Bayesian estimation-based sentiment word embedding model for sentiment analysis

Jingyao Tang | Yun Xue | Ziwen Wang | Shaoyang Hu | Tao Gong | Yinong Chen | Haoliang Zhao | Luwei Xiao

1Guangdong Provincial Key Laboratory of Quantum Engineering and Quantum Materials, School of Physics and Telecommunication Engineering, South China Normal University, Guangzhou, China
2College of Mathematics and Informatics & College of Software Engineering, South China Agricultural University, Guangzhou, China
3School of Foreign Languages, Zhejiang University of Finance & Economics, Hangzhou, Zhejiang, China
4Educational Testing Service, Princeton, New Jersey, USA
5School of Computing, Informatics and Decision Systems Engineering, Arizona State University, Tempe, USA

Abstract
Sentiment word embedding has been extensively studied and used in sentiment analysis tasks. However, most existing models have failed to differentiate high-frequency and low-frequency words. Accordingly, the sentiment information of low-frequency words is insufficiently captured, thus resulting in inaccurate sentiment word embedding and degradation of overall performance of sentiment analysis. A Bayesian estimation-based sentiment word embedding (BESWE) model, which aims to precisely extract the sentiment information of low-frequency words, has been proposed. In the model, a Bayesian estimator is constructed based on the co-occurrence probabilities and sentiment probabilities of words, and a novel loss function is defined for sentiment word embedding learning. The experimental results based on the sentiment lexicons and Movie Review dataset show that BESWE outperforms many state-of-the-art methods, for example, C&W, CBOW, GloVe, SE-HyRank and DLJT1, in sentiment analysis tasks, which demonstrate that Bayesian estimation can effectively capture the sentiment information of low-frequency words and integrate the sentiment information into the word embedding through the loss function. In addition, replacing the embedding of low-frequency words in the state-of-the-art methods with BESWE can significantly improve the performance of those methods in sentiment analysis tasks.

1 | INTRODUCTION
The past decade has witnessed the flourish of word embedding models together with their successful application in natural language processing research. Word embedding models have been widely used in the tasks of entity recognition [1], word sense disambiguation [2, 3], dependency parsing [4], machine translation [5], sentiment analysis [6, 7] and so on. Generally, the current word embedding models can be categorized into two types [8], namely the prediction-based models [9–13] and the count-based models [14–16]. The former type adopts language models to predict the next word based on its context,
whereas the latter uses the global word co-occurrence counts to derive word embeddings. As such, the word embedding models are able to capture the context information and map the semantically similar words into neighbouring points within the word embedding space [17]. Nevertheless, considering that semantically similar words may have opposite sentiment polarities [18], the widely used word embedding models have challenges in dealing with sentiment information. According to Yu et al. [19], among the top 10 semantically similar words to each target word, about 30% of them appears to have an opposite sentiment polarity to that of the target word. For this reason, research is still ongoing to develop sentiment word embedding algorithms that can appropriately tackle both semantic and sentiment information.

Recently, the sentiment word embedding models have been put forward to tackle sentiment information. Similar to the traditional word embedding models, the recently developed sentiment word embedding models can also be classified into the prediction-based models and count-based models. On the one hand, in the prediction-based models [18, 35–37], the word sentiment is taken as the local information for model learning. That is, by using the neural networks or predicting functions, the sentiment word embedding can be learnt and the sentiment polarities can be classified. On the other hand, for the count-based models [20, 21], the sentiment word embedding is derived from the word-sentiment matrices of the labelled corpora, in which case, the sentiment information is used as the global statistics.

Despite of these new models, the application of word embedding models is still limited. Current state-of-the-arts fail to dedicatedly identify and process the low-frequency words in the texts, which result in the loss of low frequency but significant information [22]. Based on the Zipf's law [23], a large number of words in any corpus are often low-frequency words [24]. According to statistics, in the Stanford Sentiment Tree (SST) corpus [25], the words with the occurrence rates lower than 5 account for 72.35% of all the words, and those lower than 10 account for 84.06%. In addition, because of the small number of low-frequency word samples, the learning process is prone to fall in over-fitting of the model. Notably, the low-frequency words carry not only sentiment information but also the noise that might affect model learning. Instead of analyzing the low-frequency words, many current sentiment word embedding models directly utilize the word information for sentiment word embedding learning. For example, in the labeled singular value decomposition (LSVD) model, the word-context matrix and the word-sentiment matrix are stitched together to obtain the sentiment word embedding through the singular value decomposition. Likewise, the LGGVe model and the DLJT1 model use both matrices as the loss function of the least squares algorithm for model learning. In this way, the sentiment word embedding models are absent of extracting the precise sentiment of low-frequency words among the word representations. Thus, this suspending issue will result in the low accuracy of sentiment polarity classification.

