applied sciences

Article

Method of Feature Reduction in Short Text Classification Based on Feature Clustering

Fangfang Li [1,†,‡], **Yao Yin** [1,‡], **Jinjing Shi** [1,*], **Xingliang Mao** [2,*] **and Ronghua Shi** [1]

[1] School of Computer Science and Engineering, Central South University, Changsha 410073, China; lifangfang@csu.edu.cn (F.L.); yyao0418@csu.edu.cn (Y.Y.); shirh@csu.edu.cn (R.S.)
[2] Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha 410073, China
* Correspondence: 214009@csu.edu.cn (J.S.); gisor@163.com (X.M.)
† Current address: Central South University, Changsha 410073, China.
‡ These authors contributed equally to this work.

Received: 15 February 2019; Accepted: 10 April 2019; Published: 16 April 2019

Abstract: One decisive problem of short text classification is the serious dimensional disaster when utilizing a statistics-based approach to construct vector spaces. Here, a feature reduction method is proposed that is based on two-stage feature clustering (TSFC), which is applied to short text classification. Features are semi-loosely clustered by combining spectral clustering with a graph traversal algorithm. Next, intra-cluster feature screening rules are designed to remove outlier feature words, which improves the effect of similar feature clusters. We classify short texts with corresponding similar feature clusters instead of original feature words. Similar feature clusters replace feature words, and the dimension of vector space is significantly reduced. Several classifiers are utilized to evaluate the effectiveness of this method. The results show that the method largely resolves the dimensional disaster and it can significantly improve the accuracy of short text classification.

Keywords: feature reduction; feature clustering; short text classification; word embedding

1. Introduction

Communication on the internet is increasingly frequent, resulting in a considerable amount of information data. Most of these data are short texts, such as microblogs and BBSs (Bulletin Board Systems). The short text classification has encountered new challenges due to their limited length, the prevalence of internet vocabulary, abbreviations, and intensifying synonyms [1,2]. Therefore, one research hotspot in natural language processing is precisely classifying these short texts and effectively analyzing their meanings.

The early work of short text classification involves converting texts into data representation, a form that computers can process, which is crucial in text classification and can directly affect classification [3]. A basic representation of a text is bag of words in which each text is signified as a vector of words. In traditional short text classification tasks, a VSM (vector space model) or a word embedding model usually represent text vectors. However, the following problems exist when applying these methods to short text classification: (1) The spatial distance between words is not considered, and dimensional disasters are easily triggered when constructing text vectors by the VSM and TF-IDF (term frequency-inverse document frequency). The VSM only pays attention to statistical information, such as word frequency. Moreover, it ignores the influence of other factors, such as synonyms on text similarity, thus reducing the accuracy of short text classification [4–6]; (2) By modeling the context and semantic relations of words, the word embedding model maps words to an abstract low-dimensional real space and generates a corresponding word vector. This is an effective way of constructing a semantic association of words [7–9]. However, in the task of short text classification, the improvement of classification accuracy is limited when applying the word embedding model alone, primarily because a short text is concise and it contains many polysemous words, noise words, and internet words. The absence of effective features makes it difficult to extract enough semantic information [10].

Feature reduction can solve the above problems. Feature reduction removes irrelevant features and extracts a feature subset without reducing the classification accuracy [11]. It is a key preprocessing step that can improve classification accuracy and reduce computation. In general, there are two ways to achieve feature reduction [12]. One approach is to only select the most important subset of features in a category and ignore the rest. However, applying this path alone does not take advantage of the relationships between features that are captured from external knowledge. Another feature reduction method is to cluster the related features and then generate a compact feature set based on the original features. This type of feature reduction builds relationships between features [13–15].

Feature clustering can, to a certain extent, achieve the semantic extension of texts. Its purpose is to enrich the semantic representation of texts without reducing the classification accuracy [16,17]. Semantic expansion changes the expression and dimension of feature vectors via auxiliary information, such as an external corpus [18], context [16], characters [19], core words [20], and attention [21], so as to improve the accuracy of text classification. There are two common routes of extending text semantics. The first way is to build an external knowledge base to enrich the semantic representation of short texts. Researchers have enriched the short texts by employing external resources, such as WordNet [22] and Wikipedia [18,23,24]. These improvements involve complex natural language processing for syntactic analysis and complex semantics and demand a mass of external data, which is rare for short text classification. The second method is to optimize the feature representation by, for example, improved word embedding [25] or feature clustering [26]. Feature clustering selects a sub-feature set in the original feature set and replaces the original feature set with a sub-feature set for correlation calculations [15]. For example, "Spider-Man", "Iron Man", and "Captain America" tend not to appear in the same article, but they fall into the same category of "Marvel characters" and they have a high degree of semantic similarity. If such words are clustered, the semantic representation capacity of a single vocabulary can be precisely and appropriately extended, and the feature dimension can be declined. However, there are two critical problems when employing this method: enhancing the efficiency of building similar feature clusters and upgrading the effectiveness of similar feature clusters.

Many studies have proven the validity of the feature reduction method. In this paper, the TSFC (two-stage feature clustering) algorithm is proposed, which employs cosine semantics as a mechanism to establish the relationship between the features and create many compact feature sets. Firstly, an improved feature clustering algorithm is used to construct similar feature subsets, and a deep relationship between the features is then established by embedding words. Finally, semantic extension of short texts is achieved. Our work focuses on how to address the above two troublesome problems in feature clustering. The enormous external knowledge is unnecessary in the approach that we proposed, which reduces a considerable amount of data requirements when compared with methods that require millions of external data, and feature dimension is significantly reduced by feature replacement.

