l=1}^{N_j^i+1} \epsilon_{j,l}^i = 1$.

After model learning, we can obtain two matrices, θ and β . θ represents the word embedding matrix, which contains basic word embeddings. Thus, the proposed cross-contextual probabilistic word embedding model takes advantage of the global context and the local contexts when learning the basic word embeddings. From the perspective of the generative process, each word is generated by the semantic distribution of the corresponding sentence, which is the local context. As shown in Definition 1, the semantic distribution of the sentence is affected by the semantic distribution of the host document, which can be treated as the global context.

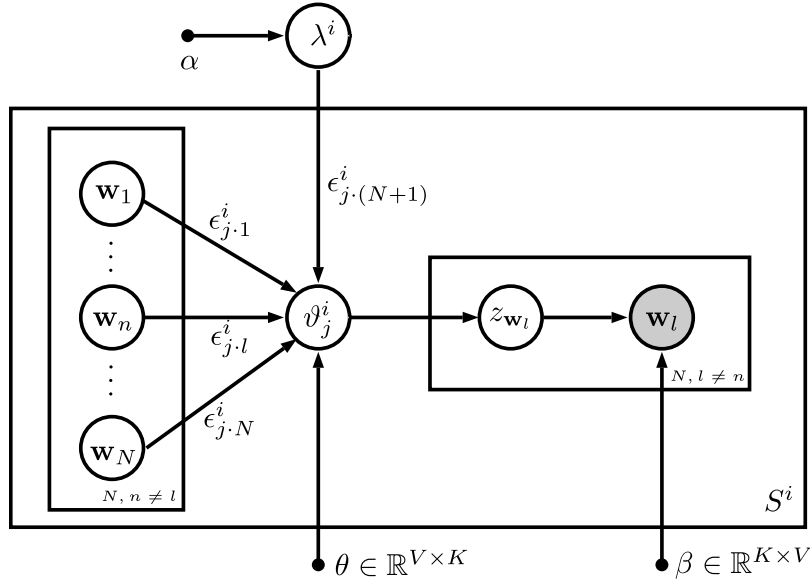


Fig. 2. The graphical model representation of the cross-contextual probabilistic word embedding model for d^i .

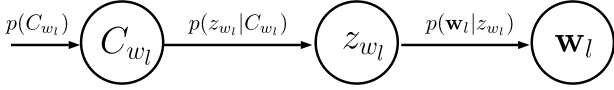


Fig. 3. The graphical model representation of the adaptive cross-contextual word embedding process.

3.3. Adaptive cross-contextual word embedding process

Consider a word \mathbf{w}_l and its contextual word set $\{\mathbf{w}_n\}_{n=1}^N$. We define the word set from the host sentence \mathbf{s} of \mathbf{w}_l , where the host sentence is treated as the context for convenience. Let $C_{\mathbf{w}_l}$ denote the context of \mathbf{w}_l in its host sentence \mathbf{s} , which contains a list of words $\{\mathbf{w}_n\}_{n=1, n \neq l}^N$. Within the generative process of the above cross-contextual probabilistic word embedding model, \mathbf{w}_l is generated by a semantics z with its distribution over the dictionary, and the semantics z is generated by the semantic distribution of the sentence, ϑ , which is obtained by the weighted average of the semantic distributions of the other words as shown in Definition 1.

By Bayes' theorem with hidden variables, we can easily obtain the conditional marginal distribution of the word \mathbf{w}_l given a set of observed variables $C_{\mathbf{w}_l}$ with an assigned semantic $z_{\mathbf{w}_l}$ as

$$p(\mathbf{w}_l | C_{\mathbf{w}_l}; z_{\mathbf{w}_l}) \propto p(\mathbf{w}_l | z_{\mathbf{w}_l}) \cdot p(z_{\mathbf{w}_l} | C_{\mathbf{w}_l}) \cdot p(C_{\mathbf{w}_l}) \\ \propto p(\mathbf{w}_l | z_{\mathbf{w}_l}) \cdot p(z_{\mathbf{w}_l} | C_{\mathbf{w}_l}),$$

where $z_{\mathbf{w}_l}$, the semantic index for \mathbf{w}_l , is a hidden variable and $p(C_{\mathbf{w}_l})$ is a constant with respect to the observed $C_{\mathbf{w}_l}$. The corresponding graphical model representation is depicted in Fig. 3.

Thus, we can get the semantic distribution of \mathbf{w}_l over the hidden semantics given the context word set $C_{\mathbf{w}_l}$ as

$$p(\mathbf{w}_l | C_{\mathbf{w}_l}) = \sum_{z \in T} p(\mathbf{w}_l | z) \cdot \sum_{n=1, n \neq l}^N p(z | \mathbf{w}_n),$$

where T denotes the semantic space and $z \in T = \{z_1, \dots, z_K\}$. Note that $p(\mathbf{w}_l | z)$ indicates the semantic probability of the word over the latent semantic index z , which is obtained from θ , and $p(z | \mathbf{w}_n)$ indicates the probability of the latent semantics over dictionary, which is defined in β . Thus, the semantic marginal

probability of the \mathbf{w}_l given the context words with the assigned semantic $z_{\mathbf{w}_l} = k$ can be written as

$$p(\mathbf{w}_l | C_{\mathbf{w}_l})_k = \frac{\theta_{v_{w_l}, k} \sum_{n=1, n \neq l}^N \beta_{k, v_{w_n}}}{\sum_{k' \in T} \theta_{v_{w_l}, k'} \sum_{n=1, n \neq l}^N \beta_{k', v_{w_n}}}, \quad (1)$$

where $k \in (1, \dots, K)$. Eq. (1) defines the update process of the adaptive word embeddings over latent semantics given a specific context. This embedding of the target word can be adjusted with the contextual words following Eq. (1), by which we can inference unlimited word embeddings in a continuous semantic space depending on various contexts.

3.4. The ACWE algorithm

Based on the update process of the adaptive word embeddings defined in Eq. (1), we show the ACWE algorithm for tailored word embedding. First, given a collection of text data with sequences of sentences, we train the proposed cross-contextual probabilistic word embedding model that encodes each word into a basic representation. We obtain the two matrices, θ and β . Next, for a target word \mathbf{w}_l with a special set of contextual words, we adjust the representation of \mathbf{w}_l following Eq. (1) to obtain the tailored word embedding.

Thus, given the target word \mathbf{w}_l and its contextual word set $\{\mathbf{w}_n\}_{n=1}^N$, the adaptive update algorithm of ACWE for \mathbf{w}_l is summarized in Algorithm 1.

Algorithm 1 Adaptive Cross-contextual Word Embedding Algorithm.

- 1: INPUT: Corpus C ; Target word \mathbf{w}_l and the contextual word set $\{\mathbf{w}_n\}_{n=1}^N$.
- 2: OUTPUT: Tailored word embedding of \mathbf{w}_l .
- 3: Train θ and β with the corpus C .
- 4: **for** Each word $w_n \in \{\mathbf{w}_n\}_{n=1}^N, n \neq l$ **do**
- 5: **for** $k \in (1, \dots, K)$ **do**
- 6: Update $\theta_{v_{w_n}, k}$ with Eq. (1).
- 7: **end for**
- 8: **end for**
- 9: **for** $k \in (1, \dots, K)$ **do**
- 10: Update $\theta_{v_{w_l}, k}$ with Eq. (1).
- 11: **end for**

We discuss the complexity of the ACWE. With the well-trained θ and β , the computational complexity of ACWE is $O((N-1) \times K + K) = O(N \times K)$, where N is the number of contextual words for the target word in a contextual text. Actually, N is small in many scenarios, thus, the computational complexity of ACWE depends on the scale of the dimension of topic space K , which is still small for real-world applications. In general, the complexity of ACWE is dominated by the cost of the proposed cross-contextual probabilistic word embedding model to train θ and β (See Step 3 in Algorithm 1). Thus, in our paper, we sort an online learning algorithm for this step to reduce the complexity of the proposed ACWE in many real applications. Moreover, θ and β can be learned off-line, which makes it more efficient and flexible to use in many real applications.