On the task of sentiment analysis, we tend to propose a novel method for sentiment word embedding learning. We design and deploy a Bayesian estimation-based sentiment word embedding (BESWE) model. Inspired by the GloVe, we introduce the sentiment probability as well as derive the loss function. Accordingly, the learnt word embedding is deemed best able to convey both the context information and the sentiment information. Seeing that the co-occurrence probability in GloVe is computed via maximum likelihood estimation, which has deficiencies in processing small samples, we thus use the Bayesian estimating method to pick up the sentiment information from the word-sentiment matrices for model learning. By introducing the prior knowledge, the Bayesian estimation can overcome the defect of insufficient sentiment information [26] and increase the robustness of the model [27]. Our model is especially more accurate in capturing the sentiment information of low-frequency words, due to its distinctive structure. On the one hand, we use a Bayesian estimator to compute the co-occurrence probabilities and the sentiment probabilities based on word-context matrices and word-sentiment matrices from the corpus. On the other hand, we exploit the learning of sentiment word embedding in a novel way by dedicatedly constructing a loss function. We compare our method against other state-of-the-art models using identical experimental setups. In these configurations, the proposed model obtains a decent working performance in various natural language processing (NLP) tasks results and is considerably better than classical approaches in low-frequency word sentiment identification.

The major contributions of this work are threefold and can be summarized as follows:

(1) Based on the Bayesian estimation principle, our sentiment probability computation method aims to extract the sentiment information of low-frequency words from the word-sentiment matrix.

(2) Our BESWE model is innovatively developed for the sentiment analysis of low-frequency words, which integrates sentiment information with word embeddings and achieves a higher accuracy in tasks involving low-frequency words and low-frequency sentences sentiment analysis.

(3) The conceptual framework of BESWE can be applied to other word embedding models. By capturing the sentiment of low-frequency words, the sentiment polarity classification accuracy of the current models can be largely improved by integrating BESWE.

The rest of this study is organized as follows. Section 2 presents the prerequisites needed for understanding the proposed model. Section 3 describes the proposed sentiment word embedding model. Section 4 presents the experimental results. A related work to our model is given in Section 5. Conclusions are drawn in Section 6.

2 | PREREQUISITE

This section introduces the basic knowledge of GloVe model and its parameter estimation principle, with the purpose of facilitating the description of subsequent model architecture.
2.1 | GloVe

The GloVe model is a word-embedding method that combines evidence from the local context and the global counts [28]. Typically, the method involves three distinguished words, probability $P$, and evidence from the local context and the global counts [28].

As long as $k$ has the similar relations to $i$ and $j$, that is, both relevant or irrelevant, the ratio $P_{ik}/P_{jk}$ would be close to 1. The information within the ratio of co-occurrence probabilities can be formulated into:

$$P_{ik}/P_{jk} = F(w_i^T \bar{w}_k - w_j^T \bar{w}_k),$$

where $w \in \mathbb{R}^n$ refers to the target word embedding and $\bar{w} \in \mathbb{R}^n$ to the context embedding.

2.2 | Parameter estimation principle

On this occasion, we tend to prove that the co-occurrence probability $P_{ik}$ can be derived from the maximum likelihood estimation.

For every single target word $i$, $x_i$ times Bernoulli experiments are conducted to extract the context independently and randomly [29]. In each experiment, there exist $V$ different outcomes. The occurrence number of the $V$th outcome, together with its probability, is represented by $x_{ik}$ and $P_{ik}$, respectively.

If the random variable $X_i = (X_{i1}, X_{i2}, \ldots, X_{iV})$ stands for the occurrence times of all the possibilities, in which $X_{ik}$ is the number of occurrence for the $k$th one, the parameter $X_i$ must obey the multinomial distribution, which can be written as:

$$X_i \sim \text{Multinomial} \left( \overline{x}_i, \overline{P}_i \right),$$

where $\overline{P}_i = (P_{i1}, P_{i2}, \ldots, P_{iV})$ and $\overline{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iV})$.

Hence, a log-likelihood function can be constructed:

$$\ln L(P_{i1}, P_{i2}, \ldots, P_{ik}, \ldots, P_{iV}),$$

$$= \ln[(P_{i1})^{x_{i1}} \cdot (P_{i2})^{x_{i2}} \cdot (P_{ik})^{x_{ik}} \cdot (P_{iV})^{x_{iV}}],$$

$$= \sum_{k=1}^{V} x_{ik} \cdot \ln P_{ik}. \quad (4)$$

To maximize the log-likelihood function in Equation (4), setting up the target function can be viewed as an equality constrained optimization problem:

$$\begin{cases}
\max & \sum_{k=1}^{V} x_{ik} \cdot \ln P_{ik} \\
\text{s.t.} & \sum_{k=1}^{V} P_{ik} = 1.
\end{cases} \quad (5)$$

Accordingly, the corresponding Lagrangian function can be formulated as:

$$J(P_{i1}, P_{i2}, \ldots, P_{iV}, \lambda) = \sum_{k=1}^{V} x_{ik} \cdot \log P_{ik} + \lambda \left( 1 - \sum_{k=1}^{V} P_{ik} \right). \quad (6)$$

together with determining the partial derivatives of $P_{ik}$

$$\frac{\partial J(P_{i1}, P_{i2}, \ldots, P_{iV}, \lambda)}{P_{ik}} = \frac{x_{ik}}{P_{ik}} - \lambda = 0 \quad (8)$$

and

$$P_{ik} = \frac{x_{ik}}{\lambda} \quad (9)$$

Seeing that the parameter $\lambda$ is with respect to

$$\sum_{k=1}^{V} P_{ik} = \sum_{k=1}^{V} \frac{x_{ik}}{\lambda} = \frac{\sum_{k=1}^{V} x_{ik}}{\lambda} = 1 \quad (10)$$

then we have

$$\lambda = \sum_{k=1}^{V} x_{ik} = x_i \quad (11)$$

It is clear however that the estimation of $P_{ik}$ is written as:

$$\hat{P}_{ik} = \frac{x_{ik}}{x_i} \quad (12)$$

Obviously, the co-occurrence probability in GloVe is equivalent to the maximum likelihood estimating outcome of the co-occurrence probability. Theoretically, the maximum likelihood estimation is a statistical estimating method on
abundant samples. In contrast, the Bayesian estimation is more effective in small sample processing [26].

