Word Representation with Salient Features

Zhang, M., Palade, V., Wang, Y. & Ji, Z.

DOI 10.1109/ACCESS.2019.2892817
ISSN 2169-3536
ESSN 2169-3536

Publisher: IEEE Open Access Options

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 9, 2018, accepted December 31, 2018, date of publication January 14, 2019, date of current version March 20, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2892817

Word Representation With Salient Features

MING ZHANG[1], VASILE PALADE[2], (Senior Member, IEEE), YAN WANG[1], AND ZHICHENG JI[1]

[1]Engineering Research Center of Internet of Things Technology Applications, Ministry of Education, Jiangnan University, Wuxi 214122, China
[2]School of Computing, Electronics and Mathematics, Coventry University, Coventry CV1 5FB, U.K.

Corresponding authors: Vasile Palade (vpalade453@gmail.com) and Yan Wang (wangyan@jiangnan.edu.cn)

This work was supported in part by the National Nature Science Foundation of China under Grant 61572238, and in part by the Outstanding Youth Foundation of Jiangsu Province under Grant BK20160001.

ABSTRACT Inspired from the idea that the contexts in which a word occurs are of different significance, this paper proposes a novel method, called word representation with Salient Features (SaFe), to represent words using salient features selected from the context words. The SaFe method employs the point-wise mutual information (PMI) method with scaled context window to measure word association between a target word and its context. Then, contexts having word associations will be selected as salient features, where the number of salient features for a given word is decided by the ratio between the number of unique contexts and the total counts of occurrences in the whole corpus. The SaFe method can be used with the positive PMI matrix (PPMI), with each row representing a word, hence the name SaFe-PPMI. Moreover, the SaFe-PPMI model can be further decomposed by using the truncated singular vector decomposition technique to obtain dense vectors. In addition to efficient computation, the new models can achieve remarkable improvements in seven semantic relatedness tasks, and they show superior performance when compared with the state-of-the-art models.

INDEX TERMS Point-wise mutual information, salient features, singular vector decomposition, word representation.

I. INTRODUCTION

Semantic relatedness is a metric estimating the degree to which two terms are related, whereas semantic similarity is a subclass of semantic relatedness, which aims to evaluate the likeness of words' meanings. Semantic relatedness includes all kinds of relationships, such as meronymy (as part of), as in tail-dog relationship, hyponymy (type-of), as in knife-cutlery relationship, and hypernymy, as in fruit-orange relationship, while semantic similarity only refers to the ''is-a'' relationship, as claimed in [1]. Recently, a variety of semantic relatedness measures have been proposed and widely used in natural language processing (NLP) tasks, such as text classification [2], information retrieval [3], word sense disambiguation [4] and discovering word senses from text [5]. The underlying factor that determines the efficiency of semantic relatedness measures relies on word representation, which can be classified into two types, i.e., the count-based distributional models and the neural-network-inspired models. Both types of models intrinsically depend on a bag-of-contexts architecture based on the hypothesis that words occurring in the same contexts tend to have similar meanings [6].

The departure point of the count-based distributional models is to extract statistical information from a large corpus, aiming to build a term-document or a term-term co-occurrence matrix [7], [8]. Apparently, the row vectors of these matrices denote the contexts in which a target word occurs. In order to improve the performance, some other reweighting transformations of co-occurrence were proposed, where PMI can be a suitable choice for the word association of a word-context pair [9]. Unlike the count-based models, the objective of neural-network-inspired models mainly concentrates on optimally predicting the corresponding contexts in which a target word tends to appear. Words are represented as real-valued vectors embedded in a low-dimensional space, which is also known as word embedding [10], [11]. A series of papers published by Mikolov *et al.* [12]–[14] described an efficient program, called word2vec, for word embedding, where the proposed Skip-Gram with Negative Sampling method (SGNS) achieved the state-of-the-art performance. The training process of the SGNS method is targeted on individual word-context pairs by predicting the log-probability that a word appears in the context of a given word. Another well-known model, called GloVe [15], was trained on the non-zero entries in a global word-context co-occurrence matrix, by optimizing a least squares regression formulation.

Clearly, all co-occurrences of word-context pairs in a corpus are observed and covered in the training processes of both the traditional distributional models and the neural-network-inspired models regardless of the various training objectives. However, some papers demonstrated that using a small portion of the word-context occurrences can obtain better performance for word representation in terms of semantic relatedness. For example, Islam and Inkpen [16] introduced the Second Order Co-occurrence PMI (SOC-PMI) method, which sorted the lists of contexts for two target words by PMI values and then calculated the relative semantic similarity by aggregating their positive PMI values (from the opposite list) of the common contexts in both lists [16]. As an extension of this research, Hassan and Mihalcea [17] made use of semantic profiles constructed from salient encyclopedic features to calculate semantic relatedness, where a profile was consisted of salient concepts appearing within contexts across a very large corpus.

Drawing inspiration from these works, it can be assumed that not all features (namely, contexts for a given word) can contribute to representing words for measuring semantic relatedness. In other words, the meaning of a word can be characterized by salient features, since certain features may be redundant or harmful. In this paper, a new method called Word Representation with Salient Features (SaFe) is presented. After sorting the features for a target word descendently by positive PMI values, a small number of features, viewed as salient features, will be decided and selected from all the contexts to represent the word. In addition, a novel approach is proposed to determine the number of salient features for each word, which is based on the ratio of the number of unique contexts of a word and the total counts of occurrences in the corpus. Moreover, two distributional models are proposed, one of which is to apply the SaFe method on the positive PMI (PPMI) matrix [18], named as SaFe-PPMI; and the other one is to utilize the truncated Singular Value Decomposition (SVD) technique to decompose the SaFe-PPMI model in order to obtain dense vectors for words [19]. Despite the efficient computation, the performance of the SaFe-PPMI model has dramatic improvements on semantic relatedness tasks, and the SaFe-SVD model can significantly outperform some state-of-the-art models, including the SGNS and GloVe models.

This paper is structured as follows: Section 2 introduces the related work and Section 3 describe the details of the two distributional models based on the SaFe method; experiments and analysis are presented in Section 4; and conclusions are given in Section 5.

II. RELATED WORK

A. COUNT-BASED DISTRIBUTIONAL MODELS

Lexical co-occurrence is constantly being viewed as a crucial indicator for semantic relatedness and has motivated a variety of methods, such as Jaccard [20], Dice [21], Ochiai [22] and PMI [23]. However, these early works only focus on measuring word association but ignore the representation for words.

The count-based distributional models aim to construct co-occurrence matrix from a corpus to generate global representations for words, where the forms of co-occurrence matrix vary in different ways. The term-document matrix proposed by Lund and Burgess [8] and the term-term matrix introduced in [7] are two early forms. Intuitively, each cell in the matrix denotes the context in which a target word occurs, and thus each row can be viewed as the representation of a word. It can be noted that the sparse and high-dimensional nature of such representations cannot efficiently generalize global information and even add a burden on computation. Therefore, the SVD technique was used to perform dimensionality reduction by matrix factorization to generate low-dimensional word representations. Suffering from the disproportionate contribution of the most frequent words like ''the'' or ''is'', it is necessary for these methods to re-weight the co-occurrence counts. Obviously, PMI is naturally regarded as a favorable choice [7], as shown in the PPMI matrix. Some other methods benefiting from the transformations of the co-occurrence counts matrix were subsequently proposed, such as the square root type transformation in the form of Hellinger PCA (HPCA) [24] and the COALS method, which transformed the co-occurrence matrix by a correlation-based normalization [25].

B. NEURAL-NETWORK-INSPIRED MODELS

The neural-network-inspired models (also referred to as prediction models) are designed to predict the corresponding contexts in a fixed-sized window for a given word. The word representations can be compressed into a semantically low-dimensional space and be denoted as dense vectors, i.e., word embeddings. The nature of word embeddings originates from the weights learned by language modeling. For example, Bengio *et al*. [10] proposed a language model based on a feedforward neural network (NNLM), where a linear projection layer and a non-linear hidden layer were combined to learn the word representations. As an extension of previous research in [10], Collobert and Weston [11] defined a unified architecture for multitask learning, including part-of-speech tags, named entity tags, semantic roles, etc. However, the full neural network used in the NNLM model suffers from expensive computation cost on a larger corpus.

Nowadays, the newly proposed models formulate the learning process of word representations as an optimization problem, such as the SGNS and Continuous Bag of Words (CBOW) models involved in the word2vec package.[1] The idea underlying these two models relied on that the log probability of a word appearing in the context of a given word was proportional to the inner product between their word vectors. Besides, the GloVe model was intrinsically based on global log-bilinear regression, taking advantages of both the global matrix factorization and the prediction

[1] https://radimrehurek.com/gensim/models/word2vec.html

model. According to the analysis in [26], the objective of the GloVe model was to explicitly factorize the word-word PMI matrix, while SGNS model was a special case of GloVe model, which implicitly factorized the same matrix. The word vectors learned by the SGNS and GloVe models are claimed as distributed representations by [12], which can capture both word similarity and word analogy properties. Recently, the neural-network-inspired models have developed in various ways. For example, some authors focused on multi-prototype representations by generating multiple vectors for each word, aiming to solve the ambiguity existing in polysemy [27], [28]. Other researches made efficient use of existing structured lexical resources or knowledge graphs, e.g., WordNet and Freebase, in order to improve the performance of word embeddings [29]–[31]. In terms of exploiting the external linguistic constraints, current specialized methods can be mainly classified into two groups: (a) joint models integrate the constraints into the learning objectives of original distributional models [43], [45]; (b) post-processing models retroactively adjust the pre-trained word vectors to satisfy the external constraints [31], [44], [46]. For example, the ELM-based model in [47] makes an efficient use of a count-based approach described in the GloVe model for text classification, where the main steps involve building a word-context matrix and then applying matrix factorization. Shi *et al.* [48] introduced a Skip-gram Topical word Embedding (STE) model to jointly learn both words and topics, thereby considering the correlations between multiple senses of different words that occur in different topics.

According to the above description, the essence of these models relies on efficiently leveraging the statistical information extracted from the occurrences of word-context pairs. However, not all context words can be considered to make contributions for constructing word representations, since the appearance of them may be occasional or useless, which will be discussed later.