Our main contributions are summarized, as follows:

- A two-stage feature clustering (TSFC) method is proposed, which takes into account the efficiency and accuracy of feature clustering.
- The original feature was replaced by similar feature clusters to express feature vectorization.
- The method that is presented in this paper can effectively reduce feature dimension.
- We conducted vast experiments by utilizing data of different types and lengths: A China Mobile SMS (Short Message Service) dataset and a Fudan News dataset.

The rest of this paper is organized, as follows: Section 2 introduces the related works of text semantic extension. Section 3 presents the proposed method, including the TSFC algorithm and the intra-cluster feature selection rules. Section 4 verifies the validity of the method with experiments with classification of short Chinese texts. We draw conclusions in Section 5.

2. Related Works

Feature reduction can reduce computation and improve the accuracy of text classification. A document analysis method, called the discriminant coefficient, was proposed to reduce the features by Xu et al. [12]. In view of the traditional serial and parallel feature fusion method shortcomings, Yu et al. proposed a dimensionality reduction method for feature vectors while using a PCA (principal component analysis) method before fusing feature vectors [13]. Li et al. proposed a general importance weighted feature selection strategy for text classification, by which its relative frequency in the document determines the importance of a feature in a document [14].

Researchers mainly carry out the semantic extension of short texts in two ways: constructing an external knowledge base and optimizing feature representation. Many have attempted using an external corpus to construct semantic ontology to optimize classification. Xu et al. [27] took the pre-built ontology as the knowledge base and introduced semantic relationships between the features. However, constructing the ontology itself requires a great deal of extra work. Desai et al. [28] utilized the currently available packages, such as WordNet, a lexical database for English from Princeton University, to help build the ontology. Although some extra effort of building the ontology can be reduced, the size and effectiveness of WordNet limited its semantic understanding. WordNet is also only English-specific. This library can no longer be used when the corpus is in another language. Pak et al. [29] proposed a contextual advertising approach that is based on Wikipedia matching, so as to embed candidate ads into related pages. Ren [30] et al. combined background knowledge with a neural network to classify texts and expanded the dimension of feature expression. Wu et al. [18] proposed an effective Wikipedia text document classification semantic matching method. This method uses a Wikipedia corpus to construct a concept space, uses heuristic rules to select concepts, and then maps texts to concepts one by one to achieve the effect of improving the text classification accuracy.

The optimization of feature representation is another effective method for semantic expansion in short texts, which optimizes within a text collection. Jiang et al. [31] combined a neural network language model and Word2Vec to increase the accuracy of emotional analysis. Lilleberg et al. [32] utilized word embedding to generate document vectors and connected these vectors as additional features to the word frequency-reverse document frequency (TF-IDF) vector to improve classification accuracy in various classifiers. Song et al. [33] employed the minimum spanning tree (MST) clustering method to select the features, and they proved that the method can improve the accuracy of text classification while ensuring efficiency. Cao et al. [26] applied a feature reduction strategy and used a k-means algorithm to cluster feature words and then classify them. However, this simple clustering method is unable to achieve an appropriate number of feature clusters or control the size of clusters. Ged et al. [15] used a loose clustering strategy that is based on graph breadth-first traversal to perform feature clustering. However, this method is inefficient and the internal correlation of the obtained feature clusters cannot be guaranteed. Vlachostergiou et al. [34] designed a depth model to learn robust, resizable representations of features from the untagged data. This method is mainly applicable to the supervision model.

3. The Proposed Approach

We put forward an effective TSFC method for the sake of overcoming the contradiction between the efficiency and accuracy of feature clustering. Figure 1 shows the framework of the TSFC approach, which consists of the following four steps. First of all, the construction of feature word bags. After text preprocessing, the feature word bags are constructed by selecting all of the appropriate feature words in the whole corpus. Secondly, similar feature clusters are constructed by TSFC. All the feature words are clustered into multiple preliminary sub-clusters by a spectral clustering algorithm in the first stage. The similar features within each feature sub-cluster are then paired. These feature pairs are loosely clustered by the method of the figure traversal algorithm in the second stage, and several similar feature clusters are obtained. Thirdly, the feature clusters are optimized. In order to weaken the influence of polysemous or irrelevant words on classification accuracy, the feature selection rules are designed to ensure the size and effect of each feature sub-cluster. The classification accuracy can be significantly improved by eliminating such words in the final cluster of similar features. Finally, several classifiers that are based on the feature vectors of similar feature clusters classify the short text corpus. In Table 1, we define the symbols used in this paper.