3 | METHODOLOGY

This section introduces the architecture of our BESWE model and its working principle.

Notation:

- For \( i \in \{1, 2, \ldots, V\} \) and \( V \) representing the number of words, we define \( x \) as the word-context matrix, together with \( x_{ik} \) and \( P_{ik} \) respectively as the occurrence counts and the occurrence probability of word \( k \) in context of word \( i \), which are from the sets \( \mathcal{X} = (x_{i1}, x_{i2}, \ldots, x_{iV}) \) and \( \mathcal{P} = (P_{i1}, P_{i2}, \ldots, P_{iV}) \).
- \( x_i = \sum_{k=1}^{V} x_{ik} \) stands for the total occurrence counts of any word within the context of word \( i \).
- For \( t_i = (t_{i0}, t_{i1}) \), we define \( t \) as the word-sentiment matrix, together with \( t_{i0} \) as the number of positive texts including word \( i \) and \( t_{i1} \) as the number of negative texts including word \( i \), where \( t_i = t_{i0} + t_{i1} \).
- For \( B_i = (1 - B_{i0}, B_{i1}) \), we define \( B_i \) as the probability of word \( i \) being positive and \( 1 - B_i \) as the probability of word \( i \) being negative.
- \( w_i \) stands for word embedding of word \( i \) as target.
- \( \tilde{w}_i \) stands for word embedding of word \( i \) as context.
- \( s_t \) stands for bias embedding of word \( i \).

3.1 | Model architecture

The architecture of the BESWE model is shown in Figure 1. Our sentiment word embedding model is developed based on the foundation of GloVe. Since the feasibility of maximum likelihood estimation for parameter calculation in GloVe is demonstrated, we start with using the co-occurrence probability for word context learning and the sentiment probability for word sentiment learning. By traversing the corpus with sentiment labels, the co-occurrence counts \( \mathcal{X} \) and sentiment counts \( \mathcal{T} \) can be obtained. Aiming at dealing with low-frequency words, the co-occurrence number is sent to Bayesian estimator for co-occurrence probability computation. Likewise, the sentiment probability can be obtained in the same manner by using sentiment counts of the word. At this stage, the estimation of \( \ln P_{ik} \) and \( \ln B_{ik} \) can be obtained, which are \( c_{ik} \) and \( e_i \), respectively. Based on these two outcomes, the loss function is constructed for sentiment word embedding learning, which is delivered as:

\[
\text{loss}(w_i, \tilde{w}_i, s_t) = \sum_{k=1}^{V} \left[ w_i^T \tilde{w}_k + w_i^T s_t - c_{ik} - e_i \right]^2.
\]  

The loss function is minimized via AdaGrad optimizer [30]. Along these processes, the learning of \( w_i, \tilde{w}_i \) and \( s_t \) is conducted where \( w_i \) stands for the sentiment word embedding of the BESWE model.

More details of the Bayesian estimator and the loss function of sentiment word embedding learning are described as follows.

3.2 | Bayesian estimator

Based on the working principle of the proposed model, the co-occurrence probability and the sentiment probability are calculated by the Bayesian estimator. Distinctively, the estimation of co-occurrence probability comes from the current D-GloVe model [16]. According to the parameter distribution assumption in Section 2.2, \( P_i \) is assumed to obey the Dirichlet prior distribution with parameter \( \alpha = (\alpha_1, \alpha_2, \ldots, \alpha_V) \), as shown in Figure 2. Thus, the Bayesian estimation of \( \ln P_{ik} \) is:

\[
c_{ik} = E_p\left[ P_i | \mathcal{X} \right] \ln P_{ik},
\]

\[
\approx \ln E_p\left[ P_i | \mathcal{X} \right] P_{ik} - \frac{V \text{ar}_p\left[ P_i | \mathcal{X} \right]}{2 E^2_p\left[ P_i | \mathcal{X} \right]} P_{ik}.
\]  

Together with

\[
\text{Var}_p\left[ P_i | \mathcal{X} \right] P_{ik} = \frac{\alpha_k + x_{ik}}{\sum_k(\alpha_k + x_{ik})} \cdot \frac{1}{\sum_k(\alpha_k + x_{ik}) + 1},
\]

\[
E_p\left[ P_i | \mathcal{X} \right] P_{ik} = \frac{\alpha_k + x_{ik}}{\sum_k(\alpha_k + x_{ik})},
\]  

and

\[
\alpha_k = \lambda_1 \cdot \frac{n_k}{\sum_k n_k},
\]

where \( n_k \) is the occurrence time of word \( k \) in the learning samples, and \( \lambda_1 \) is the regulatory factor.
In addition, we can also compute the Bayesian estimating outcome of \( \ln B_i \). The Bernoulli experiments are applied to pick up the sentiment polarity of the target word \( i \) for \( t_i \) times, which is either positive or negative. As presented above, \( B_i \) and \( t_i \) represent the probability and the occurrence time of word \( i \) being positive while \( 1 - B_i \) and \( t_i \) indicate those of negative. If \( T_i = (T_{i0}, T_{i1}) \) denotes the occurrence times of both outcomes, the random variable \( T_i \) is within the binomial distribution, that is, \( T_i \sim \text{Binomial}(T_{i1}, B_i) \). Comparatively, \( P(B_i) \) is set to obey beta distribution with the parameter \( \beta = (\beta_{i0}, \beta_{i1}) \). The distribution of aforementioned parameters is illustrated in Figure 3.