III. EFFICIENT WORD REPRESENTATION MODELS WITH SALIENT FEATURES

A. MOTIVATIONS

1) MOTIVATIONS FROM THE NEURAL-NETWORK-INSPIRED MODELS

Based on the hypothesis proposed by [32], the assembling of all contexts in which a given word does and does not occur provides a set of mutual constraints that largely determines the similarity of word meaning. Therefore, the previous models, especially the neural-network-inspired models, are preferable to use all occurrences of the word-context pairs. There existed certain preconceived stereotypes, as claimed in [33], that neural-network-inspired word embeddings had an overwhelming superiority over the count-based distributional methods on variously semantic relatedness tasks. However, considering the inner commonality existing between distributional and neural-network models, Levy *et al*. claimed that the advantages of word embeddings were primarily due to the contribution of fine-tuned hyper-parameters and sophisticatedly designed system. With the same modifications transferred to the distributional models, only local or insignificant performance difference can be observed with no global superiority biased to any single approach [34]. Surprisingly, several modifications presented in [34], including subsampling, deleting rare words, shifted PMI and context distribution smoothing (cds), had something in common, i.e., explicitly or implicitly reducing useless features. More details and analysis are presented as follows:

a) Explicit reduction of features

Subsampling: It aims to randomly remove very frequent words, like stop-words, according to a specific probability distribution.

Deleting Rare Words: The goal is to ignore the rare words in the corpus, which is performed before building word-context pairs.

b) Implicit reduction of features

Scaled Context Window: Intuitively, contexts that are closer to the target are more important, thus, both SGNS and GloVe adopt a weighting scheme that can scale the context words according to their distance from the target word, so that the effect of distant words can be weakened. For the PPMI matrix, the PMI values of these distant contexts may be relatively trivial or even negative and be ignored in some extreme cases. Hence, it can be viewed as a method of reducing features in disguise.

Shifted PMI: The number of negative samples (termed as k), is a significant hyper-parameter in SGNS, which is essential for implicitly optimizing each word-context pair: $PMI\,(w, c) - \log k$, where w and c denotes the word and the context, respectively. Levy and Goldberg applied this shift value to the distributional methods through the PPMI matrix [26]. Apparently, for some pairs having low PMI values, the effect of them may be neglected after the subtraction of $\log k$.

Context Distribution Smoothing: Aiming to smooth the distribution of contexts, all context frequencies are raised to the power of α to enlarge the probability of sampling a rare context, which conversely alleviates the bias caused by PMI values towards rare words. Thus, the distribution of features is further modified by putting less focus on rare words.

Therefore, the reduction of features plays an important role in the performance gain, so it is necessary to deeply analyze the influence of the number of selected features for word representation.

2) MOTIVATIONS FROM THE COUNT-BASED DISTRIBUTIONAL MODELS

Two most well-known count-based distributional models are the explicit PPMI matrix and its factorization using SVD technique (PPMI-SVD) in Latent Semantic Analysis (LSA) [42]. According to discussions in [9] and [34], these two models rely heavily on the positive PMI values. In other

words, the performance could deteriorate substantially on semantic similarity tasks if the contexts with negative PMI values are preserved in the matrix. From this view, a natural question is how many distinct contexts with positive PMI values a specific word has.

Herein, a matrix composed of PMI values is constructed,[2] intending to calculate the percentage of the distinct contexts with positive PMI values for each word in the vocabulary. Experimental results are shown in Fig. 1, where ws = 2, ws = 5 and ws = 10 denote that the context window sizes (ws) are set as 2, 5 and 10 and the mean values are marked as stars. Obviously, for a given word, the distinct contexts with positive PMI values take a large portion, with percentages varying in [96%, 99%] interval. It indicates that only a tiny part of contexts with negative PMI values are neglected. Considering the problem from another perspective, the scope of reducing more redundant contexts should not be limited to negative contexts, instead, more attention should be paid on discovering and removing those ineffective contexts with positive PMI values.

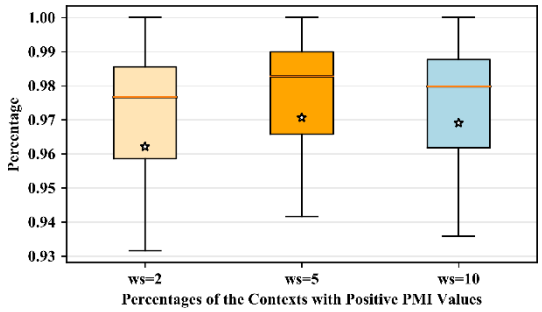

FIGURE 1. Percentages of the unique contexts with positive PMI values with different window sizes on the Wikipedia corpus.

B. THE PROPOSED SELECTION OF SALIENT FEATURES

As discussed above, reducing insignificant features for the distributional models can lead to substantial performance improvements. Two factors should be analyzed for the selection of features. The first factor is the measure of semantic relatedness between the feature and a target word, and the second one is the way to determine the number of salient features retained for word representation.

1) METHOD OF WORD RELATEDNESS: PMI WITH SCALED CONTEXT WINDOW (PMI-SCALED)

On one hand, PMI, as the first and most frequently used method, is employed here to evaluate the word relatedness, where positive values indicate that the relationships between two words are stronger than expected under an independence assumption and negative values indicate that the two words tend to appear independently. Specifically, for a given word-context pair (w, c), the PMI is defined as the log ratio between their joint probability and the product of their marginal probabilities, which can be calculated by:

$$\text{PMI (w, c)} = \log\frac{P(w,c)}{P(w)\cdot P(c)} = log\frac{f(w,c)\cdot|D|}{f(w)\cdot f(c)} \quad (1)$$

where P(.) represents the probability; f(.) represents the co-occurrence counts; D denotes the corpus and $|D|$ means the number of all tokens contained in the corpus. In terms of those word-context pairs that are never observed in the corpus, i.e., $f\ (w, c) = 0$, PMI (w, c) is set as 0 for convenience.

It can be seen that the PMI method suffers from a high bias towards the rare contexts, due to the fact that the probability of a rare context, i.e., $\tilde{P}(c)$, could be extremely low in Eq. (1), and it results in a small denominator. Generally, the rare contexts can be divided into two types: (a) co-occurring with the center word as specific collations, such as the company name ''Apple Marine'', where ''Marine'' is an unrelated context for the word ''Apple''; (b) accidentally appearing within the context window. For example, in the sentence ''The girl is reading an herbalist book'', ''herbalist'' may be a context word for the center word ''girl'', even though the relationship between them could be negligible.

It is worth noting that the contexts used as specific collations only account for a small portion of the surroundings of a given word, and most of the rare contexts are likely to be accidental occurrences, the negative impact of which should be weakened. Fortunately, it can be observed that since most of the rare contexts make no contribution to the semantic/syntactic meaning of a particular word, they tend to co-occur distantly or arbitrarily within the window of a center word. For instance, ''herbalist'' is three tokens away from ''girl'' rather than being the closest neighbor. In this case, the scaled context window approach based on a harmonic function presented in the GloVe model is implemented here to decrease the influence of the rare words. Specifically speaking, a context word that are k tokens away from the target word will be considered as 1/k of a co-occurrence.

2) METHOD OF DETERMINING THE NUMBER OF SALIENT FEATURES

According to the semantic relatedness methods described above, the original number of features for a given word is directly the number of unique words occurring in the contexts. In an extreme case, the maximum number of features for a target word should be $\min(2{*}ws * f\ (w)\ , |V|)$ if every word in the contexts is different from each other, where ws and $|V|$ denote the window size and the size of the vocabulary V, respectively. Therefore, the ratio between the actual number of features and its potentially maximum number of features can be used as an indicator for the rareness of a word. The calculation is shown in (2), where $f_{pf}(w)$ means the number of distinct features having positive PMI values for a given word w.

$$\varphi = \frac{f_{pf}\ (w)}{2 * ws * f\ (w)} \quad (2)$$

[2]The corpus is an English Wikipedia dump (December 2017) downloaded in https://dumps.wikimedia.org/, which contains 1.82 billion tokens. After preprocessing (tokenization, lowercase, etc.), 206,000 most frequent words that occur more than 100 times in the corpus are selected into the vocabulary.

Particularly, it can be assumed that the target word is common and frequent in the corpus when the ratio is small, or perhaps it is a rare word if the ratio is large. This assumption can be proved in Fig. 2, where the ratios of all words in the vocabulary along with the descendently sorted log ranks of frequency are plotted. In Fig. 2, it is obvious that the most frequent words showing in the initial phase are extremely low, whereas the relatively rare words showing in the latter part have high ratios.

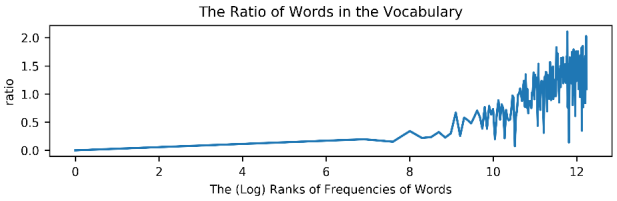

FIGURE 2. The ratio between the number of unique features and its occurrence for a given word.

From the above analysis, a question needs to be answered, i.e., how to determine the number of salient features for each word. Since the frequent words have low ratios, they always occur in similar contexts. For example, ''good'', as a common adjective, is always followed by multiple kinds of nouns, referring to living creatures, objects, places, actions, and ideas. The commonality of these nouns (i.e., contexts) lies in that they can indicate the syntactic information of ''good'', however, they have limited effect on reflecting the other aspects of word's meaning. It can be assumed that such ordinary contexts share a similar role when representing a word, thus, most of them could be removed from the list of salient features, while only those with highly scaled PMI values could be retained. On the contrary, words with high ratios are prone to appear in various contexts. The diversity and rareness of features imply the high uniqueness of words' characteristics; hence, more features should be maintained as salient features. Based on the above analysis, the number of salient features γ is determined as in Eq. (3), where α is a real value. Eq. (3) can be further simplified as Eq. (4) with $\delta = \frac{\alpha}{2*ws}$.

$$\gamma = \alpha \cdot \log(|V|)^2 \cdot \frac{f_{pf}(w)}{2 * ws * f(w)} \tag{3}$$

$$\gamma = \delta \cdot \log(|V|)^2 \cdot \frac{f_{pf}(w)}{f(w)} \tag{4}$$

3) ANALYSIS OF THE SaFe METHOD

In order to analyze the properties of the SaFe method, the number of selected features have been plotted along with the descendently sorted log ranks of the occurrence counts of words in the vocabulary. For clarity, the vocabulary is further equally partitioned into two parts, as shown in Fig. 3. (a) and (b), respectively, where ''original'' means the original number of features for a target words; and δ is represented as ''delta'' ranging from 1 to 20 with step set as 2.

