electronics

MDPI

Article

A Feature Selection Method Based on a Convolutional Neural Network for Text Classification

Jiongen Xiao [1] and Ming Hong [2,*]

[1] International Business School, Guangdong University of Finance and Economics, Guangzhou 510320, China; 20191068@gdufe.edu.cn
[2] Department of Electronic Business, South China University of Technology, Guangzhou 510006, China
* Correspondence: ming@scut.edu.cn

Abstract

Feature selection, the technique to distinguish discriminative terms among large quantities of irrelevant and noisy terms in high-dimensional text data, is the effective solution for reducing computational cost and improving the performance of text classification. To address the issues of effective feature selection in text classification, a new ranking feature selection method is developed by utilizing the deep learning model convolutional neural network (CNN). Our method uses a pre-trained three-layer CNN to capture deep features of terms and selects discriminative terms according to the combination of deep features and term frequencies, aiming to improve performance of feature selection for text classification. Moreover, the CNN training in our method is relatively fast because the loss function of the CNN does not need to reach convergence. Nine benchmark datasets and several comparative methods are introduced to validate the effectiveness of our method. Experimental results demonstrate the following: (1) Our method achieves better performance than other comparative methods in improving text classification accuracy, with significance level of 0.05 in most cases. (2) The top 20 terms selected by our method are more semantically related to the topics of datasets than comparative methods. (3) Our method outperforms comparative methods in reducing the sparsity of text. The sparsity reduction effect generally falls in [2%, 8%].

Keywords: text classification; feature selection; deep learning; convolutional neural network

check for updates

Academic Editor: Arkaitz Zubiaga

Received: 25 October 2025
Revised: 20 November 2025
Accepted: 22 November 2025
Published: 24 November 2025

Citation: Xiao, J.; Hong, M. A Feature Selection Method Based on a Convolutional Neural Network for Text Classification. *Electronics* **2025**, *14*, 4615. https://doi.org/10.3390/electronics14234615

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Text classification, the task of classifying text documents with unknown class labels according to predefined categories [1], is significant for further mining of the high commercial value [2] contained in text because it organizes the text documents into obvious classes. The applications of text classification cover various areas including public security [3], health care [4], proactive personality prediction [5], classification of job advertisements [6], and classification of patent documents [7]. Due to the potential application in various scenarios, text classification is still a hot spot for researchers. The research directions include (1) optimized classification models based on single deep neural networks [8,9] or hybrid deep neural networks [10–13]; (2) improved traditional classification models such as naïve Bayes [14]; (3) hybrid classification models based on traditional models and deep neural networks [15]; (4) classification models based on other models such as topic models [16]; (5) optimized text representation methods [17] to improve performance of classification models; and (6) auxiliary information, such as information on logical relationships in text

data [18], similarity based on graph and semantic knowledge [19], word importance [20], and embeddings of labels [21] to improve text classification performance.

The high-dimensional text data usually contains large quantities of irrelevant and noisy terms (non-discriminative terms). The non-discriminative terms cause steep computational complexity in learning processes and have negative effects on text classification performance. Feature selection is the effective technique to distinguish discriminative terms from non-discriminative ones before the task of text classification so that the dimension of text is reduced efficiently and the classification accuracy is improved effectively. Previous feature selection methods for the text domain select terms according to only semantic information such as class labels and term (document) frequencies. Recently, deep neural networks have been successfully applied in various domains including natural language processing. According to [22], deep neural networks are able to create deep feature representations through a series of complex transformations. Deep information is a kind of useful information for data mining. We think that deep information can be integrated with semantic information to improve the performance of feature selection for text classification. Taking advantage of effective deep information, this paper proposes a new ranking feature selection method termed CNNFS (**CNN**-based **F**eature **S**election). Our method is developed by using a convolutional neural network (CNN), a kind of classical deep neural network. CNNFS uses a pre-trained three-layer CNN to extract deep information of terms and performs feature selection by integrating deep information and semantic information.

Our main contribution is to illustrate an effective feature selection method, CNNFS. On the one hand, CNNFS applies a deep learning model CNN to extract deep information of terms and select discriminative terms by evaluating the integration of term frequencies and deep information. Existing studies have not considered designing effective feature selection methods by using deep information. On the other hand, CNNFS outperforms comparative feature selection methods in experiments, including improving the accuracy of text classification, selecting high proportions of semantic terms, and reducing the sparsity of text.

The other parts of this paper are organized into the following sections: Section 2 discusses previous feature selection methods in the text domain and applications of CNNs for text mining. The methodology of our method, CNNFS, is illustrated in Section 3. Section 4 covers all the details of experiments for validating the effective performance of CNNFS. Lastly, the conclusions of this paper are given in Section 5.

2. Literature Review

2.1. Feature Selection Methods in the Text Domain

Previous feature selection methods in the text domain can be categorized as ranking methods and selecting methods. Ranking methods rank terms according to ranking functions and select the top N terms to construct a discriminative subset (N is determined manually according to experiments or practical demands). Selecting methods determine the discriminative subset of terms by other algorithms automatically, without setting values of N.