Therefore, the solution of \( \ln B_i \) can be:

\[
e_i = E_p(B_i \mid T_i) [\ln B_i],
\]

\[
= \frac{V ar_p(B_i \mid T_i)}{2} - \frac{1}{2} E_p(B_i \mid T_i) [B_i] \approx \ln E_p(B_i \mid T_i) [B_i]
\]

where

\[
V ar_p(B_i \mid T_i) [B_i] = \frac{(\beta_{i1} + t_{i1}) \cdot (1 - \beta_{i1} + t_{i1})}{\sum_k (\beta_{ik} + t_{ik})},
\]

\[
E_p(B_i \mid T_i) [B_i] = \frac{\beta_{i1} + t_{i1}}{\sum_k (\beta_{ik} + t_{ik})}
\]

and

\[
\beta_{ik} = \lambda_2 \cdot \frac{m_{ik}}{\sum_k m_{ik}},
\]

where \( m_{ik} \) stands for the number of texts of the sentiment label \( k \), and \( \lambda_2 \) is the regulatory factor for word sentiment learning.

3.3 | Loss function

Considering the semantic information characterization in GloVe, we now take the ratio of sentiment probabilities to address the relationship of word sentiment. More details about the words’ sentiment relation and the ratio are presented in the Appendix 7.1. For words \( i \) and \( j \), the sentiment relation is expressed as \( B_i / B_j \). The objective function is established as:

\[
F(w_i^T s_i - w_j^T s_j) = B_i / B_j,
\]

where \( w_i \) and \( s_j \) stand for the word embedding and bias embedding of word \( j \), respectively.

Assuming that \( F \) is confirmed as the homomorphism between groups \((\mathbb{R}, +)\) and \((\mathbb{R}_{>0}, \times)\), we incorporate the sentiment information into the word embeddings. Based on Equations (2) and (22), we get:

\[
F\left(\left(w_i^T \tilde{w}_k - w_j^T \tilde{w}_k\right) + \left(w_i^T s_i - w_j^T s_j\right)\right),
\]

\[
= F\left(\left(w_i^T \tilde{w}_k - w_j^T \tilde{w}_k\right) \cdot F\left(w_i^T s_i - w_j^T s_j\right)\right),
\]

\[
= \frac{P_{ik}}{P_{jk}} \cdot \frac{B_i}{B_j}.
\]

Considering the properties of group homomorphism, we transform the above equation according to the addition commutative law into:

\[
F\left(\left(w_i^T \tilde{w}_k - w_j^T \tilde{w}_k\right) + \left(w_j^T s_i - w_i^T s_j\right)\right),
\]

\[
= F\left(\left(w_i^T \tilde{w}_k + w_j^T s_i\right) - \left(w_j^T \tilde{w}_k + w_i^T s_j\right)\right),
\]

\[
= \frac{F\left(w_i^T \tilde{w}_k + w_j^T s_i\right)}{F\left(w_i^T \tilde{w}_k + w_j^T s_j\right)} = \frac{P_{ik}}{P_{jk}} \cdot \frac{B_i}{B_j}.
\]

In line with the basic theory of GloVe, the loss function of BESWE can be constructed via the same manner:
\[
\text{loss}(\mathbf{w}_i, \mathbf{w}_k, s_i) = \sum_{k=1}^{V} \left[ w_i^T \mathbf{w}_k + w_i^T s_i - c_{ik} - e_i \right]^2,
\]

where \(c_{ik}\) and \(e_i\) are derived from Equations (14) and (18), respectively.

4 | EXPERIMENT

In this section, the working performance of the proposed BESWE model is evaluated. Some state-of-the-art word embedding models, along with the learning of specific word representations, are taken for comparison. To this end, the task of word similarity is carried out. So as to deliver the sentiment embeddings, the word- and the sentence-level sentiment analysis using different models is taken into account. The sentiment analysis tasks are further divided into three subtasks, which verify the efficiency on basic sentiment classification for all words, low-frequency word sentiment identification and integration with other baseline models. The outline of the experiments is exhibited in Figure 4.

4.1 | Experiment settings

Dataset of word embeddings: The dataset SST is used for the model training. There are five classes annotations within SST, which are originally ‘very negative’, ‘negative’, ‘neutral’, ‘positive’ and ‘very positive’. To facilitate the processing, the classes ‘positive’ and ‘very positive’ are taken to represent the polarity of positive while the ‘negative’ and ‘very negative’ represent the negative. As such, both categories are applied to the experiments.

Baseline models: We compare the effectiveness of the proposed model to other widely used models. Specifically, the models of word embeddings, such as C&W [10], CBOW [13] and GloVe [15], together with models of sentiment embeddings, including SE-HyRank [18] and DLJT1 [21], are implemented. For all the models used in this work, the word representation dimension is 50, the learning rate is set as 0.05. All the parameters associated are finetuned to get better results.