4. Model inference

The key problem in the inference of a Bayesian graphical model is to estimate the posterior distribution of latent variables conditioned on the observed data. This work resorts to the variational inference for the model inference. A tailored stochastic variational algorithm is proposed for the proposed cross-contextual probabilistic word embedding to handle large-scale corpus. The traditional variational learning method is a variational expectation-maximization procedure, which requires a full pass through the entire corpus for each iteration. One of the alternative methods is to consider mini-batches of the data per update to reduce the complexity [56–58]. For example, a stochastic inference can easily handle large-scale datasets and outperforms traditional variational inference shown in [58]. While, when the proposed model is trained by stochastic variational inference with a sequence of the mini-batches, the inference process on batches is limited by the arrival of new words. Each row of θ is the semantic distribution of each word, and the new batch of documents may contain new words whose semantic distributions are never learned. Thus, a tailored stochastic variational algorithm is proposed for the basic word embedding learning to handle large-scale corpus.

With the setting of stochastic variational inference, it is needed to define the locally maximized lower-bound for each document first. Given a document \mathbf{d}^i with S^i sentences, each sentence \mathbf{s}_j^i , $j \in \{1, \dots, S^i\}$ contains N_j^i words. For document \mathbf{d}^i , the latent variable is the semantic distribution λ^i . This work uses $\rho^i \in \mathbb{R}^{1 \times K}$ to denote the variational parameter of a Dirichlet distribution for λ^i . For sentence \mathbf{s}_j^i , the latent variables are attention vector ϵ_j^i and the semantic assignments $\{z_l\}_j^i$. Let $\xi_j^i \in \mathbb{R}^{(N_j^i+1) \times 1}$ be the variational parameter of a Dirichlet distribution for ϵ_j^i , and $\{\gamma_l\}_j^i$ be a group of variational parameters of multinomial distributions for $\{z_l\}_j^i$. For document \mathbf{d}^i , the fully factorized variational distribution is,

$$q^d(\lambda^i, \epsilon_j^i, \{z_l\}_j^i, \{\gamma_l\}_j^i) = q(\lambda^i | \rho^i) \prod_{j=1}^{S^i} q(\epsilon_j^i | \xi_j^i) \prod_{l=1}^{N_j^i} q(z_l^i | \gamma_{jl}^i). \quad (2)$$

Based on the above fully factorized variational distribution, this work maximizes the lower-bound (ELBO) to find the approximate likelihood estimations of the variational parameters in the local phase following the update equations:

$$\rho_k^i = \alpha_k^i + \sum_j \sum_l \gamma_{j,lk}^i \cdot \frac{\xi_{j,(N_j^i+1)}^i}{\sum_{l'=1}^{N_j^i+1} \xi_{j,l'}^i}, \quad (3)$$

and

$$\gamma_{j,lk}^i \propto \beta_{k,v^{w_{ij,l}} \cdot \exp\left\{\sum_{l=1}^{N_j^i} \log \theta_{v^{w_{ij,l}},k}\right\} \cdot \frac{\xi_{j,l}^i}{\sum_{l'=1}^{N_j^i+1} \xi_{j,l'}^i}} + [\Psi(\rho_k^i) - \Psi(\sum_{k'} \rho_{k'}^i)] \frac{\xi_{j,(N_j^i+1)}^i}{\sum_{l'=1}^{N_j^i+1} \xi_{j,l'}^i}, \quad (4)$$

where the subscripts of $[k, v^{w_{ij,l}}]$ and $[v^{w_{ij,l}}, k]$ denote the corresponding items in matrix β and θ , respectively. $\Psi(\cdot)$ indicates the digamma function, the first derivative of the log of the Gamma function. Also, for the attentional signals of words and the host document ξ_j^i in sentence \mathbf{s}_j^i , we maximize the terms which contain ξ using gradient descent method:

$$\begin{aligned} \mathcal{L}(\xi_j^i) = & \sum_{l'=1}^{N_j^i} \sum_{k=1}^K \gamma_{j,l'k}^i \cdot \left(\sum_{l=1}^{N_j^i} \log \theta_{v^{w_{ij,l}},k} \cdot \frac{\xi_{j,l}^i}{\sum_{l'=1}^{N_j^i+1} \xi_{j,l'}^i} + [\Psi(\rho_k^i) \right. \\ & - \Psi(\sum_{k'} \rho_{k'}^i)] \frac{\xi_{j,(N_j^i+1)}^i}{\sum_{l'=1}^{N_j^i+1} \xi_{j,l'}^i} \left. + \sum_{l=1}^{N_j^i+1} (\sum_{l'=1}^{N_j^i+1} \pi_{w_{j,l'}}^i - \xi_{j,l}^i) \right. \\ & \cdot [\Psi(\xi_{j,l}^i) - \Psi(\sum_{l'=1}^{N_j^i+1} \xi_{j,l'}^i)] - \log \Gamma(\sum_{l'=1}^{N_j^i+1} \xi_{j,l'}^i) \\ & \left. + \sum_{l'=1}^{N_j^i+1} \log \Gamma(\xi_{j,l'}^i) \right). \end{aligned} \quad (5)$$

In the training process of stochastic variational inference, we need to optimize the maximized the lower bound by subsampling the data to form noisy estimates of the natural gradient, we randomly selects mini-batches of size B in the training corpus to obtain a stochastic estimate of the lower bound, where $1 \leq B \ll M$. Consider a mini-batch \mathbf{b} with B documents in a iteration. First, we compute the local variational parameters, ρ , γ and ξ for the mini-batch \mathbf{b} . Then, we compute the intermediate global parameters of π , α , β and θ . And finally, we update the current estimate of all the global parameters with the intermediate parameters for the next iteration. Here we mainly introduce the details of the learning process of θ , which is the original word embeddings.

For θ , we compute the intermediate global parameter $\hat{\theta}$ given M replicates of each document in the \mathbf{b} , and average them in the update

$$\hat{\theta}_{vk} \propto \frac{M}{B} \sum_i \sum_j \gamma_{j,vk}^i \cdot \frac{\xi_{j,v}^i}{\sum_{l'=1}^{N_j^i+1} \xi_{j,l'}^i}. \quad (6)$$

Let w^b denote the unseen words appeared in \mathbf{b} , and w^b indicates the old words which both observed in \mathbf{b} and the previous mini-batches.

For w^b , we update the current estimate of the global θ_{w^b} with $\hat{\theta}$ directly. For w^b , we update θ_{w^b} using a weighted average of its previous values θ_{w^b} and the new value $\theta_{w^b}^b$ learned by Eq. (6) in current batch \mathbf{b} . After computing the gradient by $\nabla \theta_{w^b} = \theta_{w^b} - \theta_{w^b}^b$, we can update θ_{w^b} following:

$$\begin{aligned} \theta_{w^b} &= \theta_{w^b} - \psi^b \cdot \nabla \theta_{w^b} = \theta_{w^b} - \psi^b \cdot (\theta_{w^b} - \theta_{w^b}^b) \\ &= (1 - \psi^b) \cdot \theta_{w^b} + \psi^b \cdot \theta_{w^b}^b. \end{aligned} \quad (7)$$

where ψ^b represents the step-size in the iteration of \mathbf{b} . As described in [58], the step-size given to θ_{w^b} is obtained by:

$$\psi^b = (\tau_0 + b)^{-\eta}, \quad \tau_0 \geq 0,$$

Algorithm 2 Online variational EM algorithm.

```

1: Define  $\rho^b = (\tau_0 + b)^{-k}$ .
2: for  $b = 0$  to  $\infty$  do
3:   (E-Step:)
4:   repeat
5:     for each sentence  $\mathbf{s}_j^i$  of each document  $\mathbf{d}^i$  in  $\mathbf{b}$  do
6:       update  $\xi_j^i$ ,  $\gamma_j^i$  and  $\rho^i$ .
7:     end for
8:   until convergence
9:   (M-Step:)
10:  for each word in  $w^b$  do
11:    compute  $\theta_{w^b}$  via Eq. (6).
12:  end for
13:  for each word in  $w^b$  do
14:    update  $\theta_{w^b}$  via Eq. (7).
15:  end for
16:  update  $\beta$  via Eq. (8), and update  $\alpha$  and  $\pi$ .
17: end for

```

where $\eta \in (0.5, 1]$ controls the rate at which old values of θ_{w^b} are forgotten, and the delay $\tau_0 \geq 0$ down-weights early iterations.

