Improving Online Clustering of Chinese Technology Web News With Bag-of-Near-Synonyms

ZHE ZHANG\textsuperscript{1}, LE CHEN\textsuperscript{2}, FENGJING YIN\textsuperscript{1}, XIN ZHANG\textsuperscript{1}, AND LIXIANG GUO\textsuperscript{1}

\textsuperscript{1}Science and Technology on Information Systems Engineering Laboratory, College of Systems Engineering, National University of Defense Technology, Changsha 410073, China
\textsuperscript{2}Institute of Systems Engineering, Academy of Military Science, Beijing 100091, China

Corresponding author: Fengjing Yin (yinfengjing@nudt.edu.cn)

\begin{abstract}
In the Internet era, online clustering of technology web news can help discover scientific breakthroughs and grasp technology trends. To do that automatically, the news documents to be clustered must be represented appropriately with numerical vectors. However, traditional representations such as Term Frequency-Inverse Document Frequency (TF-IDF) cannot distinguish near-synonyms and may cause “dimension disaster.” To overcome these problems, this article proposes the Bag-of-Near-Synonyms (BoNS) model based on the idea to construct near-synonym sets using word embeddings and agglomerative clustering, and then to represent a document with a Set Frequency-Inverse Document Frequency (SF-IDF) vector in which each dimension corresponds to a near-synonym set rather than a single word. To speed up computation, we further propose the hashed version of SF-IDF and name it hSF-IDF, which employs a hash function to map each near-synonym set to a unique number as the key and hence reduces the computation of SF to linear time. In addition, we apply hSF-IDF to online clustering of Chinese technology web news and propose an improved batch-based method. Extensive experiments have been conducted on a real-world dataset. The results show that our model outperforms some strong baselines including TF-IDF, average pooling of word or character embeddings, Latent Dirichlet Allocation (LDA), and bag-of-concepts in terms of both accuracy and efficiency.
\end{abstract}

\begin{index_terms}
Information retrieval, single-pass algorithm, document representation, word embeddings, agglomerative clustering.
\end{index_terms}

\section{Introduction}

Technology web news often reports latest inventions and the launches of new products. Analyzing them can help discover scientific breakthroughs and grasp technology trends. To mine web news, online clustering techniques are widely used in existing approaches, e.g., [1]–[3], due to two facts: (1) web news is a kind of streaming data, and (2) it covers various domains and often involves emerging things, which makes it difficult, if not impossible, to pre-specify the required features like keywords or to learn them from given data samples.

Online clustering of news streams is typically conducted in an incremental manner. For each incoming news document, online clustering consists of three main steps [4]. Firstly, each document is considered as a cluster and is encoded into a vector as its numerical representation. Secondly, the similarity between the current cluster and each of the existing ones is computed. Finally the document is merged into the most similar cluster if the corresponding similarity is larger than a pre-set threshold; otherwise, a new cluster is created. In such a process, document representation is the most important part, as it determines the definition of similarity, and hence impacts notably on the clustering results. Term Frequency-Inverse Document Frequency (TF-IDF) is the mainstream representation used in existing approaches, e.g., [5]–[7]. Its key idea is to encode a document using the occurrence frequencies of a series of representative words, which are commonly selected from a large corpus and organized into a vocabulary.

However, online clustering methods using TF-IDF always suffer from two main limitations. The first is that documents reporting the same event would be grouped into different
clusters if they use many synonyms, due to the fact that TF-IDF defines a different dimension for each word in the vocabulary while ignoring their relatedness. For example, the sentences “AI changes the world.” and “Artificial Intelligence changes the globe.” express the same meaning but the similarity between their TF-IDF vectors is low because of the synonymous pairs “AI” vs. “Artificial Intelligence” and “the world” vs. “the globe.” The second limitation is that large vocabularies are often needed to reduce out of vocabulary (OOV) words, which, nevertheless, would cause huge burden to the representation computation, and thus make it difficult to perform clustering in real time. These limitations would be magnified for technology news documents, because they often report or mention emerging things which have no standard names yet and hence are usually referred to using different terms in different documents. The situation becomes even worse for Chinese technology news documents, as many of them are translated from other languages – without a unified standard, the translations provided by different news agencies usually vary greatly. As exemplified in Table 1, some reports refer to “Facebook” as “脸书” or “脸谱” in Chinese, while others directly use its full English name or the acronym “FB.”

TABLE 1. The sample in actual news where the synonyms or near-synonyms make the news confusing.

| No. | News title |
|-----|------------|
| 1   | Facebook 未来将将注隐私保护 |
|     | 脸书深陷丑闻 小扎称未来重点关注隐私保护 |
|     | FB历经丑闻,扎克伯格势必在隐私保护上投入更多的精力  |
|     | (Zuckerberg: Facebook will pay more attention to privacy protection) |
| 2   | “科创板”成为人工智能公司角逐的新战场 |
|     | 多家AI公司纷纷“科创板”群雄逐鹿 一触即发 |
|     | (“SSESTARMarket” becomes the new “battleground” for the Artificial Intelligence companies) |
| 3   | “人人生”千余员工被“炒鱿鱼”因网贷资金链断裂 |
|     | 资金链断裂,“人人生”裁员上千名员工 |
|     | (“Renrenche” lays off thousands of employees because of the lack of money) |

To overcome the above limitations, this article proposes a new document representation named Bag-of-Near-Synonyms (BoNS) to improve online clustering of Chinese technology web news. The core idea is to construct near-synonym sets using word embeddings and agglomerative clustering, and then to represent a document with a Set Frequency-Inverse Document Frequency (SF-IDF) vector in which each dimension corresponds to a near-synonym set rather than a single word. In this way, documents reporting the same event but using near-synonyms can be grouped into the same cluster. We do not use “Wordnet” (or its Chinese version named “Cilin”) because these dictionaries mainly include common words rather than the words from the technology domain. For example, “Facebook” and “Zuckerberg” are not included into them. Moreover, the updating frequency of these dictionaries is pretty low, which results in that many emerging words could not be included. While, BoNS constructs near-synonym sets from historical technology news documents, and thereby can select terms, entities and proper names commonly used in this domain. In addition, to speed up the computation of document representations, we propose the hashed SF-IDF (hSF-IDF), which employs a hash function to map each near-synonym set to a unique number as the key and hence reduces the computation of set frequency (SF) to linear time.

In summary, the main contributions of this article are as follows:

- We present a new document representation named BoNS, which uses an SF-IDF vector to encode a document, and each dimension of the vector corresponds to a near-synonym set obtained through agglomerative clustering, but not to a single word. In this way, it overcomes the limitations of TF-IDF and hence can obtain higher-quality clustering results.

- We further propose a hashed version of SF-IDF and name it hSF-IDF, which utilizes a hash function to map each near-synonym set to a unique number as the key and uses it as a prefix for each in-set word. When computing SF, we search the hash table of near-synonym sets to match the prefix. Therefore, it only takes linear time to determine which set a word belongs to and the computation of document representation can be considerably accelerated.