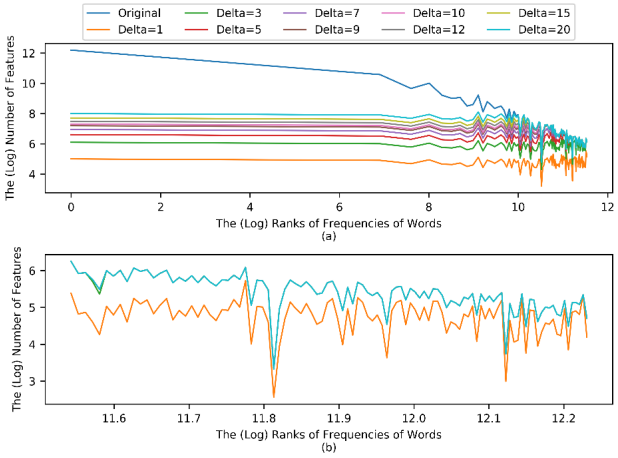

FIGURE 3. The number of salient features for words in the vocabulary.

With an eye to words with high frequency, as shown in Fig. 3. (a), it can be observed that the number of salient features can be considerably regulated by δ, while the number of salient features for words with low frequency can be hardly affected when $\delta > 1$, as seen in Fig. 3. (b). In short, the proposed SaFe method aims to reduce the redundant features of frequent words, while it is inclined to preserve the unique features of rare words.

C. THE APPLICATIONS OF THE SaFe METHOD

As mentioned above, two distributional models are proposed in this paper, i.e. the SaFe-PPMI model and the SaFe-SVD model. The SaFe-PPMI model uses the SaFe method to help the PPMI matrix reduce the redundant features, and the SaFe-SVD model employs the SVD technique to get low-dimensional representations for words.

1) WORD REPRESENTATION BY THE SaFe-PPMI MODEL

The SaFe-PPMI model is to construct a high-dimensional sparse matrix $M \in R^{|V|*|\tilde{V}|}$, where $|\tilde{V}|$ is the size of the context vocabulary in a corpus. The i_{th} row of M represents the i_{th} word w_i in the vocabulary V and the j_{th} column denotes the j_{th} potential context c_j in $\tilde{V}$. Each entry of the matrix M_{ij} represents the word relatedness between w_i and c_j calculated by the PMI-scaled method described above. Note that only positive PMI values are preserved in the PPMI matrix, indicating that the negative values and unobserved word-context pairs in the corpus are all replaced by 0.

Then, the features for each word will be reduced by the SaFe method. For each word, all contexts should be sorted descendingly according to their PMI values, and only the γ top features based on (4) will be selected as salient features for word representation.

2) WORD REPRESENTATION BY THE SaFe-SVD MODEL

SVD, as a common dimensionality reduction method, tries to find the *rank-d* decomposition regarding the L_2 loss [19]. In particular, the sparse matrix can be factorized into the

Algorithm 1 The Pseudocode of the SaFe-PPMI and the SaFe-SVD Models

1: Choose an appropriate corpus D;
2: Initialize the parameters, i.e., window size (ws), dimension (d), vocabulary size ($|V|$) and δ etc.;
3: Build the vocabulary V of the given corpus D;
4: Construct a co-occurrence matrix M $\in R^{|V|*|V|}$ of the corpus D based on a scaled context window;
5: **for** i = 1 :$|V|$ **do**
  5.1 Calculate the scaled PMI values for each nonzero context of word w_i in the i_{th} row;
  5.2 Set the scale PMI values to 0 if they are negative;
  5.3 Calculate the number of salient features φ for word w_i based on Eq. (4);
  5.4 Sort the contexts of word w_i in descending order according to their scaled PMI values;
  5.5 Select the top φ contexts as salient features of word w_i and set the other contexts to 0;
  5.6 Represent word w_i using the relative i_{th} row in the matrix M with respect to the SaFe-PPMI model.
6: Use the SVD technique to factorize the matrix M into the product of three matrices $U \cdot \Sigma \cdot V^T$;
7: Remain top d elements of Σ, and transform the sparse matrix M into $M_d = U_d \cdot \Sigma_d \cdot V_d^T$;
8: Represent the word embeddings as $W = U_d + V_d$ in terms of the SaFe-SVD model.

product of three matrices $U \cdot \Sigma \cdot V^T$, where Σ is a diagonal matrix of eigenvalues in decreasing order and U and V are orthonormal. With only the top d elements of Σ preserved, the sparse matrix will be transformed to $M_d = U_d \cdot \Sigma_d \cdot V_d^T$. Notice that the dot-products of two random rows of the matrix $W = U_d \cdot \Sigma_d$ are equal to the dot-product of the same rows of M_d, in other words, the high-dimensional rows of M_d can be replaced by the low-dimensional rows of $W = U_d \cdot \Sigma_d$ for word representation, which can dramatically save the computation.

In this paper, the idea of the SaFe-SVD model is directly applying the SVD technique to the SaFe-PPMI model. In most related papers [34], [42], the d-dimensional matrix $W = U_d \cdot \Sigma_d$ is used as a common approach to represent words. However, inspired by word representation employed in the GloVe model, the word vectors are presented by $W = U_d$. The detailed steps of the SaFe-PPMI and the SaFe-SVD models are described in **Algorithm1**.

IV. EXPERIMENTS AND ANALYSIS

A. TEST SUITES

1) TRAINING CORPUS

English Wikipedia (December 2017 dump) corpus is used in all experiments, which contains about 39.5 million sentences and 1.82 billion raw words. The preprocessing steps include removing non-alpha elements and sentence splitting, and word tokenization. There are two sets of vocabularies. One is a small vocabulary (termed as small_vocab) containing the words that occur more than100 times, resulting in 207,960 unique words and 1.79 billion tokens in the corpus, and the other vocabulary is a large one (termed as large_vocab), which contains words that occur more than 50 times, leaving 325,579 unique words and 1.80 billion tokens in the corpus. In addition, for the comparisons between the proposed models with other state-of-the-art models, two other corpora are utilized, i.e., Amazon Fine Food Reviews[3] (Amazon-Fine-Food) and Amazon Product Data[4] (Amazon-Product). The former one contains only 568,454 reviews, and the latter one has nearly 82.83 million reviews. In order to get a large corpus, the Amazon-Product and Wikipedia corpora are combined together for comparison purposes, termed as Product-Wikipedia. After the same preprocessing as with the Wikipedia dump and the removal of words appearing less than 100 times in the corpora, the Amazon-Product has a vocabulary of 308,181 unique words with 6.9 billion tokens retained, and the vocabulary of the Product-Wikipedia includes 439,041 unique words with 8.7 billion tokens left in the corpus. Since the Amazon-Fine-Food is a small corpus, only those words appearing less than 50 times are removed, leaving 15,856 unique words and 44 million tokens.

2) TRAINING SETTINGS

ws, referring to window size, is set to 2, 5 and 10, and the dimensions for the SaFe-SVD, SGNS and GloVe models are set as 300 and 600. The number of epochs for the SGNS model is 5, while the number of iterations for the GloVe model is set as 50 and 100, when the corresponding dimensions are 300 and 600. With pre-extracted word-context pairs supplied, the GloVe and SGNS models are trained with pre-built modules called genism[5] and glove-python,[6] respectively. For the sake of clear expression, the percentage of salient features selected for word representation is listed in Table 1, where "num_features" denotes the total number of salient features selected for all words. The percentages span from 9.52% to 97.57%, with δ increasing from 1 to 400. Note that each element in the PPMI matrix is computed by using the PMI-scaled method.

3) EVALUATION DATASETS

Experiments are conducted on seven semantic relatedness datasets, as ashown in Table 2. Each dataset contains a

[3]https://snap.stanford.edu/data/web-FineFoods.html

[4]http://jmcauley.ucsd.edu/data/amazon/

[5]https://radimrehurek.com/gensim/models/word2vec.html

[6]https://nlp.stanford.edu/projects/glove/

TABLE 1. The percentages of salient features.

δ	1	2	3	4	5	6	7	8	9	10
num _Features	2.52e+07	4.65e+07	6.08e+07	7.17e+07	8.05e+07	8.79e+07	9.45e+07	1e+08	1.05e+08	1.1e+08
percentage	9.52%	17.56%	22.99%	27.08%	30.41%	33.24%	35.70%	37.88%	39.85%	41.65%
δ	11	12	13	14	15	16	17	18	19	20
num _Features	1.15e+08	1.19e+08	1.22e+08	1.26e+08	1.29e+08	1.32e+08	1.35e+08	1.38e+08	1.41e+08	1.43e+08
percentage	43.30%	44.83%	46.25%	47.59%	48.85%	50.04%	51.16%	52.22%	53.24%	54.21%
δ	40	60	80	100	120	140	160	180	200	220
num _Features	1.79e+08	1.99e+08	2.13e+08	2.23e+08	2.3e+08	2.36e+08	2.39e+08	2.43e+08	2.46e+08	2.49e+08
percentage	67.57%	75.31%	80.48%	84.18%	86.94%	89.05%	90.71%	92.03%	93.09%	93.96%
δ	240	260	280	300	320	340	360	380	400	
num _Features	2.5e+08	2.52e+08	2.53e+08	2.54e+08	2.55e+08	2.56e+08	2.56e+08	2.58e+08	2.58e+08	
percentage	94.66%	95.24%	95.74%	96.17%	96.53%	96.85%	97.12%	97.37%	97.59%	

TABLE 2. Evaluation datasets used in the comparisons.

Name	Number of Pairs	References
MEN-LEM	3000	[35]
WS-353	353	[36]
WSS-353	203	[37]
WSR-353	252	[38]
RG-65	65	[39]
RW	2034	[40]
SimLex-999	999	[41]

number of word pairs marked with human-annotated relatedness scores. For example, the relatedness scores of word pairs listed in the SimLex-999 dataset range from 0 to 10; and the unrelated pair (''dirty'', ''narrow'') is annotated as a low value, i.e., 0.3, while the semantically similar pair (''vanish'', ''disappear'') is marked as a high value, i.e., 9.8. In this case, the way to evaluate the performance of the proposed models is to compare their measurements of relatedness on word pairs with a gold standard, where the score for each word pair is measured by the cosine similarity between the normalized representations of words. Then, the general quality of a particular model can be obtained by computing the Spearman's rank correlation coefficient between the calculated relatedness scores and the human judgments.

B. EXPERIMENTAL ANALYSIS OF THE SaFe-PPMI MODEL

Various hyper-parameter settings, including δ, ws, and vocabulary size, will be discussed, aiming at fully analyzing the properties of the SaFe-PPMI model. For all experiments conducted on the SaFe-PPMI model, the range of δ is in [1] and [20] with step set as 1, and the number of salient features accounts for 9.52% to 54.21% of all the features contained in the PPMI matrix. The window size is set as 2 and the vocabulary used is small_vocab. Experimental results are presented in Tables 3-5, and it is observed that the SaFe method has substantial impact on the performance of the PPMI model. Without complicated hyper-parameter turning, the SaFe-PPMI model is capable of obtaining remarkable advantages over the PPMI model.