Similarly, the β can be also updated by:

$$\beta_{vk} = (1 - \psi^b) \cdot \beta_{vk} + \psi^b \cdot \frac{M}{B} \sum_{i=1}^B \sum_{j=1}^{S^i} \sum_{l=1}^{N_j^i} \gamma_{j,lk}^i \cdot (w_{jl}^i)^v. \quad (8)$$

Also, for each mini-batch, we use gradient descent method by taking derivative of the terms with respect to π and α to compute the intermediate parameters of them, respectively. For the sentence \mathbf{s}_j^i , the involved terms which contain π are:

$$\begin{aligned} \mathcal{L}(\pi_{w_j^i}) = & \log \Gamma\left(\sum_{l=1}^{N_j^{i+1}} \pi_{w_{j,l}^i}\right) - \sum_{l=1}^{N_j^{i+1}} \log \Gamma(\pi_{w_{j,l}^i}) \\ & + \sum_{l=1}^{N_j^{i+1}} (\pi_{w_{j,l}^i} - 1)(\Psi(\xi_{j,l}^i) - \Psi(\sum_{l'=1}^{N_j^{i+1}} \xi_{j,l'}^i)). \end{aligned} \quad (9)$$

Note that the $\pi_{w_{j,l}^i}$ indicates the π_{V+1} for all sentences.

For each document \mathbf{d}^i , the involved terms which contain α are:

$$\begin{aligned} \mathcal{L}(\alpha^i) = & \log \Gamma\left(\sum_{k=1}^K \alpha_k^i\right) - \sum_{k=1}^K \log \Gamma(\alpha_k^i) \\ & + \sum_{k=1}^K (\alpha_k^i - 1)(\Psi(\rho_k^i) - \Psi(\sum_{k'=1}^K \rho_{k'}^i)). \end{aligned} \quad (10)$$

Finally, we update the current global parameters π and α as same as β . We describe the online learning algorithm in Algorithm 2.

5. Model analysis and comparison

5.1. Model analysis on matrix factorization

Nonnegative matrix factorization (NMF) on topic modeling has been proven to be equivalent to optimizing the same objective function as PLSA [59]. This section will analyze the proposed ACWE in the view of matrix factorization.

Consider a corpus with M documents denoted by $C = \{\mathbf{d}^1, \mathbf{d}^2, \dots, \mathbf{d}^M\}$, where \mathbf{d}^i , $i \in \{1, \dots, M\}$, indicates the i th

document in the corpus. The dictionary contains V words. Let F be the document-word matrix, and $F_{iv} = F(\mathbf{d}^i, \mathbf{w}^v)$ denotes the frequency of word \mathbf{w}^v in document \mathbf{d}^i , where $v \in \{1, \dots, V\}$.

For topic modeling, PLSA tries to factorize the matrix F into two different nonnegative matrices U and H with a fixed K , where K is the number of latent topics. $U \in \mathbb{R}^{M \times K}$ is the matrix of the distributions of documents over latent topics, and $H \in \mathbb{R}^{V \times K}$ is the matrix of the distributions of latent topics over the dictionary. In brief, topic models can be viewed as one of the matrix factorization processes:

$$F(\mathbf{d}^i, \mathbf{w}^v) = UH^T.$$

In the proposed model, the sentences are represented as the contexts based on the ‘bag-of-words’ assumption. As mentioned, let $T \in \{0, 1\}^{M \times S}$ denote the document-sentence matrix where S is the total number of sentences in the corpus. $T_{ij} = T(\mathbf{d}^i, \mathbf{s}_j^i)$ is a binary value that indicates whether \mathbf{d}^i contains \mathbf{s}_j^i or not. Based on the assumption of the proposed model, the semantic distribution matrix of the sentences is $\vartheta_{S \times K}$. Hence, $F(\mathbf{d}^i, \mathbf{w}^v)$ can be factorized as:

$$F(\mathbf{d}^i, \mathbf{w}^v) = T\vartheta H^T,$$

where the main difference from PLSA is that the proposed model defines the semantic generation process on the sentence level.

Let $A \in \mathbb{R}^{S \times V}$ denote the sentence-word matrix, and $A_{jv} = A(\mathbf{s}_j^i, \mathbf{w}^v)$ denote the attention value of the word \mathbf{w}^v in sentence \mathbf{s}_j^i . Based on the assumption of the proposed model, the semantic distribution matrix of the sentences $\vartheta_{S \times K}$ can be calculated as follows:

$$\vartheta = AU_\theta,$$

where each row in $U_\theta \in \mathbb{R}^{V \times K}$ denotes the semantic distribution of one word from the dictionary. Note that the attention value of the host document is ignored for ease of presentation.

Thus, the proposed model can be shown in the way of matrix factorization as follows:

$$F(\mathbf{d}^i, \mathbf{w}^v) = TAU_\theta H^T.$$

It is interesting to note that the proposed ACWE obtains the nonnegative probabilistic word embeddings U_θ through a process of matrix factorization, which implies that the word embeddings are distributions over latent semantics.

5.2. Model comparison

There are two main types of models for learning word embeddings. The first type includes the global matrix factorization approaches, e.g., latent semantic analysis (LSA) and non-negative sparse embedding (NNSE). The second includes the local content window approaches such as the skip-gram model and its extensions. The proposed ACWE leverages these two types of methods to learn word embeddings. As discussed earlier, the ACWE benefits from the global statistical information to train the global word embeddings (i.e., θ). It also trains on separate local context windows (sentences) to learn adaptive word embeddings.

Many vector-space models of lexical semantics create a single ‘prototype’ embedding to represent the meaning of a word or learn multi-prototype word embeddings. Some recent studies attempt to train multi-prototype word embeddings by clustering context window features [10,11], or determining the number of word embeddings through topics [30,34], or using a specific probability process such as the Chinese restaurant process [12,25]. Differently, the ACWE makes no restricted assumptions to learn multi-prototype word embeddings. Every word has an original global embedding learned from the document-level information,

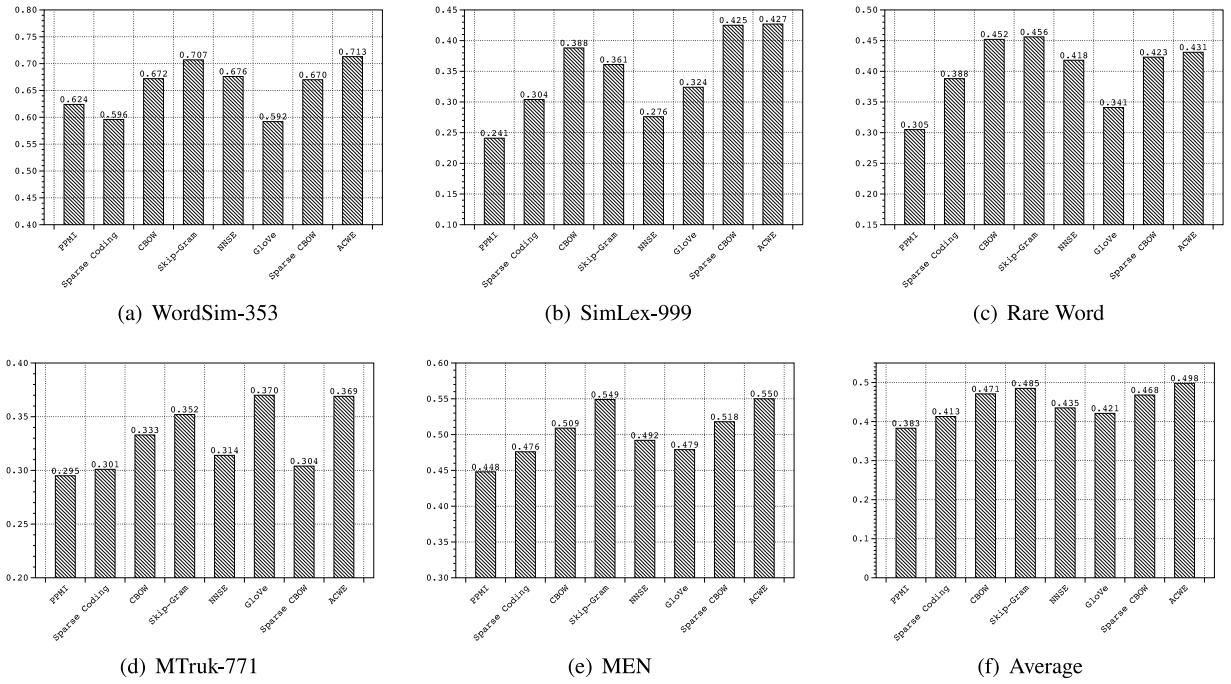


Fig. 4. The results of Spearman correlation coefficient for the word similarity task on different datasets from (a) to (e). (f) is the results of the weighted average on the five datasets. The higher values indicate better performances.

