JIS

Research Paper

Journal of Information Science
1–16
© The Author(s) 2018
Reprints and permissions:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/0165551518786676
journals.sagepub.com/home/jis
SAGE

Detecting new Chinese words from massive domain texts with word embedding

Yu Qian
School of Management and Economics, University of Electronic Science and Technology of China, P.R. China

Yang Du
School of Management and Economics, University of Electronic Science and Technology of China, P.R. China

Xiongwen Deng
School of Management and Economics, University of Electronic Science and Technology of China, P.R. China

Baojun Ma
Research Center for Big Data Management & Intelligent Decision, School of Economics and Management, Beijing University of Posts and Telecommunications, P.R. China

Qiongwei Ye
School of Business, Yunnan University of Finance and Economics, P.R. China; School of Economics and Management, Tsinghua University, P.R. China

Hua Yuan
School of Management and Economics, University of Electronic Science and Technology of China, P.R. China

Abstract

Textual information retrieval (TIR) is based on the relationship between word units. Traditional word segmentation techniques attempt to discern the word units accurately from texts; however, they are unable to appropriately and efficiently identify all new words. Identification of new words, especially in languages such as Chinese, remains a challenge. In recent years, word embedding methods have used numerical word vectors to retain the semantic and correlated information between words in a corpus. In this article, we propose the word-embedding-based method (WEBM), a novel method that combines word embedding and frequent *n*-gram string mining for discovering new words from domain corpora. First, we mapped all word units in a domain corpus to a high-dimension word vector space. Second, we used a frequent *n*-gram word string mining method to identify a set of candidates for new words. We designed a pruning strategy based on the word vectors to quantify the possibility of a word string being a new word, thereby allowing the evaluation of candidates based on the similarity of word units in the same string. In a comparative study, our experimental results revealed that WEBM had a great advantage in detecting new words from massive Chinese corpora.

Keywords

Natural language processing; new word detection; similarity measurement; textual information retrieval; word embedding

Corresponding authors:
Baojun Ma, Research Center for Big Data Management & Intelligent Decision, School of Economics and Management, Beijing University of Posts and Telecommunications, Beijing 100876, P.R. China.
Email: mabaojun@bupt.edu.cn

Qiongwei Ye, School of Business, Yunnan University of Finance and Economics, Kunming, 650221, P.R. China.
Email: yqw@ynufe.edu.cn

1. Introduction

Textual information retrieval (TIR) is one of the most common forms of information retrieval. TIR is based on the concept that a document is formed from word units [1,2] or atomic words [3], and these units can be considered as good content descriptors. Obtaining the correct word units from massive amounts of text is essential in the field of TIR. In general, word segmentation is the classic approach used to accomplish this type of task [4,5], which involves the tokenisation of a string (portion) of written language into component units at the initial stage of natural language processing (NLP) [6].

However, Sproat and Emerson [7] reported that emerging new words have a great impact on both the performance and accuracy of word segmentation tools because, in the literature of NLP, most word segmentation methods are implemented through a word dictionary and the corresponding matching/searching algorithm [8]. One common problem regarding these approaches is that the dictionary must be updated continuously in order to maintain its ability to detect new words; however, updating a dictionary consumes both time and resources and may not address the challenge of recognising new words that spring up unpredictably. Therefore, even the best word segmentation software cannot obtain all of the words accurately from a textual corpus. In addition, regardless of the negative impact on the performance of the word segmentation software, new words in a corpus may have practical applications for analysing the sentiment [9], recommendations [10], detecting trends [11,12] and analysing ancient literature [13,14]. Notably, new word detection is essential for TIR.

In general, two concepts are used to describe the results of new word detection, which include the new generated words and the unlisted words. In lexicology, the new generated words refer to new vocabulary that accompanies changes in the social environment. Unlisted words refer to all words that have not been included in a collection, for example, the dictionary used in a word segmentation tool [15]. Chinese text (especially sentences) reveal no clear separation between words unlike English text [16,17]; therefore, the word segmentation tools are apt to divide the Chinese text into a set of more finely grained units [18], even to the level of characters. For example, the word phrase ‘火车站’ (‘train station’) would be separated into two parts: ‘火车’ (‘train’) and ‘站’ (‘station’). Obviously, such result may have an impact on the task of TIR. In this work, ‘new words’ refers primarily to the words that are domain-specific/time-sensitive terms, most of which have not yet been correctly recognised by the segmentation algorithms. Accordingly, the purpose of this article is to present an efficient way to identify the potential new words that have been inappropriately segmented by the software and to retrieve them from the over fine-grained word elements. Hereafter, we call these potential elements of the new word, ‘word units’. Furthermore, we use ‘word string’ to consider the order of their position in a sentence.

In the literature, two types of supervised and unsupervised methods are used to detect new words. The supervised methods are mainly based on prior knowledge or on prelabelled data provided manually by the domain experts; however, this work is expensive. In contrast, the unsupervised methods for detecting new words are based on the distribution of word units in the corpus, such as word correlations. Most unsupervised methods proposed in the literature suffer from poor stability and lack of reusability. In this work, we noticed that (1) most new words are composed of two (or more) word units and (2) if a word is inappropriately segmented, then the word units of the same word may be expected to have a high frequency of co-occurrence and a high proximity in the position. The basic idea of this work is to detect new words from a corpus by considering the advantages of the co-occurrence frequency and position proximity of paired word units. In linguistics, word embedding is discussed in relation to distributional semantics. The underlying idea of word embedding is that ‘a word is characterized by the company it keeps’ [19]. This theory has been regarded by NLP as a guide for quantifying the semantic similarities between linguistic items, based on their occurrence (i.e. position, correlation and distribution) in a large corpus. Once a word embedding model has been trained, all the word units in a corpus can be mapped onto a set of vectors composed entirely of numbers (i.e. word vectors). Interestingly, we noticed that the word units separated from the same word reveal a considerably high similarity in the new word vector representation space. For example, the word ‘张勇’ (‘Zhang Yong’), a Chinese person’s name, is always inaccurately segmented by the NLP software into two distinct elements ‘张/勇’ (‘Zhang/Yong’); however, a well-trained neural network mode can embed the new representations (vectors) of ‘张’ (‘Zhang’) and ‘勇’ (‘Yong’), which are highly similar to each other. In this research, we propose a novel method for detecting potential new words in the Chinese texts. Initially, the frequent *n*-gram word strings in a domain corpus are extracted as the candidates of new words. Simultaneously, each word unit in the corpus is represented as a vector by a well-trained neural network model. Eventually, based on the similarity between the vectors of any two candidate strings, we evaluated the possibility that their combination is a new word and pruned the candidates that did not satisfy the evaluation conditions.

This article offers three main contributions. First, we propose a novel framework for new word detection from a large-scale online corpus. To the best of our knowledge, our approach is the first to apply a word embedding method to detect new words. Second, we present a frequent *n*-gram word string mining method for extracting candidates of new words. Third, we design a pruning strategy based on the similarity of word vectors to evaluate if the combination of any two candidate strings is a new word. The remainder of this article is organised as follows. Section 2 provides a brief summary of the related work. Section 3 describes the research framework and the proposed methods in detail. Section 4 reveals the experimental results for the various methods, and section 5 discusses our results and certain limitations of this research. Our conclusions are presented in section 6.

2. Related work

Detection of new words is an important part of TIR, which is connected technically with NLP research in computer science. As such, new word detection has attracted the attention of several researchers and has obtained significant results. In academic literature, solutions for new word detection are divided into two categories, supervised and unsupervised methods [20].

As aforementioned, the supervised methods generally require prior knowledge. Accordingly, Fu and Luke [21] proposed a method for discovering new words based on pattern matching using a set of annotated corpora. First, they labelled portions of the corpora manually as the training sets, in which the vocabulary, vocabulary links and word generation patterns were highlighted. Second, they introduced the hidden Markov model (HMM) to predict new words in the word segmentation results. The main issues that arose with this approach were the high computational complexity and the large number of corpora that had to be labelled manually. Goh et al. [22] treated new word detection as a supervised classification problem. They initially introduced the HMM model for segmenting the texts and prelabelling the words coarsely. Furthermore, a support vector machine (SVM) was trained to extract the key annotations. Eventually, the characters of these key annotations were used to identify the new words from the segmentation results; however, this method suffered from a heavy initial word segmentation problem that led to errors in the follow-up processing work. Xu and Gu [23] presented a method based on the famous SVM classifier. They used prior knowledge in their initial word segmentation, and for tagging parts of speech (POS), which were trained into a new word vector space for easy use by the SVM method. Considering that the characteristics of the potential new words in a corpus may have different segmentation boundaries, Chen et al. [24] proposed a set of statistical features to identify the boundary for new words and then used the conditional random field (CRF) method to synthesise these features and obtain new words in a corpus. Although they reported that their approach achieved excellent experimental results, notably, their method required complex calculation and manual annotation to train for data labelling and establish the feature space.