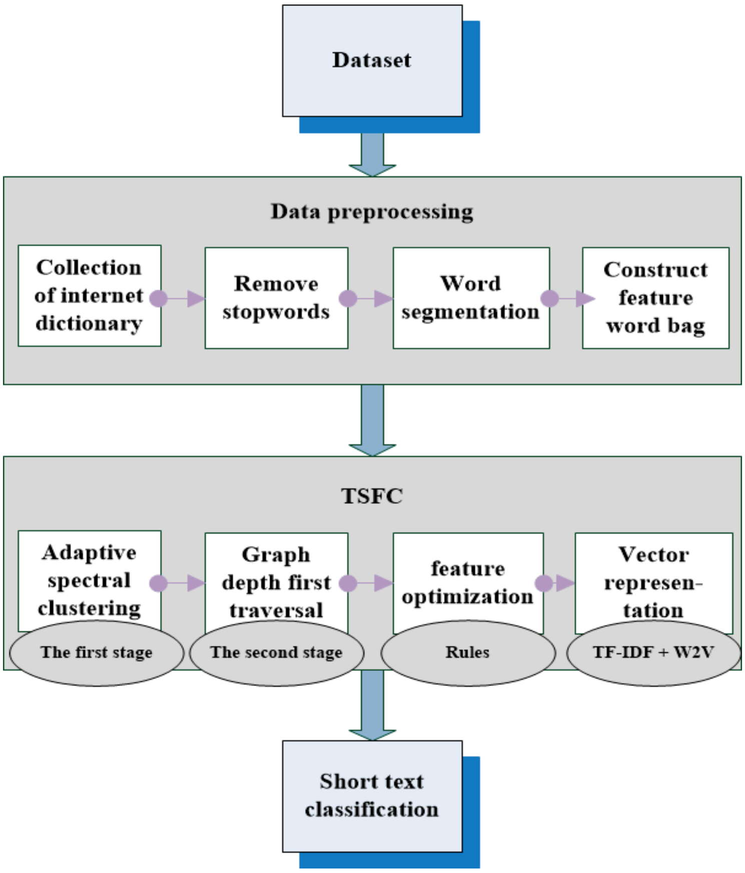

Figure 1. The framework of the two-stage feature clustering (TSFC) approach.

Table 1. Symbols and their definitions.

Symbol	Definition
C_i	Initial feature cluster
C_s	Similar feature cluster
W_r	Feature words in similar feature clusters
T	Word bags
D	An unclassified document set consisting of text documents
V(w)	The vector representation of the feature word w

3.1. Construct the Feature Word Bags

The experimental corpus of this paper is the Fudan Chinese text classification dataset and the China Mobile SMS dataset. The steps of text preprocessing are as follows:

- Collection of an internet dictionary. Social network texts are more informal and colloquial than formal texts. They contain many new internet buzzwords, such as “laotie” and “lanshouxianggu”. The internet dictionary is a user-defined dictionary. It includes internet terms that contribute to the precision of word segmentation.

- Stopword removal. Stopwords are words that contain no concrete meanings (e.g., prepositions, pronouns, and conjunctions). Therefore, stopwords need to be removed from each document, so as to avert a negative impact on the method. A Chinese stopword list with 2929 entries is applied here.
- Word segmentation. Chinese word segmentation refers to the cutting of a Chinese character sequence into individual words. Word segmentation is the process of regrouping consecutive word sequences into compound word sequences. The NLTK (Natural Language Toolkit) [35] segmentation kit is used in this method.
- Construct feature word bags. In all corpuses, after word segmentation, feature words whose word frequency is higher than the set threshold and that are without repetition are selected to construct the feature word bags.

3.2. *Two-Stage Feature Clustering*

The TSFC method contains two stages. The sub-clusters obtained in the first stage are called initial feature clusters, and those that are obtained in the second stage are called similar feature clusters. Word bags are cut into multiple initial feature clusters in the first stage. The purpose of this preliminary clustering is to obtain more semantically similar features in a smaller feature set, namely the sub-cluster that was obtained in this step. Its advantage is that it can significantly reduce the size of the overall calculation of feature clustering, because a large number of paired comparison calculations are needed in the clustering process. In addition, the size of the final similar feature clusters can be controlled to some extent. In the second stage, the graph traversal algorithm is used to directly connect semantically similar features and merge them into clusters. This method can cover more feature points than the traditional clustering algorithm and avoid the loss of effective information. However, this method has no distance or density constraint, which is emphasized in the traditional clustering method, so we call it loose clustering. TSFC is a semi-loose clustering strategy, when combined with the strict clustering in the first stage. This is a compromise, because the accuracy of traditional clustering methods to construct similar feature clusters is not satisfactory, and the computation that is required by the graph-based search method is enormous.

3.2.1. Adaptive Spectral Clustering

At the first stage, adaptive spectral clustering is utilized. Spectral clustering works by first transforming the data from Cartesian space into similarity space and then clustering in similarity space. The original data is projected into the new coordinate space, which encodes information regarding how nearby data points are. The similarity transformation reduces the dimensionality of space and, loosely speaking, pre-clusters the data into orthogonal dimensions. This pre-clustering is non-linear and it allows for arbitrarily connected non-convex sample space, which is an advantage of spectral clustering. Another advantage of spectral clustering is that it can improve the time efficiency of constructing similar feature clusters and ensure the effectiveness of each similar feature cluster.

We use an adaptive multipath spectral clustering algorithm in this approach. According to the adaptive strategy that was proposed by Liu.et al. [36], different values can be obtained for different local densities, so a more accurate number of class clusters can be obtained according to the dataset. It is worth mentioning that the TSFC algorithm is not sensitive to the initial clustering K value, and the initial K value can be adjusted to a roughly appropriate value, according to the size of the dataset. The focus of our work is the feature clustering in the second stage. The purpose of the first stage of clustering is only to divide the feature word bag into multiple initial feature clusters. The reason is that initial clustering only needs to divide the text feature word space into several small blocks and ensure that the cluster is neither too large nor too small; while, the two-stage graph search can connect each feature word. In other words, the cluster partition error of the first stage can be corrected in a certain sense in the second stage.

3.2.2. Loose Clustering Based on Graph Depth-First Traversal

The clustering in the second stage is carried out within each initial feature cluster that was obtained in the first stage. This phase consists of the following steps:

Step 1. Pairing. In our approach, pairing is the process of connecting any two words whose cosine similarity is higher than the threshold values of 0.35, 0.4, 0.45, 0.5, 0.6, or other values. The similarity is calculated while using the word's 300-dimension vector derived from the Word2Vec model. The Word2Vec model is trained by a 5GB Wikipedia Chinese corpus. For example, the words "strawberry" and "grapes" have a similarity of 0.46, as calculated from Word2Vec, so "strawberry" and "grapes" are paired when the similarity threshold is 0.45. The similarity between "strawberry" and "grapes" is estimated by calculating the cosine similarity between the vectors that are represented by *V* and *E*; that is, sim (*V*, *E*) = cosine (*V*, *E*), defined as follows:

$$cosine(V,E) = \frac{\sum_{i=1}^{n}\left(v_i \times v_{E_i}\right)}{\sqrt{\sum_{i=1}^{n} V_i^2 \times \sum_{i=1}^{n} V_E^2}} \tag{1}$$

where *V* and *E* are the vector of length *n*. Therefore, we declare that, if the cosine similarity of feature word *V* and *E* exceeds the threshold, then *V* and *E* are paired.