1) MODEL ANALYSIS: THE NUMBER OF SALIENT FEATURES

The experimental results are listed in Table 3 with the last column (i.e., PPMI) showing the results of the PPMI model. Besides, the convergence curves with increasing δ are plotted in Fig. 4.

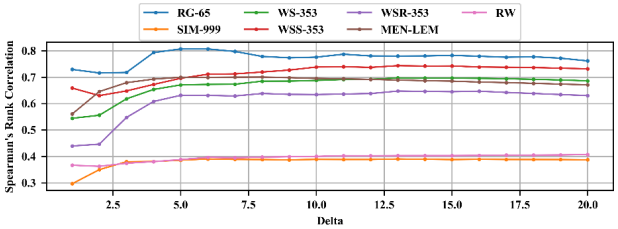

FIGURE 4. Convergence curves of the SaFe-PPMI model based on various δ across different tasks.

From Table 3, it is clear that the SaFe-PPMI model consistently outperforms the PPMI model on all semantic relatedness tasks by a minimum margin of 5.2 percent, especially for the improvements on the WS-353 and WSR-353 datasets, where the margins have reached up to 12.3 and 12.6 percent, respectively. The RG-65 and MEN-LEM datasets can achieve the best performance when $\delta = 5$, and $\delta = 6$ is the best setting for the SimLex-999 datasets. Three related datasets, i.e. WS-353, WSS-353 and WSR-353, have the best

TABLE 3. Performance comparisons across different tasks of the SaFe-PPMI model with the small_vocab (best results in bold with underline).

δ	1	2	3	4	5	6	7	8	9	10	
RG-65	0.729	0.716	0.718	0.793	***0.807***	0.807	0.797	0.779	0.773	0.776	
SimLex-999	0.296	0.350	0.380	0.380	0.386	***0.390***	0.388	0.388	0.387	0.389	
WS-353	0.544	0.556	0.618	0.653	0.671	0.673	0.673	0.685	0.685	0.688	
WSS-353	0.659	0.630	0.648	0.672	0.696	0.711	0.712	0.719	0.727	0.738	
WSR-353	0.439	0.446	0.547	0.608	0.631	0.631	0.629	0.638	0.635	0.634	
MEN-LEM	0.561	0.645	0.679	0.693	***0.700***	0.699	0.700	0.700	0.698	0.695	
RW	0.366	0.362	0.374	0.380	0.388	0.396	0.395	0.397	0.399	0.400	
δ	11	12	13	14	15	16	17	18	19	20	PPMI
RG-65	0.787	0.780	0.780	0.780	0.783	0.779	0.776	0.777	0.772	0.762	*0.747*
SimLex-999	0.388	0.388	0.390	0.389	0.388	0.389	0.388	0.388	0.388	0.387	*0.308*
WS-353	0.691	0.692	***0.697***	0.696	0.696	0.696	0.694	0.692	0.689	0.686	*0.574*
WSS-353	0.740	0.737	***0.744***	0.742	0.742	0.739	0.738	0.736	0.734	0.732	*0.663*
WSR-353	0.636	0.638	***0.647***	0.646	0.645	0.647	0.642	0.638	0.634	0.630	*0.521*
MEN-LEM	0.694	0.691	0.689	0.687	0.685	0.682	0.679	0.677	0.674	0.671	*0.648*
RW	0.402	0.402	0.403	0.403	0.403	0.404	0.405	0.405	0.406	***0.407***	*0.333*

TABLE 4. Statistical results across different tasks of the SaFe-PPMI model with the small_vocab (best results in bold with underline).

ws	Methods	RG-65	SimLex-999	WS-353	WSS-353	WSR-353	MEN-LEM	RW
2	SaFe-Worst	0.716	0.296	0.544	0.630	0.439	0.561	0.362
	SaFe-Best	0.807	***0.390***	0.697	***0.744***	0.647	0.700	***0.407***
	SaFe-Mean	0.774	0.381	0.669	0.715	0.612	0.680	0.395
	PPMI	0.747	0.308	0.574	0.663	0.521	0.648	0.333
5	SaFe-Worst	0.717	0.318	0.574	0.670	0.474	0.625	0.363
	SaFe-Best	***0.838***	0.359	***0.705***	0.742	***0.658***	***0.702***	0.376
	SaFe-Mean	0.785	0.352	0.684	0.727	0.629	0.688	0.368
	PPMI	0.654	0.206	0.472	0.543	0.449	0.591	0.243
10	SaFe-Worst	0.684	0.318	0.581	0.675	0.457	0.626	0.348
	SaFe-Best	0.810	0.339	0.693	0.731	0.641	0.681	0.357
	SaFe-Mean	0.749	0.334	0.674	0.711	0.613	0.667	0.352
	PPMI	0.573	0.154	0.403	0.459	0.402	0.542	0.195

TABLE 5. Statistical results across different tasks of the SaFe-PPMI model with large_vocab (best results in bold with underline).

ws	Methods	RG-65	SimLex-999	WS-353	WSS-353	WSR-353	MEN-LEM	RW
2	SaFe-Worst	0.702	0.278	0.476	0.631	0.324	0.494	0.307
	SaFe-Best	0.824	***0.397***	0.693	***0.753***	0.640	0.690	***0.380***
	SaFe-Mean	0.780	0.376	0.660	0.729	0.591	0.661	0.362
	PPMI	0.757	0.325	0.596	0.685	0.537	0.659	0.334
5	SaFe-Worst	0.735	0.314	0.505	0.659	0.374	0.569	0.332
	SaFe-Best	***0.848***	0.367	***0.696***	0.752	***0.643***	***0.696***	0.352
	SaFe-Mean	0.797	0.352	0.674	0.735	0.609	0.675	0.345
	PPMI	0.689	0.232	0.508	0.581	0.477	0.617	0.264
10	SaFe-Worst	0.686	0.326	0.545	0.680	0.413	0.597	0.332
	SaFe-Best	0.838	0.348	0.682	0.737	0.624	0.677	0.335
	SaFe-Mean	0.755	0.336	0.667	0.721	0.598	0.657	0.334
	PPMI	0.614	0.181	0.447	0.509	0.433	0.575	0.223

performance when δ is 13. Although no specific δ value can completely dominate the results, the curves in Fig. 4 show a common tendency for a rising δ on different datasets. Except for the minor difference in the beginning, the performance of the SaFe-PPMI model improves dramatically until δ steadily increases to a specific value, and then the performance will

slightly deteriorate when δ becomes larger. In other words, the efficiency of the SaFe-PPMI model is highly influenced by the setting of δ. Particularly, the performance cannot be satisfying if δ is too small, indicating that less features could lose crucial information hidden in the corpus. On the other hand, excessive features may be useless or even harmful for improvements. In conclusion, the performance of SaFe-PPMI matrix can dramatically benefit from the SaFe method.

2) MODEL ANALYSIS: WINDOW SIZE

This subsection aims to investigate the effect of window size (termed as ws) on the SaFe-PPMI model. In order to save space and elaborate clearly, the results of different δ values for a given ws are presented in statistical forms, including the worst value (SaFe-Worst), the mean value (SaFe-Mean) and the best value (SaFe-Best), as shown in Table 4. With regard to the best values on different datasets, the SimLex-999, WSS353 and RW datasets can obtain the best performance when ws $= 2$, and the other datasets perform the best when ws $= 5$. Note that the best results on the RG-65, the SimLex-999 and the RW-353 datasets depend on the ws to a large extent, while there are no distinct gaps among the comparisons on other datasets in regard to the setting of ws.

To further provide contrasts regarding the number of salient features, the δ values of the best results for each ws are plotted in Fig. 5. Apparently, the δ values for ws $= 2$ are the highest with the exception of that on the SimLex-999 dataset, inferring that more salient features are required, when compared with other window sizes. Moreover, ws $= 10$ has the lowest δ values, with less salient features needed for the best results, however, its performance is the worst among all. The low δ value needed by a large window size implies that there may be numerous superfluous and repetitive contexts extracted from the corpus, which can only partially interpret the meaning of a word. Considering that the best results obtained by ws $= 5$ outperforms those of ws $= 2$ and ws $= 10$, and the corresponding δ values on different datasets are moderate, ws $= 5$ can be considered as a suitable choice for the SaFe-PPMI model.

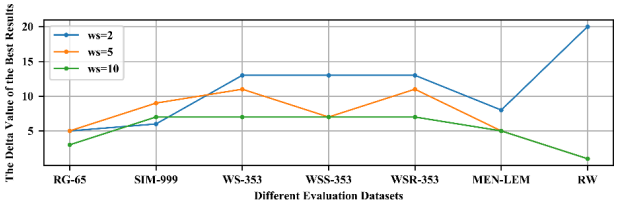

FIGURE 5. δ values of the best results.

3) MODEL ANALYSIS: VOCABULARY SIZE

To further investigate the efficiency of the SaFe-PPMI model, this group of experiments are performed on the large_vocab. Similar to Table 4, the results in Table 5 show that the SaFe-PPMI model with ws $= 2$ reach the best performance on the SimLex-999, WSS-353, and RW datasets, and the other datasets have the best performance when ws $= 5$. Thus, ws $= 2$ and 5 can achieve constant superiority over different vocabularies. On the other hand, there is a consensus in previous research that the rare words should be removed from the vocabulary because of their negative impact on the performance of the models. There are two ways to discard the rare words; one is to retain a set of most frequent words in the vocabulary [15], while the other way is to filter those words that appear less than fixed times (e.g. 100) in the corpus [34]. As defined above, the large_vocab contains a large number of infrequent words with frequency ranging from 50 to 100, therefore, it is more likely to be interfered with the rare words, comparing with the small_vocab. However, on the RG-65, SimLex-999 and WSS-353 datasets, the best results obtained by the large_vocab on the SaFe-PPMI model are better than those of the small_vocab. It implies that the SaFe-PPMI model has advantages on ignoring the interference caused by the rare words, reflecting that the SaFe method has better generalization ability of extracting valuable information.

Moreover, the curves of the worst, the mean and the best results on six datasets with various ws settings across different vocabularies have been presented in Fig. 6, where S2, S5, S10 denotes the small_vocab with ws $= 2, 5, 10$; and L2, L5, L10 denotes the large_vocab with ws $= 2, 5, 10$. It is obvious that the SaFe-PPMI model outperforms the PPMI model by a large gap. In addition to the superiority of the best results achieved by the SaFe-PPMI model, even its mean results can overwhelmingly surpass the results of the PPMI model across all datasets.