Task 1, Word similarity measure: The capacity of word embedding models is verified via the processing in standard word similarity. Comprehensively, the average working performance on word similarity tasks is obtained on the dataset EN-WS-553-ALL, EN-WS-553-SIM and SCWS. We utilize the word embeddings to calculate the similarity score of the word pairs first. Together with the standard similarity score provided by the dataset, we shall thus compute the correlation coefficient. The evaluation metric in this task is the computed correlation coefficient, which is detailed elsewhere [31].

Task 2, Word-level sentiment analysis: On this occasion, the support vector machine classifier is used for training the words vectors, with each word vector representing specific sentiment. The popular sentiment polarity lexicons, NRC and MPQA [32], are taken as the ground truth of the word sentiments. The number of positive and negative words for MPQA is 2301 and 4151 while those for NRC is 2231 and 3324. The N-fold cross validation with the value \(N = 5\) and \(N = 10\) is performed on all datasets. The working performance refers to the classification accuracy of the models.

The word-level sentiment analysis is carried out via the following three subtasks.

Task 2.1, Basic word-level sentiment analysis: The usual task for evaluating the performance is to capture the sentiment information of the words. Thus, we generate the word embeddings and apply them to the sentiment classification of the sentiment lexicon.

Task 2.2, Low-frequency word sentiment analysis: Since the Bayesian estimation principle is able to deal with low-frequency words, we pick the words with the frequencies lower than 5 and 10 from the SST corpus for investigation. In this way, the low-frequency word embeddings are collected for sentiment analysis.

Task 2.3, BESWE integration with other models: In order to improve the classification accuracy, we integrate BESWE with the baseline models. A specific word embedding set is developed, which contains low-frequency word embeddings from BESWE and nonlow-frequency word embeddings from any other baseline model. By using these word embeddings for model learning, the sentiment analysis is carried out.

Task 3, Sentence-level sentiment analysis: Considering the sentiment analysis for sentence, the Movie Review dataset [33] is used, which contains 10,622 samples with the proportion of each polarity 1:1. We use a convolutional neural network (CNN), namely text-CNN, with its online implementation [34]. The inputs of text-CNN are word embeddings. The samples are divided into training, validating and testing, with the ratio of 6:2:2. The training episode is set as 200 epochs using the default settings. Based on the validation outcomes, we apply the optimal model to testing. The evaluation metrics are the classification accuracy and F1 score.

Similarly, there are also three subtasks like in the word-level sentiment analysis.

Task 3.1, Basic sentence-level sentiment analysis: As stated in Task 2.1, we take the words from Movie Review for sentiment analysis.

Task 3.2, Low-frequency sentence sentiment analysis: In this task, we select the low-frequency words whose occupation is over 10% in the sentence to be the low-frequency sentence samples. Thereby the low-frequency sentences are taken for sentiment analysis.

Task 3.3, BESWE integration with other models: Similar to Task 2.3, the integration of BESWE with each baseline model is devised to deal with both the low-frequency words and the nonlow-frequency words in the sentences from the Movie Review dataset.

4.2 | Experimental results

Task 1, Word similarity measure: The outcomes of the word similarity task are given in Table 1. It is apparent that our
BESWE method outperforms the other methods on all three datasets, verifying the capability of capturing sufficient semantic information for analysis. The sentiment word embeddings are generally more competitive than the basic word embeddings on the task of word similarity. The main reason is that, by exploiting the sentiment information, semantically more accurate word embeddings are obtained to optimize the working performance. Moreover, the Bayesian estimation
principle is effective in tackling the large number of low-frequency words in the corpus, based on which more accurate sentiment word embeddings are accessible. Accordingly, one can easily see a considerable gap between our model and the baseline models.

Task 2.1, Basic word-level sentiment analysis: The word-level analysis is carried out on the dataset of single-word entries. For basic word-level sentiment analysis, our model is proved to be a competitive alternative to classical word embedding models (Table 2). Furthermore, the BESWE model obtains the best outcome with 10-fold cross validation on the dataset NRC.

Task 2.2, Low-frequency word sentiment analysis: The BESWE model shows a better accuracy in tackling low-frequency words, as shown in Table 3. For the frequencies lower than $5$ (LF5) and $10$ (LF10), our model shows its stability in sentiment polarity classification, compared to the baseline models. As mentioned in the result of Task 1, the low-frequency word processing ability contributes to establishing precise sentiment word embeddings. By contrast, there is also a performance gap between BESWE and other algorithms for the dataset NRC.

Task 2.3, BESWE integration with other models: In line with the results of Task 2.2, we combine the proposed model to baseline models, not only to show its superiority, but also to optimize the word-level sentiment analysis outcomes of baseline models. As long as the BESWE algorithm is capable of processing low-frequency words, the low-frequency word embeddings (i.e. LF5 and LF10 from BESWE) are incorporated into the nonlow-frequency word embeddings from the baseline. The outcomes of the integration method are reported in Table 4. The classification accuracy is significantly improved for most models. The maximum performance gap of 4.44% is observed in the integration of DLJT1 and BESWE against the basic DLJT1 for the dataset MPQA.