Semantically related words are combined by direct links that are based on a loose clustering strategy. This strategy can intuitively select suitable words to form a similar feature cluster, rather than cluster all of the words together in one. Obviously, not all of the words are suitable for entering a similar feature cluster, because they do not have a high semantic similarity with respect to any other words in the feature word bag. At the same time, it avoids the influence of an improper number of clusters on the clustering results. In addition, this method can cover more effective keywords, because the complex and strict clustering algorithm is not used.

Step 2. Building the adjacency list. The feature adjacency list is composed of pairs of feature words with similarities of 0.4, 0.45, 0.5, 0.55, and 0.6, respectively. This structure can also be represented as an undirected linkage graph, which is the basis for determining the final cluster of similar features. Within each initial feature cluster, the feature adjacency list is composed of paired feature pairs. Figure 2 shows the graph of feature pairs with undirected connections.

The points in the figure are feature words, the lines between the two points represent the pairing relationship, and the numbers represent the cosine similarity. Since Chinese word vectors are used in this method, English words may not represent the elements in the graph, but it is a word in Chinese. The optimal similarity threshold is determined by the word vector model. The optimal similarity threshold will be different with different word vector models. The results of different similar feature clusters caused by different thresholds will be further described in the following experiments.

Step 3. Clustering. Depth-first traversal is performed on the adjacency list that is obtained in Step 2. Each iteration traversal forms a similar feature cluster, and multiple similar feature clusters are then obtained. Note that our traversal does not directly jump to a new node that has not been visited. During each traversal, if all existing nodes have been visited and no new feature pairs are found, the nodes that have been visited will be clustered into a cluster and the next iteration will then begin. Within each initial feature cluster, a similar feature cluster is constructed by traversing the feature adjacency list. Once these clusters are formed, these clusters will be used in text vectorization. This method of feature reduction is driven by the idea that it is not necessary to keep all of the semantically similar features separate. Instead, those similar features can be reduced to a new feature and made equivalent to an element in the process of text vectorization. For example, the features "Monster", "Devil", and "Ghost" have high cosine similarity, so they are clustered into a similar feature cluster, which is represented by "M-D-G". In the classification process, "Monster", "Devil", and "Ghost" are replaced with "M-D-G", so the total feature dimension is reduced by 2. Specific examples of text substitution are described in the next section.

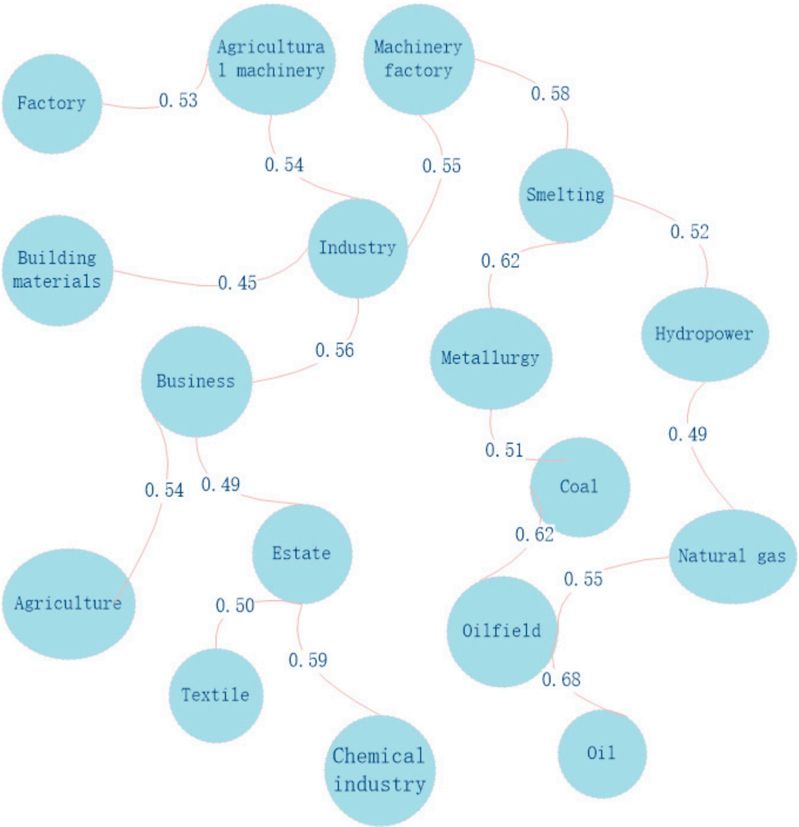

Figure 2. Feature pair adjacency graph.

Algorithm 1 shows the details of the TSFC algorithm.