- We apply the proposed BoNS model to online clustering of Chinese technology web news and propose an improved batch-based method. Extensive experiments have been conducted on a real-world dataset. The results show that our model is superior to some strong baselines including TF-IDF, average pooling of word or character embeddings, Latent Dirichlet Allocation(LDA), and bag-of-concepts in terms of both accuracy and efficiency.

The remainder of this article is organized as follows. Section 2 gives a review of related work. Section 3 describes the BoNS model. The details of experiments are presented in Section 4, followed by the results and analyses of experiments given in Section 5. We conclude this article and discuss possible future work in Section 6.

II. RELATED WORK

Two broad classes of related work, viz. topic detection and tracking (TDT) and document representation, are briefly surveyed in the following.

A. TOPIC DETECTION AND TRACKING

TDT is proposed in 1996 jointly by DARPA, CMU, and so on, which is intended to explore techniques for detecting the appearance of new topics and evolution of them [8].
James Allan defines five tasks of TDT: Story Segmentation, Topic Detection, Topic Tracking, First Story Detection, and Story Link Detection [9]. Among these tasks, Topic Detection and Topic Tracking relate most closely to online clustering of news streams, and are often approached to using online single-pass [10]–[14] or Online Latent Dirichlet Allocation (OLDA) [15]–[17]. The former splits the incoming news stream into batches, employs TF-IDF to represent each document, performs single-pass clustering in each batch, and finally merges similar clusters across neighboring batches. OLDA also works in a batch-based manner, but employs LDA instead to conduct intra-batch clustering. Specifically, it trains a new LDA model in each batch and uses it as a prior one for the training in successive batch. In addition, it keeps adjusting the learned topics through sampling words in the documents in a new batch according to the topic distribution of the current model. However, LDA models topics to be multinomial distributions over words and documents to be mixtures sampled from topics, and hence is very complex. In contrast, TF-IDF represents a document using the frequency distribution of a set of representative words, which is much simpler but often achieves comparable or even better clustering quality, and thus becomes the most commonly used one in TDT. Later researches in TDT present various improvements for single-pass algorithm. Yin et al. [10] propose the Incremental Algorithm for Clustering Internet Texts (ICIT) algorithm that chooses only nouns and verbs to construct the vocabulary, and uses average-link as the linkage mechanism to enhance clustering accuracy. Xiaolin et al. [11] introduce “topic seed” to simplify the computation of similarity. Zhe et al. [12] employ sliding time window to weaken the dependence of clustering results on the input order of the documents. Yan et al. [13] represent documents with incremental TF-IWF-IDF, which is defined as the product of TF-IDF and the inverse word frequency (IWF) computed over the current time slice. Huang et al. [14] replace the TF-IDF based representation commonly used in single-pass with LDA. Specifically, they train an LDA model in an offline process and represent each document to be clustered using a vector encoding the topic distribution, in which topics are further encoded as the corresponding distributions of a set of representative words. To summarize, some of these improvements, e.g., [10]–[12], try to simplify the single-pass clustering process, and others, e.g., [13], [14], attempt to modify the documents representation. However, all of them still represent document based on term distributions which define a distinct dimension for each word and hence could not deal with synonyms or near-synonyms. Different from them, the BoNS model proposed in this article uses the distribution over near-synonym sets rather than individual words.

B. DOCUMENT REPRESENTATION

As aforementioned, document representation is very important for online clustering. Thus, we review some typical document representation models in the following.

1) BAG-OF-WORDS (BoW)

This model is a simplified document representation commonly used in Natural language processing (NLP) and Information retrieval (IR), in which a document is represented using the normalized term frequency (TF). TF-IDF, as an improved version of BoW, multiplies TF by IDF of words to get rid of the influence of stop words which appear frequently but have little value for differentiating documents. A common disadvantage of TF-IDF and BoW, as mentioned above, is that they define an individual dimension for each word, which makes the resulting vectors to be sparse and high-dimensional, and usually causes the documents reporting the same event but using many near-synonyms to have very different representation vectors that cannot be grouped into the same cluster.

2) TOPIC MODELS

Topic models are based upon the idea that documents are mixtures of topics, where a topic is a probability distribution over terms [18]. Generally, they map documents from a large matrix of term-document into a low-dimension latent semantic space, and then represent documents based on this latent space. Typical models include Latent semantic analysis (LSA), Probabilistic latent semantic analysis (pLSA) and LDA. They differ from each other mainly in how to get the latent semantic space. LSA, also called Latent semantic indexing (LSI) [19], uses Singular-value decomposition (SVD) on the large term-document matrix to decompose it into a latent space, of which each dimension can be regarded as a latent topic. pLSA is proposed to improve LSA, which uses a probabilistic model rather than SVD to get latent semantic space [20]. It introduces a latent variable and utilizes EM algorithm to estimate the parameters. Blei et al. [21] extend pLSA to LDA by introducing a Dirichlet prior. Specifically, in LDA, it is assumed that there are k underlying latent topics according to which documents are generated, and each topic is represented as a multinomial distribution over terms. Documents are modeled by sampling a mixture of topics via a hidden Dirichlet random variable that specifies a probability distribution on a latent, low-dimensional topic space. Compared with pLSA, LDA is less likely to overfit. However, all these topic models are based on term distribution, and hence suffer from the common limitation of high computational complexity.

3) DISTRIBUTED REPRESENTATIONS

This family of models is based on the distributed hypothesis that the words occurring in a similar context tend to have similar meanings [22]. Different from BoW, they map words or characters into low-dimensional and dense vector spaces using the weight matrices learned in course of training language models. The resulting word or character vectors are called embeddings. Word2vec proposed by Mikolov et al. [23] in 2013 is a seminal one which enjoys the widest application. Various techniques have been studied to
derive vector representations of documents from the embeddings of the words or characters they contain. Among them, the simplest is to use weighted average of word embeddings, also called Word Average Pooling (WAP). Some others such as weighting different words [24] and doc2vec [25] are popular as well both in academia and industry.

However, there are still some limitations with them worth noting. WAP and the like employ the weighted average of word or character embeddings to represent a document, which is essentially a point in the corresponding word or character space, and hence ignores the distribution of words or characters, leading to less representative document vectors. Others which directly map the documents into some embedding space, e.g., doc2vec and the like, are not suitable for streaming data. Specifically, if we use such models trained in an offline process, many streaming documents cannot be looked up in the vocabulary of the document embeddings and thus become OOV ones that could not be represented as numerical vectors. Also, it is difficult to train those models online because the training processes cost pretty huge time, which would violate the real time principle of online clustering.