Table 1

The results of Spearman correlation coefficient on SCWS with baseline models which consider word polysemy.

MP-VSM	NTSG	PM-MP	MSSG	TWE	Multiple-WP	STE	ELMo	ACWE ₀	ACWE ₁
0.594	0.685	0.636	0.692	0.681	0.657	0.680	0.703	0.720	0.733

Table 2

Some cases to show the word similarity by raking the top 9 most similar words, where the numbers are the corresponding cosine distances.

	Ranking lists with the corresponding cosine distances
Education	(students, 0.9915), (school, 0.9904), (university, 0.9890), (year, 0.9883), (college, 0.9883), (post, 0.9880), (report, 0.9879), (public, 0.9877), (teaching, 0.9875)
China	(largest, 0.9867), (Chinese, 0.9856), (Singapore, 0.9846), (united, 0.9843), (Asia, 0.9841), (commission, 0.9834), (kingdom, 0.9832), (Russia, 0.9830), (employees, 0.9824)
Movie	(music, 0.9874), (stars, 0.9871), (writer, 0.9851), (famous, 0.9848), (film, 0.9847), (broadcast, 0.9842), (drama, 0.9841), (song, 0.9840), (actor, 0.9839)
University	(school, 0.9930), (academic, 0.9928), (founded, 0.9926), (students, 0.9924), (college, 0.9921), (year, 0.9899), (science, 0.9890), (education, 0.9889), (international, 0.9888)
Programming	(computers, 0.9829), (MIT, 0.9819), (intelligence, 0.9818), (implement, 0.9815), (computing, 0.9814), (artificial, 0.9810), (technologies, 0.9806), (digital, 0.9805), (communication, 0.9801)
Health	(medical, 0.992), (patients, 0.9887), (food, 0.9882), (medicine, 0.9872), (serves, 0.9863), (social, 0.9861), (families, 0.9860), (largest, 0.9859), (states, 0.9858)
County	(district, 0.987), (west, 0.986), (located, 0.985), (valley, 0.985), (river, 0.985), (city, 0.985), (bay, 0.984), (historic, 0.983), (governor, 0.983)
Bus	(port, 0.985), (corporation, 0.985), (north, 0.983), (station, 0.982), (airport, 0.981), (road, 0.981), (buses, 0.980), (industries, 0.980), (park, 0.979)

and an adjusted word embedding is generated through the original global embedding in its present context. Thus, the ACWE tries to learn the adaptive word embedding to address the problem of word polysemy.

Compared with ELMo [5], the word embeddings learned by ACWE are nonnegative, as word embeddings θ are generated by a Dirichlet distribution. As described in [60], a nonnegative assumption for word embedding could be efficient in improving word embedding interpretability. In particular, the two main characteristics of the word embeddings are learned by the proposed model. First, the global word embedding is represented by a nonnegative vector because it is defined as a probability

distribution over all the latent semantics. The nonnegative word embedding is highly interpretable since each semantics learned by the proposed method is defined by a distribution over explainable vocabulary. Second, a local and tailored word embedding is generated in a special context to capture more accurate semantics of the targeted word, a process which is adaptively adjusted based on the global word embedding. This method has been innovative because it uses topic models to create word embedding. The proposed method can achieve much better interpretability and improved flexibility in adjusting semantics. Case studies will be presented in Section 6 to demonstrate its capacity of capturing

and representing word polysemy as well as its advantage in interpretability.

6. Experiments

In this section, we evaluate the proposed model on word similarity, polysemy induction and text classification tasks. We also demonstrate the visualizations of the semantics of words to show the capacity of the proposed ACWE model on word polysemy and interpretability.

6.1. Experimental settings and training configuration

The widely used Wikipedia is taken as the corpus to train all the models and a snapshot of the whole Wikipedia is used in our experiments. It contains about 531,306 pages. Infrequent words have been removed from this corpus, and a dictionary of about 10,810 frequent words is obtained. The pure digit words are removed as well as stop words. The abstract of each article is used as a document. After splitting the documents into sentences and removing the sentences in which the number of words is fewer than 5, a corpus is obtained with 2.19M sentences, which is called Wikipedia-L. A small subset is also built from Wikipedia which is called Wikipedia-S. 68 categories are selected as class labels, such as education, science, military, and so on. For each category, 250 documents are randomly selected from Wikipedia-L. Then, only the first sentence of each document is kept. After removing the sentences with fewer than 10 words, 16,070 sentences (articles) are obtained with the corresponding labels. This corpus is used for the text classification tasks. Meanwhile, the data from Internet Movie database (IMDb)¹ is also used. 31,108 movies with the storylines and the genres are included. The storylines are treated as the texts, and the 29 genres are treated as the labels for the text classification task.

The proposed ACWE is trained with $K = 200$, which means that the words are embedded into a 200-dimensional semantic space. Particularly, the proposed model is first trained on the small corpus, Wikipedia-S, with the criterion of topic coherence described in [61,62] to learn an initial β and θ . Then, the parameters are updated by the proposed stochastic variational learning algorithm on Wikipedia-L. The source code of the proposed ACWE is available online, as well as all the scripts and data of the experiments.²

6.2. Experiments on word similarity

In this part of the experiments, the quantitative comparisons of the proposed ACWE method with other baselines are demonstrated on the word similarity task, which is to measure how well the model captures the similarity between two words. This task is introduced in [11], which evaluates the performance of a model by calculating the Spearman's rank correlation between the ranking of ground truth similarity scores and that based on the similarity scores produced by the model. There are six benchmark datasets.

(1) WordSim-353 [63] is a standard dataset for evaluating word vector representations. It consists of a list of word types, the similarity of which is rated in an integral scale from 1 to 10. Monosemic and polysemic words are included.

(2) SimLex-999 [64] contains 999 pairs of nouns, verbs and adjectives. SimLex-999 provides a way of measuring how well models capture the word similarity. Note that the word similarity

measured by SimLex-999 is about the meaning of words and concepts but not relationship or association between two words.

(3) Rare Word [65] is a word set focusing on rare words to complement existing ones and it contains 2034 word pairs.

(4) MTruk-771 [66] dataset that contains 771 word pairs whose similarity is crowdsourced from Amazon Mechanical Turk.

(5) MEN [67] benchmark consists of 3000 word pairs which are randomly selected from words that occur at least 700 times in a freely available corpus.

(6) Stanford Contextual Word Similarity (SCWS) dataset [11] consists of 2003 word pairs and their sentential contexts with human judgments.

The proposed ACWE model is evaluated on the word similarity task by comparing with GloVe, Skip-Gram and CBOW, positive pointwise mutual information (PPMI), NNSE, Sparse Coding and Sparse CBOW. Note that the above baseline models can learn sparse or dense word embeddings without considering the word polysemy.

The baselines which consider word polysemy are also compared, such as Multi-Prototype Vector-Space Models (MP-VSM) [10], Multiple-WP [11], PM-MP [29], Multiple-sense Skip-Gram (MSSG) [12], Topical Word Embeddings (TWE) [30], Neural Tensor Skip-Gram model (NTSG) [34], Skip-Gram Topical word Embedding (STE) [68] and ELMo [5] on the SCWS dataset. They are trained on Wikipedia with the same dimensions of word embeddings. For ELMo, bi-directional LSTM is considered and the dimension is 100 for each direction. For the proposed ACWE, we use ACWE₀ to denote the model which learns the global word embeddings without adjusting the word embeddings adaptively. For the SCWS dataset, the context is given for the targeted word in each word pair. Thus, we recalculate the word embedding of \mathbf{w} given the context via Eq. (1) and this approach is named by ACWE₁.