Unsupervised methods for word detection aim to reduce the amount of required manual work by applying either a rule-based or statistical index-based method. Rule-based methods use either unique or previously recognised language rules to match potential new words in the text. In their work, Wang et al. [25] adopted a filter to obtain general 2–8-gram word strings from the initial segmentation as candidates. Furthermore, they introduced the statistical matching rules to identify new words. Zheng et al. [10] proposed a method for detecting new words in a specific area based on the behaviour of users in the targeted area. In this work, input of vocabulary and representative words by experts was essential.

In general, unsupervised methods include two filtering steps to identify the terms frequently repeated in the corpus as candidates. New words are distinguished among the candidates using statistical indices. Pecina and Schlesinger [26] conducted a large number of lexical recognition experiments using 55 different statistics to identify 2-gram word strings. Their experimental results indicated that mutual information (MI) is the best measurement for evaluating the lexical relevance among words in texts. Taking this idea further, Du et al. [27] presented an automatic *n*-gram new word detection method by combining a pointwise mutual information (PMI) algorithm with a small number of rules. In an NLP environment, if the combination of several word strings indicates a new word, then these word strings should appear repeatedly in various contexts in the corpus. Based on this general idea, Huang and Powers [28] proposed a method for discovering new words by evaluating the information entropy around the candidates.

In summary, several methods proposed in the TIR literature require considerable pretagging work, so that the algorithm may achieve a good performance. In addition to the high labour and time costs, these methods suffer from training set availability, feature optimisation and annotation consistency [29]. In our study, we proposed a novel method to detect new words efficiently from the domain corpora by combining methods of word embedding and frequent *n*-gram strings mining.

3. The new word detection method

3.1. The research framework

In this section, we present a novel method aimed at identifying new words in the Chinese corpora, namely, the word-embedding-based method (WEBM). The process of WEBM is presented in Figure 1.

WEBM begins by extracting massive corpus online for a targeted domain (such as finance, sports or music). All the crawled texts are used to form the initial dataset of $D = \{D_1, ..., D_i, ..., D_{|D|}\}$, where D_i denotes the ith text and $|D|$ represents the total number of documents in D.

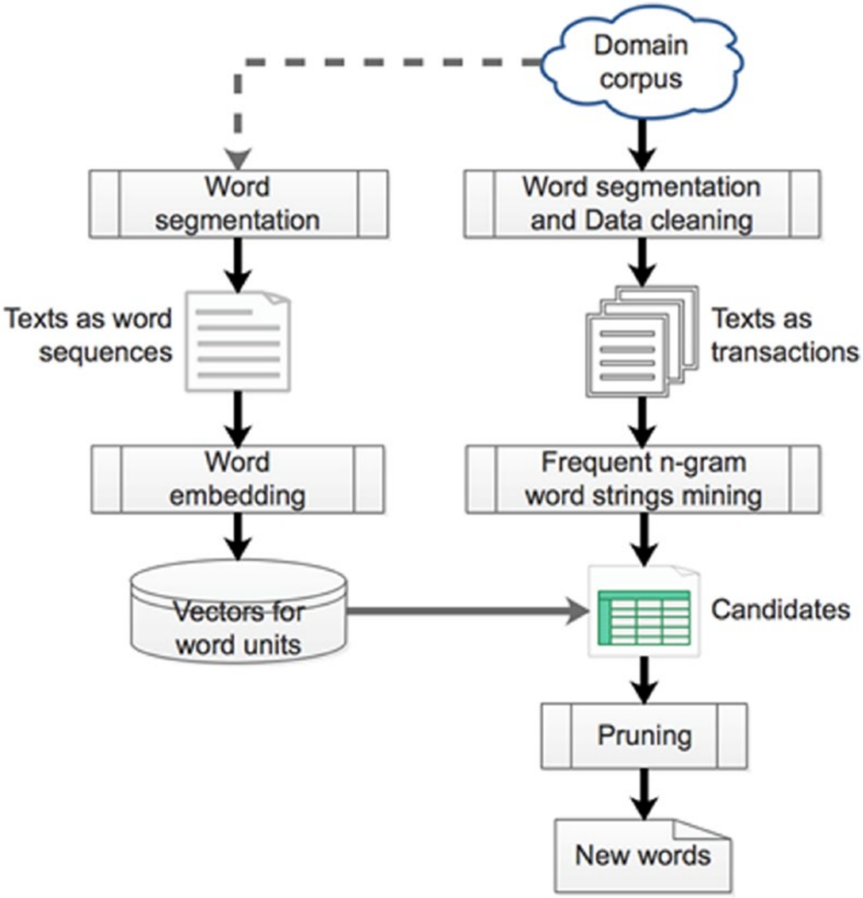

Figure 1. The new words detection process.

The subsequent data processing steps are implemented in parallel, with two paths (see Figure 1). In the process described on the left path, a word segmentation method is used to divide the entire text into word units. Moreover, a word embedding method is introduced to map each word unit to a word vector. In the process described on the right path, word segmentation and data cleaning should be done first, as that in the normal NLP task. As a result, WEBM converts each text into a transaction data record, in which the word units extracted from the text were used as items of transaction data. Furthermore, a frequent pattern mining method is introduced to mine the frequent *n*-gram word strings. Herein, an *n*-gram word string (where *n* is greater than 1) signifies that *n* word units have appeared frequently and simultaneously in the same (or similar) semantic relationships, such as a sentence. Eventually, a pruning algorithm is presented as a filter to obtain the correct new words from the candidates.

3.2. Word segmentation and data preprocessing

Word segmentation is used to extract the word units from a corpus that has been crawled online. The process of word segmentation usually involves the NLP task of tokenising a set of input texts into word units [6,30]. We use third-party software for word segmentation because word segmentation is not the focus of this study. After word segmentation, each text in the corpus will be converted into a form of word sequences (transaction), which corresponds to two different purposes, that is, word embedding and *n*-gram word string mining.

To keep more contextual information of word units after word segmentation, the *i*th text $D_i \in D$ is segmented by software directly into m_i word units without losing any information

$$D_i = \left\{w_{i1}, \ldots, w_{ij}, \ldots, w_{im_i}\right\} \tag{1}$$

where w_{ij} means the *j*th word unit in D_i.

Nevertheless, to improve the performance of *n*-gram word string mining, the noise phrases, stop words and meaningless symbols in corpus are removed [31], and only those word units that might be a part of the new words are reserved. To that end, each text $D_i \in D$ is scanned twice. In the first round of the scan, the content of D_i is divided into small parts by segmentation software, and in the second round of the scan, the duplicated items in D_i are removed so that only one instance remains. Eventually, a transactional dataset T based on D is generated as $T = \{T_1, ..., T_i, ..., T_{|T|}\}$, where $|T|$ indicates the total number of texts and T_i is a transactional record with n_i items. Here

$$T_i = \left\{t_{i1}, ..., t_{ij}, ..., t_{in_i}\right\} \tag{2}$$

which can be calculated as $T_i = \{D_i \setminus S\}$. Here, t_{ij} means the *j*th word units in T_i and S is a predefined dictionary for both punctuation marks and stop words.

Notably, each item in T_i should maintain the same position order as that in D_i. Therefore, T_i is far more than a word sequence, it is a word string. For example, if $D_i = \left\{'A', 'B', 'A', 'A', 'C'\right\}$, then the duplicate '*A*'s are removed leaving a single '*A*', and $T_i = \{'A', 'B', 'C'\}$, in which the position of each word unit is preserved as that in D_i.