4) BAG-OF-CONCEPTS (BoC)
This model is proposed recently by Kim et al. [26] and aims to exploit the relative merits of word2vec and BoW. Specifically, the word embeddings pre-trained using word2vec are grouped first by k-means clustering to construct the dictionary of “concepts.” Then, similar to TF-IDF, the model represents each document using the resulting concepts by defining the Concept Frequency-Inverse Document Frequency (CF-IDF) as the required document representation. In such way, it inherits the low dimensionality of word2vec and the interpretability of BoW, and thus can acquire explicit understanding the content of documents. It is noteworthy that BoC is originally designed for document classification. The authors train the model on a static corpus, represent the documents in the corpus with the resulting concepts, and then classify them into some predefined categories. However, for online clustering, we need to train the concepts offline and then use it in the online process, which is quite different from text classification. In addition, there are still two main problems with BoC: First, the clustering algorithm used in BoC is k-means which needs us to pre-specify the concept number “k” appropriately – if “k” is too small, plenty of noise (the words belonging to other concepts) will be introduced into each of the resulting concepts; while if it is too large, one concept would be subdivided into several clusters. Since we often have no knowledge about how many concepts contained in the input documents before processing them, it is apt to set an inappropriate value for k, and thus hurt the performance of document representations. Second, the computation of CF-IDF in BoC is through counting and adding word frequency with respect to each “concept”, which involves matching each word in the document to be represented to all concepts, together with the words they contained, and thereby costs considerable time. These problems have not been addressed by Kim et al. in [26], as they focus on classifying documents in a static corpus, but are of special importance to the online clustering of continuously incoming and often huge quantity of web news documents.

In comparison, the BoNS model proposed in this article similarly combines word embeddings and clustering but differs notably from BoC in three aspects: First, our model employs agglomerative clustering instead of k-means to discover the near-synonym sets contained in the given corpus, and thus does not need to pre-set “k.” It groups words with a threshold of the similarity between words, which are far easier to pre-specify. Second, we propose the hSF-IDF which uses a hash function to accelerate the computation of the SF-IDF vector of a given document, and is more appropriate for online clustering where efficiency, besides accuracy, is an important goal we need to achieve. Third, BoNS is originally designed for and can be readily integrated with online clustering – we construct near-synonym sets in an offline process and use them to represent streaming documents in online process. The experiments confirm that our model outperforms BoC substantially in terms of both accuracy and efficiency.

III. METHODOLOGY
As illustrated in Fig.1, the framework proposed in this article for online clustering of technology web news with BoNS consists of two main parts, i.e., near-synonym sets construction and online clustering. The former is an offline process. It firstly selects representative words from a large corpus of history news. Then it retrieves the embeddings of representative words from a pre-trained word embeddings vocabulary, and finally constructs the required near-synonym sets via agglomerative clustering. The latter process implements online clustering by splitting incoming news documents into batches in chronological order and then grouping them into clusters in a batch-based manner. Specifically, it first computes the representation vector of each incoming document using the near-synonym sets generated in the offline process, then groups the vectors within each batch using the single-pass algorithm, and finally merges similar clusters in neighboring batches.

It is worth mentioning that, different from English, Chinese texts have no delimiter between words, and often contain more stop words. Therefore, some preprocessing is specially required before further analysis. Specifically, in the proposed framework, all input documents must undergo the following preprocessing (which is not depicted in Fig.1 for simplicity) before entering the module of near-synonym sets construction or online clustering. First, the irrelevant contents (such as meaningless English letters and external links) are filtered with regular expressions. Second, word segmentation is conducted with some NLP toolkit such as “jieba”. Finally stop words are removed according to a dictionary collected from the web.

\[1\text{https://github.com/fxsjy/jieba}\]

\[2\text{https://github.com/goto456/stopwords}\]
A. NEAR-SYNONYM SETS CONSTRUCTION

The core idea underlying this part is that, since two words whose representation vectors are close to each other in an embedding space, e.g., that trained using word2vec, tend to have similar meanings, we can obtain sets of near-synonyms through clustering such word embedding vectors. As shown in Fig.1, this may be implemented in three main steps: The first is to select representative words from a large corpus of historical technology web news documents. Observed that technology news topics differ from each other mainly in the verbs and nouns they contain, we perform POS (Part-of-Speeches) filtering to retain only verbs and nouns as candidates. Traditionally, highly-frequent words are often chosen as the representatives, but, in this way, plenty of stop words lacking representativeness may be included. To circumvent this, we rank the candidates in descending order of their LTU weights\(^3\) [27], [28], and then choose highly-weighted ones in the rank list to construct the vocabulary required. LTU weight is a variant of TF-IDF, both of which are methods for term weighting. It introduces document length in the denominator to counteract the factor that the frequencies of the terms will be higher in longer documents. Formally, the LTU weight of word \(i\) with respect to document \(j\) is defined as

\[
\text{LTU}_{ij} = \frac{(\log(f_{ij}) + 1.0) \times \log(N)}{0.8 + 0.2 \times \frac{d_j}{\text{avg.\_dl}}}.
\]

where \(f_{ij}\) is the normalized frequency for word \(i\) in document \(j\), \(n_i\) is the number of document containing word \(i\), \(N\) is the total number of documents in the given dataset, \(\text{avg.\_dl}\) is the averaged document length in the given dataset, \(d_j\) is the length of document \(j\), and \(\frac{d_j}{\text{avg.\_dl}}\) acts as the penalty term of document length. \(\text{LTU}_{ij}\) is defined as the maximal LTU weight over all documents containing word \(i\) and is regarded as the representativeness metric for word \(i\). The formal definition is

\[
\text{LTU}_{i} = \max\{\text{LTU}_{ij_1}, \text{LTU}_{ij_2}, \ldots, \text{LTU}_{ijn}\}. \tag{2}
\]

The second step is to retrieve the embeddings vector of each representative word from the vocabulary of pre-trained word embeddings while discarding the OOV ones. The final step is to group these embedding vectors into clusters, called near-synonym sets, through an agglomerative clustering module. Traditional algorithms [29] start with each data object as a separate cluster, and then merge repeatedly the closest two until only one cluster is left or some other termination condition is satisfied. They are commonly very simple but suffer too much from the poor efficiency, especially when the amount of data becomes large, as they need to evaluate the similarity between each pair of clusters and only choose the closest one in every iteration. To overcome this problem, we utilize the improved hierarchical agglomerative clustering [30], which combines random sampling with partitioning and uses the centroid distance as the linkage mechanism. Specifically, a random sample drawn from all words is first partitioned, and then each partition is partially clustered, after that the partial clusters are then again clustered in a second pass to yield the final results. Cosine distance is used to measure the similarity between the normalized word embedding vectors, which is defined as

\[
\cos(w_i, w_j) = w_{i1}w_{j1} + w_{i2}w_{j2} + \ldots + w_{in}w_{jn} \tag{3}
\]

where \(w_i\) and \(w_j\) are the normalized embedding vectors of words \(i\) and \(j\), \(w_{i1}, \ldots, w_{in}\) are the elements of \(w_i\) and \(w_{j1}, \ldots, w_{jn}\) are those of \(w_j\). In addition, two criteria are used in combination to determine whether or not the closest pair of words (or clusters) should be merged: if the cosine distance between the two words is lower than a pre-defined threshold \(\delta_1\), they are not similar enough to be near-synonyms and
thus will not be merged; otherwise, they are assigned to the same (new) group only if they have the same part-of-speech. Some sample results of the agglomerative clustering are given in Fig. 2, which shows that the proposed algorithm is effective for finding sets of near-synonyms.