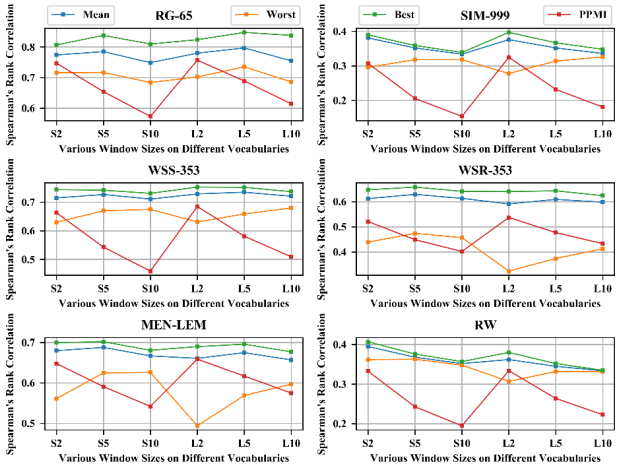

FIGURE 6. The best performance with various window sizes across different vocabularies.

C. EXPERIMENTAL ANALYSIS OF THE SaFe-SVD MODEL

The following experiments are conducted to investigate the performance of the SaFe-SVD model, where three elements will be probed, including the δ, ws and the dimensionality. Considering the comparable performance between different vocabularies, only the small_vocab will be analyzed for reference. Due to the observation in [26], SVD has been validated

TABLE 6. Performance comparisons across different tasks for the SaFe-SVD model with ws set as 2 (best results in bold with underline).

δ	20	40	60	80	100	120	140	160	180	200	220
RG-65	0.744	0.750	0.769	0.776	0.766	0.771	0.774	0.778	0.776	0.777	0.777
SimLex-999	0.389	0.399	0.404	0.406	0.407	0.408	0.409	0.410	0.410	0.411	0.411
WS-353	0.704	0.705	0.711	0.713	0.715	0.717	0.718	0.718	0.718	***0.719***	0.719
WSS-353	0.755	0.752	0.751	0.756	0.758	0.760	0.761	0.762	0.762	0.763	0.764
WSR-353	0.642	0.636	0.643	0.646	0.648	0.649	0.652	0.652	0.652	***0.654***	0.654
MEN-LEM	0.737	0.742	0.744	0.746	0.747	0.748	0.749	0.750	0.750	0.751	0.751
RW	0.439	0.443	0.447	0.449	0.450	0.451	0.452	0.453	0.454	0.454	0.454
δ	240	260	280	300	320	340	360	380	400	SVD	SaFe-PPMI
RG-65	0.782	0.780	***0.783***	0.782	0.781	0.781	0.780	0.782	0.782	*0.781*	0.807
SimLex-999	0.411	0.411	0.411	0.411	0.411	0.411	0.411	0.411	0.411	***0.412***	0.390
WS-353	0.719	0.719	0.719	0.718	0.718	0.718	0.718	0.718	0.718	*0.718*	0.697
WSS-353	0.764	0.765	***0.766***	0.766	0.766	0.765	0.765	0.765	0.765	*0.764*	0.744
WSR-353	0.654	0.654	0.654	0.654	0.654	0.654	0.654	0.653	0.654	***0.654***	0.647
MEN-LEM	0.751	0.751	0.751	0.751	0.751	0.751	0.751	***0.752***	0.752	***0.752***	0.700
RW	0.454	0.454	***0.455***	0.455	0.455	0.455	0.455	0.455	0.455	***0.455***	0.407

to be an efficient choice for discovering semantic relatedness, but less suitable for exploring analogies when compared to neural-network-inspired models, thus, no consideration will be given to the evaluation of word analogy.

1) MODEL ANALYSIS: THE NUMBER OF SALIENT FEATURES

Here, this group of experiments tends to explore the effect of the number of salient features on the performance of the SaFe-SVD model. Due to the fact that the a small δ represents that excessive features will be filtered for each word and thus results in a spare SaFe-PPMI matrix with most of the entries being zeros, which is harmful for the generation ability of the SVD technique used in the SaFe-SVD model. Under this situation, the values chosen for δ are supposed to be bigger than that used in the SaFe-PPMI model, ranging from 20 to 400 here, with step stride set as 20. The percentage of salient features is from 54.21% to 97.59%. Besides, ws is set to 2 and dimension is 300.

Table 6 reveals that the SaFe-SVD model has competitive performance when compared to the SVD model. Precisely, the SaFe-SVD model outperforms the SVD model on the RG-65, WS-353 and WSS-353 datasets, and it has the same performance on another three datasets, i.e., the WSR-353, MEN-LEM and RW. The SVD model can only surpass the SaFe-SVD model on the SimLex-999 dataset by a negligible superiority. As seen from Table 6, two datasets, namely the WS-353 and the WSR-353, can achieve the best performance when δ reaches up to 200, and the SaFe-SVD model perform the best on the RG-65, WSS-353 and RW datasets, when $\delta = 280$. The MEN-LEM dataset exhibits the best performance when δ is 380. Although the large δ values for the best results means that more salient features are required, the better or at least competitive performance obtained by the SaFe-SVD model demonstrates that there is a small number of features in the corpus, which can have a negative effect on the performance and can be removed by the SaFe method.

Moreover, the best results of the SaFe-PPMI model across different datasets are also listed in the final column (''SaFe-PPMI'') in Table 6. As declared above, the SaFe-PPMI model uses less features than the SaFe-SVD model, with δ merely ranging from 1 to 20. However, it performs worse than the SaFe-SVD model on 6 out of 7 datasets, and only has the best performance on the RG-65 dataset. In general, the SaFe-SVD model is better for applications on different NLP tasks.

The convergence curves of the SaFe-SVD model based on increasing δ values across different evaluation datasets are shown in Fig. 7. Similar to the SaFe-PPMI model, the performance of the SaFe-SVD model also has an upward trend when δ is rising. However, the degree of improvements is not considerable, and the difference between the worst and the best results is trivial, with the mean margin being merely 1.85 percent. It can be inferred from the results that the SaFe method should be advocated for those models for which less computation is preferable.

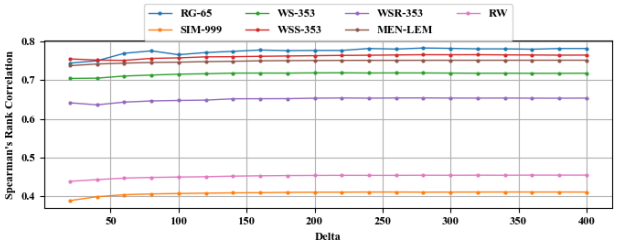

FIGURE 7. Convergence curves across different tasks of the SaFe-PPMI model based on various δ.

2) MODEL ANALYSIS: WINDOW SIZE

With the purpose of examining the impact of window size, ws is set to 5 and 10 for the SaFe-SVD model, and the corresponding results are listed in Tables 7 and 8, respectively. The dimension is set as 300.

TABLE 7. Performance comparisons across different tasks for the SaFe-SVD model with ws set as 5 (best results in bold with underline).

δ	20	40	60	80	100	120	140	160	180	200	
RG-65	0.775	***0.782***	0.775	0.776	0.779	0.777	0.771	0.773	0.775	0.775	
SimLex-999	0.374	0.384	0.388	0.391	0.394	0.395	0.396	0.397	0.398	0.398	
WS-353	0.732	0.736	0.736	0.739	0.741	0.742	0.743	0.744	0.744	***0.745***	
WSS-353	0.769	0.765	0.768	0.772	0.775	0.775	***0.776***	0.774	0.773	0.774	
WSR-353	0.676	0.680	0.678	0.682	0.685	0.687	0.689	0.690	0.691	***0.692***	
MEN-LEM	0.750	0.754	0.755	0.757	0.758	0.759	0.760	0.761	0.761	0.762	
RW	0.451	0.452	0.455	0.454	0.456	0.456	0.458	0.458	0.459	0.460	
δ	220	240	260	280	300	320	340	360	380	400	SVD
RG-65	0.771	0.773	0.773	0.772	0.771	0.772	0.771	0.772	0.772	0.772	*0.773*
SimLex-999	0.398	***0.399***	0.399	0.399	0.399	0.399	0.399	0.399	0.399	0.399	*0.399*
WS-353	0.745	0.744	0.744	0.744	0.744	0.745	0.745	0.745	0.745	0.745	*0.744*
WSS-353	0.774	0.774	0.774	0.775	0.775	0.776	0.776	0.776	0.776	0.776	*0.776*
WSR-353	0.691	0.691	0.690	0.691	0.691	0.691	0.692	0.692	0.691	0.691	*0.690*
MEN-LEM	0.762	0.762	0.762	***0.763***	0.763	0.763	0.763	0.763	0.763	0.763	*0.763*
RW	0.460	***0.461***	0.461	0.461	0.461	0.461	0.461	0.461	0.461	0.461	*0.461*

TABLE 8. Performance comparisons across different tasks for the SaFe-SVD model with ws set as 10 (best results in bold with underline).