Fig. 4 and Table 1 summarize the results on the word similarity tasks, where the proposed ACWE model outperforms the baselines except Word2Vector (CBOW and Skip-Gram) on Rare Word. The main reason is that many words in Rare Word are non-polysemous. While, the weighted average value of ACWE is still high, as shown as in Fig. 4(f), where the weight values are from the word numbers in each benchmark datasets. The experiments on word similarity demonstrate that the global word embeddings learned from the proposed ACWE capture the effective features from the semantic level, which is the main benefit of the usage of topic modeling. In Fig. 4, ACWE₁ achieves the best result, which demonstrates that the word embeddings can be improved significantly by the contexts, when the contexts are available in the SCWS dataset.

To visualize the learned global word embeddings, some cases are given to show the word similarity. Specifically, we rank the cosine distances of some target words with other words in the dictionary and present the top 9 most similar words in Table 2. The values in Table 2 are the corresponding cosine distances between two words. Experiments indicate that global word embeddings effectively capture the similarities among the words, and the words of similar meanings have similar embeddings. Thus, the capacity for capturing the semantics of the proposed model enables us to learn high-quality word embeddings with an unsupervised learning framework.

6.3. Experiments on adaptive word embeddings

The main contribution of this paper is to learn adaptive word embeddings within different contexts. Thus, in this part of the experiments, the performance of word embeddings are shown within different contexts by using the proposed ACWE algorithm. Some cases of polysemous words in different contexts using the

¹ <https://www.imdb.com>.

² <http://www.shuangyin.li/acwe/>.

Table 3

Adaptive word embeddings with different contexts. The “papers”, “biomedical” and “light” are demonstrated.

Books or papers printed today, by the same publisher, and from the same type as when they were first published, are still the first editions of these books to a bibliographer.
–0.140819 [published, book, written, wrote, edition]
–2.286497 [journal, peer, reviewed, scientific, academic]
–3.945425 [type, volume, frequently, visual, notably]
–5.231559 [included, magazine, leading, editor, press]
–6.079971 [records, record, index, literature, reference]
I know of a research group in a university where students submit some academic papers without their professor having read them, let alone contributing to the work.
–0.823609 [research, project, foundation, led, projects]
–1.037011 [journal, peer, reviewed, scientific, academic]
–2.664416 [university, professor, faculty, Harvard, department]
–3.089699 [field, study, studies, scientific, fields]
–3.266090 [students, student, teaching, teachers, teacher]
Biomedical definition, application of natural sciences, especially the biological and physiological sciences, to clinical medicine.
–1.009875 [field, study, studies, scientific, fields]
–1.171692 [biology, molecular, biological, genetics, ecology]
–2.172841 [medical, medicine, clinical, patient, surgery]
–2.231589 [institute, established, center, private, institution]
–3.357441 [science, fellow, MIT, Stanford, laboratory]
The treatments available at biomedical center include natural herbs, special diet, vitamins and minerals, lifestyle counseling, positive attitude, and conventional medical treatments when indicated.
–0.826321 [center, Massachusetts, Boston, Dr., Md.]
–0.864242 [medical, medicine, clinical, patient, surgery]
–2.540809 [include, applications, processing, large, techniques]
–3.798415 [natural, areas, land, environmental, environment]
–4.364615 [disease, treatment, effects, cancer, risk]
Light is electromagnetic radiation within a certain portion of the electromagnetic spectrum.
–0.129370 [nuclear, light, radiation, magnetic, experiments]
–3.438747 [term, refers, word, meaning, means]
–3.585760 [image, images, color, vision, camera]
–3.752776 [line, station, railway, operated, bus]
–4.770996 [energy, mass, particles, electron, atomic]
A railbus is a light weight passenger rail vehicle that shares many aspects of its construction with a bus.
–0.440478 [line, station, railway, operated, bus]
–1.815225 [body, exercise, lower, weight, strength]
–2.777080 [process, single, typically, multiple, result]
–3.266629 [original, play, stage, theater, tragedy]
–4.287032 [construction, formed, cross, bridge, replaced]
Heavy weights are good for developing strength and targeting specific muscle, and light weights are good for build and maintain lean muscle.
–1.226888 [common, specific, terms, concept, object]
–1.251034 [body, exercise, lower, weight, strength]
–2.309753 [cell, cells, blood, growth, muscle]
–2.868760 [process, single, typically, multiple, result]
–2.941860 [due, high, low, quality, additional]

proposed ACWE algorithm are shown in Table 3. We compute the word embeddings of three words in different contexts to show the adaptive adjustment of our model in different contexts. The values in Table 3 denote the log-probabilities. Note that the basic word embeddings of the three words are shown as in Table 4, and Table 3 shows the results after updating the word embeddings adaptively. These experiments show the process that the word semantics are changed with the different contexts surrounded.

6.4. Experiments on polysemy induction

In this section, this work evaluates the performance of the adaptive word embeddings by following the methodology of polysemy induction [15]. The sentence classification task is chosen to investigate the effectiveness of the proposed model. For this task, we use the average over all the word embedding vectors in the sentence as its representation. Wikipedia-S and IMDb are used as the test sets. For each storyline in IMDb, we only keep the words which are in the dictionary trained by Wikipedia-L.

According to [15], There are three steps:

(1) We test the original word embeddings of our model and the embeddings obtained by other baseline models, which can be treated as a single representation for each word. For our model,

the word embeddings θ are learned on the large-scale corpus, by setting the semantic number $T = 200$. As described above, we call it ACWE₀. Particularly, ACWE₀ learns the global word embeddings without adjusting the word embeddings adaptively.

(2) The word embeddings are updated given the corresponding contexts to get the adaptive embeddings, which computes multiple representations for each polysemous word. In detail, the word embeddings are updated in each sentence to get the adaptive word embeddings within different contexts. That is, for each word \mathbf{w} , we let all the other words in the text as the context words \mathbf{w}_c for \mathbf{w} . We recalculate the word embedding of \mathbf{w} given \mathbf{w}_c via Eq. (1) and this approach is named by ACWE₁. This process is run for all the words in each sentence, and average all the new word embeddings as the embedding of the host sentence in the classification task.

(3) The task is evaluated using the basic word embeddings and the adaptive word embeddings on the same text classification task to show the performance gains. Specifically, the sentence embeddings are treated as the sentence features to train an SVM. 80% sentences are used for training and the rest is for testing to evaluate the performance of sentence classification.

For comparison, we average the embeddings of all the words in a sentence as the sentence representation for the baseline

Table 4

The cases to show the word polysemy and interpretability with the top 5 semantics of each word and the top 5 words in each semantic ranked by the log-probabilities.

	Ranking of semantics
County	−2.33009 [county, national, historic, located, district] −2.787075 [park, river, valley, lake, located] −2.986754 [local, authority, city, area, region] −3.159863 [south, west, north, east, England] −3.184729 [house, historic, style, story, places]
Papers	−3.121343 [journal, peer, reviewed, editor, published] −3.132917 [born, American, January, September, December] −3.271758 [author, books, science, work, German] −3.421435 [book, published, work, English, history] −3.453628 [university, professor, academic, philosophy, studies]
Comedians	−2.149376 [television, show, aired, episode, episodes] −2.225051 [American, radio, writer, television, show] −2.585478 [produced, series, film, films, short] −2.797709 [released, series, video, TV, DVD] −2.946048 [film, directed, drama, comedy, starring]
Genomics	−1.422867 [biology, molecular, cell, gene, protein] −1.610026 [species, evolution, biological, natural, humans] −2.701144 [human, theory, study, social, individual] −3.071058 [human, brain, mental, cognitive, psychology] −3.19488 [concept, terms, object, defined, objects]
Prof	−3.045664 [professor, university, science, scientist, computer] −3.12606 [university, professor, academic, philosophy, studies] −3.262579 [born, American, January, September, December] −3.352558 [author, books, science, work, German] −3.425612 [people, group, including, world, country]
Health	−2.975491 [care, health, services, hospital, patients] −3.141724 [blood, symptoms, risk, vaccine, pregnancy] −3.219858 [development, health, organization, global, European] −3.27917 [high, includes, including, related, level] −3.314238 [medical, medicine, center, health, clinical]
Light	−2.918804 [station, line, rail, bus, transit] −3.116566 [light, energy, device, speed, motion] −3.234566 [human, theory, study, social, individual] −3.317204 [called, term, considered, word, form] −3.396814 [concept, terms, object, defined, objects]
Computers	−2.97187 [system, systems, developed, based, control] −3.094254 [design, power, technology, electronic, equipment] −3.240674 [computer, computing, computers, graphics, dos] −3.366713 [mobile, devices, phone, software, solutions] −3.406292 [network, open, access, information, Internet]
Biomedical	−2.986134 [research, project, institute, foundation, projects] −2.988148 [science, physics, field, scientific, sciences] −3.168841 [biology, molecular, cell, gene, protein] −3.203789 [human, theory, study, social, individual] −3.217035 [medical, medicine, center, health, clinical]

models, which is the same as the proposed model. For AdaGram [25], the best word vectors are chosen given the contexts. The other comparisons include GloVe, Skip-Gram, NNSE, AdaGram, ELMo and Bert.