Note that, we construct near-synonym sets using historical technology news documents rather than some defined topics due to the fact that news topics are constantly evolving, in particular that novel ones keep emerging. While, for a given domain, e.g., the technology domain targeted in this article, some terms, entities and proper names commonly used remain relatively steady.

B. ONLINE CLUSTERING

As aforementioned, online clustering is implemented in a batch-based manner, in which a batch is defined as a fixed number of news documents published within a continuous time span. As shown in Fig.1, the workflow consists of three modules, i.e., document representation computing, intra-batch clustering, and cross-batch merging, each of which is detailed in the following. It should be noted that, before computing the representation, each incoming document must undergo some pre-processing similar to that has been conducted on the history news (see the start of section III for details).

1) DOCUMENT REPRESENTATION COMPUTING

In this module, we use the near-synonym sets generated in section III-A to represent the documents to be clustered. Specifically, we define SF-IDF as the quantitative representation for each incoming document, in which the SF of near-synonym set \( s_i \) with respect to document \( d_j \) is

\[
SF(s_i, d_j) = \frac{|w \in (s_i \cap d_j)|}{|w \in d_j|},
\]

(4)

where the numerator \( |w \in (s_i \cap d_j)| \) on the right is the number of words in \( s_i \) that appears in \( d_j \) and the denominator \( |w \in d_j| \) is the total number of all words in \( d_j \). Since near-synonym sets that occur frequently in many documents are not good for discriminating these documents, the IDF is further introduced to filtered them out, which is defined as

\[
IDF_{ij} = \log \frac{|D| \times n}{|d \in D; s_i \in d|},
\]

(5)

where \( |D| \) is the number of documents in the batch under consideration, \( n \) is the number of words in \( s_i \), and \( |d \in D; s_i \in d| \) is the number of documents in this batch that contain words in \( s_i \). It is different from traditional IDF in the definition of the numerator, which is defined to be \( |D| \) in traditional IDF. In our definition, we count the document frequencies of the words within \( s_i \), which appear within the current batch, and then add them up as the denominator. Thus, \( |D| \) is needed to be further multiplied by \( n \), otherwise the denominator could be larger than the numerator and hence yields a negative IDF.

As a matter of fact, the time complexity of computing SF defined in (4) is \( O(m\log l) \) where \( m \) denotes the length of the vocabulary and \( l \) is the maximum number of words contained in the sets included in the vocabulary. Thus, it often takes much more time to compute the SF-IDF vector of a document, compared with the computation of TF-IDF, especially when some near-synonym sets contain a lot of words, namely \( l \) is large. To speed up computation, we extend SF-IDF to hSF-IDF with a hash function, which maps each near-synonym set to a unique number as the key, and then uses it as a prefix for each in-set word. Hence the code of each word consists of two parts: a prefix that is the key of the near-synonym set the word belongs to, and its local id in the set. Finally we use a hash table \( T \) to store the prefixes of all selected words. Note that these operations are performed offline, and thus nearly do not extend the running time of online clustering. When computing the SF of each word in an incoming document, we search \( T \) to get its prefix, and then search the hash table of near-synonym sets to match the prefix so that we can determine which set the word belongs to. As observed in our experiments, the time to execute the hash function in online clustering is so little that it can be ignored when evaluating the time complexity. Hence, the time complexity of computing SF is reduced to be in linear time. Furthermore, to deal with potential hash collisions, the pseudo-random probing strategy is adopted. The hash code of \( s_i \) is computed as follows,

\[
H_i = (H(\cdot) + r_j) \mod \text{length}(s_i),
\]

(6)

where \( H_i \) is the hash code of \( s_i \), \( H(\cdot) \) is the hash function and \( r_j \) is the \( j \)-th number in a random sequence \( RS \). When the hash collision happens, a pseudo-random sequence will be generated to update the hash code.

2) INTRA-BATCH CLUSTERING

The flowchart of this module is given in Fig.3. As shown, it is based on single-pass algorithm. To be more specific, the cluster set \( C \) of a given batch is initialized to be an empty one, and then the documents in the batch are clustered in a sequential manner: The module loads the representation
vector of an unprocessed news document \( \alpha \) in the current batch, and evaluates the cosine distances, similar to that defined in (3), between (the representation vectors of) \( \alpha \) and the centers of the clusters in \( C \). The closest cluster \( c_1 \) is selected and the corresponding cosine distance \( \theta \) is compared with a pre-specified threshold \( \delta_2 \). If \( \theta \) is larger than \( \delta_2 \), \( \alpha \) will be merged into \( c_1 \) and its center vector will be updated; otherwise, a one-document cluster will be created and included into \( C \). The process is repeated until all documents in the batch are clustered.

3) CROSS-BATCH MERGING
As the last step in online clustering, we merge similar clusters across neighboring batches, which also works on-line. First, we compute the cosine distance between the center vectors of each pair of clusters across two neighboring batches, i.e., the current one and its immediate predecessor. If the distance is larger than a pre-specified threshold, then the two component clusters are merged into a single one. This is repeated until all such pairs are compared. Note that, since each batch is defined to contain only a limited number of documents (to avoid introducing large lag into online clustering), we choose to use \( \delta_2 \) here for merging decision, which is the same to that used in intra-batch clustering.

IV. EXPERIMENTAL SETUP
We evaluate our BoNS model using a real-world dataset through a set of carefully designed experiments. This section details the dataset, evaluation metrics, baselines and implementations of these experiments.

A. DATASET
Currently, there is no dataset publicly available for online clustering of Chinese technology web news. Therefore, to evaluate the proposed model, we construct a real-world dataset by harvesting news documents from web portals. The documents are all crawled from the technology channels of some famous Chinese web portals such as NetEase Technology, Sina Science and Technology, Tencent Science and Technology between February 8 and March 11, 2019. In such way, we obtain a total of 2,320 news documents, which are then filtered, processed and annotated manually by three annotators in this field. As a result, a dataset containing 436 documents and 19 annotated topics is finally generated. Among these documents, the longest one contains more than 5,000 words while the shortest only contains 48 words. The topics and the number of documents they contain are given in Table 2.

To train word embeddings and select representative words, we construct another corpus of historical technology news by crawling webpages from the same portals and channels mentioned above. After de-duplication and filtering irrelevant content like external links, the collected documents amount to a total of 58,664, and are all published between January 11, 2018 and February 7, 2019, which means that the resulting corpus has no overlap with the one constructed above for clustering evaluation. These documents are used to select representative words. For embedding training, we further mix them with the Chinese wiki corpus to obtain an even larger one.