2.1.1. Ranking Methods

Various methods evaluate relationships between terms and classes. Bahassine et al. [23] introduced an improved version of Chi-square (CHI), for the purpose of balancing the selected terms from each class. Cekik and Uysal [24] designed a new method based on the concept of rough sets. Zhou et al. [25] combined term frequencies and document frequencies in both classes and datasets to design a new method. Amazal and Kissi [26] proposed a method based on MI, variances of frequencies of terms in classes, and maximum

frequencies of terms. Parlak and Uysal [27] proposed a new method which performed feature selection according to the integration of conditional probabilities of terms given classes and conditional probabilities of classes given terms. Cekik and Uysal [28] presented a feature selection which selected discriminative terms by projecting the relationship among a term and any two of the classes to a two-dimensional integer space. Parlak [29] designed a new method by using conditional probabilities of terms given classes and joint probabilities of terms and classes. Jin et al. [30] proposed three new feature selection methods. The three methods were, respectively, designed based on term frequencies in classes, document frequencies in classes, and the integration of the two. Verma and Sahu [31] proposed a method for multi-label text classification. The method selected features based on relevance between features and labels. Ige and Gan [32] proposed a two-objective method which selected features by ensemble filters and then reduced the quantity of the selected features. Mohanrasu et al. [33] developed a ridge-regression-based method which evaluated relationship between features by a relation matrix. Cekik [34] introduced a method based on the knowledge of class, document, and document–class. Farek and Benajdja [35] designed a method which considered mutual information and semantic correlation. Liu et al. [36] developed several improved versions of feature selection methods based on frequency and probability.

There are methods which perform feature selection by term frequencies or document frequencies. Ashokkumar et al. [37] developed a new method which combined the information of TF-IDF (term frequency–inverse document frequency), weights of nouns, and the reciprocal of the total number of terms in documents. Garg [38] introduced a feature selection method based on a co-occurrence term network. A node of the network represented a term, while an edge represented the relationship between two adjacent terms. Okkalioglu [39] proposed a method based on the distances between terms.

Some studies pay attention to redundancy of terms and explanation. Lazhar and Amira [40] developed a method based on the assumption that semantically related terms lead to redundancy. Rehman et al. [41] proposed a method to address the problem of explanation of CNNs.

2.1.2. Selecting Methods

Selecting methods apply various kinds of methods to select discriminative subsets of terms, instead of setting the parameter N manually.

Evolutionary algorithms can be used for selecting discriminative subsets of terms. Chantar et al. [42] proposed an improved binary grey wolf optimizer and used it to design a new feature selection method. Alalyani and Larabi-Marie-Sainte [43] proposed a new method based on the firefly algorithm. Thirumoorthy and Muneeswaran [44] used a hybrid poor and rich optimization algorithm to select optimal subsets in their feature selection method. Hosseinalipour et al. [45] proposed two methods which were developed based on two versions of the farmland fertility algorithm. Adel et al. [46] proposed a new method based on the binary bat algorithm. Priya and Karthika [47] developed an embedded method to generate an optimal feature subset and then to determine the importance of features. Singh et al. [48] proposed a method for improving the global solution convergence. Wu et al. [49] designed a new method based on their proposed binary black hole algorithm. Kaya et al. [50] integrated migrating birds optimization feature selection with text classifiers to improve accuracy and efficiency. Msallam et al. [51] introduce three binary-fire-hawk-based methods for unsupervised feature selection. Nachaoui et al. [52] developed an improved PSO (particle swarm optimization) method which incorporated several components including weight, constriction factor, chaotic map, and a fitness function for penalty. Dhal and Azad [53] proposed a two-stage method which performed filter feature selection

and then multi-objective wrapper feature selection. Farek and Benaidja [54] proposed an improved binary PSO method based on mutual information and a local search method.

Discriminative subsets of terms can also be selected by various other methods. El-Hajj and Hajj [55] introduced a method which transformed feature selection to an optimal problem. The method selected discriminative terms by maximizing the difference between classes and minimizing the difference between terms in the same class. Saeed and Al Aghbari [56] proposed a method based on association rules to generate discriminative subsets of terms for each class. Similarly, Farghaly and Abd El-Hafeez [57] presented a method which used the Apriori algorithm to recognize discriminative subsets of terms for each class. Sagbas [58] introduced a method which generated subsets by using a greedy search on ranked features. Liu et al. [59] proposed a two-stage method for text classification, based on the designs of multiple objects and multiple filters.

2.1.3. Summary

Both ranking methods and selecting methods perform feature selection mainly according to supervised information (relationships between terms and classes) and unsupervised information (e.g., document frequencies and term frequencies). These two kinds of information are semantic information. Supervised information is helpful to evaluate the significance of terms under classes. However, supervised information is hard to obtain because most text documents are not assigned class labels in practical scenarios. Moreover, supervised information may bring negative effects on feature selection in imbalanced datasets because the selected subsets of terms may contain more information of frequent classes. Unsupervised information is naturally contained in text documents, but it is limited because some important information (e.g., position information of terms in sentences) of text documents is filtered out.