Algorithm 1. Two-Stage Feature Clustering

```
Input: a set of unclassified text documents
Output: similar feature clusters
1. for each doc ∈ corpus do:
      for each term ∈ doc do:
   terms ← pre-processing all term with NLP methods
      end
word bags ← terms
      end
2. adaptive spectral clustering in word bags:
Similar feature clusters $C_i$ ← Word bags
3. for each term ∈ $C_i$ do:
        for each word ∈ term do:
        if cosine similarity of any two words > similarity threshold then:
        adjacency list ← paired words
        similar feature clusters $C_s$ ← depth-first traversal in adjacent list
        end
```

3.3. Sub-Cluster Feature Selection

Obviously, in our method, not all of the feature words should be clustered into a similar feature cluster, which is a negative effect of feature clustering. In some texts, it may yield richer semantics to words that are not related to the subject, that is, a whole cluster of semantics. Therefore, in some cases, the influence of ambiguous or irrelevant words is magnified. In addition, due to the loosely clustered pairing strategy, some of the related feature pairs may be less compact in the entire feature cluster, and some points of these feature pairs are actually noise points in this cluster. These outliers have a bad effect on the classification results. In order to improve the quality of each similar feature cluster, we designed an intra-cluster feature selection rule. Feature screening is carried out in each cluster after the similar feature sub-clusters are obtained. The principle of screening is to calculate the average similarity and the maximum similarity of each feature term and the whole cluster. The concept of average similarity is defined, as follows:

Average similarity: the average cosine similarity of a feature to all features in a cluster.

Therefore, the main point of our rule can be summarized, as follows: the words with average similarity less than the threshold value will be deleted. Here, the threshold is equal to the similarity threshold of feature word pairing in Step 1 of the previous section.

3.4. Vectorization

The purpose of the TSFC method is to obtain similar feature clusters, that is, high similarity feature sets. The next step of text classification is to vectorize the feature words. TF-IDF and word embedding are used to verify the effectiveness of our method. It is an intuitive way to demonstrate the effect of our method. TF-IDF and word embedding are the most common text vectorization methods. At the same time, they are regarded as the basis of other improved vectorization methods.

A Corpus D consists of a set of texts, $D = \{d_1, d_2, \cdots, d_n\}$ and the word bags $\{T = \left\{w_1, w_2, \cdots, w_m, w_{r_1}, w_{r_2}, \cdots, w_{r_k}\right\}$, where k is the number of all elements in similar feature clusters, m is the number of other words in the feature bag, and n represents the dimension of the vector space.

When the TF-IDF is used, there are the following rules for feature vectorization:

$$d_i = \left\{w_1, w_2, \cdots, w_{r_1}, w_{r_2}, \cdots\right\} \rightarrow d_i = \left\{w_1, w_2, \cdots, c_{s_1}, c_{s_2}, \cdots\right\} \tag{2}$$

W_r is the feature word in the corresponding similar feature cluster C_s. That is to say, in the process of calculating TF and IDF, if a feature word belongs to a similar feature cluster, then it will be replaced by a corresponding similar feature cluster. For a similar feature cluster with a capacity of 20, 20 feature words will be replaced by it, so the total feature dimension is reduced by 19.

When the TF-IDF is used, there are the following rules for feature vectorization:

$$V_{(d_i)} = \left\{\frac{V_{(W_1,W_2\cdots)} + V_{(Wr_1,Wr_2\cdots)}}{l}\right\} \rightarrow V_{(d_i)} = \left\{\frac{V_{(W_1,W_2\cdots)} + V_{(Cs_1,Cs_2\cdots)}}{l}\right\} \tag{3}$$

The vector of each similar feature cluster is the mean of all the feature word vectors in the cluster. In the process of text vectorization, similarly, if a feature word belongs to a similar feature cluster, then a corresponding similar feature cluster will replace it. The essence of this method is to make the semantically close feature vectors closer to each other in Cartesian coordinates, that is, they are replaced by a central point (similar feature cluster). Table 2 shows a practical example of feature substitution.

Table 2. Example of feature replacement.

The Original Text	After the Replacement
Intel dollar risk fund investment generation	*Cs*-1, *Cs*-2, *Cs*-3 investment generation
New knowledge from science books	*Cs*-4 and *Cs*-5 introduce *Cs*-6 *Cs*-7

4. Experimental Validation

4.1. Dataset and Data Preprocessing

Two datasets are used in our experiment, and they cover different categories and sizes, as shown in Table 3. The first is the China Mobile SMS dataset, which was obtained from a project, including normal SMS and five types of spam SMS. The other is a Fudan News dataset, which contains five single-label categories, and the balanced data size of each category after our processing. The purpose of employing these datasets is related to its considerable feature size, which makes it an appropriate candidate for evaluating the outcome of our approach. The SMS dataset includes five categories: front, advertising, credit card, loan, and other. The Fudan News datasets includes six types of data, and these categories include sports, politics, economics, art, history, and computers.

Table 3. Dataset and feature size.

Dataset	Testing Size	Training Size	Feature Size	Categories
China Mobile SMS	23541	76001	7800	5
Fudan News Data	14332	36026	9537	6

4.2. Performance of the TSFC Algorithm

In this paper, four aspects of the TSFC algorithm are evaluated: the effect of spectral clustering, the scale of similar feature clusters, the effect of feature reduction, and the classification accuracy.

4.2.1. Results of Spectral Clustering

The first step of constructing similar feature clusters is to cut the feature set into several sub-clusters by using a spectral clustering algorithm, and the number of clustering centers K is the only parameter to be adjusted. In Figure 3, the broken line shows the time relationship between the number of clustering centers and the construction of similar feature clusters, and it shows the final classification F1 score of the TSFC method under different K values. If K is too large or too small, then the size of similar feature clusters will not be optimal. Therefore, when K is moderate, the best results will be obtained. With the increase of K, the time to obtain similar feature clusters is shorter, because, the larger the initial feature cluster is, the more computational work to construct similar feature clusters is. In our method and date set, the optimal K fluctuates by around 10, depending on the size of the dataset.