3.3. Word embedding

An effective way to improve the computability of text data for NLP is to map each word unit to an appropriate vector space of numbers. There are two types of well-known vectors for NLP in the literature: one-hot representation and word vectors. The latter is a distributed representation of words in a corpus based on the following statistical language model [32]

$$p\left(w_1^T\right) = \prod_{t=1}^{T} p\left(w_t \middle| w_1^{t-1}\right) \tag{3}$$

where w_t represents the *t*th word, and $w_1^T = \left(w_1, w_2, \ldots, w_{T-1}, w_T\right)$ represents the word sequence before w_t. This general approach, referred to as word embedding, has performed well in generating the numerical word vector representations in a variety of NLP tasks; however, given a word sequence generated by segmentation software as relation (2), calculation of relationship (3) is a difficult task when n_i becomes larger. Recently, several training methods have been proposed to seek efficient ways to represent the word units as appropriate vectors. In particular, two models – the continuous bag-of-words (CBOW) and continuous Skip-gram models proposed by Mikolov et al. [33,34] – have revealed significant advantages in efficiency in training neural network models for word representation. The main idea of CBOW is to predict the representation of a target word that appears in the middle of other words, by combining the representations of the surrounding words in a sentence or a document. In comparison, the training objective of the Skip-gram model is to predict the surrounding words using the word representation in the middle.

By adopting the well-trained CBOW and Skip-gram models, we can obtain a numerical vector to represent word w_i in D by mapping it onto a vector of $vec(w_i)$

$$w_i \to vec\left(w_i\right)^K \tag{4}$$

where K denotes the dimension of the word vector, which usually is suggested as a value between 50 and 200. In the following, we use $W(D)$ to denote all the word vectors for words in D.

3.4. Mining frequent n-gram word strings

If there are *n* word units that occur simultaneously and frequently in the same, or similar, text environments in a corpus, then there is a considerably high probability that their combination might become a potential new word [28]. We call such a combination an '*n*-gram word string', where *n* represents the number of word units in the string. For example, the text 'CEO张勇表示' ('CEO Zhang Yong said') would be turned into 'CEO/张/勇/表示' ('CEO/Zhang/Yong/said') after word segmentation. If the combination of '张勇表示' ('Zhang Yong said') appears frequently in multiple texts, then a frequent 3-gram word string of {'张', '勇', '表示'} ({'Zhang', 'Yong', 'said'}) can be obtained. In the method proposed in this work, the task of *n*-gram word string mining will be transferred into an equivalent task of finding the frequent *n*-itemsets (patterns) from the corpus.

The frequent patterns are itemsets that appear in a dataset no less than a user-specified threshold. In recent years, frequent pattern mining has been introduced in NLP tasks as a necessary preprocessing method for extracting quality phrases from corpora [14,35]. In our work, the task of frequent pattern mining is conducted on the transaction dataset of *T*. Given

Algorithm 1. Frequent *n*-gram word string mining.

```
Input: $T$, $min_s$;
Output: A set of $n$-gram string: $NGS$;
 1: $n = 1$;
 2: while $FP^{(n)}(T) \neq \phi$ do
 3:     $n \leftarrow n+1$;
 4:     Mining $FP^{(n)}(T)$ with threshold $min_s$;
 5: end while
 6: for $i = 1$ to $|T|$ do
 7:   for $j = 1$ to $|T_i| - 1$ do
 8:      $k = 1$;
 9:      while $\{\overline{t_{ij} \ldots t_{ij+k}}\} \in FP^{(k+1)}(T)$ is TRUE do
10:          $k \leftarrow k+1$;
11:      end while
12:      if $k > 1$ then
13:          $NGS \leftarrow \{\overline{t_{ij} \ldots t_{ij+k-1}}\}$;
14:      end if
15:    end for
16: end for
17: return $NGS$
```

Table 1. Sample for candidate word strings.

T_i	Candidate word strings
T_1	{'Zhang', 'Yong', 'Said'} ({'张', '勇', '表示'})
T_2	{'Alibaba', 'declared'} ({'阿里巴巴', '宣布'})
T_3	{'integrity', 'department'} ({'廉政', '部'})
T_4	{'propose', 'privatization', 'offer'} ({'提出', '私有化', '要约'})

the database T and the predefined threshold of *minimum support* $= min_s$, the object of frequent pattern mining is to collect all the n-itemsets whose frequency is larger than min_s into set $FP^{(n)}(T)$. In contrast to the traditional mining tasks, the order of the word units in an n-gram string needs to be preserved. Thus, we presented Algorithm 1 as a new mining method.

In Algorithm 1, all of the frequent itemsets (lines 1–5) are extracted first. Notably, any frequent pattern mining technique is feasible, theoretically, for the $FP^{(n)}$ (T) mining task. Furthermore, for the ith transaction T_i, its cutting-string $\{t_{ij} \ldots t_{ij+k}\}$ is matched with the frequent patterns in $FP^{(k+1)}$. If it matches, then $\{t_{ij} \ldots t_{ij+k}\}$ is a k-gram word string (lines 6–16). Eventually, the calculation result is represented as a set of n-gram strings, namely NGS (line 17). Some sample word strings could be found by the mining algorithm as presented in Table 1.

Algorithm 1 can help us find a large volume of n-gram word strings ($n = 2, \ldots, k$); however, these word strings cannot be regarded as new words directly due to noise. In Table 1, for example, '张勇' ('Zhang Yong') in the 3-gram word string of {'张', '勇', '表示'} can serve as a new word, but the total string of '张勇表示' ('Zhang Yong said') cannot be a new word. Therefore, we need a pruning method to remove the noise word unit '表示' ('said').

3.5. Word vector–based pruning strategy

Although a new word may be segmented inappropriately into several word units by the word segmentation software, parts of these word units still reveal some relationships in the corpus. These relationships mainly reflect their various correlations, such as context, location, grammatical function and even frequency. As discussed earlier in this article, the vector embedded by a neural network for a word unit in a corpus might be a good representation of these correlations. As a result, if some word units are inappropriately separated from the same new word, their word vectors would reveal some similarity. For example, in our experiments, the sequence of {'张', '勇', '表示'} ({'Zhang', 'Yong', 'said'}) is found as a 3-gram word string, in which the cosine similarity between the two-word vectors corresponding to '张' (Zhang) and '勇' (Yong) is 0.38, whereas the cosine similarity between the word vectors corresponding to '勇' (Yong) and '表示' (said) is 0.01. These results indicate that the combination of '张' and '勇' has a high probability of being a new word, which is far greater than that of the combination of '勇' and '表示'.

Algorithm 2. Word-vector-based pruning.

```
Input: $W(D)$, $NGS$, $f_0$;
Output: A set of new words: $NW$;
 1: $NW = \phi$;
 2: while $NGS \neq \phi$ do
 3:    Pick up one word string $s \in NGS$;
 4:    Pick up word vectors $vec(s_i)_{i=1,\ldots,|s|} \in W(D)$;
 5:    FLAG = 1;
 6:    for $i = 1$ to $|s| - 1$ do
 7:       if $f(vec(s_i), vec(s_{i+1})) \geq f_0$ then
 8:          Continue;
 9:       else
10:          if $i - FLAG \geq 1$ then
11:             $NW \leftarrow \left\{ s_{FLAG} \ldots s_i \right\}$;
12:             $FLAG \leftarrow (i+1)$;
13:          end if
14:       end if
15:    end for
16:    $NGS \leftarrow NGS \setminus s$;
17: end while
17: return NW
```

Based on the word segmentation technology, the semantic similarity of word units in the same word and the principle of word representation technology, we constructed a novel new word pruning strategy to obtain correct new words from the candidate word strings generated by Algorithm 1. In this strategy, we introduce a function $f(\cdot)$ to measure the similarity between any two-word vectors. By using $vec(w_1)$ and $vec(w_2)$ to denote the vectors of two-word units of w_1 and w_2 in the same n-gram word string, if the similarity between $vec(w_1)$ and $vec(w_2)$ is greater than or equal to threshold f_0, that is

$$f\left(vec\left(w_1\right), vec\left(w_2\right)\right) \geq f_0 \tag{5}$$

then the word string $\{w_1 w_2\}$ as a new word will be reserved. Otherwise, the combination of $\{w_1 w_2\}$ should be pruned. Algorithm 2 presents the processes of the word-vector-based pruning strategy.