B. EVALUATION METRICS
Since what we focus on is the results of online clustering of web news, we use the most common metrics in IR for evaluating clustering, i.e., precision (P), recall (R) and F1-score (F1). They are defined, respectively, as

\[
P = \frac{TP}{TP + FP},
\]

\[
R = \frac{TP}{TP + FN},
\]

\[
F1 = \frac{2PR}{P + R},
\]

where \( TP \) refers to True Positives, \( FP \) stands for False Positives, \( FN \) is False Negatives, and \( TN \) refers to True Negatives.

\[4\] https://tech.163.com/

\[5\] https://tech.sina.com.cn/

\[6\] https://new.qq.com/ch/tech/

\[7\] https://github.com/mattzheng/ChineseWiki
TABLE 2. The topics of the dataset.

| No. | Topic | Count |
|-----|-------|-------|
| 1   | 熊猫直播破产 (PandaTV went broke) | 28 |
| 2   | 苏宁收购万达百货 (Suning E-commerce merged Wanda Department Store) | 26 |
| 3   | 美国枪支监管法案 (The extradition of Wanzhou Meng acquired by America government) | 19 |
| 4   | 网约车司机（Online nurse service spring up） | 21 |
| 5   | 科技板设立 (SSSESTARMarket was set up) | 38 |
| 6   | “锤子科技”破产 (Smartisan Company went broke) | 26 |
| 7   | Facebook数据泄露 (Data leakage of Facebook) | 21 |
| 8   | 星巴克黄爪杯爆款 (The hunger marketing of the Starbucks cat-claw mug) | 10 |
| 9   | 斗鱼上市 (DouYuTV applied for listing) | 13 |
| 10  | 故宫夜场开放 (The Palace Museum open the evening show) | 22 |
| 11  | 小米9发布 (Mi 9 launched) | 28 |
| 12  | 埃航波音737空难 (The air crash of Boeing737) | 12 |
| 13  | “咪蒙”被封禁 ("MiMeng" was banned in social network) | 20 |
| 14  | 华为起诉美国政府 (Huawei prosecuted American government) | 22 |
| 15  | “人人车”裁员风波 (Layoffs of "Reenrenche") | 34 |
| 16  | 亚马逊停止纽约总部建设 (Amazon stopped the headquarters in New York) | 26 |
| 17  | 《流浪地球》火爆 (The Wandering Earth get a box office hit) | 29 |
| 18  | “龙”飞船试飞成功 (The success of the SpaceX’s Crew Dragon) | 16 |
| 19  | P2P爆雷 (P2P——network loan station is in chaos) | 25 |
|     | Total | 436 |

Furthermore, online clustering is highly related to TDT; so we also use the metrics established by National Institute of Standards and Technology (NIST) [31] for TDT evaluation in the experiments, which measure the performance of clustering using the false alarm ($P_{false}$), missed detection ($P_{miss}$), and the detection cost function ($C_{cost}$) [32]. They are defined as

$$P_{miss} = \frac{FN}{TP + FN},$$

$$P_{false} = \frac{FP}{FP + TN},$$

$$C_{cost} = C_{miss} \times P_{miss} \times P_t + C_{false} \times P_{false} \times (1 - P_t),$$

where $C_{miss}$ is weighting coefficient called missing cost, $C_{false}$ is another weighting coefficient called false positive cost, $P_t$ is the prior probability of topic occurrence. Generally, $C_{miss}$ is much larger than $C_{false}$, indicating that miss rate is more significant than false positive rate.

C. BASELINES

We apply several typical document representation models to online single-pass and compare the F1, $C_{cost}$, and time consumption of these methods with those of our proposed model. The baselines are as follows:

- BoC: the representation model proposed by Kim et al. [26], which is slightly adapted to fit the online clustering scenario by training a BoC model on each batch of documents and then represent them using the corresponding CF-IDF vectors.
- LDA: the representation used in [14], which trains LDA model in an offline process to represent the documents with topic distributions.
- WAP: the simplest document representation based on word2vec [24], which is the weighted average of all the embeddings of the words contained in the given document.
- TF-IDF: the most popular representation used in TDT [5], which computes the TF in a document and multiplies it by the IDF to obtain the document representation vector.
- BERT-based: representations utilizing BERT, which is a typical Transformer-based approach and has achieved great success in many NLP tasks [33]. In our experiments, we employ two BERT-based representations inspired by [34]. The first one uses word level embeddings in a manner similar to WAP. The second one uses sentence level embeddings by encoding each sentence into an embedding vector, and then computing the average of sentence vectors as the document vector.

D. IMPLEMENTATION DETAILS

All experiments are performed in Python3.6 on a computer with a 2.10GHz Intel(R) Xeon(R) CPU and 64GB of memory. When running time is measured, the test is repeated over 5 times and the average is reported. Word segmentation in experiments is conducted with the toolkit “jieba.” Word2vec is used to train word embeddings as it is simple and also used in the experiments of BoC in [26], Word2vec and LDA are trained using Gensim library, and BoC is trained using the toolkit provided by Kim et al. [26]. The agglomerative clustering in sklear library is adapted for our experiments. Two version of BERT models [35] are used in our experiments, namely BERT-base (12-layer, 768-hidden, 12-heads) and BERT-large (24-layer, 1024-hidden, 16-heads).

Parameter settings of the experiments are given in Table 3. Some parameters including $C_{miss}$, $C_{false}$, and $P_t$ are set according to the previous standards established in TDT2004 [31]. As for the rest, $\delta_1$ is set according to some preliminary experiments which show that the number of near-synonym sets gets small when $\delta_1$ is too large and the noisy ones will become more when $\delta_1$ is too small.

8 https://radimrehurek.com/gensim/  
9 https://github.com/hank110/bagofconcepts  
10 https://scikit-learn.org/  
11 https://github.com/google-research/bert
We set \textit{batch size} as 100 because our dataset size is 436, and with such \textit{batch size} the clustering will be more efficient. \textit{Word embedding size} for BoNS and BoC are both set from 100 to 500 with the step-size being 100, which is in agreement with the settings in [26]. Considering that the vocabulary of near-synonym sets will not make sense if its size is too small, we vary, in the experiments, the \textit{Document vector dimension} from 1000 up to 3500 with step 500. In fact, the dimension of document vectors means differently for each document representation – for BoNS and TF-IDF, it is the size of the vocabulary, for BoC it equals the pre-specified “k” of k-means clustering, viz., the number of “concepts”, and for LDA and WAP, it equals the dimension of the hyper-parameters when training the model.