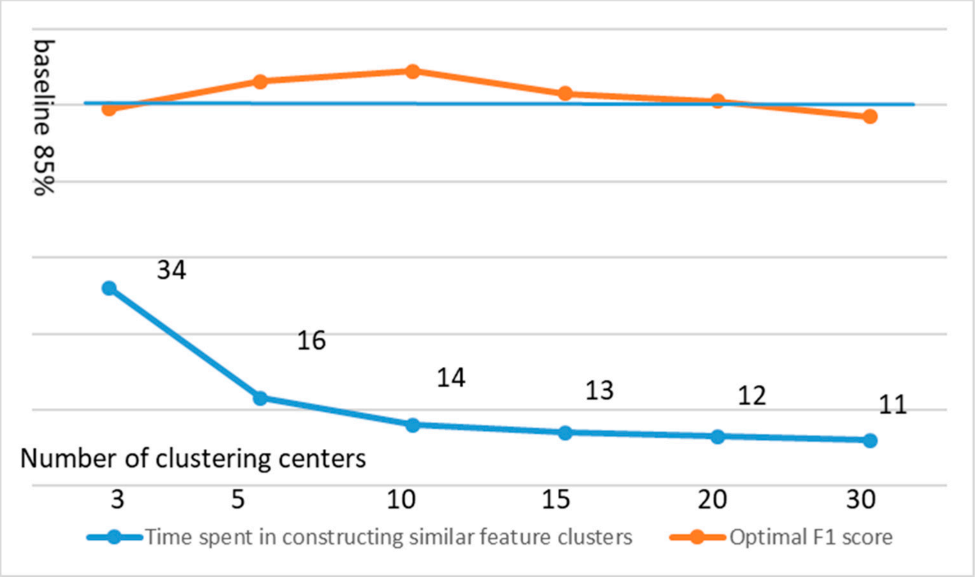

Figure 3. Spectral clustering results. The time refers to the cost of constructing similar feature clusters in minutes. The baseline for evaluating the F1 score is obtained using the TF-IDF method.

4.2.2. Size of Similar Feature Clusters

Within each sub-cluster of spectral clustering, multiple different similarity thresholds are used to connect the features. The word vector model calculates the cosine similarity between two features, so the optimal similarity threshold is different when different word vector models are utilized. In order to verify the universal applicability of this method, the word vectors that are trained by Wikipedia's corpus are utilized instead of the specific word vectors in a certain field, which proves that our method has good portability.

Figure 4 shows the results of similar feature clusters. Similarity thresholds of 0.6, 0.55, 0.5, 0.45, and 0.4 are used. Two important factors are plotted in the figure to reveal the effect of the clustering: the cluster size distribution and the total number of similar feature clusters. With the increase in similarity threshold, the size of the cluster of similar features decreases, but this reduction is not particularly obvious in the three intervals of 45%, 50%, and 55%. The cluster with a capacity that exceeds 20 is the smallest, and clusters with a capacity of 2–5 account for more than half of the total. However, when similar features are further increased, the size of feature clusters begins to decrease. The influence of feature clusters that are too large or too small on the classification results is also limited, which will be discussed later. When the threshold value is 40%, all of the features in the similar feature clusters cover about 50% of the feature word bags, which implies a very desirable feature reduction result. The essence of similar feature clusters is feature reduction and semantic extension. Determining the optimal similarity threshold depends on the actual task. Usually, the threshold is acceptable when similar feature clusters can cover 40%–50% of the total feature words.

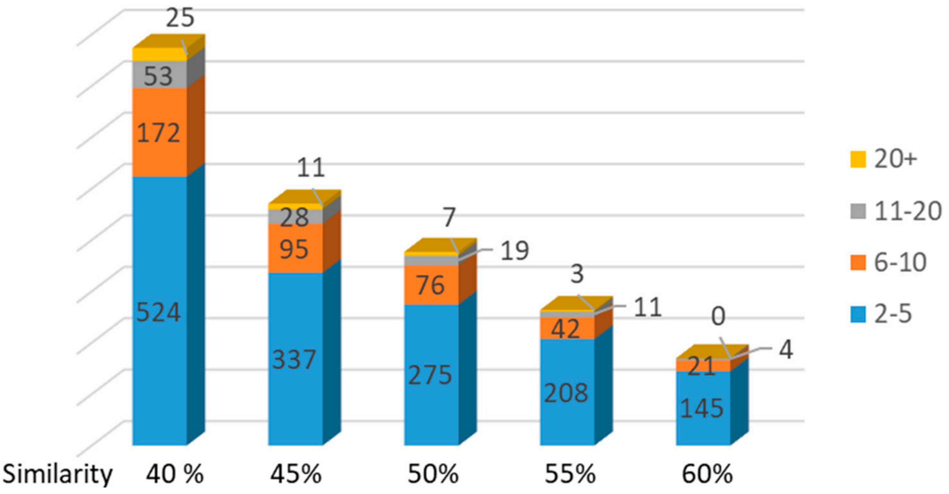

Figure 4. Size of similar feature clusters. The legend shows the number of similar feature clusters that are of different sizes.

4.2.3. Classification Effectiveness Evaluation

Table 4 shows the F1 score in classifiers, such as the naive Bayes (NB), the support vector machine (SVM), and logistic regression (LR). The ratio of the training set to the test set is 4:1 in the classification experiment, and a five-fold cross validation is employed. A significant trend is that, if the similarity threshold is too high or too low, the improvement of classification results will be limited. This is because, when the similarity threshold is too high, the size of similar feature clusters is very small and it only affects a few features. When the similarity threshold is too low, the paired features are not close enough in the similarity feature cluster, and the size of the similarity feature cluster is too large, which results in inappropriate feature replacement. When TF-IDF is utilized, the optimal similarity threshold is 50% and the classifier is SVM or NB, while the optimal threshold used by LR is 45%, and the maximum F1 score is increased to 3.04%. However, a high similarity threshold results in an F1 score that is lower than the baseline. It is worth mentioning that the representation of feature reduction is what we expect. The lower the similarity threshold, the stronger the feature reduction, which can reach a maximum of 27.54%. When word embedding is used for vectorization, the maximum F1 score increases by 6.7%, but when the threshold is 40 and 60%, the F1 score is lower than that of baseline.