For each n-gram word string $s \in NGS$, all word vectors for its word units (line 4) will be considered first. Furthermore, the similarity between $vec(s_i)$ and $vec(s_i+_1)$ ($i = 1, \ldots, |s|$) will be calculated to find the longest combination for a potential new word (lines 5–17). After the pruning, the inappropriate combinations in NGS, such as '勇表示' in word string $s = \{$'张', '勇', '表示'$\}$, can be removed.

4. Data experiments

4.1. Datasets and experimental setup

The research goal of this study is to detect new words from massive domain texts. Considerably, four corpora have been extracted from different domains for experiments. The first two domains are regarding finance and sports. Two corpora are extracted separately from www.sina.com and www.sohu.com, which are two well-known online news providers in China. The texts presented on these two sites are typically generated by professional reporters; hence, the words and phrases used are expected to be more standardised. The two other domains are tourism and music, for which, we have collected reviews from www.mafengwo.com and http://music.163.com/, respectively. All the texts on these two sites are contributed voluntarily by users and are user-generated content (UGC). We presumed it would be extremely challenging to detect new words from UGC since there are no guidelines for regulating the users' writing styles.

We introduced the third-party software of Jieba (https://github.com/fxsjy/jieba) to conduct the task of word segmentation because it employs the state-of-the-art models and has revealed strong performance in segmenting Chinese texts. The statistical information of the four datasets is presented in Table 2.

Using the above four datasets, we conducted a series of experiments to examine the feasibility and the advantages of the method proposed in this work. Our work comprised two parts: conducting test experiments to examine the process of WEBM (sections 4.2–4.5) and conducting the comparative experiments to measure the performance of WEBM against other state-of-the-art methods (section 4.6). Notably, in implementing the WEBM method, we first introduce a neural network model to transform all the data in a corpus into a set of word vectors and then conduct Algorithms 1 and 2 to

Table 2. Statistical information of the four datasets used in the experiments.

Dataset	Domain	URL	Size (M)	# Sentences	# Word units
D_F	Finance	http://finance.sina.com.cn/chanjing/	20.2	137,052	4,088,153
D_S	Sport	http://sports.sohu.com/guojizuqiu_a.shtml	11.3	115,245	1,658,567
D_T	Tourism	http://www.mafengwo.cn	59.4	515,345	12,481,842
D_M	Music	http://music.163.com/	75.2	1,003,401	14,910,600

obtain the potential new words. In addition, it is important to determine how to obtain the ground-truth for detecting new words from a real-world dataset. Without losing generality, we follow the common practice in NLP research and use an 'expert(s) annotation' strategy in the experiments to evaluate the final results; however, it is known that the manual evaluation by experts is quite difficult and time consuming. Therefore, in the following experiments, we implement the task of new word detection only on a small part of the corpus (i.e. test set), so that the experts could make quick and reasonable judgments. Considerably, four test sets are generated and in each test set, 2000 texts are extracted from the corresponding domain dataset. Furthermore, three experts are invited to read the texts in each test set and to select the new words from the texts as the ground-truth for the corresponding domains.

4.2. Word embedding experiments

In the literature, the training processes of neural network models for word embedding are extremely time consuming and resource intensive. In 2013, Google presented an efficient tool called word2vec[1] for training word embedding models. It has been proven that the performance of word2vec is much faster and better than the other methods. In experiments of word embedding for WEBM, we would like to introduce word2vec to generate the word vectors.

In order to receive a high quality of word vectors with word2vec, the window size, vector dimension and training model parameters must be set appropriately. In our experiments, we set the window size to 5 and the vector dimension to 200 [36]. As for the training model, both the Skip-gram and CBOW models are well supported by the word2vec toolkit. To make the optimal choice for the experiments, we tested the performance of each in WEBM.

In data mining tasks, it is necessary to balance the performance of the mining method in terms of the precision and recall. If the goal is to pursue a high rate of precision, then the performance on recall must be sacrificed and vice versa. Following this line of thought, we examined the performances of the Skip-gram and CBOW models in WEBM. Initially, the Skip-gram and CBOW models were used separately in word2vec to generate two sets of word vectors. Moreover, each set of word vectors was used to detect new words from the four test sets. The performances of precision and recall on each test set are illustrated in Figure 2. The results revealed that the Skip-gram model achieved a better performance than the CBOW model. Therefore, this study used the Skip-gram model in word2vec to generate the word vectors.

4.3. Frequent n-gram word string mining

In this section of our work, we mined frequent *n*-gram word strings from the test sets to determine the candidates for new words. To simplify the calculation, we adopted the frequent pattern mining algorithm used in the association analysis [37]. In this algorithm, a threshold of minimum support (or corresponding *support count*[2]) is used to filter the noise patterns in the dataset. For example, if we set the minimum support threshold as 3, an *n*-gram word string whose frequency is greater than 3 is deemed a frequent *n*-gram word string. Notably, the support count of a pattern refers to the frequency of a word string in a corpus.

The existing research on frequent pattern mining demonstrated that the effect of the support count is essential, but quite subtle. If the value is set to be considerably small, then thousands of patterns may be generated with too much noise, whereas if the value of the support count is too large, that setting may lead to limited results [38].

Considering the aforementioned concerns, we explored the total number of *n*-gram ($n \geq 2$) word strings that could be gathered from each test set under different minimum support (minimum frequency) levels, with a range from 3 to 60. The results are presented in Figure 3. Notably, in all four test sets, most *n*-gram word strings could be found by setting the threshold of support count at 10. Only a few word strings had a frequency larger than 10. These results indicated that the appropriate value of minimum support for mining frequent *n*-gram word strings may range from 3 to 10, which corresponded to the value of minimum support from 0.15% to 0.5%. In our experiments, the minimum support count was set to 3 to mine new words as much as possible.

In addition, the results in Figure 3 also revealed that the average frequency of *n*-gram word strings generated by professional reporters (Figure 3(a) and (b)) was higher than those generated by the nonprofessional writers (Figure 3(c)

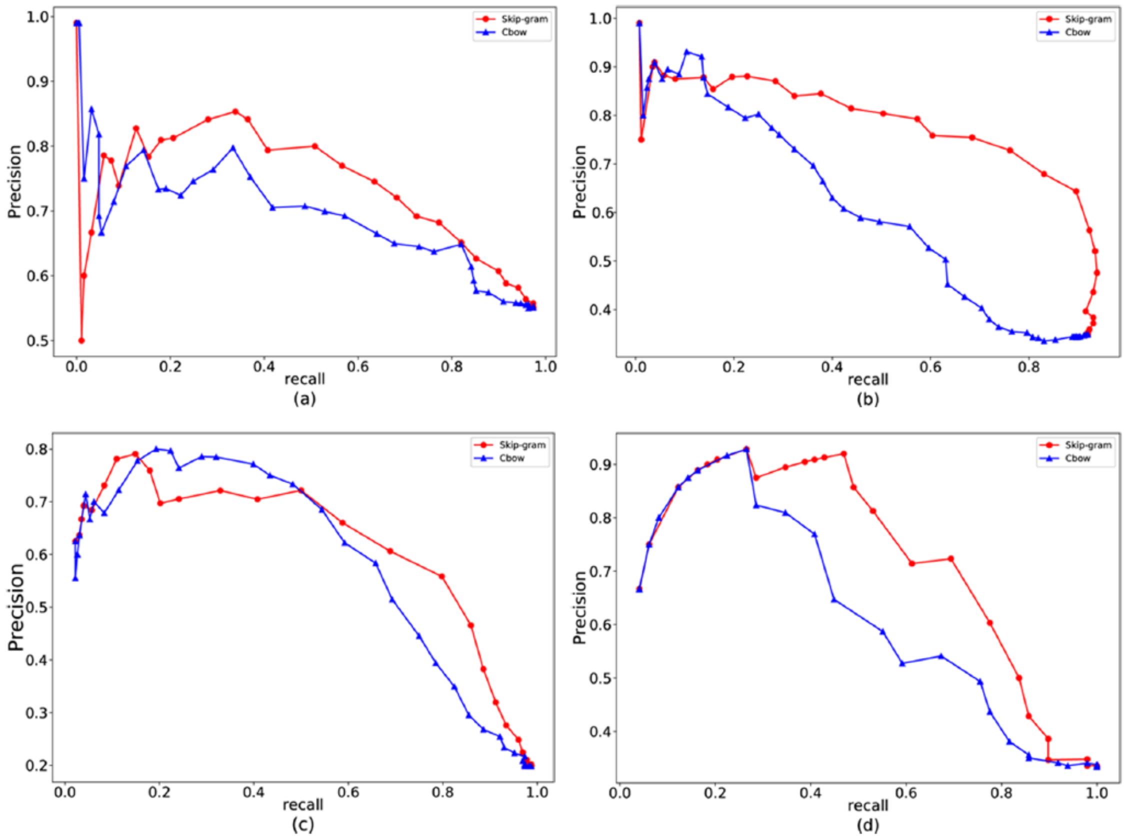

Figure 2. The precision-recall comparison of the Skip-gram and CBOW model on test sets: (a) finance news, (b) sport news, (c) tourism review and (d) music review.

and (d)). In particular, in the normative language environment, new words were used more frequent than in the other language environments, which imply that new words may be easier to find in a normative language environment.