Ranking methods rank terms by evaluation functions and select subsets of terms by using the manual parameter N. Ranking methods have two advantages. On the one hand, the evaluation of each term can be performed separately. One the other hand, the sizes of discriminative subsets of terms can be controlled manually, according to practical necessity. In the aspect of core ideas, selecting methods in our literature review can be divided into two types. One of the two types ranks terms and selects discriminative subsets using other algorithms (e.g., probability models or optimization algorithms) according to the ranked terms. The other type selects discriminative subsets directly by using optimization algorithms. Compared with ranking methods, selecting methods create much computational complexity. Moreover, most of the selecting methods are not flexible to control the sizes of discriminating subsets. In summary, ranking methods are more flexible and efficient than selecting methods.

2.2. CNNs for Text Mining

CNNs have been widely used in tasks related to images, including image compression [60], image classification [61], and detection of image forgery [62]. At the same time, CNNs have also been used in the tasks of text representation, text classification, and semantic analysis such as sentiment analysis and opinion analysis.

2.2.1. Text Representation

CNNs have been used as the tool for achieving text representation in previous studies. Wang et al. [63] developed a complex CNN with probability distribution settings for text representation. Xu et al. [64] designed a text representation solution by integrating a CNN and a variational autoencoder. Gasmi et al. [65] conducted the task of effectively retrieving medical images by using a CNN to achieve representation of queries. Liu et al. [66] and Wahyu Trisna et al. [67] utilized a CNN to discover pattens to better understand a text.

2.2.2. Text Classification

CNNs can be used as text classifiers. Wu et al. [68] developed a CNN which could capture context information for text classification. Xu et al. [69] developed a CNN classification model which applied context information of words as extra input data. Alsaleh and Larabi-Marie-Sainte [43] proposed a classifier based on GA and a CNN. GA was applied to optimize the parameters of the CNN. Butt et al. [70] proposed an improved CNN model. This model was mainly constructed by multiple convolution layers, a maxpooling layer, an averaging pooling layer, and a fully connected layer. Liang et al. [71] proposed a CNN with multiple attention settings for text classification. Liu et al. [72] developed a CNN classifier with multiple convolution layers and multiple pooling layers. Zhao et al. [73] also proposed a similar CNN classifier. Qorich and EL Quazzani [74] used a CNN structure of three convolution layers, a pooling layer, and three fully connected layers for text sentiment classification. Thekkekara et al. [75] developed a classification model for detecting depression. The model was constructed by combining a CNN, long short-term memory (LSTM), and an attention mechanism. Guo et al. [76] utilized TextCNN as a student model for their proposed distillation model.

CNNs can also be used to extract deep features of terms for text classification. Liu et al. [77] developed a complex classifier which used a CNN to extract a representation of input data. Zeng et al. [78] proposed a CNN with several kernels to process input text data. Lyu and Liu [79] introduced a text classification model based on a recurrent neural network (RNN) and a CNN. In this model, the CNN was used for capturing the importance of words. Wang et al. [80] used a CNN with an attention mechanism to discover related words in text data. Liu et al. [81] used a graph CNN to capture sentiment features of text in their proposed classification model. Zhou et al. [82] used a CNN framework to capture semantic information at multiple levels for text classification. Xiong et al. [83] and Li et al. [84] utilized a CNN to extract local features from text.

2.2.3. Semantics Analysis

CNNs have been applied in sentiment analysis. Huang et al. [85] predicted sentiment strength by using a CNN to analyze three types of embeddings of words. Krishnan et al. [86] conducted sentiment analysis by using a CNN which was improved by an optimization algorithm. Usama et al. [87] used a CNN to capture high-level features from input data. Wang et al. [88] integrated two types of CNN for their sentiment analysis model. One CNN was used to analyze context information, while the other one was used for capturing sentiment information. Ghorbanali et al. [89] proposed an ensemble network of multiple CNNs for multimodal sentiment analysis. Huang et al. [90] developed a model which analyzed embedding of sentences and sentiment words by using parallel CNN structures. Murugaiyan and Uyyala [91] proposed a serial CNN structure for recognizing emotion from speeches. Mutinda et al. [92] designed a model for sentiment analysis by integrating BERT and CNN. Alnowaiser [93] utilized a CNN to extract features for citation sentiment analysis. He and Abisado [94] proposed a model for sentiment analysis on short film comments. The model utilized a CNN to generate local features of word vectors.

CNNs have also been involved in various other tasks of semantics analysis. Chen et al. [95] applied a CNN to detect verbal aggression on Twitter. Qiu et al. [96] presented a model based on a CNN for automatically discriminating legal documents. Heo et al. [97] proposed a model for predicting atrial fibrillation. The model used a CNN to extract features. Li and Yin [98] compared the performance of different models based on a CNN and RNN in the task of analyzing semantic features. Jian et al. [99] proposed a hybrid model for measuring the readability of English text. The model was designed based on a CNN, RNN, and attention mechanism. Qiu et al. [100] proposed a CNN structure

with a mechanism for filtering words for extracting spatial relation from text. Boukhers et al. [101] proposed a CNN model for ICD code prediction of clinical text. The model is constructed by parallel structures of multiple convolution layers. Muppudathi and Krishnasamy [102] proposed an improved CNN model for detecting anomalies on social media. The improvement in the model reflected on the pooling operation and the activation function. Fan [103] utilized a CNN to generate local features for detecting cyberbullying. Zeng [104] developed a CNN-based model to analyze emotional semantics from text. Faseeh et al. [105] proposed a hybrid model for identifying duplicate questions. The model learned local and long-term dependencies by using a CNN and LSTM, respectively. Wu et al. [106] developed a multimodal CNN for fake news detection.