For GloVe, the released implementation is used,³ and trained on Wikipedia-L with the dimension of word embedding set to 200. For the ELMo, the released implementation⁴ is used, and the ELMo model is retrained on Wikipedia-L. For the Bert, the pre-trained Bert is used, which is trained on Wikipedia and Book-Corpus⁵. The pre-trained Bert is fine-tuned with Wikipedia-L. Note that, the dimension of word embedding in the pre-trained Bert is 768, which is larger than the proposed model and the other comparisons. For all the comparisons, a bag-of-words averaging is employed to produce the sentence embedding.

Fig. 5 shows the evaluation results of sentence classification on Wikipedia-S and IMDb. Both ACWE₀ and ACWE₁ outperform

all baseline methods significantly on Wikipedia-S. These experiments demonstrate that the way of adaptively adjusting the word embedding with the contexts is effective, where the adaptive cross-contextual word embedding process can improve the capacity of capturing the latent features for word embedding with the neighbored words.

In addition, the methodology of polysemy induction is one of the popular approaches to evaluate the performance of polysemy. ACWE₀ presents the global embeddings without considering the adaptive cross-contextual process. Compared with the other models, ACWE₀ also get better performances on the sentence classification. The main reason is that the polysemy has been already captured when the global embeddings are learned in the proposed model. With the adaptive cross-contextual word embedding process, the proposed ACWE further improves the capacity of modeling the word polysemy. Compared with ACWE₀, ACWE₁ based on the adaptive word embeddings leads to significant performance improvement in sentence classification tasks.

To further show the effectiveness of the proposed adaptive word embeddings, we adjust the ratio of the updated words in a text. We test with the ratio of $r = \{0.1, \dots, 1\}$ on sentence classification tasks with Wikipedia-S, where r indicates the

³ <https://nlp.stanford.edu/projects/glove/>.

⁴ <https://allennlp.org/elmo>.

⁵ <https://github.com/google-research/bert>.

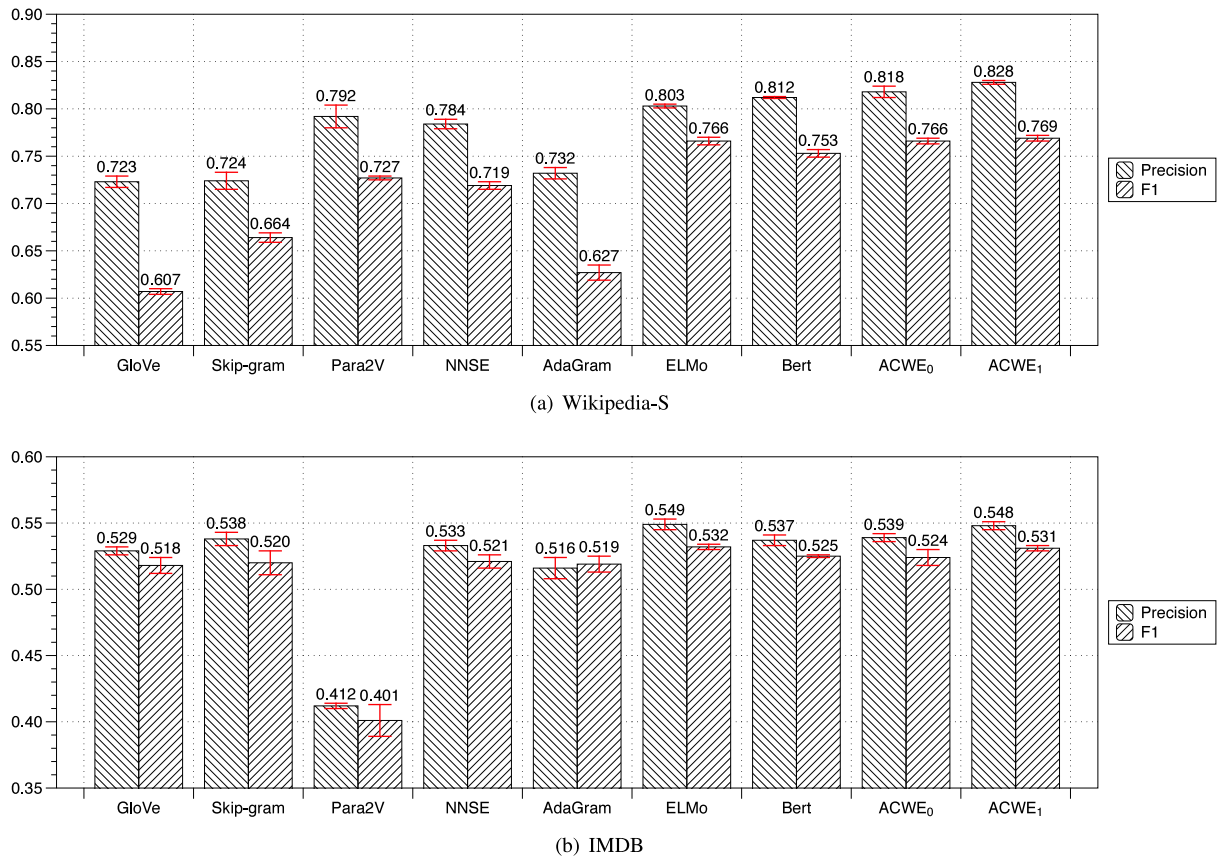


Fig. 5. Sentence classification results for different models on Wikipedia-S (up) and IMDb (Down) with 5-fold cross-validation. Error bars: standard deviation.

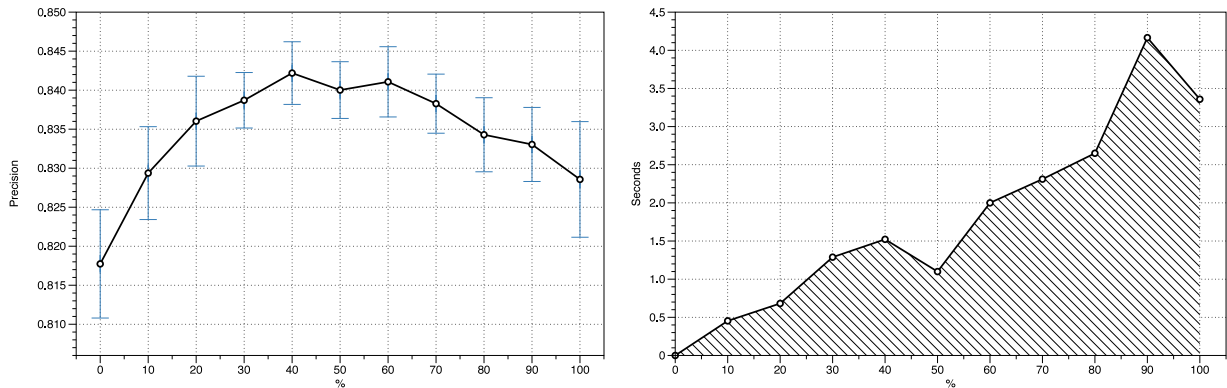


Fig. 6. (Left) Sentence classification results with different ratios on Wikipedia-S. (Right) The average time for each sentence with different ratios.

proportion of the words that are adaptively updated given the context. Note that $r = 0$ corresponds to ACWE₀ and $r = 1$ implies ACWE₁. Fig. 6 shows the precision on the sentence classification by varying r . We can see that the adaptive word embeddings can improve the performance of the sentence classification task. Also, we notice that, in Fig. 6, the best results come from the ratio between 0.4 and 0.6, which means that we do not need to adjust all the word embeddings. The main reason is that not all the words in a sentence are polysemous.