δ	20	40	60	80	100	120	140	160	180	200	
RG-65	0.768	0.770	0.769	0.775	0.779	0.779	0.784	0.782	0.783	0.783	
SimLex-999	0.362	0.370	0.374	0.377	0.379	0.380	0.380	0.381	0.381	0.382	
WS-353	0.727	0.731	0.737	0.740	0.743	0.746	0.746	0.748	0.748	0.748	
WSS-353	0.762	0.762	0.764	0.767	0.768	0.774	0.775	0.774	0.774	0.774	
WSR-353	0.673	0.673	0.680	0.688	0.693	0.696	0.695	***0.697***	0.695	0.695	
MEN-LEM	0.755	0.758	0.760	0.762	0.763	0.764	0.765	0.765	0.766	0.766	
RW	0.452	0.455	0.455	0.457	0.460	0.461	0.460	0.461	0.462	0.463	
δ	220	240	260	280	300	320	340	360	380	400	SVD
RG-65	0.780	0.781	0.783	0.784	0.784	0.784	0.785	0.785	0.781	***0.786***	*0.784*
SimLex-999	0.382	0.382	0.382	0.382	0.382	***0.383***	0.383	0.383	0.383	0.383	*0.383*
WS-353	0.748	***0.749***	0.748	0.747	0.747	0.748	0.748	0.748	0.748	0.748	*0.748*
WSS-353	0.774	0.775	0.775	0.774	0.774	0.775	0.775	0.775	0.775	***0.776***	*0.774*
WSR-353	0.695	0.696	0.695	0.694	0.695	0.695	0.697	0.696	0.697	0.697	*0.694*
MEN-LEM	0.766	0.766	0.766	0.766	0.766	0.766	***0.767***	0.767	0.767	0.767	*0.767*
RW	0.463	***0.464***	0.464	0.464	0.464	0.464	0.464	0.464	0.464	0.464	*0.464*

As shown in Table 7, four datasets, i.e. the SimLex-999, WS-353, WSR-353 and RW, can obtain the best performance with δ set to 200 and 240, each of which performs the best on two datasets. The RG-65, WSS-353 and MEN-LEM datasets exhibit the best performance when δ is set to 40, 140 and 280, respectively. When compared with the SVD model, the SaFe-SVD is better on the RG-65, WS-353 and WSR-353 datasets, and there is no difference between their performance on the other datasets. For the results of ws $= 10$ shown in Table 8, both of $\delta = 240$ and 400 can obtain the best performance on two datasets. The former performs the best on the WS-353 and RW datasets, and the latter achieves the best on the RG-65 and WSS-353 datasets. The other datasets show the best performance on different δ values, containing 160, 320 and 340. The SaFe-SVD model with ws $= 10$ can surpass the SVD model on four datasets, i.e. the RG-65, WS-353, WSS-353 and WSR-353; besides, the same performance can be found on the remaining datasets. Overall, the SaFe-SVD model has superiority over the SVD model for both ws $= 5$ and 10.

In order to compare thoroughly, the best results and their corresponding δ values for the SaFe-SVD model with

TABLE 9. Performance comparisons across different tasks for the SaFe-SVD model with dimension set as 600 (best results in bold with underline).

δ	20	40	60	80	100	120	140	160	180	200	
RG-65	0.781	0.801	0.799	0.810	0.810	0.807	0.810	0.809	0.808	0.812	
SimLex-999	0.415	0.427	0.432	0.436	0.439	0.441	0.441	0.441	0.441	0.442	
WS-353	0.722	0.736	0.742	0.743	0.746	0.746	0.748	0.748	0.749	0.749	
WSS-353	0.776	0.786	0.788	0.789	0.789	0.786	0.788	0.787	0.787	0.788	
WSR-353	0.652	0.667	0.671	0.674	0.679	0.680	0.682	0.683	0.682	0.682	
MEN-LEM	0.758	0.763	0.765	0.767	0.768	0.769	0.770	0.771	0.772	0.772	
RW	0.485	0.489	0.490	0.492	0.492	0.493	0.494	0.495	0.495	0.495	
δ	220	240	260	280	300	320	340	360	380	400	SVD
RG-65	0.810	0.808	0.808	0.810	0.813	0.812	0.813	***0.814***	0.814	0.814	*0.810*
SimLex-999	0.443	0.443	0.444	0.444	0.444	***0.445***	0.444	0.444	0.444	0.444	*0.444*
WS-353	0.748	0.748	0.749	0.749	***0.750***	0.750	0.750	0.750	0.750	0.750	*0.749*
WSS-353	0.787	0.787	0.788	0.787	0.789	0.788	***0.790***	0.789	0.789	0.789	*0.788*
WSR-353	0.680	0.681	0.681	0.682	***0.685***	0.685	0.685	0.684	0.684	0.684	*0.683*
MEN-LEM	0.772	0.772	0.772	***0.773***	0.773	0.773	0.773	0.773	0.773	0.773	***0.773***
RW	0.495	0.495	0.495	0.495	0.495	0.495	0.495	0.495	***0.496***	0.495	*0.495*

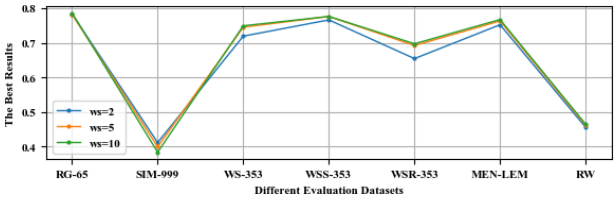

FIGURE 8. The best results of the SaFe-SVD model with various window sizes.

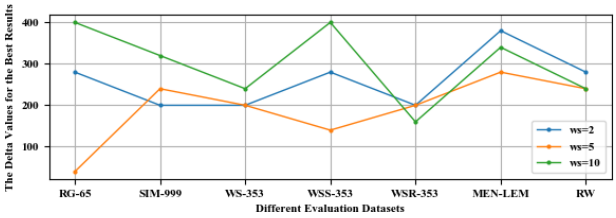

FIGURE 9. The δ values of the best results of the SaFe-SVD model with various window sizes.

ws = 2, 5, 10 are showed in Fig. 8 and Fig. 9, respectively. As inferred from Fig. 9, the SaFe-SVD model with ws = 2 performs the worst, while ws = 10 shows the best performance and have a slight advantage over ws = 5. In Fig. 9, the SaFe-SVD model with ws = 10 requires the most salient features on 4 out of 7 datasets, and ws = 5 needs the least salient features on 5 datasets. Considering the comparable performance of ws = 2 and 5, ws = 5 is recommended when computation cost is a burden.

3) MODEL ANALYSIS: DIMENSIONALITY

This subsection is aimed at discussing the effect of dimensionality on the SaFe-SVD model, where the dimension here is set to 600. Due to the well-performed results listed in Table 5, ws is set to 5.

Through the comparisons presented in Table 9, the SaFe-SVD model can perform better than the SVD model in all cases, except for the results related to the MEN-LEM dataset on which the same performance is given. This indicates that the SaFe-SVD model can consistently have superior performance regardless of the dimensionality. More comparisons, i.e., the difference between different dimensions, are shown in Fig. 10, where dim = 300 and dim = 600 represent that the dimensions are set to 300 and 600, respectively. The curves show that dim = 600 can outperform dim = 300 on six datasets by a significant margin.

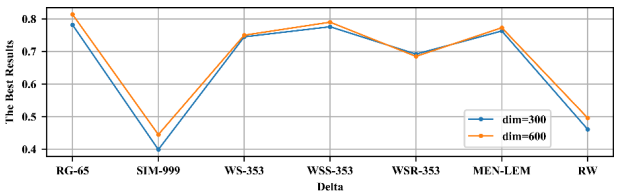

FIGURE 10. The best results of the SaFe-SVD model with different dimensions.

D. PERFORMANCE COMPARISONS OF THE SaFe-SVD MODEL WITH STATE-OF-THE-ART MODELS

1) COMPARISONS OF THE PERFORMANCE ON DIFFERENT CORPORA

The performance of the SaFe-SVD, SGNS and GloVe models across seven datasets using four corpora, including the Amazon-Fine-Food, Wikipedia, Amazon-Product and Product-Wikipedia, are listed in Table 10, where SaFe-SVD-best, SaFe-SVD-mean and SaFe-SVD-worst denote the best, average and worst results obtained by the SaFe-SVD model. The SaFe-SVD model can keep continuous superiorities over other models on three datasets, i.e. the SimLex-999, WS-353 and MEN-LEM datasets. With an eye to each corpus,

TABLE 10. Performance comparisons among the SaFe-SVD, SGNS and GloVe models on different corpora.

Corpora	Models	RG-65	SimLex-999	WS-353	WSS-353	WSR-353	MEN-LEM	RW
Amazon-Fine-Food	SaFe-SVD-max	***0.731***	***0.325***	***0.480***	***0.584***	***0.388***	***0.579***	***0.544***
	SaFe-SVD-mean	0.721	0.319	0.475	0.578	0.384	0.578	0.537
	SaFe-SVD-min	0.673	0.318	0.473	0.574	0.379	0.573	0.532
	SGNS	0.640	0.250	0.414	0.457	0.306	0.554	0.476
	GloVe	-0.041	0.022	0.160	0.191	0.096	0.190	0.237
Wikipedia	SaFe-SVD-max	0.782	***0.399***	***0.745***	0.776	***0.692***	***0.763***	0.461
	SaFe-SVD-mean	0.774	0.395	0.742	0.774	0.688	0.760	0.459
	SaFe-SVD-min	0.771	0.374	0.732	0.765	0.676	0.750	0.451
	SGNS	***0.806***	0.367	0.707	***0.780***	0.622	0.763	***0.470***
	GloVe	0.481	0.235	0.504	0.561	0.482	0.531	0.080
Amazon-Product	SaFe-SVD-max	0.758	***0.365***	***0.709***	***0.755***	***0.640***	***0.780***	***0.526***
	SaFe-SVD-mean	0.746	0.340	0.699	0.751	0.625	0.769	0.516
	SaFe-SVD-min	0.707	0.264	0.660	0.736	0.564	0.724	0.456
	SGNS	***0.759***	0.318	0.631	0.707	0.534	0.754	0.488
	GloVe	0.458	0.252	0.463	0.517	0.471	0.603	0.119
Product-Wikipedia	SaFe-SVD-max	0.770	***0.332***	***0.716***	***0.750***	0.797	***0.652***	***0.777***
	SaFe-SVD-mean	0.757	0.304	0.694	0.740	0.782	0.623	0.761
	SaFe-SVD-min	0.698	0.216	0.599	0.684	0.720	0.492	0.692
	SGNS	***0.781***	0.303	0.626	0.707	***0.811***	0.531	0.752
	GloVe	0.657	0.250	0.538	0.622	0.482	0.628	0.128

TABLE 11. Performance Comparisons among the SaFe-SVD, SGNS and GloVe models with dimension set as 300.

Datasets		RG-65	SimLex-999	WS-353	WSS-353	WSR-353	MEN-LEM	RW
ws = 2	SaFe-SVD	0.783	***0.411***	0.719	0.766	0.654	0.752	0.455
	SGNS	***0.806***	0.406	0.712	***0.790***	0.627	0.763	***0.488***
	GloVe	0.402	0.147	0.370	0.422	0.367	0.456	0.029
ws = 5	SaFe-SVD	0.782	0.399	0.745	0.776	0.692	0.763	0.461
	SGNS	***0.806***	0.367	0.707	0.780	0.622	0.763	0.470
	GloVe	0.481	0.235	0.504	0.561	0.482	0.531	0.080
ws = 10	SaFe-SVD	0.786	0.383	***0.749***	0.776	***0.697***	***0.767***	0.464
	SGNS	0.794	0.340	0.700	0.772	0.622	0.760	0.457
	GloVe	0.521	0.236	0.521	0.592	0.500	0.574	0.102

the SaFe-SVD model can perform best on at least four datasets. In particular, the SaFe-SVD model shows good performance on the Amazon-Fine-Food and Amazon-Product corpora, performing best on seven and six datasets, respectively. Since such Amazon-related corpora collect numerous online reviews that contain intricately affective states and subjective information, such as personal opinions, attitudes and emotions, they show biases towards sentiment characteristics, which is different from the news pages included in the Wikipedia. Thus, comparing with other two models, the SaFe-SVD model is much more capable of discovering the sentiment information involved in the short and concise reviews.