Task 3.1, Basic sentence-level sentiment analysis: The effectiveness of the proposed model is further evaluated on the sentence-level sentiment analysis tasks. From Table 5, we see that the BESWE model has a better accuracy than any other baseline methods.

Task 3.2, Low-frequency sentence sentiment analysis: In line with the outcome on low-frequency words, the sentiment analysis on low-frequency sentence exceeds the baselines in both evaluation settings. The highest classification accuracy of the BESWE, based on the word frequencies lower than 5 (LF5), is 81.25%, as shown in Table 6. Meanwhile, the maximum performance gap of 9.75% is observed against the DLJT1. Comparing Tables 5 and 6, the improvement in accuracy against the baseline in Table 6 outperforms that in Table 5. One possible explanation is that the low-frequency sentences in Task 3.2 contain more low-frequency words. Since the sentiment word embeddings learnt via the Bayesian principle is considerably more informative, it is reasonable to expect better performance in low-frequency words analysis, as it is the case. Besides, the accuracy on ‘LF5’ in Table 6 over-performs that of ‘LF10’, which further verifies the significance of our model.

Task 3.3, BESWE integration with other models: Similar to Task 2.3, since the working performance on low-frequency sentence is highlighted, we integrate the corresponding word embeddings to improve the classification accuracy of the baseline. The outcomes of the integration methods, which combine the baseline and the proposed BESWE, are reported in Table 7. In contrast, these outcomes fail to exceed that from the direct use of the proposed model. That is, for the sentiment analysis of low-frequency words, BESWE always obtains the best and most consistent results in the identification of sentiment polarity. Whereas, the application of BESWE can narrow the performance gap. For the C&W model, a 10% improvement is presented.

Effects of $\lambda_1$ and $\lambda_2$: The hyperparameters in the BESWE model include the regulatory factors $\lambda_1$ and $\lambda_2$ used to represent the semantic and sentiment information. In this experiment, to obtain the optimal settings, we vary the values of $\lambda_1$ and $\lambda_2$ within $\{1, 0.75, 0.5, 0.25, 0.1, 0.05, 0.02, 0.01\}$ to learn the BESWE model. In this way, we get 64 different BESWE models.

The results on the low-frequency sentence sentiment analysis against different hyperparameter settings are shown in Figures 5 and 6. The former indicates the accuracy on the lower-than-5-frequency words while the latter indicates that of the lower-than-10-frequency words. The variation of $\lambda_1$ does not cause a significant difference, while that of $\lambda_2$ has a negative correlation with the accuracy. According to Figure 5, the highest accuracy of the sentence-level sentiment analysis is 81.25% at the point $\lambda_1 \in \{0.05, 0.75\}$ and $\lambda_2 = 0.01$. Likewise, in Figure 6, the optimal values of $\lambda_1$ and $\lambda_2$ are both 0.01, which lead to an accuracy of 80.50%.

| Table 1: Word similarity results |
|---------------------------------
| **Model** | **EN-WS-353-ALL** | **EN-WS-353-SIM** | **SCWS** |
| C&W | 8.36 | 10.34 | 39.64 |
| CBOW | 3.50 | 6.53 | 49.17 |
| GloVe | 7.31 | 8.77 | 40.14 |
| SE-HyRank | 7.51 | 11.52 | 45.55 |
| DLJT1 | 17.38 | 20.55 | 43.76 |
| **BESWE** | **17.85** | **24.88** | **49.25** |

| Table 2: Basic word-level sentiment analysis results |
|---------------------------------------------|
| **Model** | **MPQA** | **NRC** |
| **Method** | **5-Fold** | **10-Fold** | **5-Fold** | **10-Fold** |
| C&W | 63.86 | 63.87 | 57.86 | 56.50 |
| CBOW | 63.86 | 63.87 | 57.21 | 56.71 |
| GloVe | 62.30 | 63.33 | 58.21 | 59.64 |
| SE-HyRank | 61.60 | **64.65** | 59.79 | 61.07 |
| DLJT1 | **66.67** | 62.93 | **63.00** | 61.29 |
| **BESWE** | 64.25 | 63.94 | 58.71 | **64.71** |
To sum up, these experimental results clarify the effectiveness of the proposed sentiment word embedding. The BESWE model outperforms other state-of-the-art models in word similarity measure. In the sentiment analysis at both word and sentence levels, our method still shows comparable outcomes. Specifically, our model produces considerably better results than the baseline methods on sentiment information capturing of both low-frequency words and low-frequency sentences. Moreover, by integrating the low-frequency word embeddings from BESWE to other...
models, the classification accuracies of the baseline models improve to a large extent.

5 | RELATED WORK

5.1 | Word embeddings

As pointed out in the Introduction section, both prediction-based word embedding models and count-based word embedding models are applied to learn word embeddings [8]. Fundamentally, Bengio et al. [9], establish neural network language model to predict target words using preceding contexts, and thus to learn the word embeddings. Following this theory, Collobert and Weston [10,11] put forward a CNN to predict the target word, based on not only preceding but also succeeding contexts. As currently edge-cutting prediction-based methods, CBOW and skip-gram models [12,13] are of simple single-layer architecture. Both of these models can efficiently compute word embeddings from large-scale datasets. Besides, researchers also focus on using the global word context co-occurrence counts in the corpus for learning word embeddings, which are taken as the basis of count-based methods. Deerwester et al. [14] novelty propose the latent semantic analysis model for exploiting word-document co-occurrence counting to learn word embedding for the first time. As a most widespread model, GloVe [15] computes the word-word co-occurrence counts via a specific weighted least squares model. A consistent and competitive result can be obtained on the tasks of sentiment analysis.