4.4. Candidate pruning and parameter tuning

Based on the candidates generated by Algorithm 1, we can run the pruning algorithm (Algorithm 2) on the candidate dataset (frequent *n*-gram word strings) to identify the potential new words. It is known that the similarity measure used in Algorithm 2, the function of $f(\cdot)$, may have a significant impact on the performance of WEBM. In the literature of data mining, the most commonly used similarity measures for vector comparison are cosine similarity, Euclidean distance and Manhattan distance [39,40]. In this work, we need to conduct a set of experimental analyses to compare the performance of these measures. Therefore, each of the three measures of similarity are introduced systematically into Algorithm 2 to prune the candidates. Eventually, the precision and recall performance of WEBM was reported based on the different similarity metrics.

Figure 4 represents the precision-recall results of the four test sets, from which we can draw some interesting and helpful information for new word detection. First, for any dataset, if we needed higher precision, the performance of WEBM on recall was usually poor, as indicated by the initial sections of the curves in Figure 4. Second, the cosine similarity worked much better than the Euclidean or Manhattan distance for pruning candidates, especially in the case of detecting new words in the UGC dataset (see Figure 4(c) and (d)). Based on the experimental results displayed in Figure 4, we chose the cosine similarity to measure the similarity between two-word vectors in WEBM for the process of pruning candidates. The optimal threshold of f_0 is set based on the following exploratory experiments.

As demonstrated by the experimental results in Table 3, when the value of f_0 increased from 0.1 to 0.4, the *precision* value gradually increased. When f_0 ranged from 0.4 to 0.7, the *precision* value was stable. When $f_0 > 0.7$, the *precision* value began to decline. In contrast to the performance of *precision* on different f_0, the *recall* value dropped sharply when f_0 exceeded 0.6. Based on these results, $f_0 = 0.4$ may be a feasible value for the cosine similarity in WEBM.

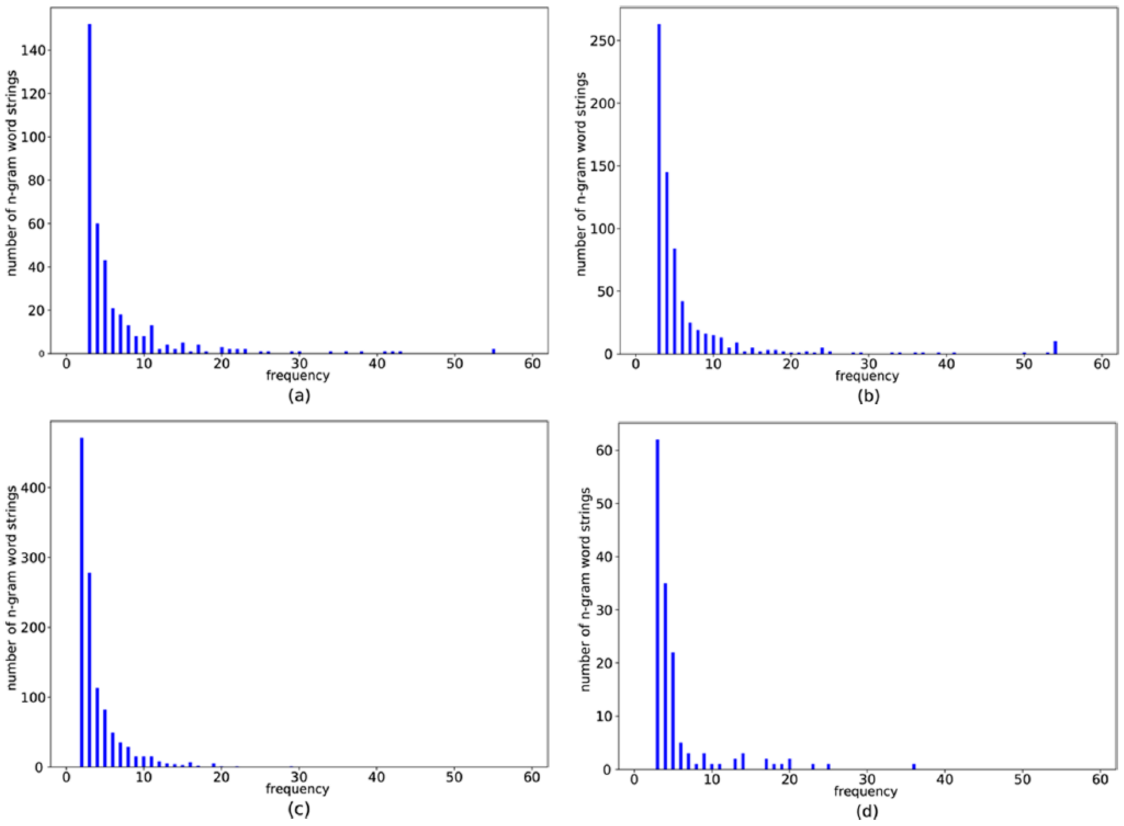

Figure 3. The frequency distributions of the mined *n*-gram word strings: (a) finance news, (b) sport news, (c) tourism review and (d) music review.

4.5. Results of implementing new word detection

The performance of WEBM regarding the detection of new words on the four test sets is summarised in Table 4. The test corpus of finance news, for example, has a total of 604 candidate *n*-gram word strings which were mined, and 216 new words were identified by the pruning algorithm, as compared with 278 new words marked by the experts. Similar results were obtained by WEBM on the three other test corpora.

We were interested in the types of new words that would be discovered by WEBM. As a demonstration, Table 5 reveals some interesting new words that first were inappropriately segmented by the software and then detected and reassembled by our method. For example, the detected new word '蘑菇街' ('Company Mogujie') was always segmented inappropriately by the third-party software as two parts, '蘑菇' ('Mushroom') and '街' ('Street'), marked as '蘑菇/街'.

Interestingly, WEBM demonstrated a special capacity to recover most of the domain-related named entities that are usually not included in the dictionaries of word segmentation software. Examples included the names of VIPs and organisations (finance domain), names and nicknames of athletes (sport domain), names of local foods and buildings (tourism domain), and names of music artists and the titles of newly published songs (music domain). Furthermore, in the tourism and music domains (the UGC corpora), some newly generated popular slang terms were extracted appropriately by WEBM, such as '抖腿' ('Shake legs'), '点赞' ('Thumbs up'), '二柱子' ('Erzhuzi'), '文字范' ('Fonta'), '老北京' ('Old Peking') and '良心价' ('Conscience price'), among others.

4.6. Comparative experiments

To illustrate the effectiveness of WEBM in detecting new words, we also conducted a series of experiments on the test sets that compared our proposed approach with several state-of-the-art methods. In the literature, several machine learning algorithms have been used for new word detection. Since WEBM is unsupervised, we first used two popular unsupervised methods for comparison: the PMI method [27,41] and the left-right information entropy (LE) method [42–44].