2.2.4. Summary

CNNs have been used as text classifiers or feature extractors to discover deep features of terms in text mining. However, the deep features extracted by CNNs are unable to be explicitly interpreted. The use of deep features discovered by CNNs in feature selection can fill the gap between implicit deep information and explicit semantic information. Moreover, deep features extracted by CNNs are also the supplementary information for the limited information contained in term frequencies.

2.3. Research Motivation

This paper is motivated by the following discussions on related work.

Ranking methods are more flexible than selecting methods. Ranking methods are flexible and efficient at evaluating terms separately and controlling the dimensions of discriminative subsets conveniently. Compared with ranking methods, some of the selecting methods must rank terms by using ranking methods before selecting discriminative subsets. Furthermore, selecting methods select discriminative subsets by using complex techniques such as probability models or optimization algorithms. In summary, selecting methods create much computational complexity, compared with ranking methods, so we design our method CNNFS as a ranking method.

Previous feature selection methods use supervised information or unsupervised information for feature selection. The two kinds of semantic information are limited. Firstly, supervised information is helpful to evaluate significance of terms under classes, but it is hard to obtain because most text documents in practical scenarios are not assigned class labels. Secondly, supervised information may bring negative effects on feature selection in imbalanced datasets. The selected subset of terms may contain more information about frequent classes. Last but not least, unsupervised information is naturally contained in text documents, but it is limited because semantic information (e.g., position information of terms in sentences) of text documents is filtered out, especially in the bag-of-words model. According to the above discussion, CNNFS is designed to perform feature selection by using unsupervised information.

Deep information extracted from terms by a CNN can be utilized for improving the performance of feature selection. CNNs have been demonstrated as an effective deep model to extract deep features from images and have also been used in text mining. Deep information extracted by CNNs can be useful supplementary information for the extension of limited supervised information or unsupervised information to achieve better feature selection performance. Based on the above discussion, CNNFS integrates unsupervised information and deep information extracted by a CNN to perform feature selection.

3. Methodology

This section discusses the methodology of this paper. Section 3.1 presents the research objectives of this paper based on the discussion in Section 2.3. Section 3.2 introduces the structure of our three-layer CNN. The structure of our CNN is specially designed for selecting discriminative terms in text. Section 3.4 illustrates our methodology to use the three-layer CNN for feature selection.

3.1. Research Objectives

According to Section 2.3, we determine the following four research objectives.

Objective 1: We aim to design a ranking method, CNNFS. A ranking method is flexible to evaluate the effectiveness of terms separately and control the size of discriminative subsets of terms.

Objective 2: CNNFS performs feature selection according to the combination of unsupervised term frequency information and deep information extracted by a CNN to achieve better performance.

Objective 3: CNNFS trains the CNN by using a flexible strategy to emphasize the efficiency of feature selection.

Objective 4: CNNFS achieves good performance in improving text classification accuracy, recognizing high proportion of semantic terms, and reducing sparsity of text.

Objectives 1 to 3 are achieved in the design of our method, CNNFS (Section 3.4). Objective 4 is achieved in experiments (Section 4).

3.2. Feature Selection Preliminary

This section describes the task of feature selection mathematically. Let $D = \{D_i | i = 1, 2, \ldots, |D|\}$ and $T = \{T_j | j = 1, 2, \ldots, |T|\}$ denote a set containing $|D|$ text documents and a set containing $|T|$ terms, respectively. All the terms in T belong to the text documents in D. Under the bag-of-words model, text documents in D are transformed to a $|D| \times |T|$ document–term matrix (shown in Figure 1). The ith text document is represented as $D_i = [t_{i1}, t_{i2}, \ldots, t_{i|T|}]^T$. The jth term in T is represented as $T_j = \left[t_{11}, t_{21}, \ldots, t_{|D|j}\right]$. t_{ij} represents the frequency of term T_j in document D_i. For text classification, feature selection is responsible for the task of identifying discriminative terms from T to construct a small subset $T'(T' \subseteq T, |T'| \ll |T|)$. Classifiers are able to achieve better performance by learning from T' instead of T because T' contains many fewer non-discriminative terms. Moreover, the training complexity is much lower because T' is much smaller than T.

3.3. Structure of the Three-Layer CNN

CNNFS trains a three-layer CNN to discover deep features in T_j. The structure of the CNN is shown in Figure 1. The CNN accepts T_j as input and outputs the deep feature $F(T_j)$. The three layers, namely, the smoothing layer, extending layer, and compressing layer, are specially designed to perform different processes for T_j. The smoothing layer is designed for smoothing T_j and reducing its sparsity of term frequencies. The extending layer is designed for extending information of T_j. The compressing layer is designed for further processing and compressing the output of extending layer so that the dimensions of T_j and $F(T_j)$ are the same. Table 1 summarizes the CNN architecture and training parameters.