In summary, our method achieves a maximum feature reduction of 27% and a maximum F1 score improvement of 3% with TF-IDF.

Table 4. Result of classification methods.

SIMILARITY	SVM (F-Score)	NB (F-Score)	LR (F-Score)	Feature Reduction
TF-IDF+TSFC	**China Mobile SMS**			
Baseline	85.12%	83.55%	80.45%	-
40%	85.22%	84.78%	82.11%	22.9%
45%	87.70%	84.59%	82.35%	20.7%
50%	88.16%	85.60%	81.72%	17.0%
55%	85.49%	84.35%	81.09%	12.5%
60%	85.45%	81.43%	80.21%	7.1%
TF-IDF+TSFC	**Fudan News**			
Baseline	93.54%	91.02%	87.87%	-
40%	93.89%	91.27%	85.20%	27.54%
45%	95.54%	95.74%	87.21%	23.11%
50%	96.69%	93.58%	87.87%	17.98%
55%	96.18%	92.11%	88.01%	15.25%
60%	93.82%	91.94%	87.49%	10.80%
Word2vec+TSFC	**China Mobile SMS**			
Baseline	83.23%	83.49%	81.13%	-
40%	79.50%	80.04%	79.48%	-
45%	83.31%	84.56%	81.87%	-
50%	89.93%	86.20%	84.99%	-
55%	86.17%	84.35%	83.19%	-
60%	82.21%	83.48%	82.25%	-
Word2vec+TSFC	**Fudan News Data**			
Baseline	91.67%	90.00%	86.86%	-
40%	87.81%	87.26%	86.19%	-
45%	91.74%	89.52%	87.51%	-
50%	93.19%	92.47%	88.92%	-
55%	93.05%	90.46%	87.28%	-
60%	91.82%	89.84%	86.95%	-

5. Conclusions

A two-stage feature clustering (TSFC) approach that is based on a semi-loose strategy is proposed. This method employs a word-embedded model to build feature sample space, and an improved feature clustering strategy and sub-cluster optimization rules are used to build appropriate similar feature clusters, which are utilized to replace the original feature vectors for short text classification. The evaluation experiments demonstrate that the efficiency and accuracy of the TSFC are improved and that the method has good extensibility. At the same time, this method significantly reduces the feature dimension. For corpora in different fields, a Word2Vec model that is trained by applying the corresponding corpus, which allows for better short text classification results, calculates the cosine similarity of the feature. Future work may focus on improving the semi-loose clustering method, so as to improve the feature reduction effect of our similar feature clustering. The feature clustering method can be regarded as an effective complement to other fields, including feature selection, information extraction, and association rule mining.

6. Patents

The method of this paper involves a patent under review named "an optimized short text classification method".

Author Contributions: X.M. proposed the idea, Y.Y. designed and performed the experiments and analyzed the results, X.M. and Y.Y. drafted the initial manuscript, and Y.Y., J.S., R.S., and F.L. revised the final manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (Grant Nos. 61401519, 61872390), the Natural Science Foundation of Hunan Province (Grant Nos. 2016JJ4119, 2017JJ3415), and the Postdoctoral Science Foundation of China (Grant No. 2016M592450).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zheng, C.T.; Liu, C.; Wong, H.S. Corpus-based topic diffusion for short text clustering. *Neurocomputing* **2018**, *275*, 2444–2458. [CrossRef]
2. Sriram, B.; Fuhry, D.; Demir, E.; Ferhatosmanoglu, H.; Demirbas, M. Short text classification in twitter to improve information filtering. In Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Geneva, Switzerland, 19–23 July 2010.
3. Jia, C.Y.; Carson, M.B.; Wang, X.Y.; Yu, J. Concept decompositions for short text clustering by identifying word communities. *Pattern Recognit.* **2018**, *76*, 1–13. [CrossRef]
4. Zhang, D.L.; Wang, D.S.; Zheng, W.M. Chinese text classification system based on VSM. *J. Tsinghua Univ.* **2003**, *43*, 1288–1290.
5. Xia, T.; Du, Y. Improve VSM text classification by title vector based document representation method. In Proceedings of the 6th International Conference on Computer Science & Education, Singapore, 3–5 August 2011; pp. 210–213.
6. Zhang, Z.; Fan, X.Z. Improved VSM based on Chinese text categorization. *Comput. Eng. Design* **2006**, *21*.
7. Bojanowski, P.; Grave, E.; Joulin, A. Enriching Word Vectors with Subword Information. *Trans. Assoc. Comput. Linguist.* **2016**, *5*, 135–146. [CrossRef]
8. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781v3.
9. Le, Q.; Mikolov, T. Distributed Representations of Sentences and Documents. In Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 21–26 June 2014.
10. Du, H.; Xu, X.; Cheng, X.; Wu, D.; Liu, Y.; Yu, Z. Aspect-specific sentimental word embedding for sentiment analysis of online reviews. In Proceedings of the 25th International Conference Companion on World Wide Web Conferences Steering Committee, Montreal, QC, Canada, 11–15 April 2016; pp. 29–30.
11. Heisele, B.; Serre, T.; Prentice, S. Hierarchical classification and feature reduction for fast face detection with support vector machines. *Pattern Recognit.* **2003**, *36*, 2007–2017. [CrossRef]
12. Gao, L.J.; Chien, B.C. Feature Reduction for Text Categorization Using Cluster-Based Discriminant Coefficient. In Proceedings of the Conference on Technologies and Applications of Artificial Intelligence, Tainan, Taiwan, 16–18 November 2012.
13. Yu, Y.; Zhu, Q. The method of multi-step dimensionality reduction and parallel feature fusion in clothing recognition. In Proceedings of the International Conference on Artificial Intelligence and Robotics and the International Conference on Automation, Control and Robotics Engineering, Kitakyushu, Japan, 13–15 July 2016.
14. Li, B. Importance weighted feature selection strategy for text classification. In Proceedings of the International Conference on Asian Language Processing (IALP), Tainan, Taiwan, 21–23 November 2016.
15. Ge, L.H.; Moh, T.-S. Improving Text Classification with Word Embedding. In Proceedings of the IEEE International Conference on Big Data, Boston, MA, USA, 11–14 December 2018.
16. Biemann, C.; Osswald, R. Automatic Extension of Feature-based Semantic Lexicons via Contextual Attributes. In *From Data and Information Analysis to Knowledge Engineering*; Spiliopoulou, M., Kruse, R., Borgelt, C., Nürnberger, A., Gaul, W., Eds.; Springer: Berlin/Heidelberg, Germany, 2006.
17. Song, Q.; Ni, J.; Wang, G. A fast clustering-based feature subset selection algorithm for high-dimensional data. *IEEE Trans. Knowl. Data Eng.* **2013**, *25*, 1–14. [CrossRef]
18. Wu, Z.; Zhu, H.; Li, G. An efficient Wikipedia semantic matching approach to text document classification. *Inf. Sci.* **2017**, *393*, 15–28. [CrossRef]
19. Zhang, X.; Zhao, J.B.; Yann, L. Character-level convolutional networks for text classification. In Proceedings of the Conference on Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015.
20. Yuan, F.; Yuan, J.Y. Naive Bayes Chinese text classification based on core words of class. *J. Shandong Univ.* **2006**, *41*, 46–49.