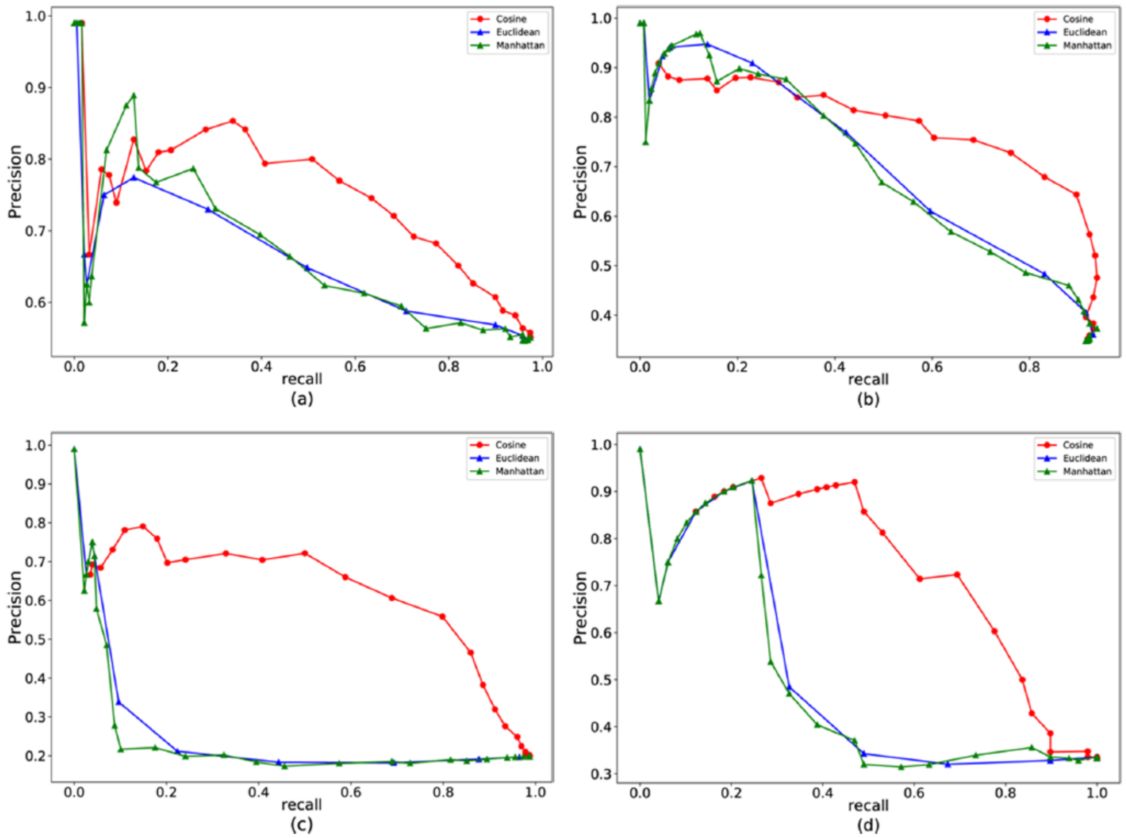

Figure 4. Comparing the performance of different similarity metrics in candidate pruning on test sets: (a) finance news, (b) sport news, (c) tourism review and (d) music review.

Table 3. Impacts of f_0 on the performance of WEBM.

f_0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Precision	0.5769	0.625	0.6667	**0.6842**	**0.6875**	**0.6875**	**0.7**	0.625	0.5
Recall	**0.0893**	**0.0893**	**0.0833**	**0.0774**	0.0654	0.0655	0.0417	0.0298	0.0126
F-value	**0.1546**	**0.1562**	**0.1481**	**0.1390**	0.1196	0.1196	0.0787	0.0568	0.0248

WEBM: word-embedding-based method.
Note: The bold values highlight these highest values for each measure, which are basically stable. By highlighting the highest values with different values of f_0, it is easy to infer that f_0 = 0.4 may be a feasible value for the cosine similarity in WEBM, which takes into account the optimality of the three measures.

Table 4. New word detection results from test set for each domain.

Domain	Size of test set	# of *n*-gram candidates	# of new words (by WEBM)	# of new words (by experts)
Finance	2000	604	216	278
Sport	2000	678	296	260
Tourism	2000	1139	268	228
Music	2000	147	44	49

WEBM: word-embedding-based method.

In the PMI-based method, before candidate pruning, the results used in PMI were the same as those used in WEBM. The significant difference between PMI and WEBM exists in the subsequent pruning process, in which the PMI value between two-word units was calculated and then used to filter the noise from the frequent *n*-gram word strings.

In the LE method, the left entropy was obtained by computing the information entropy for all possible words to the left of a word string and then summing the results. The right entropy was calculated in a similar way for all words to the right

Table 5. Ten sample new words detected in the test set for each domain.

Finance	Sport	Tourism	Music
蘑菇/街	石原/直/树	猪扒/包	李/健
(Company Mogujie)	(ShiYuanZhiShu)	(Pork package)	(Lee Jian)
穷/游/网	平野/甲/斐	猪脚/煲	螺旋/丸
(Qiongyouwang)	(PingYeJiaFei)	(Pig pot)	(Spiral pill)
对冲/基金	银河/战舰	英式/下午/茶	B/站
(Hedge-Fund)	(Galacticos)	(British afternoon tea)	(B station)
持股/比例	锋/霸	梅子/粉	单曲/循环
(Share-holding ratio)	(Front Hegemony)	(Plum Jelly)	(Single cycle)
林/翰	互/交白卷	酸/萝卜	贝加尔湖/畔
(Lin Han)	(Goalless)	(Acid/radish)	(Banks of Lake Baikal)
邵/晓/锋	哈/维	土/楼群	吓/得
(Shao Xiaofeng)	(Harvey)	(Soil building group)	(xia de)
姜/鹏	AC/米兰	芝士/排骨	佐/助
(Jiang Peng)	(AC Milan)	(Cheese ribs)	(Sasuke)
魅/族	拉/基蒂/奇	艇/仔/粥	红旗/歌舞团
(Mei zu)	(Rakitic)	(Tingzai porridge)	(Hongqi Song and Dance Troupe)
孙/玉成	张/稀/哲	老/酸奶	曲/库/
(Sun Yucheng)	(Zhang Xizhe)	(Old/yogurt)	(Song library)
乐/视网	安菲尔德/球场	妙/香/居	电/音
Letv	(Anfield Stadium)	(Miaoxiangju home)	(Electric tone)

of the same word string. Eventually, both the left and the right entropy were summed as an indicator to determine whether a combination of the two adjacent word strings was a new word.

In addition, the method of CRF is regarded as the best self-standing method for detecting new words [4,24]. CRF is particularly good at fine-grained named entity recognition [45]. Accordingly, we introduced the CRF method in the comparative experiments as well. The implementation of CRF was based mainly on the CRF++ tool,[3] in which we chose the five words just before and after the present location, as well as their associated POS, as features.

For comparison, we utilised three measurements, precision, recall and *F*-measure, to evaluate the performance of the methods for detecting new words from a corpus, as given

$$Precision = \frac{\#\, new\ words\ correctly\ identified}{\#\, recognized\ \ words} \tag{6}$$

$$Recall = \frac{\#\, new\ words\ correctly\ identified}{\#\, new\ words\ in\ the\ corpus} \tag{7}$$

$$F\text{-}measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{8}$$

The results of the comparative experiments are presented in Table 6. As discussed in Table 6, WEBM performed much better on both precision and recall than the other methods. One explanation for the results may be that WEBM depends more on the co-occurrence of word strings. Similarly, we can see that PMI and LE can achieve higher accuracy in new word detection under some textual environments; however, these two methods are more susceptible to the noise relation of co-occurrence among word units. Since noisy data are unavoidable in large corpora, especially in a UGC dataset, it was difficult for the PMI and LE methods to reveal high effectiveness on all four test sets. Notably, the CRF method may not have achieved its best performance here because of the lack of optimisation for its feature selection.

In addition, the significant difference between the WEBM and PMI exists in their different pruning processes. We further examined the performance of these two methods on new word detection. In the experiment, we extracted six test sets from the financial corpus. The sizes of these test sets ranged from 1000 to 2000. The experimental results reveal that WEBM is superior to PMI in accuracy, recall and *F*-value. Figure 5 is a visual comparison of *F*-values.

Table 6. The comparative results of the methods.

Method	Precision	Recall	*F*-measure
PMI	61.8%	55.9%	58.7%
LE	59.1%	51.3%	54.9%
CRF	72.2%	50.6%	61.1%
WEBM	81.2%	60.1%	69.1%

PMI: pointwise mutual information; LE: left-right information entropy; CRF: conditional random field; WEBM: word-embedding-based method.