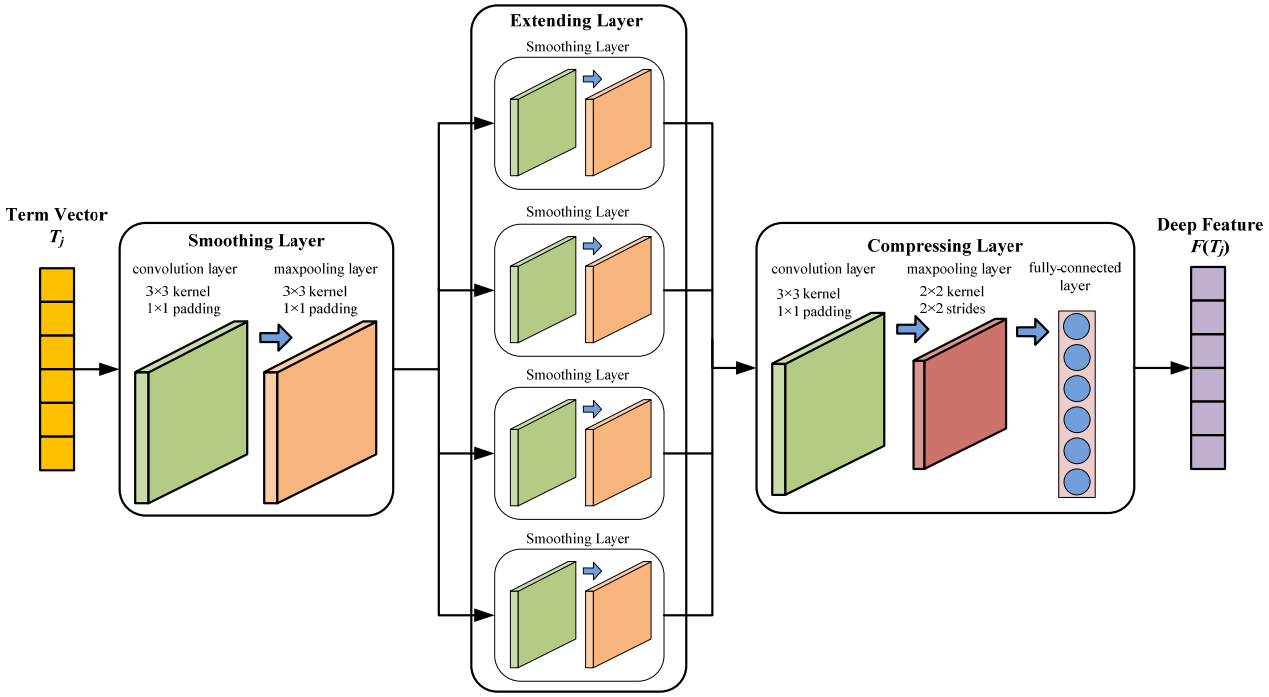


Figure 1. Structure of the three-layer CNN.

Table 1. Details of the CNN architecture and training parameters.

Smoothing Layer		Extending Layer		Compression Layer	
Layer	**Params**	**Layer**	**Params**	**Layer**	**Params**
A convolution layer	3 × 3 kernel 1 × 1 padding	Four smoothing layers		A convolution layer	3 × 3 kernel 1 × 1 padding
A maxpooling layer	3 × 3 kernel 1 × 1 padding			A maxpooling layer	2 × 2 kernel 2 × 2 strides
				A fully connected layer	$\|D\|\times\|D\|$ weight matrix

3.3.1. Basic Operations

There are three kinds of operations involved in our CNN structure: the $p \times p$ padding, the $k \times k$ kernel, and the $s \times s$ strides. These operations are used to adjust values of term vectors and control the dimensions of the output deep features.

3.3.2. Basic Layers

(1) Smoothing Layer

The smoothing layer is the first layer of our CNN, and it contains a convolution layer and a maxpooling layer. The input of this layer is the term vector $T_j = \left[t_{11}, t_{21}, \ldots, t_{|D|j}\right]$, and the output is the feature vector of T_j. The convolution layer and the maxpooling layer are used for discovering deep information of T_j. Furthermore, the convolution layer can also ease the problem of sparsity of T_j. Figure 2 illustrates structure of the smoothing layer. The convolution layer first processes the input $|D| \times 1$ T_j and creates a $|D| \times 1$ feature. Next, the maxpooling layer processes the feature and creates the final feature $SL(T_j)$. The * operator is the convolution operator.

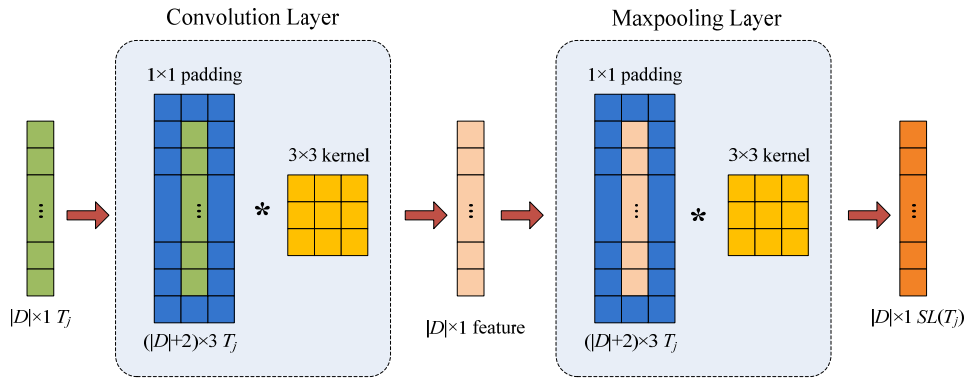


Figure 2. Structure of the smoothing layer.