2) COMPARISONS OF THE PERFORMANCE ON VARIOUS WINDOW SIZES

This set of experiments are designed to validate the effectiveness of the SaFe-SVD model when compared with the SGNS and the GloVe models. The comparisons for dim = 300 and 600 are presented in Tables 11 and 12, respectively. Note that the performance of the SaFe-SVD model is only related to the best results achieved.

From Table 11 it can be noted that the GloVe model performs the worst in all cases. Based on the analysis above, the SaFe-SVD model performs the best when ws = 10, obtaining the best performance on the WS-353, WSR-353 and MEN-LEM datasets. On the contrary, the SGNS model shows a downward trend when ws is increasing, and it has the best performance on the RG-65, WSS-353 and RW datasets when ws = 2. By the way, SimLex-999 dataset achieves the best performance on the SaFe-SVD model when ws = 2. In consideration of the various settings for the best results, it seems that there is no apparent commonality among the comparisons.

However, the above analysis indicates that the SaFe-SVD model can perform better when dim = 600. According to the

TABLE 12. Performance Comparisons among the SaFe-SVD, SGNS and GloVe models with dimension set as 600.

Datasets		RG-65	SimLex-999	WS-353	WSS-353	WSR-353	MEN-LEM	RW
ws = 2	SaFe-SVD	0.809	***0.459***	0.738	0.789	0.664	0.759	0.483
	SGNS	***0.829***	0.434	0.733	***0.795***	0.657	0.771	0.495
	GloVe	0.412	0.251	0.509	0.572	0.494	0.557	0.090
ws = 5	SaFe-SVD	0.814	0.445	***0.750***	0.790	0.685	0.773	***0.496***
	SGNS	0.813	0.389	0.722	0.789	0.646	0.769	0.484
	GloVe	0.593	0.290	0.572	0.630	0.543	0.647	0.127
ws = 10	SaFe-SVD	0.825	0.432	0.745	0.787	***0.687***	***0.774***	0.494
	SGNS	0.811	0.360	0.709	0.775	0.640	0.764	0.465
	GloVe	0.647	0.285	0.606	0.660	0.578	0.653	0.144

TABLE 13. Run-time of different models on various corpora.

Models	Amazon-Fine-Food	Amazon-Product	Wikipedia	Product-Wikipedia
SaFe-PPMI	4m	45m	38m	1h 39m
SaFe-SVD	10m	1h 11m	40m	1h 41m
SGNS	4m	9h 38m	6h 8m	12h 28m
GloVe	17m	20h 52m	11h 28m	34h 48m

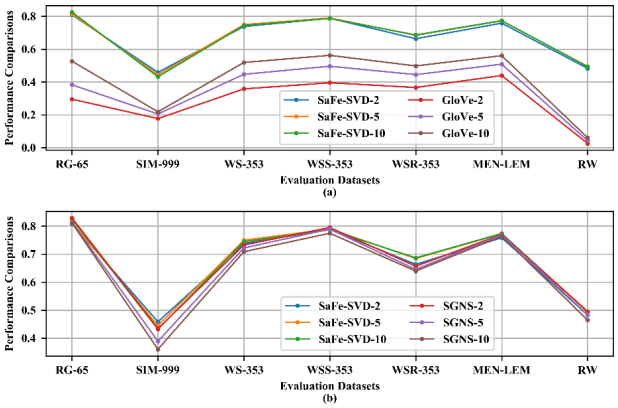

FIGURE 11. The performance curves for the SaFe-SVD, SGNS and GloVe models with dimension set as 600.

results listed in Table 12 the SaFe-SVD model can perform the best on five datasets, ignoring the various window sizes. The SGNS model can obtain the best performance on the RG-65 and WSS-353 datasets when ws = 2. Besides, the GloVe model has a decreasing trend when ws is rising, and it continually performs the worst among all comparisons. For full explanation, the results have been plotted in Fig. 11, where the top half (i.e. Fig. 11. (a)) shows the comparisons with the GloVe model and the below half (i.e. Fig. 11. (b)) presents the comparisons with the SGNS model. In addition, SaFe-SVD-2 in Fig. 11 denotes that ws is set to 2 for the SaFe-SVD model, and this naming rule is used for other legends. Clearly, gaps between the curves of the SaFe-SVD model and the GloVe model are large. Moreover, the general performance of the SaFe-SVD model is better than the SGNS model, especially on the SimLex-999, WS-353 and WSR-353 datasets.

3) RUN-TIME COMPARISONS

The run-time of the SaFe-PPMI model depends on processing the co-occurrence matrix based on the SaFe method, which can be easily dealt with by parallel computing across multiple machines, and the SaFe-SVD model needs to add the training time consumed by the SVD technique. The run-time of different models performed on diverse corpora is listed in Table 13, corresponding to the experiments in Section IV.D.1, where ws = 5 and dim = 300. Note that the SaFe-PPMI and the SaFe-SVD models are performed on a single thread of an Intel Broadwell CPU (128GB RAM), while the GloVe and the SGNS models are conducted with ten threads. Since the run-time of the proposed models vary due to the impact of δ, the maximum times (i.e., $\delta = 200$) are given in Table 13 where "h" and "m" represent "hours" and "minutes", respectively. There is no doubt that the application of the SaFe method does not cost much time, and the Safe-PPMI and SaFe_SVD models can save plenty of time with less demands for equipment, compared to other models.

E. EXPLANATION FOR THE UNSATISFYING PERFORMANCE OF THE GloVe MODEL

Unfortunately, the GloVe model exhibits low performance in most cases, especially on the RG-65, SimLex-999 and RW datasets, which contradicts the results presented in the original paper [15] . However, the hyper-parameters of the GloVe model in this paper are set as the same with the original paper, except that stop-words, such as "this", "is", "her" and "it" etc., are not removed from the corpora used here. Since this paper mainly discusses the necessity of reducing worthless features, thus, for authenticity and fairness, only those contexts with extremely low frequency are cleared and the stop-words are not removed for all experiments.

TABLE 14. The performance of the GloVe model on the amazon food dataset with dimension set as 300.

		RG-65	SimLex-999	WS-353	WSS-353	WSR-353	MEN-LEM	RW
ws=2	W	-0.068	0.007	0.102	0.146	0.054	***0.169***	0.146
	WO	***0.518***	***0.098***	***0.208***	***0.316***	***0.315***	0.084	***0.375***
ws=5	W	-0.041	0.022	0.160	0.191	0.096	***0.190***	0.237
	WO	***0.614***	***0.098***	***0.256***	***0.357***	***0.476***	0.132	***0.420***
ws=10	W	0.205	-0.026	0.078	0.122	0.022	***0.212***	0.138
	WO	***0.520***	***0.082***	***0.257***	***0.370***	***0.273***	0.156	***0.409***

TABLE 15. The performance of the GloVe model on the amazon food dataset with dimension set as 600.

		RG-65	SimLex-999	WS-353	WSS-353	WSR-353	MEN-LEM	RW
ws=2	W	-0.298	0.012	0.164	0.266	***0.077***	***0.202***	0.166
	WO	***0.265***	***0.117***	***0.214***	***0.295***	0.070	0.106	***0.398***
ws=5	W	0.109	0.027	0.103	0.150	0.076	***0.241***	0.143
	WO	***0.506***	***0.078***	***0.297***	***0.397***	***0.308***	0.185	***0.410***
ws=10	W	-0.060	-0.004	0.176	0.252	0.066	***0.254***	0.196
	WO	***0.614***	***0.094***	***0.291***	***0.407***	***0.273***	0.165	***0.416***

In order to validate that the performance of the GloVe model is influenced by the retention of the stop-words, an experiment is performed on the Amazon-Fine-Food corpus with the stop-words retained or removed. The results are given in Tables 14 and 15 with the dimension set as 300 and 600, respectively, where "W" and "WO" denote that the stop-words are included or not. Noticeably, without the stop-words, the performance of the GloVe model can have remarkable improvements on almost all datasets, except for the MEN-LEM dataset. It demonstrates that the good performance of the GloVe model is obtained by the fine-tuning of the hyper-parameters and well-selected contexts, implying that capturing salient features could be an essential step before training the word embedding.

V. CONCLUSION

In this paper, a novel method called SaFe is presented to select salient features for a target word, which is built on the notion that the meaning of a word can be characterized by significant contexts. Two models have been proposed to represent words, i.e. the SaFe-PPMI model and the SaFe-SVD model, where the former model represents a word as a row in the matrix and the latter one represents words as dense vectors in a low-dimensional space. In terms of semantic relatedness tasks, the SaFe-SVD model outperforms the SaFe-PPMI model, therefore, it is more suitable for many NLP applications. Furthermore, the proposed SaFe method can be further applied to other neural-network-inspired models, such as the GloVe model, which can help eliminate the negative impact of useless contexts. The experiments reveal that the SaFe method can significantly improve the performance of the PPMI model on semantic relatedness tasks, and the SaFe-SVD model can consistently outperform the state-of-the-art models, such as the SGNS and GloVe models.

REFERENCES

[1] A. Ballatore, M. Bertolotto, and D. C. Wilson, "An evaluative baseline for geo-semantic relatedness and similarity," *GeoInformatica*, vol. 18, no. 4, pp. 747–767, Oct. 2014.

[2] E. Gabrilovich and S. Markovitch, "Computing semantic relatedness using Wikipedia-based explicit semantic analysis," in *Proc. 20th Int. Joint Conf. Artif. Intell.*, Hyderabad, India, 2007, pp. 1606–1611.

[3] P. Resnik, "Using information content to evaluate semantic similarity in a taxonomy," in *Proc. 14th Int. Joint Conf. Artif. Intell.*, Montreal, QC, Canada, 1995, pp. 448–453.

[4] S. Patwardhan, S. Banerjee, and T. Pedersen, "Using measures of semantic relatedness for word sense disambiguation," in *Proc. 4th Int. Conf. Comput. Linguistics Intell. Text Process.*, Mexico City, Mexico, 2003, pp. 241–257.

[5] P. Pantel and D. Lin, "Discovering word senses from text," in *Proc. 8th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Edmonton, AB, Canada, 2002, pp. 613–619.