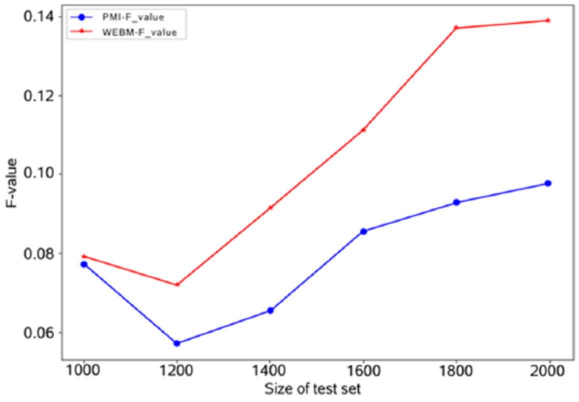

Figure 5. The comparison of *F*-values on financial test sets.

The advantage of WEBM is that it considers both the relation of co-occurrence frequency and location adjacency between word units. In particular, the use of the word embedding method causes WEBM to retain more context information [36,46] and then provides some capability for noise tolerance in new word detection. Based on a comparison of the forms of the new words that were discovered by these methods, we found that WEBM was able to identify more low frequency words than the other methods.

5. Discussion

The approach presented in this work is focused mainly on two sub-tasks: candidate generation based on frequent pattern mining and candidate pruning based on word embedding. The study results presented by Mikolov et al. [33] reported that the higher the dimension of the word vector, the better the semantic effects. Alternatively, more training data leads to longer training time. Therefore, the following experiment explored the effect of corpus size on the performance of WEBM (Table 7).

As seen from the aforementioned results, with the increase in corpus size, the precision tended to be stable after a brief rise, while the recall and *F*-value tended to be steady. In general, as more documents were trained, the quality of the word vectors (and the related word representation) increased; however, based on the aforementioned experimental results, we can conclude that if the quality of the word vector reaches a certain high level, WEBM will have difficulty in improving its performance in new word detection by adding more corpora, that is, by improving the quality of word vectors. Clearly, one limitation of the WEBM is the mechanism of generating candidate sets, which needs to be improved in the future.

Since the most atomic unit of Chinese words include single characters, we implemented the following experiment to explore the WEBM performance with retrieving new words from a set of Chinese characters. Initially, we segmented the texts of the finance test set into a finer granularity formation of Chinese characters. Furthermore, we mapped these characters to word vectors using the Skip-gram model. Moreover, we introduced WEBM to identify the real new words from the test sets. Eventually, WEBM extracted 147 word strings as candidates and reported 20 new words, of which 7 were correct. These results revealed that WEBM has the capability to retrieve new words from a set of Chinese characters; however, the performance was considerably poor, precision = 0.35, recall = 0.0021 and *F*-value = 0.0042. WEBM revealed such bad performances because the distribution of characters in a corpus differs from the distribution of words (consisting of several characters); hence, the contextual information learned by a word embedding method is extremely messy. Therefore, the other limitation of WEBM is that, like other methods, its performance is also affected by the segmentation results.

Table 7. The effects of corpus size on the performance of WEBM.

Number of texts	1000	1500	2000	2500	3000	3500	4000
Corpus size (MB)	7.06	10.53	13.84	17.21	20.49	23.99	27.82
Precision	0.5926	0.5600	0.6087	0.6364	0.6667	0.6500	0.6667
Recall	0.0952	0.0833	0.0833	0.0833	0.0833	0.0774	0.0833
F-value	0.1641	0.1451	0.1466	0.1474	0.1481	0.1383	0.1481

WEBM: word-embedding-based method.

6. Conclusion and future work

With the evolution of social and linguistic environments, an increasing number of new words are being constantly generated. These new words deliver important information for communications; hence, they need to be identified correctly by the TIR system. On one hand, NLP-related information retrieval relies heavily on word segmentation to obtain the basic word units used in texts; whereas, on the other hand, traditional word segmentation toolkits are unable to identify all new words accurately and efficiently. Therefore, the identification of new words remains a challenge for TIR.

In recent years, innovative word embedding methods have been able to map words in a corpus efficiently to numerical vector space. These new representations, word vectors, can retain more semantic information and correlations between words. Inspired by this idea, this article proposed a novel unsupervised method, WEBM, for detecting new words from a massive domain corpus. In general, WEBM combines three methods, word embedding, frequent *n*-gram string mining and noise pruning strategy. By working with the word embedding method, WEBM can consider a wide range of correlations between word units, rather than relying only on their co-occurrence. By working with the *n*-gram string mining method and pruning strategy, WEBM can effectively capture the frequent word strings in the text and filter out the noisy relationship. To the best of our knowledge, this approach is the first to introduce word embedding for new word detection. Our experimental results on four domain corpora revealed that the performance of the proposed method (WEBM) was superior to the results from the traditional unsupervised methods.

Future research will explore measures for increasing the volume of the corpora and strive to improve the vector representation and candidate generation. In addition, more benchmarked methods will be introduced into comparative experiments.

Acknowledgements

The authors thank the editors and the anonymous reviewers for their kind comments and suggestions.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship and/or publication of this article.

Funding

This work was partly supported by the National Natural Science Foundation of China (Nos 71671027, 71572029, 71490723, 71271044, 71402007 and 71772017), the Beijing Municipal Social Science Foundation (No. 17GLB009), the Scientific Research Foundation by MOE and CMCC (China Mobile Communications Corporation) (No. MCM20170505), the Fundamental Research Funds for the Central Universities (No. 2018XKJC04). Thanks also to the supports of Yunnan Science and Technology Fund (2017FA034), Yunnan Province Young Academic and Technical Leader Candidate Program (2018HB), Yunnan Provincial E-Business Entrepreneur Innovation Interactive Space (2017DS012) and Kunming Key Laboratory of E-Business and Alternative Finance (KGF[2018]18).

Notes

1. https://code.google.com/archive/p/word2vec/
2. Here, the relation between the concept of *support* and *support count* is defined as *support*=*support_count*/2000.
3. https://taku910.github.io/crfpp/

References

[1] Choi K-S, Isahara H, Kanazaki K, et al. Word segmentation standard in Chinese, Japanese and Korean. In: *Proceedings of the 7th workshop on Asian language resources*, Singapore, 6–7 August 2009, pp. 179–186. Stroudsburg, PA: Association for Computational Linguistics.

[2] Wang Y, Kazama J, Tsuruoka Y, et al. Adapting Chinese word segmentation for machine translation based on short units. In: *Proceedings of the 7th conference on international language resources and evaluation (LREC'10)*, Valletta, 17–23 May 2010, pp. 1758–1764. Luxembourg: European Language Resources Association (ELRA).

[3] Liu Q, Wu L, Yang Z, et al. Domain phrase identification using atomic word formation in Chinese text. Knowl-Based Syst 2011; 24: 1254–1260.

[4] Peng F, Feng F and McCallum A. Chinese segmentation and new word detection using conditional random fields. In: *Proceedings of the 20th international conference on computational linguistics (COLING'2004)*, Geneva, 23–27 August 2004, p. 562. Stroudsburg, PA: Association for Computational Linguistics.

[5] Thet TT, Na J-C and Ko Ko W. Word segmentation for the Myanmar language. *J Inf Sci* 2008; 34: 688–704.

[6] Gao J, Li M, Wu A, et al. Chinese word segmentation and named entity recognition: a pragmatic approach. *Comput Ling* 2005; 31: 531–574.

[7] Sproat R and Emerson T. The first international Chinese word segmentation bakeoff. In: *Proceedings of the second SIGHAN workshop on Chinese language processing*, Sapporo, Japan, 11–12 July 2003, pp. 133–143. Stroudsburg, PA: Association for Computational Linguistics.

[8] Huang X, Peng F, Schuurmans D, et al. Applying machine learning to text segmentation for information retrieval. *Inf Retr* 2003; 6: 333–362.

[9] Huang M, Ye B, Wang Y, et al. New word detection for sentiment analysis. In: *Proceedings of the 52nd annual meeting of the Association for Computational Linguistics*, Baltimore, MD, 23–24 June 2014, pp. 531–541. Stroudsburg, PA: Association for Computational Linguistics.