(2) Extending Layer

The extending layer (shown in Figure 3) is constructed by four independent smoothing layers. Each smoothing layer processes the input $|D| \times 1\ SL(T_j)$ and creates an output $|D| \times 1\ SL(SL(T_j))$. Four of the four $|D| \times 1\ SL(SL(T_j))$s are concatenated by rows to create two $|D| \times 2\ SL(SL(T_j))$s. The two $|D| \times 2\ SL(SL(T_j))$s are further concatenated by columns to create the final output, $2|D| \times 2\ SL(SL(T_j))$. The extending layer aims to discover further deep information about $SL(T_j)$ by the smoothing layer. The concatenation operation is used to create more variation and offer more relationships among elements in T_j for the up-coming compressing layer to discover more deep information. The * operator is the convolution operator.

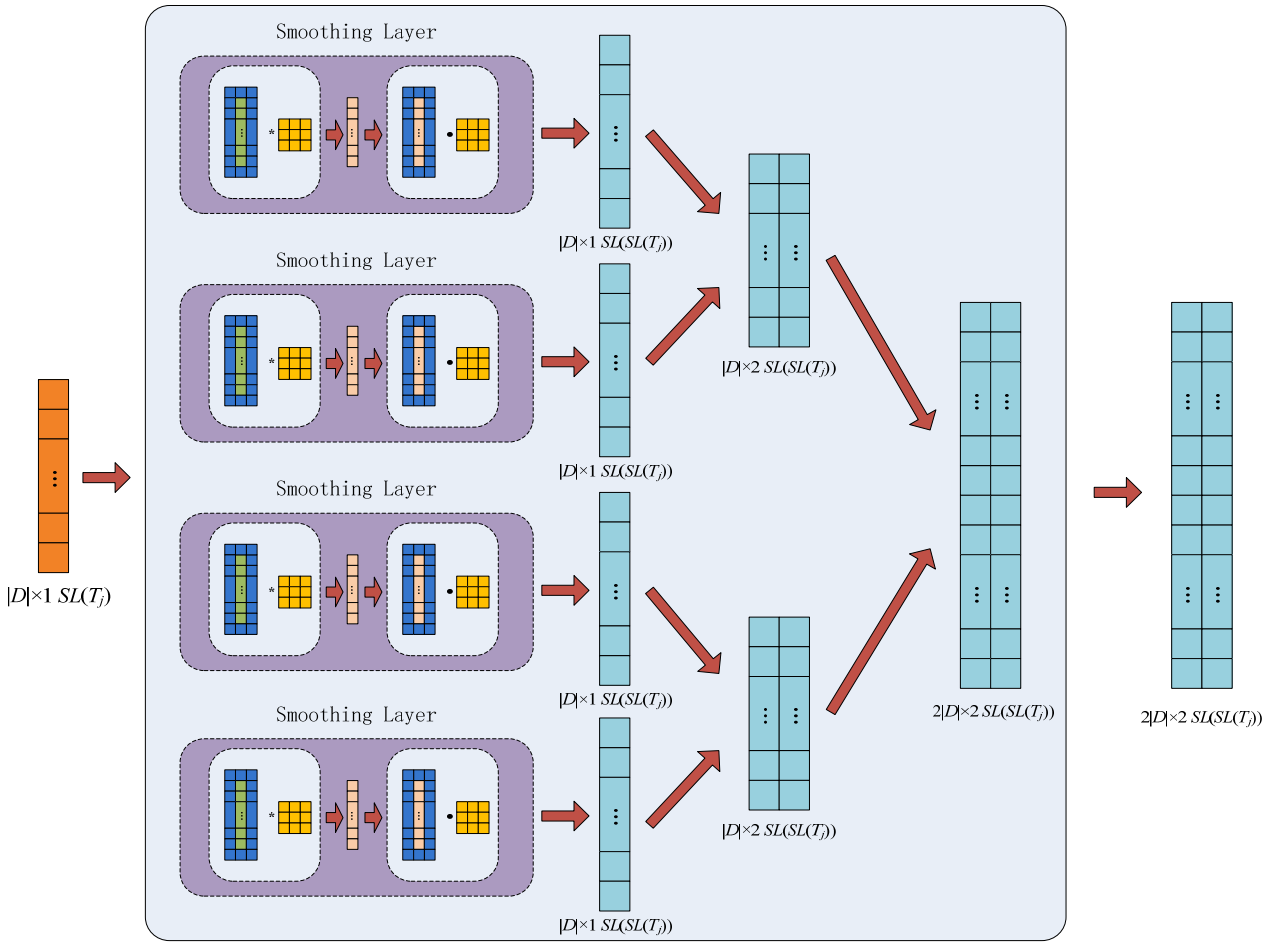


Figure 3. Structure of the extending layer.

(3) Compressing Layer

The final compressing layer (shown in Figure 4) processes the $2|D| \times 2\ SL(SL(T_j))$. The convolution layer is used to discover more deep information from $2|D| \times 2\ SL(SL(T_j))$. The maxpooling layer is used to compress the $2|D| \times 2$ output of the convolution layer to a final $|D| \times 1$ deep feature. Moreover, the maxpooling layer also reduces the sparsity of features to some extent. The $|D| \times 1$ deep feature is further processed by a fully connected layer to create the final deep feature, $F(T_j)$. We can see that $F(T_j)$ is of the same dimensions as T_j. The * operator is the convolution operator.