[6] J. R. Firth, *A Synopsis of Linguistic Theory*. Oxford, U.K.: Oxford Univ. Press, Oxford, 1957, pp. 11–13.

[7] T. K. Landauer and S. T. Dutnais, "A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge," *Psychol. Rev.*, vol. 104, no. 2, pp. 211–240, Apr. 1997.

[8] K. Lund and C. Burgess, "Producing high-dimensional semantic spaces from lexical co-occurrence," *Behav. Res. Methods, Instrum., Comput.*, vol. 28, no. 2, pp. 203–208, Jun. 1996.

[9] J. A. Bullinaria and J. P. Levy, "Extracting semantic representations from word co-occurrence statistics: A computational study," *Behav. Res. Methods*, vol. 39, no. 3, pp. 510–526, Aug. 2007.

[10] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, Feb. 2003.

[11] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proc. 25th Int. Conf. Mach. Learn. (ICML)*, Helsinki, Finland, 2008, pp. 160–167.

[12] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Scottsdale, Arizona, 2013, pp. 1301–3781.

[13] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. 26th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, Lake Tahoe, NV, USA, 2013, pp. 3111–3119.

[14] T. Mikolov, W. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, Atlanta, GA, USA, 2013, pp. 746–751.

[15] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Doha, Qatar, 2014, pp. 1532–1543.
[16] A. Islam and D. Inkpen, "Second order co-occurrence PMI for determining the semantic similarity of words," in *Proc. Int. Conf. Lang. Resour. Eval.*, Genoa, Italy, 2006, pp. 1033–1038.
[17] S. Hassan and R. Mihalcea, "Semantic relatedness using salient semantic analysis," in *Proc. 25th AAAI Conf. Artif. Intell.*, San Francisco, CA, USA, 2011, pp. 884–889.
[18] P. D. Turney and P. Pantel, "From frequency to meaning: Vector space models of semantics," *J. Artif. Intell. Res.*, vol. 37, pp. 141–188, Feb. 2010.
[19] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, no. 3, pp. 211–218, Sep. 1936.
[20] P. Jaccard, "The distribution of the flora in the alpine zone," *New Phytologist*, vol. 11, no. 2, pp. 37–50, Feb. 1912.
[21] L. R. Dice, "Measures of the amount of ecologic association between species," *Ecology*, vol. 26, no. 3, pp. 297–302, Jul. 1945.
[22] S. Janson and J. Vegelius, "Measures of ecological association," *Oecologia*, vol. 49, no. 3, pp. 371–376, Jul. 1981.
[23] K. W. Church and P. Hanks, "Word association norms, mutual information, and lexicography," in *Proc. 27th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, Vancouver, BC, Canada, 1989, pp. 76–83.
[24] R. Lebret and R. Collobert, "Word embeddings through Hellinger PCA," in *Proc. Eur. Chapter Assoc. Comput. Linguistics (EACL)*, Gothenburg, Sweden, 2014, pp. 482–490.
[25] D. L. T. Rohde, L. M. Gonnerman, and D. C. Plaut, "An improved model of semantic similarity based on lexical co-occurrence," *Commun. ACM*, vol. 8, no. 116, pp. 627–633, Nov. 2006.
[26] O. Levy and Y. Goldberg, "Neural word embeddings as implicit matrix factorization," in *Proc. Neural Inf. Process. Syst.*, Montreal, QC, Canada, 2014, pp. 2177–2185.
[27] Y. Liu, Z. Liu, T.-S. Chua, and M. Sun, "Topical word embeddings," in *Proc. 29th AAAI Conf. Artif. Intell., 27th Innov. Appl. Artif. Intell. Conf.*, Austin, TX, USA, 2015, pp. 2418–2424.
[28] A. Neelakantan, J. Shankar, A. Passos, and A. McCallum, "Efficient non-parametric estimation of multiple embeddings per word in vector space," in *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, Doha, Qatar, 2015, pp. 1059–1069.
[29] M. Yu and M. Dredze, "Improving lexical embeddings with semantic knowledge," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics*, Baltimore, MD, USA, vol. 2, 2014, pp. 545–550.
[30] C. Xu *et al.*, "RC-NET: A general framework for incorporating knowledge into word representations," in *Proc. 23rd ACM Int. Conf. Inf. Knowl. Manage. (ACM)*, Shanghai, China, 2014, pp. 1219–1228.
[31] M. Faruqui, J. Dodge, S. K. Jauhar, C. Dyer, E. Hovy, and N. A. Smith, "Retrofitting word vectors to semantic lexicons," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol. (NAACL-HLT)*, Denver, CO, USA, 2015, pp. 1606–1615.
[32] T. K. Landauer, P. W. Foltz, and D. Laham, "An introduction to latent semantic analysis," *Discourse Process.*, vol. 25, nos. 2–3, pp. 259–284, 1998.
[33] M. Baroni, G. Dinu, and G. Kruszewski, "Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics*, Baltimore, MD, USA, vol. 1, 2014, pp. 238–247.
[34] O. Levy, Y. Goldberg, and I. Dagan, "Improving distributional similarity with lessons learned from word embeddings," *Trans. Assoc. Comput. Linguistics*, vol. 3, pp. 211–225, May 2015.
[35] E. Bruni *et al.*, "Distributional semantics in technicolor," in *Proc. 50th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, Jeju Island, South Korea, 2012, pp. 136–145.
[36] L. Finkelstein *et al.*, "Placing search in context: The concept revisited," *ACM Trans. Inf. Syst.*, vol. 20, no. 1, pp. 116–131, 2002.
[37] T. Zesch, C. Müller, and I. Gurevych, "Using wiktionary for computing semantic relatedness," in *Proc. 23rd Nat. Conf. Artif. Intell.*, Chicago, IL, USA, 2008, pp. 861–866.
[38] E. Agirre, E. Alfonseca, K. Hall, J. Kravalova, M. Pasca, and A. Soroa, "A study on similarity and relatedness using distributional and wordnet-based approaches," in *Proc. Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2009, pp. 19–27.
[39] H. Rubenstein and J. B. Goodenough, "Contextual correlates of synonymy," *Commun. ACM*, vol. 8, no. 10, pp. 627–633, 1965.
[40] M. Luong, R. Socher, and C. D. Manning, "Better word representations with recursive neural networks for morphology," in *Proc. 7th Conf. Comput. Natural Lang. Learn.*, Sofia, Bulgaria, 2013, pp. 104–113.
[41] F. Hill, R. Reichart, and A. Korhonen, "SimLex-999: Evaluating semantic models with (genuine) similarity estimation," *Comput. Linguistics*, vol. 41, no. 4, pp. 665–695, 2014.
[42] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *J. Amer. Soc. Inf. Sci.*, vol. 41, pp. 391–407, Sep. 1990.
[43] D. Kiela, F. Hill, and S. Clark, "Specializing word embeddings for similarity or relatedness," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Lisbon, Portugal, Sep. 2015, pp. 2044–2048.
[44] D. Tang, F. Wei, B. Qin, N. Yang, T. Liu, and M. Zhou, "Sentiment embeddings with applications to sentiment analysis," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 2, pp. 496–509, Feb. 2016.
[45] K. A. Nguyen, S. S. Walde, and N. T. Vu, "Integrating distributional lexical contrast into word embeddings for antonymsynonym distinction," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, Berlin, Germany, vol. 2, Aug. 2016, pp. 454–459.
[46] N. Mrkšić, D. Ó Séaghdha, T.-H. Wen, B. Thomson, and S. Young, "Neural belief tracker: Data-driven dialogue state tracking," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, Vancouver, BC, Canada, vol. 1, Jul. 2017, pp. 1777–1788.
[47] P. Lauren, G. Qu, J. Yang, P. Watta, G.-B. Huang, and A. Lendasse, "Generating word embeddings from an extreme learning machine for sentiment analysis and sequence labeling tasks," *Cogn. Comput.*, vol. 10, no. 4, pp. 625–638, 2018.
[48] B. Shi, W. Lam, S. Jameel, S. Schockaert, and K. P. Lai, "Jointly learning word embeddings and latent topics," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Tokyo, Japan, Aug. 2017, pp. 375–384.

MING ZHANG received the B.S. degree in electronic information engineering from Jiangnan University, Wuxi, China, in 2012, and the M.S. degree in control science and engineering from Jiangnan University, where she is currently pursuing the Ph.D. degree in control science and engineering.

She was a Visiting Ph.D. Student at Coventry University, U.K., for two years. She is currently working on novel methods for word representation using machine learning and statistics. Her research interests include evolutionary computing and sentiment analysis.

VASILE PALADE was a Lecturer with the Department of Computer Science, University of Oxford, Oxford, U.K., for many years. He joined the School of Computing, Coventry University, Coventry, U.K., in 2013. He has authored more than 130 papers in journals and conference proceedings and books on machine learning/computational intelligence and applications. His research interests include machine learning/computational intelligence, neural networks, deep learning, neuro-fuzzy systems, and various nature-inspired algorithms. He has delivered keynote talks and has chaired international conferences on machine learning and applications. He is a member of the IEEE Computational Intelligence Society and an Associate Editor for several reputed journals.

YAN WANG received the Ph.D. degree in control theory and control engineering from the Nanjing University of Science and Technology, Nanjing, China, in 2006.

From 2013 to 2014, she was a Visiting Researcher with the School of Electrical and Computer Engineering, Louisiana State University, Baton Rouge, LA, USA. She is currently a Professor of control science with Jiangnan University and the Head of the Provincial Excellent Team of Innovative Research. She has authored more than 80 articles. She holds 11 invention patents and one industry standard. Her research interests include intelligent manufacturing using perception and collaboration technologies.

Dr. Wang was a recipient of the Yangtse River Scholar of the Ministry of Education of China and the Provincial Science Fund for Distinguished Young Scholars, Jiangsu, China.

ZHICHENG JI received the Ph.D. degree in power electronics and drives from the China University of Mining and Technology, China, in 2004.

From 2003 to 2004, he was a Visiting Researcher with the University of Toronto, Canada. He is currently a Professor of control science with Jiangnan University and the Deputy Director of the Information Department, 7th Science Technology Committee, Ministry of Education of China. He has authored one book and more than 200 articles. He holds more than 10 invention patents. His research interests include intelligent manufacturing, new energy, and control techniques for the Internet of Things.

Dr. Ji's awards and honors include, twice, the First Class Award in Research Achievements (Science and Technology), Ministry of Education, China, in 2011 and 2016, and, three times, the First Class Award in Teaching Achievement, Jiangsu, China, in 2007, 2009, and 2013.