The first and an essential step in clustering is to represent documents with our BoNS model and the baseline ones. As for BoNS and BoC, we encode words into numerical vectors with different \textit{Word embedding size} in Table 3 to construct near-synonym sets and concepts, and compute the hSF-IDF and CF-IDF vectors of documents under consideration. TF-IDF also needs to construct a vocabulary of representative words and then compute TF-IDF vectors as the document representations, but it does not use word embeddings. WAP and LDA are trained using \textit{Gensim library} with different \textit{Document vector dimensions}. Then, we conduct single-pass clustering using document vectors of fixed dimension with different threshold \(\delta_2\) and select the best one as the result for that dimension. For example, in Fig.4 and Fig.5, the F1 and \(C_{\text{cost}}\) of LDA and TF-IDF with 2000 \textit{Document vector dimension} reach the best accuracy at different thresholds – 0.085 for LDA and 0.09 for TF-IDF. Finally, to comprehensively compare BoNS with WAP, TF-IDF, LDA, and BoC, we change the value of \textit{Document vector dimension} and conduct experiments in the same manner as above used. As for BoNS and BoC, we compute the average results of 100-500 \textit{Word embedding size} as the final.

We set \textit{batch size} as 100 because our dataset size is 436, and with such \textit{batch size} the clustering will be more efficient. \textit{Word embedding size} for BoNS and BoC are both set from 100 to 500 with the step-size being 100, which is in agreement with the settings in [26]. Considering that the vocabulary of near-synonym sets will not make sense if its size is too small, we vary, in the experiments, the \textit{Document vector dimension} from 1000 up to 3500 with step 500. In fact, the dimension of document vectors means differently for each document representation – for BoNS and TF-IDF, it is the size of the vocabulary, for BoC it equals the pre-specified “k” of k-means clustering, viz., the number of “concepts”, and for LDA and WAP, it equals the dimension of the hyper-parameters when training the model.

The first and an essential step in clustering is to represent documents with our BoNS model and the baseline ones. As for BoNS and BoC, we encode words into numerical vectors with different \textit{Word embedding size} in Table 3 to construct near-synonym sets and concepts, and compute the hSF-IDF and CF-IDF vectors of documents under consideration. TF-IDF also needs to construct a vocabulary of representative words and then compute TF-IDF vectors as the document representations, but it does not use word embeddings. WAP and LDA are trained using \textit{Gensim library} with different \textit{Document vector dimensions}. Then, we conduct single-pass clustering using document vectors of fixed dimension with different threshold \(\delta_2\) and select the best one as the result for that dimension. For example, in Fig.4 and Fig.5, the F1 and \(C_{\text{cost}}\) of LDA and TF-IDF with 2000 \textit{Document vector dimension} reach the best accuracy at different thresholds – 0.085 for LDA and 0.09 for TF-IDF. Finally, to comprehensively compare BoNS with WAP, TF-IDF, LDA, and BoC, we change the value of \textit{Document vector dimension} and conduct experiments in the same manner as above used. As for BoNS and BoC, we compute the average results of 100-500 \textit{Word embedding size} as the final.
Z. Zhang et al.: Improving Online Clustering of Chinese Technology Web News With BoNS

**FIGURE 6.** Per-batch time consumption of online clustering.

TABLE 5. The $C_{cost}(\times 10^{-3})$ with different dimensions of document vectors.

| Dimension | Method            | 1000  | 1500  | 2000  | 2500  | 3000  | 3500  |
|-----------|-------------------|-------|-------|-------|-------|-------|-------|
| 2500      | BoC+ single-pass  | 6.79  | 6.29  | 6.13  | 6.37  | 5.90  | 5.62  |
| 2500      | LDA+ single-pass  | 5.82  | 4.70  | 5.58  | 6.09  | 5.33  | 4.07  |
| 2500      | WAP+ single-pass  | 4.52  | 4.83  | 3.68  | 4.44  | 4.57  | 4.53  |
| 2500      | TF-IDF+ single-pass | 4.89 | 4.94  | 4.36  | 4.13  | 3.24  | 3.35  |
| 2500      | BoNS+ single-pass | 4.76  | 3.47  | 3.20  | 3.34  | 3.09  | 2.86  |

**TABLE 6.** The results of clustering using BoNS and BERT-based models.

|                      | F1 (%) | $C_{cost}(\times 10^{-3})$ | Running time (seconds per batch) |
|----------------------|--------|-----------------------------|----------------------------------|
| BoNS                 | 89.71  | 2.86                        | 0.22                             |
| BERT-base (word level) | 87.62  | 3.84                        | 4.2                              |
| BERT-base (sentence level) | 87.18  | 3.70                        | 0.39                             |
| BERT-large (word level) | 89.02  | 2.91                        | 20.82                            |
| BERT-large (sentence level) | 88.40  | 3.29                        | 1.78                             |

TF-IDF. Besides, it also has much lower $C_{cost}$: 47.38% than BoC, 37.16% than LDA, 29.82% than WAP, and 18.96% than TF-IDF. Obviously, the near-synonym sets vocabulary would have poor coverage when it is too short, namely there could be many OOV words, which may further lead the representation vectors of documents to be less different from each other. This is the reason why the clustering accuracy is the lowest when *Document vector dimension* is 1000. When the vocabulary becomes longer, its representativeness grows up and BoNS could be more accurate. However, longer representation vectors unnecessarily need more computation, and hence increase the overall running time. Besides, as aforementioned in Section II, no matter too big or too small, the inappropriate “k” of BoC always will hurt document representation and the results of online clustering. Thus, the performance of BoC is worst among these baselines.

In Table 6, it shows that the best performing BoNS (when *Document vector dimension* is 3500) is superior to all BERT-based representations with respect to F1 and $C_{cost}$. We think this is because BERT-based representations are essentially average pooling of word level or sentence level embeddings that put no emphasis on representative words. Besides, a very large corpus collected from Wiki is used to train BERT-base and BERT-large, which is an open domain corpus and hence may cause terms specific for the technology domain to be OOV ones.

**B. RUNNING TIME**

Our BoNS model also reduces the running time on the dataset. We record the time consumption of clustering the documents in the evaluation dataset constructed above using different document representations, and present the results in Fig.6(a). As an exception, the running times of clustering using BERT-based representations have not been plotted in this figure. This is mainly because the training of BERT often costs huge amount of computation and normally needs TPU [33]. Thus we use the BERT models trained and provided by [35], which have only two fixed dimensions, namely 784 of BERT-base and 1024 of BERT-large, and give the corresponding running times in Table 6. As shown in the table, the efficiency of BERT-base with sentence level embeddings is comparable to that of BoNS, but BERT-large and BERT-base with word level embeddings cost much more time than BoNS. With respect to the efficiency of the other representations, it is shown in Fig.6(a) that combining LDA, TF-IDF or BoNS (with hSF-IDF) with single-pass costs much less time than using WAP and BoC. We find the latter ones spend too much time on loading models and looking up the huge vocabulary. In addition, it is noteworthy that the curves of BoNS (hSF-IDF) and TF-IDF lie very close to each other.
TABLE 7. The news reports belonging to the topic “Smartisan Company went broke.”