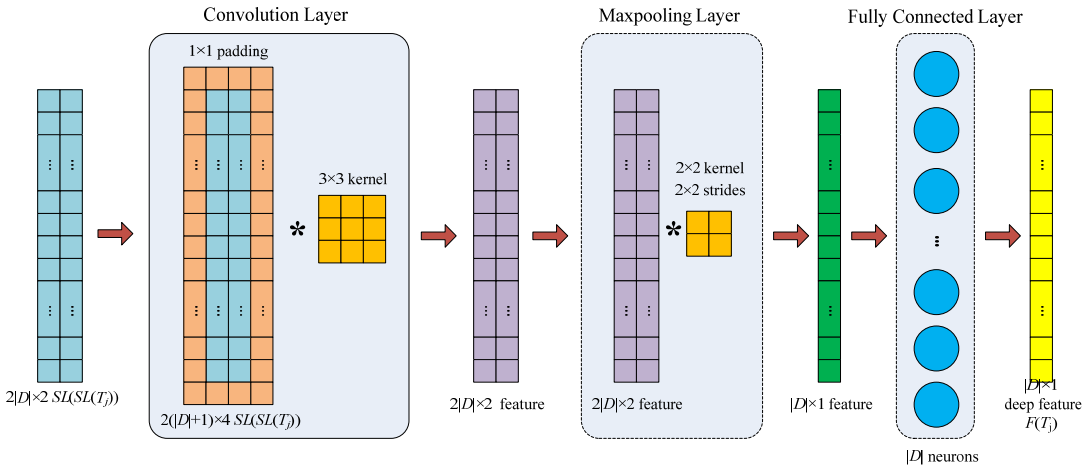


Figure 4. Structure of the compressing layer.

3.4. The Proposed Feature Selection Method, CNNFS

The flow chart of feature selection using CNNFS is shown in Figure 5. CNNFS performs feature selection in two stages: (1) First stage, a three-layer CNN (shown in Figure 3) is trained. (2) Second stage, discriminative terms are selected according to the trained CNN in the first stage.

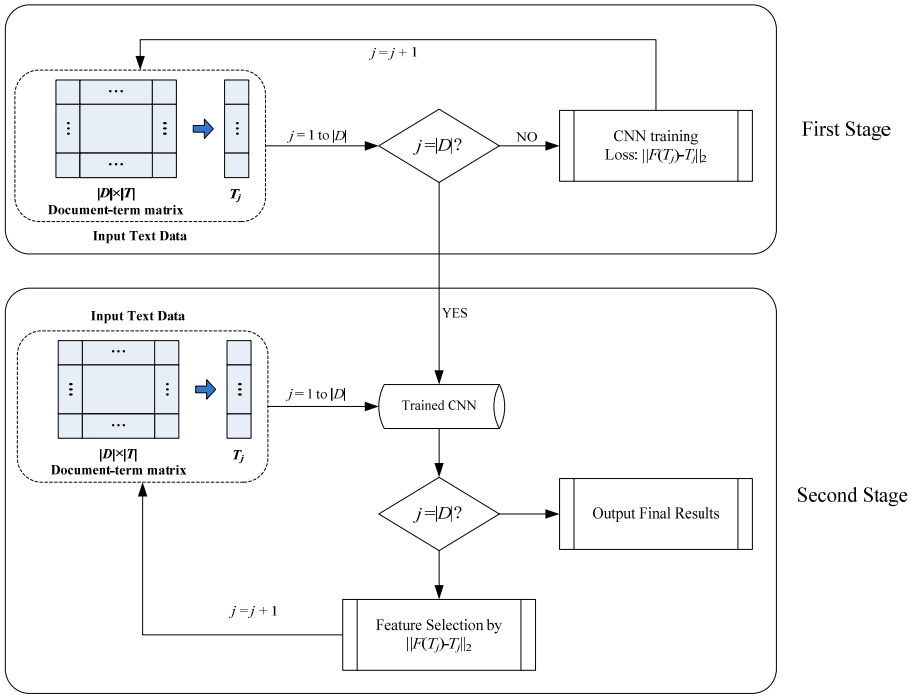


Figure 5. The flow chart of feature selection using CNNFS.

(1) First Stage

Each $T_j = \left[t_{11}, t_{21}, \ldots, t_{|D|j}\right]$ in the document–term matrix is input one by one to train the CNN. The output of the CNN is a deep feature of T_j: $F(T_j) = \left[f_{11}, f_{21}, \ldots, f_{|D|j}\right]$. $\left\|F(T_j) - T_j\right\|_2$ evaluates the difference between T_j and $F(T_j)$. A large value of $\left\|F(T_j) - T_j\right\|_2$ indicates that T_j contains a lot of deep information.