[10] Zheng Y, Liu Z, Sun M, et al. Incorporating user behaviors in new word detection. In: *Proceedings of the twenty-first international joint conference on artificial intelligence (IJCAI-09)*, Pasadena, CA, 2009, pp. 2101–2106. San Francisco, CA: Morgan Kaufmann Publishers.

[11] Asur S, Huberman BA, Szabo G, et al. Trends in social media: persistence and decay. In: *Proceedings of the 5th international AAAI conference on weblogs and social media*, Barcelona, 17–21 July 2011, pp. 434–437. Menlo Park, CA: AAAI Press.

[12] Zhang L, Zhao J and Xu K. Who creates trends in online social media: the crowd or opinion leaders? *J Comput Mediat Commun* 2016; 21: 1–16.

[13] Qiu L and Zhang Y. Word segmentation for Chinese novels. In: *Proceedings of the twenty-ninth AAAI conference on artificial intelligence (AAAI'15)*, Austin, TX, 25–30 January 2015, pp. 2440–2446. Menlo Park, CA: AAAI Press.

[14] Xie T, Wu B and Wang B. New word detection in ancient Chinese literature. In: *Asia-Pacific web (APWeb) and web-age information management (WAIM) joint conference on web and big data*, Beijing, China, 7–9 July 2017, pp. 260–275. Cham: Springer.

[15] Li H, Huang C-N, Gao J, et al. The use of SVM for Chinese new word identification. In: *First international joint conference on natural language processing (IJCNLP 2004)*, Hainan Island, China, 22–24 March 2004, pp. 723–732. Cham: Springer.

[16] Ma B, Zhang N, Liu G, et al. Semantic search for public opinions on urban affairs: a probabilistic topic modeling-based approach. *Inf Process Manage* 2016; 52: 430–445.

[17] Liu Y and Lin C. A new method to compose long unknown Chinese keywords. *J Inf Sci* 2012; 38: 366–382.

[18] Cheng F, Duh K and Matsumoto Y. Synthetic word parsing improves Chinese word segmentation. In: *Proceedings of the 53rd annual meeting of the Association for Computational Linguistics and the 7th international joint conference on natural language processing*, Beijing, China, 26–31 July 2015, pp. 262–267.

[19] Firth JR. A synopsis of linguistic theory, 1930–1955. In: *Studies in linguistic analysis*. Oxford: Philological Society, 1957, pp. 1–32.

[20] Liang Y, Yin P and Yiu SM. New word detection and tagging on Chinese Twitter stream. In: Madria S and Hara T (eds) *Big data analytics and knowledge discovery*. Cham: Springer, 2015, pp. 310–321.

[21] Fu G and Luke K-K. Chinese unknown word identification using class-based LM. In: *First international joint conference on natural language processing (IJCNLP 2004)*, Hainan Island, China, 22–24 March 2004, pp. 704–713. Berlin: Springer.

[22] Goh C-L, Asahara M and Matsumoto Y. Machine learning-based methods to Chinese unknown word detection and POS tag guessing. *J Chinese Lang Comput* 2006; 16: 185–206.

[23] Xu Y and Gu H. New word recognition based on support vector machines and constraints. In: *2nd international conference on information science and control engineering (ICISCE'2015)*, Shanghai, China, 24–26 April 2015, pp. 341–344. New York: IEEE.

[24] Chen F, Liu Y, Wei C, et al. Open domain new word detection using condition random field method. *J Soft* 2013; 24: 1051–1060.

[25] Wang M-C, Huang C-R and Chen K-J. The identification and classification of unknown words in Chinese: an n-grams-based approach. In: Akira I and Yoshihiko N (eds) *Festschrift for professor Akira Ikeya*. Tokyo, Japan: The Logico-linguistics Society of Japan, 1995, pp. 113–123.

[26] Pecina P and Schlesinger P. Combining association measures for collocation extraction. In: *Proceedings of the COLING/ACL on main conference poster sessions*, Sydney, NSW, Australia, 17–18 July 2006, pp. 651–658. Stroudsburg, PA: Association for Computational Linguistics.

[27] Du L, Li X and Yu G. New word detection based on an improved PMI algorithm for enhancing Chinese segmentation system. *Acta Sci Natur Univ Pekinensis* 2016; 52: 35–40.

[28] Huang JH and Powers D. Chinese word segmentation based on contextual entropy. In: *Proceedings of the 17th Asian Pacific conference on language, information and computation*, Sentosa, Singapore, 1–3 October 2003, pp. 152–158.

[29] Grouin C, Lavergne T and Névéol A. Optimizing annotation efforts to build reliable annotated corpora for training statistical models. In: *Proceedings of the 8th linguistic annotation workshop*, Dublin, 23–24 August 2014, pp. 54–58. Stroudsburg, PA: Association for Computational Linguistics.

[30] Xu H, Yuan H, Ma B, et al. Where to go and what to play: towards summarizing popular information from massive tourism blogs. *J Inf Sci* 2015; 41: 830–854.

[31] Stavrianou A, Andritsos P and Nicoloyannis N. Overview and semantic issues of text mining. *ACM Sigmod Rec* 2007; 36: 23–34.

[32] Bengio Y, Ducharme R, Vincent P, et al. A neural probabilistic language model. *J Mach Learn Res* 2003; 3: 1137–1155.

[33] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space, 2003, https://arxiv.org/abs/1301.3781

[34] Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality. In: *Proceedings of the 26th international conference on neural information processing systems (NIPS'2013)*, Lake Tahoe, NV, 5–10 December 2013, pp. 3111–3119. Red Hood, NY: Curran Associates.

[35] Deng K, Bol PK, Li KJ, et al. On the unsupervised analysis of domain-specific Chinese texts. *Proc Nat Acad Sci U S A* 2016; 113: 6154–6159.

[36] Finley GP, Farmer S and Pakhomov SV. What analogies reveal about word vectors and their compositionality. In: *Proceedings of the 6th joint conference on lexical and computational semantics (*SEM 2017)*, Vancouver, BC, Canada, 3–4 August 2017, pp. 1–11. Stroudsburg, PA: Association for Computational Linguistics.

[37] Agrawal R, Imieliński T and Swami A. Mining association rules between sets of items in large databases. In: *Proceedings of the ACM SIGMOD international conference on management of data (SIGMOD'1993)*, Washington, DC, 25–28 May 1993, pp. 207–216. New York: ACM.

[38] Wang J, Han J, Lu Y, et al. TFP: an efficient algorithm for mining top-k frequent closed itemsets. *IEEE T Knowl Data En* 2005; 17: 652–663.

[39] Han J and Pei MK. *Data mining: concepts and techniques*. 3rd ed. San Francisco, CA: Morgan Kaufmann Publishers, 2011.

[40] Liu B. *Web data mining: exploring hyperlinks, contents, and usage data*. 2nd ed. Berlin; Heidelberg: Springer, 2011.

[41] Zeng D, Wei D, Chau M, et al. Domain-specific Chinese word segmentation using suffix tree and mutual information. *Inf Syst Front* 2011; 13: 115–125.

[42] Sornlertlamvanich V, Potipiti T and Charoenporn T. Automatic corpus-based Thai word extraction with the C4.5 learning algorithm. In: *Proceedings of the 18th conference on computational linguistics (COLING'2000)*, Saarbrucken, 31 July–4 August 2000, pp. 802–807. Stroudsburg, PA: Association for Computational Linguistics.

[43] Mei L, Huang H, Wei X, et al. A novel unsupervised method for new word extraction. *Sci China Inf Sci* 2016; 59: 92102.

[44] Chen A and Sun M. Domain-specific new words detection in Chinese. In: *Proceedings of the 6th joint conference on lexical and computational semantics (*SEM 2017)*, Vancouver, BC, Canada, 3–4 August 2017, pp. 44–53. Stroudsburg, PA: Association for Computational Linguistics.

[45] Han ALF, Wong DF and Chao LS. Chinese named entity recognition with conditional random fields in the light of Chinese characteristics. In: Kłopotek MA, Koronacki J and Marciniak M, et al. (eds) *Language processing and intelligent information systems*. Berlin: Springer, 2013, pp. 57–68.

[46] Baroni M, Dinu G and 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*, Baltimore, MD, 23–25 June 2014, pp. 238–247. Stroudsburg, PA: Association for Computational Linguistics.