5.2 | Sentiment word embeddings

Likewise, prediction-based sentiment word embedding models and count-based sentiment word embedding models are dedicatedly studied. In 2011, Maas et al. [35] apply a logistic regression as the predictor to learn sentiment word embeddings. On the other hand, an approach based on recursive autoencoders [36] is designed that year to learn the vector representations of phrases and full sentences, which exploits the vector representations at each node of the hierarchy and uses softmax classifier for sentiment label prediction. Tang et al. [18] propose the hybrid ranking method for learning sentiment embeddings by regulating the traditional C&W model, which encodes sentiment information in the continuous representation of words. With the development of deep learning networks, Lan et al. [37] construct a CNN to detect the semantic and sentiment information. Hereafter, two kinds of information are integrated to generate sentiment word vectors. In terms of count-based sentiment word embedding models, Li et al. [21] incorporate the sentiment count into model learning by proposing a variety of count-based models (e.g. DLJT1) on the foundation of GloVe. Furthermore, LSVD and LGloVe [20] are developed as the improvement of SVD and GloVe, respectively. In these models, the application of word-label counts facilitates the learning of sentiment word embedding with label information.

Different from these works, our work mainly focuses on resolving the issues of low-frequency word sentiment analysis. Little attention is paid to this topic currently within the NLP domain. Our model targets at extracting the sentiment information by obtaining a more accurate sentiment word embedding. Notably, our model is orthogonal to the aforementioned models. Our model overperforms the state-of-the-art while the adaptation of our approach into these methods also results in a better working performance.

6 | CONCLUSIONS

In this work, a novel BESWE model is designed and deployed on the tasks of sentiment analysis. Aiming to obtain both the semantic and the sentiment information, a Bayesian estimator is developed to compute the co-occurrence probability and the sentiment probability. Furthermore, a loss function for the purpose of sentiment word embedding learning is constructed. We test our model on a variety of tasks to evaluate its working performance. Experimental results indicate that the BESWE model is a comparable alternative to the state-of-the-art methods in word similarity identification, word- and sentence-level sentiment analysis. Specifically, our model outperforms other methods on low-frequency word and low-frequency sentence sentiment polarity classification to demonstrate its efficacy. By integrating the BESWE into the baseline, the classification accuracy can be improved considerably compared to the basic models.

This study offers a creative and practical method for both the semantic and sentiment information capturing.
Distinctively, our model shows its superiority in dealing with low-frequency words and thus results in a higher accuracy in sentiment analysis.

ACKNOWLEDGEMENTS
This work was supported by the National Statistical Science Research Project of China under Grant No. 2016LY98, the Science and Technology Department of Guangdong Province in China under Grant Nos. 2016A010101003 and 2016A010101021 and 2016A0101010122, the Characteristic Innovation Projects of Guangdong Colleges and Universities (Nos. 2018KTSCX049 and 2018GKTSCX069), the Science and Technology Plan Project of Guangzhou under Grant Nos. 201802010033 and 201903010013, the Bidding Project of Laboratory of Language Engineering and Computing of Guangdong University of Foreign Studies (No. LEC2019 ZBKT005).