| No. | news title                                                                 |
|-----|---------------------------------------------------------------------------|
| 1   | Yonghao Luo withdraws from the “liaotianbao” shareholder’s ranks (Wang Weicheng’s ultimate beneficiary) |
| 2   | (Bullet SMS is completely over? Yonghao Luo withdraws from the ranks of “liaotianbao” shareholders)          |
| 3   | (The “liaotianbao” (original bullet SMS) team disbanded, Yonghao Luo has withdrawn)                            |
| 4   | The “liaotianbao” team has been disbanded and it probably become the shortest-lived entrepreneurial product of Yonghao Luo |
| 5   | The “liaotianbao” team was dissolved and dismissed 150 million yuan a year ago                                  |
| 6   | (Liaotianbao) layoffs 85% Yonghao Luo’s next outlet is electronic cigarette                                        |
| 7   | (On the collapse of “liaotianbao” : born in the new, died in the retention)                                       |
| 8   | (Liaotianbao) layoffs 85% or Yonghao Luo use electronic cigarette equity pledge to compensate employees?          |
| 9   | (Yonghao Luo withdrew The bullet text message really turned into a bullet?)                                       |
| 10  | Yonghao Luo looked at himself in 7 years ago                                                                              |
| 11  | (Advantages Technology or will acquire hammer smooth breathing technology Extended ecological chain layout)        |
| 12  | (Yonghao Luo sells hammer air purifier business Jiangfeng Liu takes over)                                               |
| 13  | (Yonghao Luo sells hammer empty net business: former glory president Jiangfeng Liu will take over)              |

This reveals that the running times of BoNS and TF-IDF are very close and much lower than the other three. To have a closer look at the time consumption of them and that of the naive BoNS model (i.e., SF-ISF), we further plot their running time curves together in Fig.6(b). It is obvious from this figure that SF-IDF is the slowest as its computation is the most complex. However, after applying hash mapping, the time consumption of the improved version, i.e., hSF-IDF, decreases significantly and becomes comparable to and slightly better than that of TF-IDF, which proves that the proposed improvement can reduce the running time remarkably.

C. CASE STUDY

We further randomly select a topic for case study and analyze the online clustering results of LDA, TF-IDF, and BoNS, whose performances have been shown above to be good in terms of both running time and accuracy. The chosen topic “Smartisan Company went broke” is about the bankruptcy of the company named “Smartisan” and the titles of the reports it contains are given in Table 7. As shown in Fig.7, there are 21 TPs, 5 FNs, and 1 FP (the news belongs to another topic) in the results of BoNS. Notably, BoNS recalls more news reports than TF-IDF and LDA, as the latter two both recall 16 but
miss 11 reports. In addition, most news reports that TF-IDF misses (No.1, 17, 18, and 19) contain the word “Yonghao Luo”; while, for LDA, it clusters them correctly but misses many reports containing the word “Smartisan” (No.11, 12, 13, 20, and 21). “Yonghao Luo” is the name of the president of the company “Smartisan”, which is often used to refer to the company in the real world, so it can be regarded as the near-synonym, or rather an alias, of “Smartisan.” BoNS recalls most of the reports that contain “Yonghao Luo” or “Smartisan” and are missed by TF-IDF and LDA, which demonstrates that BoNS is effective for distinguishing near-synonyms. The false positive one is entitled “Can Mi 9 break through among mobile phone manufacturers?” and belongs to the topic “Mi 9 launched”, but also mentions the mobile phone manufacture “Smartisan Company.” It is truly difficult to winnow out even for human annotators.

In summary, our BoNS model has the best performance in terms of both accuracy and efficiency when the dimension of document representation vectors is over 1500. What is more, it could effectively deal with documents reporting the same event but using near-synonyms.

VI. CONCLUSION AND DISCUSSION

In this article, we propose a new document representation named BoNS and utilize it to deal with the near-synonyms in online clustering of Chinese technology web news. We further give a hashed version of SF-IDF named hSF-IDF to accelerate the computation of document representation. To validate the effectiveness of the proposed model, we construct a real-world dataset and conduct extensive experiments on it. The results show that our model outperforms some strong baselines in terms of both accuracy and efficiency.

In our article, we use simplified Chinese corpus to train the embeddings and hence the model works for simplified Chinese. While, if we add some tradition Chinese texts to the corpus and use it to train embeddings and construct vocabulary, the proposed model also can deal with tradition Chinese news. What is more, if Chinese corpus is changed to other languages corpus, e.g., English or Spanish, this model also can be applicable, which only slightly differs in the preprocessing that English and Spanish texts do not need word segmentation.

However, BoNS still has the limitation that it cannot address the problem of polysemy. This is because our model represents the words on the basis of word2vec, which gives each word a single representation vector, and hence could express only one meaning. We note that the problem of polysemy has been attacked with contextualized model in some other tasks such as Named entity recognition [36] and Language Modeling [37], and plan to address this problem in online clustering in the future. In addition, since natural languages are keeping evolving, the vocabulary of near-synonym sets should be updated timely. To do that, some mechanism should be developed to monitor novel words or phrases and to use the detected ones to extend the vocabulary. We also plan to explore this issue in our future work.