Meanwhile, the efficiency of the proposed model is tested on Wikipedia-S. Fig. 6 also shows the average of the update time for each sentence, where the average word number in each sentence is 148. According to the results, we can see that it takes less than 2 s to update the word embeddings in a long sentence with the ratio of 0.6. The computational complexity of ACWE is $O(N \times K)$, where N is the number of contextual words for the

target word. As the discussion in Section 3.4, N is always small in many scenarios, thus, the proposed ACWE is efficient in the real-world text classification tasks.

6.5. Experiments on word polysemy and interpretability

The polysemy and interpretability of a word embedding are visualized by showing the top semantics assigned to the word, where the top semantics can be explained by the largest probabilities of words in the dictionary. The top semantics ranked by the probabilities in θ can be treated as the multiple senses of the word, which is an inherent advantage of topic models.

Table 4 shows some words with the top 5 semantics, and each semantics is represented by the top 5 words, where the values ahead are the log-probabilities of each semantics for the target word. We can see that each word has some main semantics over

the latent semantic space, which matches the assumption of word polysemy. Benefit from the probability inference of ACWE, the probability values of each main semantics for a polysemous word can be obtained, where each semantics can be explained by a set of words explicitly. This characteristic of this interpretability does not exist in the models based on deep neural networks, such as Skip-Gram, ELMo or Bert. Meanwhile, based on the property of the probabilistic word embeddings, the semantics of each polysemous word can be gathered in different contexts and show the different specific meanings. More cases can be shown with our scripts (see footnote 2).

7. Conclusions

Understanding the meaning of words is of great importance in many applications, including helping machines to understand the text, classifying text, and building knowledge graphs on web information retrieval. Word embedding learning is one of the methods to understand words. There are many unresolved issues in the field of word embedding. In this paper, we have identified a critical issue in most word embedding learning methods, which is that they are not adaptive to capture word polysemy and build different representations for different senses of the same word.

To address the problem of word polysemy, this paper explored the potential of using contextual information to obtain different senses for the same word. Based on topic modeling, this paper proposed an adaptive cross-contextual word embedding model (ACWE). The proposed ACWE first understands a word in its general senses and creates a global word embedding, then the ACWE will take advantage of the contextual information of the word in different contexts, and adjusts the word embedding to generate different local word embeddings. The local word embedding allows the proposed method to use contextual information representing the senses of the word in the corresponding context. Thus, the ACWE is capable to capture the polysemy of a word and build multiple word embeddings to represent the word's different senses in different contexts. Due to the underlying use of nonnegative vectors, word embeddings produced by the ACWE are highly interpretable. Comprehensive experiments have been conducted to evaluate the performance of the proposed ACWE model. Considering word polysemy on six popular benchmark datasets, the results demonstrated that ACWE outperformed state-of-the-art methods.

In this paper, we also extended ACWE with an online algorithm to fit in the document stream scenario. Using the online algorithm, the proposed ACWE can handle the problem of the large-scale corpus and provides a more effective and extensive semantics of the texts to handle all senses appearing in the polysemy. By training on a large-scale corpus, i.e., Wikipedia, the extended ACWE has been validated by experiments. The results showed that it can achieve higher precision and F1 on sentence classifications than the models without considering word polysemy, and can also compete with methods based on deep neural networks.

Because the semantic/topic number, K , is predefined in the proposed ACWE, the capacity of the proposed model to capture the latent and fine-grained semantics is limited, especially when the semantics are highly correlated or hierarchical. Therefore, to model the multiple senses more accurately, in the future work we focus on extending the proposed ACWE model to learn an infinite and sparse semantic space.

CRedit authorship contribution statement

Shuangyin Li: Conceptualization, Methodology, Software, Writing - original draft. **Rong Pan:** Supervision. **Haoyu Luo:** Reviewing and Validation. **Xiao Liu:** Reviewing and Validation. **Gansen Zhao:** Supervision, Validation and editing, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by National Natural Science Foundation of China (No. 62006083) and National Key Research and Development Program of China (2020YFA0712500). This work was also partly supported by Basic and Applied Basic Research Fund of Guangdong Province (No. 2019B1515120085). Rong Pan and Gansen Zhao are the corresponding authors.

References

- [1] C.D. Manning, P. Raghavan, H. Schütze, et al., *Introduction to Information Retrieval*, Cambridge university press Cambridge, 2008.
- [2] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: *NIPS*, 2013.
- [3] J. Pennington, R. Socher, C.D. Manning, Glove: Global vectors for word representation, in: *EMNLP*, 2014.
- [4] B. Murphy, P.P. Talukdar, T. Mitchell, Learning effective and interpretable semantic models using non-negative sparse embedding, in: *COLING 2012*, 2012.
- [5] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, in: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018.
- [6] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2018, arXiv preprint [arXiv:1810.04805](https://arxiv.org/abs/1810.04805).
- [7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.
- [8] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, Q.V. Le, Xlnet: Generalized autoregressive pretraining for language understanding, 2019, arXiv preprint [arXiv:1906.08237](https://arxiv.org/abs/1906.08237).
- [9] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, 2019, arXiv preprint [arXiv:1907.11692](https://arxiv.org/abs/1907.11692).
- [10] J. Reisinger, R.J. Mooney, Multi-prototype vector-space models of word meaning, in: *ACL*, 2010.
- [11] E.H. Huang, R. Socher, C.D. Manning, A.Y. Ng, Improving word representations via global context and multiple word prototypes, in: *ACL*, 2012.
- [12] A. Neelakantan, J. Shankar, A. Passos, A. McCallum, Efficient non-parametric estimation of multiple embeddings per word in vector space, in: *EMNLP*, 2014.
- [13] X. Chen, Z. Liu, M. Sun, A unified model for word sense representation and disambiguation, in: *EMNLP*, 2014.
- [14] S. Arora, Y. Li, Y. Liang, T. Ma, A. Risteski, Linear algebraic structure of word senses, with applications to polysemy, *Trans. Assoc. Comput. Linguist.* 6 (2018) 483–495.
- [15] H. Dubossarsky, E. Grossman, D. Weinshall, Coming to your senses: on controls and evaluation sets in polysemy research, in: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.
- [16] Y. Bengio, R. Ducharme, P. Vincent, C. Jauvin, A neural probabilistic language model, *JMLR* (2003).
- [17] M. Faruqui, Y. Tsvetkov, D. Yogatama, C. Dyer, N. Smith, Sparse overcomplete word vector representations, 2015, arXiv.
- [18] F. Sun, J. Guo, Y. Lan, J. Xu, X. Cheng, Sparse word embeddings using l1 regularized online learning, in: *IJCAI*, 2016.
- [19] B. Hu, B. Tang, Q. Chen, L. Kang, A novel word embedding learning model using the dissociation between nouns and verbs, *Neurocomputing* 171 (2016) 1108–1117, <http://dx.doi.org/10.1016/j.neucom.2015.07.046>.
- [20] M.E. Peters, W. Ammar, C. Bhagavatula, R. Power, Semi-supervised sequence tagging with bidirectional language models, *ACL* (2017).
- [21] B. Scarlino, T. Pasini, R. Navigli, SenseBERT: Context-enhanced sense embeddings for multilingual word sense disambiguation, in: *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, the Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, the Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020*, New York, NY, USA, February 7–12, 2020, AAAI Press, 2020, pp. 8758–8765.