ORCID
Jingtao Tang https://orcid.org/0000-0003-1651-8480

REFERENCES
1. Turian, J., Ratinov, L., Bengio, Y.: Word representations: a simple and general method for semi-supervised learning. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pp. 384–394 (2010)
2. Iacobacci, I., Pilehvar, M.T., Navigli, R.: Embeddings for word sense disambiguation: an evaluation study. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, vol. 1, pp. 897–907 (2016)
3. Dave, V.S., et al.: Neural-brane: neural Bayesian personalized ranking for attributed network embedding. Data Sci. Eng. 4, 119–131 (2019)
4. Bansal, M., Gimpel, K., Livescu, K.: Tailoring continuous word representations for dependency parsing. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, vol. 2, pp. 809–815 (2014)
5. Ding, Y., et al.: Visualizing and understanding neural machine translation. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, vol. 1, pp. 1150–1159 (2017)
6. Wang, X., et al.: Predicting polarities of tweets by composing word embeddings with long short-term memory. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, vol. 1, pp. 1343–1353 (2015)
7. Xiao, L., et al.: Targeted sentiment classification based on attentional encoding and graph convolutional networks. Appl. Sci. 10, 957 (2020)
8. Baroni, M., Dinu, G., Kruszewski, G.: Don’t count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, vol. 1, pp. 238–247. Long Papers, (2014)
9. Bengio, Y., et al.: A neural probabilistic language model. J. Mach. Learn. Res. 3, 1137–1155 (2003)
10. Ronan, C., Jason, W.: A unified architecture for natural language processing: deep neural networks with multitask learning. In: International Conference on Machine Learning, pp. 160–167 (2008)
11. Ronan, C., et al.: Natural language processing (almost) from scratch. J. Mach. Learn. Res. 2493–2537 (2011)
12. Mikolov, T., et al.: Efficient estimation of word representations in vector space. In: International Conference on Learning Representations (2013)
13. Mikolov, T., et al.: Distributed representations of words and phrases and their compositionality. In: Proceedings of the 26th International Conference on Neural Information Processing Systems, vol. 2, pp. 3111–3119 (2013)
14. Deerwester, S., et al.: Indexing by latent semantic analysis. J. Am. Soc. Inf. Sci. 41, 391–407 (1990)
15. Pennington, J., Socher, R., Manning, C.: Glove: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014)
16. Jameel, S., Schockaert, S.: D-GloVe: a feasible least squares model for estimating word embedding densities. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pp. 1849–1860 (2016)
17. Fu, P., et al.: Learning sentiment-specific word embedding via global sentiment representation. In: Thirty-Second AAAI Conference on Artificial Intelligence, pp. 4808–4815 (2018)
18. Tang, D., et al.: Sentiment embeddings with applications to sentiment analysis. IEEE Trans. Knowl. Data Eng. 28, 496–509 (2016)
19. Yu, L.C., et al.: Refining Word Embeddings Using Intensity Scores for Sentiment Analysis, vol. 26, pp. 671–681. IEEE Press (2018)
20. Yang, L., et al.: Improving word representations with document labels. IEEE/ACM Trans. Audio, Speech, Lang. Process. 25, 863–870 (2017)
21. Li, Y., et al.: Learning word representations for sentiment analysis. Cognit. Comput. 9, 843–851 (2017)
22. Wu, Y., et al.: CASR: a context-aware residual network for single-image super-resolution. Neural. Comput. Appl. 1–16 (2019)
23. Zipf, George K.: Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology. Ravenio Books (2016)
24. Guo, J., et al.: Revisiting embedding features for simple semi-supervised learning. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 110–120 (2014)
25. Socher, R., et al.: Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 1631–1642 (2013)
26. Phoong, S.Y., Ismail, M.T.: A comparison between Bayesian and maximum likelihood estimations in estimating finite mixture model for financial data. J. Sleep Med. 44, 1033–1039 (2015)
27. Wu, Y., et al.: Local and global Bayesian network based model for flood prediction. In: 2018 24th International Conference on Pattern Recognition (ICPR), pp. 225–230 (2018)
28. Sergio, J., et al.: word2met: WordNet-based word representation rivaling neural word embedding for lexical similarity and sentiment analysis. IEEE Comput. Intell. Mag. 14, 41–53 (2019)
29. Dijacić, PM., Huang, Y.: Estimation of a Bernoulli parameter p from imperfect trials. IEEE Signal Process. Lett. 7, 160–163 (2000)
30. John, D., Elad, H., Yoram, S.: Adaptive subgradient methods for online learning and stochastic optimization. In: COLT 2010 – the 23rd Conference on Learning Theory, pp. 257–269 (2010)
31. Levy, O., Goldberg, Y., Dagan, I.: Improving distributional similarity with lessons learned from word embeddings. Trans. Assoc. Comput. Linguist. 3, 211–225 (2015)
32. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing contextual polarity in phrase-level sentiment analysis. In: Proceedings of the conference on human language technology and empirical methods in natural language processing, pp. 347–354 (2005)
33. Pang, B., Lee, L.: Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, pp. 115–124 (2005)
34. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1746–1751 (2014)
35. Maas, A.L., et al.: Learning word vectors for sentiment analysis. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1746–1751 (2014)
36. Socher, R., et al.: Semi-supervised recursive autoencoders for predicting sentiment distributions. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pp. 151–161 (2011)
37. Lan, M., et al.: Three convolutional neural network-based models for learning sentiment word vectors towards sentiment analysis. In: 2016 International Joint Conference on Neural Networks (IJCNN), pp. 3172–3179 (2016)
APPENDIX

SENTIMENT RELATION OF DIFFERENT WORDS USING BAYESIAN ESTIMATION

For $W_i = \text{good}$ and $W_j = \text{bad}$, we have $B_i/B_j > 1$, $(1 - B_i)/(1 - B_j) < 1$, that is, $W_i = \text{good}$ is more positive than $W_j = \text{bad}$.

For $W_i = \text{good}$ and $W_j = \text{great}$, we have $B_i/B_j \approx 1$, $(1 - B_i)/(1 - B_j) \approx 1$, that is, $W_i = \text{good}$ and $W_j = \text{great}$ are of similar sentiment polarities, which are both positive.

For $W_i = \text{then}$ and $W_j = \text{home}$, we have $B_i/B_j \approx 1$, $(1 - B_i)/(1 - B_j) \approx 1$, that is, $W_i = \text{then}$ and $W_j = \text{home}$ are of similar sentiment polarities, which are both neutral.

Specific sentiment relations calculated by Bayesian estimation are presented in Table 8.

| Probability and ratio | $W_i = \text{good}$ | $W_i = \text{good}$ | $W_i = \text{then}$ | $W_i = \text{home}$ |
|-----------------------|---------------------|---------------------|---------------------|---------------------|
| $B_i$                 | 0.5872              | 0.5872              | 0.3819              |                     |
| $B_j$                 | 0.0886              | 0.7053              | 0.4864              |                     |
| $B_i/B_j$             | 6.6272              | 0.8326              | 0.7852              |                     |
| $(1 - B_i)/(1 - B_j)$ | 0.1509              | 1.2011              | 1.2735              |                     |