REFERENCES

[1] J. Allan, R. Papka, and V. Lavrenko, “On-line new event detection and tracking,” in Proc. 21st Annua. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR), 1998, pp. 37–45.
[2] R. Kosala and H. Blockeel, “Web mining research: A survey,” ACM SIGKDD Explorations Newsletter, vol. 2, no. 1, pp. 1–15, 2000.
[3] J. Beringer and E. Hülsemeyer, “Online clustering of parallel data models,” Data Knowl. Eng. Extend. Workshop, Jan. 1998, pp. 180–184, Aug. 2000.
[4] C. Wang, M. Zhang, S. Ma, and L. Ru, “Automatic online news issue construction in Web environment,” in Proc. 17th Int. Conf. World Wide Web (WWW), 2008, pp. 457–466.
[5] R. Papka et al., “On-line new event detection using singlepass clustering,” Univ. Massachusetts Amherst, Amherst, MA, USA, Tech. Rep. 290941, 290954, 1998, vol. 10.
[6] Y. Yang, T. Pierce, and J. Carbonell, “A study of retrospective and on-line event detection,” in Proc. 21st Annua. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR), 1998, pp. 28–36.
[7] X.-Y. Dai, Q.-C. Chen, X.-L. Wang, and J. Xu, “Online topic detection and tracking of financial news based on hierarchical clustering,” in Proc. Int. Conf. Mach. Learn. Cybern., vol. 6, Jul. 2010, pp. 3341–3346.
[8] J. Allan, J. G. Carbonell, G. Doddington, J. Yamron, and Y. Yang, “Topic detection and tracking pilot study final report,” in Proc. DARPA Broadcast News Transcription Understanding Workshop, Jan. 1998, pp. 194–218.
[9] J. Allan, Topic Detection and Tracking: Event-Based Information Organization, vol. 12. New York, NY, USA: Springer, 2012.
[10] F.-J. Yin, W.-D. Xiao, B. Ge, and F.-F. Li, “Incremental algorithm for clustering texts in Internet-oriented topic detection,” Jisuanji Yingyong Yanjiu, vol. 28, no. 1, pp. 54–57, 2011.
[11] Y. Xiao lin, Z. Xiao, K. Nan, and Z. Feng chao, “An improved single-pass clustering algorithm Internet-oriented network topic detection,” in Proc. 4th Int. Conf. Intell. Control Inf. Process. (ICICIP), Jun. 2013, pp. 560–564.
[12] G. Zhe, J. Zhe, L. Shoushan, T. Bin, N. Xin bin, and X. Yang, “An adaptive topic tracking approach based on single-pass clustering with sliding time window,” in Proc. Int. Conf. Comput. Sci. Netw. Technol., vol. 2, Dec. 2011, pp. 1311–1314.
[13] D. Yan, E. Hu, and B. Hu, “An improved single-pass algorithm for Chinese microblog topic detection and tracking,” in Proc. IEEE Int. Congr. Big Data (BigData Congress), Jun. 2016, pp. 251–258.
[14] B. Huang, Y. Yang, A. Mahmood, and H. Wang, “Microblog topic detection based on LDA model and single-pass clustering,” in Proc. Int. Conf. Rough Sets Current Trends Comput. New York, NY, USA: Springer, 2012, pp. 166–171.
[15] G. Xu, Y. Meng, Z. Chen, X. Qiu, C. Wang, and H. Yao, “Research on topic detection and tracking for online news texts,” IEEE Access, vol. 7, pp. 58407–58418, 2019.
[16] L. AliSUMait, D. Barbará, and C. Domeniconi, “On-line LDA: Adaptive topic models for mining text streams with applications to topic detection and tracking,” in Proc. 8th IEEE Int. Conf. Data Mining, Dec. 2008, pp. 3–12.
[17] M. D. Hoffman, D. M. Blei, and F. Bach, “Online learning for latent Dirichlet allocation,” in Proc. 23rd Int. Conf. Neural Inf. Process. Syst. (NIPS), vol. 1. Red Hook, NY, USA: Curran Associates Inc., 2010, pp. 856–864.
[18] D. Blei, “Probabilistic topic models,” in Proc. 17th ACM SIGKDD Int. Conf. Tuts. (KDD Tuts.), 2011, pp. 77–84.
[19] 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, no. 6, pp. 391–407, 1990.
[20] T. Hofmann, “Probabilistic latent semantic analysis,” in Proc. 15th Conf. Uncertainty Artif. Intell. (UAI). San Francisco, CA, USA: Morgan Kaufmann, Jan. 1999, pp. 289–296.
[21] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent Dirichlet allocation,” J. Mach. Learn. Res., vol. 3, pp. 993–1022, Mar. 2003.
[22] 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.
[23] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in Proc. Adv. Neural Inf. Process. Syst., vol. 2013, pp. 3111–3119.
[24] S. Zhang, H. Wang, X. Zhang, J. Cheng, P. Li, and Z. Ding, “Embedding-based feature extraction methods for chinese sentiment classification,” in Proc. IEEE 2nd Int. Conf. Data Sci. Cyberspace (DSC), Jun. 2017, pp. 569–577.
[25] Q. Le and T. Mikolov, “Distributed representations of sentences and documents,” in Proc. Int. Conf. Mach. Learn., 2014, pp. 1188–1196.

[26] H. K. Kim, H. Kim, and S. Cho, “Bag-of-concepts: Comprehending document representation through clustering words in distributed representation,” Neurocomputing, vol. 266, pp. 336–352, Nov. 2017.

[27] J. Reed, Y. Jiao, T. Potok, B. Klump, M. Elmore, and A. Hurson, “TF-ICF: A new term weighting scheme for clustering dynamic data streams,” in Proc. 5th Int. Conf. Mach. Learn. Appl. (ICMLA), Dec. 2006, pp. 258–263.

[28] R. Jin, C. Falusos, and A. G. Hauptmann, “Meta-scoring: Automatically evaluating term weighting schemes in IR without precision-recall,” in Proc. 24th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR), 2001, pp. 83–89.

[29] G. Karypis, E.-H. Han, and V. Kumar, “Chameleon: Hierarchical clustering using dynamic modeling,” Computer, vol. 32, no. 8, pp. 68–75, 1999.

[30] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” 2018, arXiv:1810.04805. [Online]. Available: http://arxiv.org/abs/1810.04805

[31] J. Park, C. Park, J. Kim, M. Cho, and S. Park, “ADC: Advanced document clustering using contextualized representations,” Expert Syst. Appl., vol. 137, pp. 157–166, Dec. 2019.

[32] Y. Cui, W. Che, T. Liu, B. Qin, Z. Yang, S. Wang, and G. Hu, “Pre-training with whole word masking for Chinese BERT,” 2019, arXiv:1906.08101. [Online]. Available: http://arxiv.org/abs/1906.08101

[33] Q. Cai, “Research on Chinese naming recognition model based on BERT embedding,” in Proc. IEEE 10th Int. Conf. Softw. Eng. Service Sci. (ICSESS), Oct. 2019, pp. 1–4.

[34] G. Wiedemann, S. Remus, A. Chawla, and C. Biemann, “Does BERT make any sense? Interpretable word sense disambiguation with contextualized embeddings,” in Proc. 15th Conf. Natural Lang. Process. (KONVENS), 2019, pp. 1–10.

LE CHEN received the M.E. degree in mechatronic engineering from Air Force Engineering University (AFEU), China, in 2005. He is currently a Senior Engineer with the Institute of Systems Engineering, Academy of Military Sciences (AMS). His research interest includes information system engineering.

FENGJING YIN received the Ph.D. degree in management science and engineering from the National University of Defense Technology (NUDT), Changsha, China, in 2011. He is currently a Lecturer with the College of Systems Engineering, National University of Defense Technology. His research interests include social networks analysis and data mining.

XIN ZHANG received the B.S. and Ph.D. degrees in system engineering from the National University of Defense Technology (NUDT), China, in 2000 and 2006, respectively. He is currently a Professor with the Science and Technology on Information Systems Engineering Laboratory, College of Systems Engineering, NUDT. His research interests include cross-model data mining, information extraction, and event analysis.

LIANG GUO was born in Wuhan, China, in 1991. He received the B.Eng. degree in systems engineering and the M.Eng. degree in management science and engineering from the National University of Defense Technology (NUDT), Changsha, China, in 2014 and 2016, respectively, where he is currently pursuing the Ph.D. degree with the College of Systems Engineering. His research interests include information extraction, data mining, and text analytics.

ZHE ZHANG was born in Hebei, China, in 1995. He received the bachelor’s degree in software engineering from the Beijing Institute of Technology (BIT), Beijing, China, in 2018. He is currently pursuing the master’s degree in management science and engineering with the National University of Defense Technology (NUDT). His research interests include natural language processing, data mining, and text analysis.