- [22] B. McCann, J. Bradbury, C. Xiong, R. Socher, Learned in translation: Contextualized word vectors, in: *Advances in Neural Information Processing Systems*, 2017, pp. 6294–6305.
- [23] O. Melamud, J. Goldberger, I. Dagan, context2vec: Learning generic context embedding with bidirectional lstm, in: *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, 2016, pp. 51–61.
- [24] J. Li, D. Jurafsky, Do Multi-Sense Embeddings Improve Natural Language Understanding? in: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1722–1732.
- [25] S. Bartunov, D. Kondrashkin, A. Osokin, D. Vetrov, Breaking sticks and ambiguities with adaptive skip-gram, in: *Artificial Intelligence and Statistics*, 2016.
- [26] Y. Yao, J. Zhang, F. Shen, W. Yang, P. Huang, Z. Tang, Discovering and distinguishing multiple visual senses for polysemous words, in: *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [27] V. Vukotić, C. Raymond, Mining polysemous triplets with recurrent neural networks for spoken language understanding, in: *International Conference on Spoken Language Processing (Interspeech)* 2019, Graz, Austria, 2019, <https://hal.archives-ouvertes.fr/hal-02170709>.
- [28] Z.-I. Ye, H.-x. Zhao, Syntactic word embedding based on dependency syntax and polysemous analysis, *Front. Inf. Technol. Electron. Eng.* 19 (4) (2018) 524–535.
- [29] F. Tian, H. Dai, J. Bian, B. Gao, R. Zhang, E. Chen, T.-Y. Liu, A probabilistic model for learning multi-prototype word embeddings, in: *COLING*, 2014.
- [30] Y. Liu, Z. Liu, T.-S. Chua, M. Sun, Topical word embeddings, in: *AAAI*, 2015.
- [31] R. Das, M. Zaheer, C. Dyer, Gaussian lda for topic models with word embeddings, in: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2015.
- [32] J. Guo, W. Che, H. Wang, T. Liu, Learning sense-specific word embeddings by exploiting bilingual resources, in: *COLING*, 2014.
- [33] Z. Wu, C.L. Giles, Sense-aware semantic analysis: A multi-prototype word representation model using wikipedia, in: *AAAI*, 2015.
- [34] P. Liu, X. Qiu, X. Huang, Learning context-sensitive word embeddings with neural tensor skip-gram model, in: *IJCAI*, 2015.
- [35] B. Salehi, P. Cook, T. Baldwin, A word embedding approach to predicting the compositionality of multiword expressions, in: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, Denver, Colorado, 2015, pp. 977–983, <http://dx.doi.org/10.3115/v1/N15-1099>, <https://www.aclweb.org/anthology/N15-1099>.
- [36] T. Ruas, W.I. Grosky, A. Aizawa, Multi-sense embeddings through a word sense disambiguation process, *Expert Syst. Appl.* 136 (2019) 288–303, <http://dx.doi.org/10.1016/j.eswa.2019.06.026>.
- [37] B. Athiwaratkun, A.G. Wilson, A. Anandkumar, Probabilistic fasttext for multi-sense word embeddings, in: I. Gurevych, Y. Miyao (Eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, ACL 2018, Melbourne, Australia, July 15–20, 2018, Volume 1: Long Papers, Association for Computational Linguistics, 2018, pp. 1–11.
- [38] K. Ashihara, T. Kajiura, Y. Arase, S. Uchida, Contextualized word representations for multi-sense embedding, in: S. Politzer-Ahles, Y. Hsu, C. Huang, Y. Yao (Eds.), *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*, PACLIC 2018, Hong Kong, December 1–3, 2018, Association for Computational Linguistics, 2018.
- [39] A. Ferrari, B. Donati, S. Gnesi, Detecting domain-specific ambiguities: an NLP approach based on wikipedia crawling and word embeddings, in: *2017 IEEE 25th International Requirements Engineering Conference Workshops (REW)*, IEEE, 2017, pp. 393–399.
- [40] W. Wang, N. Niu, H. Liu, Z. Niu, Enhancing automated requirements traceability by resolving polysemy, in: *2018 IEEE 26th International Requirements Engineering Conference (RE)*, IEEE, 2018, pp. 40–51.
- [41] A. Ferrari, A. Esuli, S. Gnesi, Identification of cross-domain ambiguity with language models, in: *2018 5th International Workshop on Artificial Intelligence for Requirements Engineering (AIRE)*, IEEE, 2018, pp. 31–38.
- [42] D. Toews, L. Holland, Determining domain-specific differences of polysemous words using context information, in: *Joint Proceedings of REFSQ-2019 Workshops, Doctoral Symposium, Live Studies Track, and Poster Track Co-Located with the 25th International Conference on Requirements Engineering: Foundation for Software Quality (REFSQ 2019)*, Essen, Germany, March 18th, 2019.
- [43] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent Dirichlet allocation, *JMLR* (2003).
- [44] S. Li, Y. Zhang, R. Pan, M. Mao, Y. Yang, Recurrent attentional topic model, in: *31th AAAI Conference on Artificial Intelligence (AAAI-17)*, Association for the Advancement of Artificial Intelligence (AAAI), 2017, pp. 3223–3229.
- [45] S. Li, Y. Zhang, R. Pan, K. Mo, Adaptive probabilistic word embedding, in: *Proceedings of the Web Conference 2020*, in: *WWW '20*, Association for Computing Machinery, New York, NY, USA, 2020, pp. 651–661, <http://dx.doi.org/10.1145/3366423.3380147>.
- [46] K. Tanaka, A. Niimi, Word topic prediction model for polysemous words and unknown words using a topic model, in: *Intelligent Computing-Proceedings of the Computing Conference*, Springer, 2019, pp. 860–866.
- [47] S. Kunii, H. Shinnou, Use of combined topic models in unsupervised domain adaptation for word sense disambiguation, in: *Proceedings of the 27th Pacific Asia Conference on Language, Information, and Computation (PACLIC 27)*, 2013, pp. 415–422.
- [48] Y. Xiao, Z. Fan, C. Tan, Q. Xu, W. Zhu, F. Cheng, Sense-based topic word embedding model for item recommendation, *IEEE Access* 7 (2019) 44748–44760.
- [49] X. Zhang, R. Feng, W. Liang, Short text topic model with word embeddings and context information, in: *International Conference on Computing and Information Technology*, Springer, 2018, pp. 55–64.
- [50] S. Li, Y. Zhang, R. Pan, Bi-directional recurrent attentional topic model, *ACM Trans. Knowl. Discov. Data* 14 (6) (2020) <http://dx.doi.org/10.1145/3412371>.
- [51] D.S. Chaplot, R. Salakhutdinov, Knowledge-based word sense disambiguation using topic models, in: *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [52] H. Xu, W. Wang, W. Liu, L. Carin, Distilled wasserstein learning for word embedding and topic modeling, in: *Advances in Neural Information Processing Systems*, 2018, pp. 1716–1725.
- [53] T. Hofmann, Probabilistic latent semantic indexing, in: *SIGIR*, 1999.
- [54] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, 2014, arXiv preprint [arXiv:1409.0473](https://arxiv.org/abs/1409.0473).
- [55] S. Li, J. Li, R. Pan, Tag-weighted topic model for mining semi-structured documents, in: *IJCAI*, 2013.
- [56] O. Bousquet, L. Bottou, The tradeoffs of large scale learning, in: *NIPS*, 2008.
- [57] P. Liang, D. Klein, Online EM for unsupervised models, in: *ACL*, 2009.
- [58] M.D. Hoffman, D.M. Blei, C. Wang, J. Paisley, Stochastic variational inference, *J. Mach. Learn. Res.* (2013).
- [59] T. Hofmann, Probabilistic latent semantic analysis, in: *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann Publishers Inc., 1999, pp. 289–296.
- [60] H. Luo, Z. Liu, H.-B. Luan, M. Sun, Online learning of interpretable word embeddings, in: *EMNLP*, 2015.
- [61] D. Mimno, H.M. Wallach, E. Talley, M. Leenders, A. McCallum, Optimizing semantic coherence in topic models, in: *EMNLP*, 2011.
- [62] D. Newman, J.H. Lau, K. Grieser, T. Baldwin, Automatic evaluation of topic coherence, in: *ACL*, 2010.
- [63] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, G. Solan, G. Wolfman, E. Ruppín, Placing search in context: The concept revisited, in: *WWW*, 2001.
- [64] F. Hill, R. Reichart, A. Korhonen, Simlex-999: Evaluating semantic models with (genuine) similarity estimation, *Comput. Linguist.* (2016).
- [65] T. Luong, R. Socher, C.D. Manning, Better word representations with recursive neural networks for morphology, in: *CoNLL*, 2013.
- [66] G. Halawi, G. Dror, E. Gabrilovich, Y. Koren, Large-scale learning of word relatedness with constraints, in: *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2012.
- [67] E. Bruni, N.-K. Tran, M. Baroni, Multimodal distributional semantics, *J. Artificial Intelligence Res.* (2014).
- [68] B. Shi, W. Lam, S. Jameel, S. Schockaert, K.P. Lai, Jointly learning word embeddings and latent topics, in: *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2017.