21. Zheng, J.; Cai, F.; Shao, T.; Chen, H. Self-Interaction Attention Mechanism-Based Text Representation for Document Classification. *Appl. Sci.* **2018**, *8*, 613. [CrossRef]
22. Wei, T.; Lu, Y.; Chang, H. A semantic approach for text clustering using WordNet and lexical chains. *Expert Syst. Appl.* **2015**, *42*, 2264–2275. [CrossRef]
23. Qureshi, M.A. Utilising Wikipedia for Text Mining Applications. *ACM SIGIR Forum* **2016**, *49*, 150–151. [CrossRef]
24. Ray, S.K.; Singh, S.; Joshi, B.P. A semantic approach for question classification using wordnet and Wikipedia. *Pattern Recognit. Lett.* **2010**, *31*, 1935–1943. [CrossRef]
25. Xu, H.; Ming, D.; Zhu, D.; Kotov, A.; Carcone, A.I.; Naar-King, S. Text Classification with Topic-based Word Embedding and Convolutional Neural Networks. In Proceedings of the International Conference on Bioinformatics, Computational Biology, and Health Informatics, Seattle, WA, USA, 2–5 Octobet 2016.
26. Cao, Q.M.; Guo, Q.; Wang, Y.L. Text clustering using VSM with feature clusters. *Neural Comput. Appl.* **2015**, *26*, 995–1003.
27. Xu, G.X.; Wang, C.Z.; Wang, L.H.; Zhou, Y.H.; Li, W.K.; Xu, H. Semantic classification method for network tibetan corpus. *Clust. Comput.* **2017**, *20*, 155–165. [CrossRef]
28. Desai, S.S.; Laxminarayana, J.A. WordNet and Semantic similarity based approach for document clustering. In Proceedings of the International Conference on Computation System & Information Technology for Sustainable Solutions, Bangalore, India, 12 December 2016; pp. 312–317.
29. Pak, A.N.; Chung, C.W. A Wikipedia Matching Approach to Contextual Advertising. *WWWJ* **2010**, *13*, 251–274. [CrossRef]
30. Ren, F.; Deng, J. Background Knowledge Based Multi-Stream Neural Network for Text Classification. *Appl. Sci.* **2018**, *8*, 2472. [CrossRef]
31. Jiang, S.; Lewris, J.; Voltmer, M. Integrating rich document representations for text classification. In Proceedings of the IEEE Systems and Information Engineering Design Symposium (SIEDS), Charlottesville, VA, USA, 29 April 2016; pp. 303–308.
32. Lilleberg, J.; Yun, Z.; Zhang, Y. Support vector machines and Word2vec for text classification with semantic features. In Proceedings of the 14th International Conference on Cognitive Informatics & Cognitive Computing, Beijing, China, 6–8 July 2015; pp. 136–140.
33. Song, Y.; Wang, H.; Wang, Z. Short Text Conceptualization Using a Probabilistic Knowledgebase. In Proceedings of the International Joint Conference on Artificial Intelligence, Barcelona, Spain, 16–22 July 2011; pp. 2330–2336.
34. Vlachostergiou, A.; Caridakis, G.; Mylonas, P.; Stafylopatis, A. Learning Representations of Natural Language Texts with Generative Adversarial Networks at Document, Sentence, and Aspect Level. *Algorithms* **2018**, *11*, 164. [CrossRef]
35. Loper, E.; Bird, S. NLTK: The natural language toolkit. *arXiv*, 2002; arXiv:cs/0205028.
36. Liu, X.Y.; Li, J.W.; Hong, Y.U. Adaptive Spectral Clustering Based on Shared Nearest Neighbors. *J. Chin. Comput. Syst.* **2011**, *32*, 1876–1880.