(2) Second Stage

In this stage, each T_j is input to the trained CNN to achieve its deep feature $F(T_j)$. The score of T_j $CNNFS(T_j)$ is calculated according to Formula (1). As mentioned above, a large value of $\left\|F(T_j) - T_j\right\|_2$ indicates that T_j contains a lot of deep information, so terms with large values of $CNNFS(T_j)$ are considered discriminative. The deep information is reflected in $F(T_j)$.

$$CNNFS(T_j) = \left\|F(T_j) - T_j\right\|_2 \tag{1}$$

(3) Algorithm

The process of performing CNNFS for feature selection is illustrated by the algorithm shown in Algorithm 1. The first stage starts at line 2 and stops at line 6. Function initializeCNN() initializes the CNN in Figure 3. Line 3 to line 6 trains the CNN using T_js. The CNN training stops at $j = |D|$. Our training strategy has three advantages in simplifying the training process and reducing training time. Firstly, the stop point ($j = |D|$) of CNN training is definite. Secondly, each term vector only needs to be input once. Finally, the convergence of the loss function is unnecessary. Line 7 to line 12 shows the second stage. $F(T_j)$ is calculated for each T_j by the trained CNN. The scores of T_j are stored in FS, the vector of length $|T|$. Elements in FS are then sorted in decreasing order. The output subset T' contains the top N discriminative terms in FS. N is a manual parameter which is set according to experiments or practical needs.

Algorithm 1. Algorithm of feature selection using CNNFS

Input: dtm, the $|D| \times |T|$ document-term matrix
N, the number of selected discriminating terms
Output: T', the subset of discriminating terms

```
1  Begin
2    # first stage
3    CNN = initializeCNN();
4    For j = 1 →|D|
5      TrainCNN(T_j, CNN);
6    End For
7    # second stage
8    FS = vector(length =|T|);
9    For j = 1 →|D|
10     F(T_j) = CNN(T_j);
11     FS[T_j] = ||F(T_j) − T_j||_2;
12   End For
13   # output results
14   T' = sort(FS, decreasing = True)[1 : N];
15   Return T';
16 End
```

(4) Complexity

According to [107], the complexity of a certain number of traditional supervised and unsupervised filter methods is $O(|D|\times|T|) + O[s(|T|)]$. $|D|$ represents the number of text documents. $|T|$ represents the total number of unique words in the $|D|$ documents. $O(|D|\times|T|)$ represents the complexity of scoring the $|T|$ unique words. $O[s(|T|)]$ represents the complexity of sorting the $|T|$ unique words according to their scores. According to Algorithm 1, the complexity of CNNFS is also $O(|D|\times|T|) + O[s(|T|)]$. Moreover, CNNFS has the advantages of utilizing more deep information than traditional filter methods.

4. Experiments

4.1. Datasets

Nine benchmark datasets are applied for comparison experiments. The details of these nine datasets are illustrated in Table 2. The first two datasets (CF and CR), respectively, contain user favorite description text and user review text of 12 different Toyota cars. They are generated from the original OpinRank Review dataset, which is applied in [108]. CNAE, utilized in [109,110], consists of business description texts about Brazilian companies, which are categorized into nine groups. IMDB, consisting of positive and negative movie reviews, is one of the subsets in the Sentiment Labeled Sentences Dataset in [111]. KDC and TTC are datasets which contains Turkish text of articles and news. KDC is utilized for experiments in [112–114]. TTC is introduced in [115]. The aforementioned datasets are obtained from the famous UCI repository. The WEBKB dataset [116,117] is a collection of web pages of computer science departments of four universities in January 1997. The R8 and R52 datasets are the subsets of the Reuters-21578 dataset [116,118–121], a collection of economic news published by Reuters in 1978.

Table 2. Details of the nine datasets.

Name of Dataset	Quantity of Documents	Quantity of Terms	Number of Classes	Description
CF	1007	1824	12	User favorite description text of 12 different Toyota cars
CR	1047	3956	12	User review text of 12 different Toyota cars
CNAE	1080	856	9	Business description text of Brazilian companies from nine groups
IMDB	1000	2422	2	Positive and negative movie review text
KDC	4007	13,201	8	Turkish text of articles and news
TTC	3600	3208	6	
WEBKB	4199	7684	4	Collection of web pages of computer science departments of four universities
R8	7674	17,231	8	Subset generated from the Reuters-21578 dataset
R52	9100	19,080	52	

There are two reasons for choosing these datasets in our comparative experiments: (1) The diversity of different topics (e.g., cars, business, movies, computers, and news) in the datasets guarantees the robustness and generalization of the results in comparative

experiments. (2) A certain number of researchers have used these datasets in experiments of several text mining tasks (e.g., text classification, opinion analysis, and information retrieval), so the reliability of these datasets can be guaranteed.

4.2. Evaluation Metrics

Support vector machine (SVM), the effective classification algorithm, and multinomial naïve Bayes (MNB), the classical text classification algorithm, are utilized to train classifiers in experiments. Classification accuracy [22] is used as an evaluation metric classifier. Tenfold cross-validation is applied in order to guarantee more accurate classification performance.

Six comparative methods (shown in Table 3) are introduced in our experiments. The NDM method was recently proposed by [122]. NDM is described in the view of binary class in the original paper, but it can be extended for the case of multi-class in the view of weighted averaging. The other five methods are usually used as the comparative methods in previous studies. All six feature selection methods select terms with high scores as discriminative terms.

Table 3. Comparative feature selection methods.

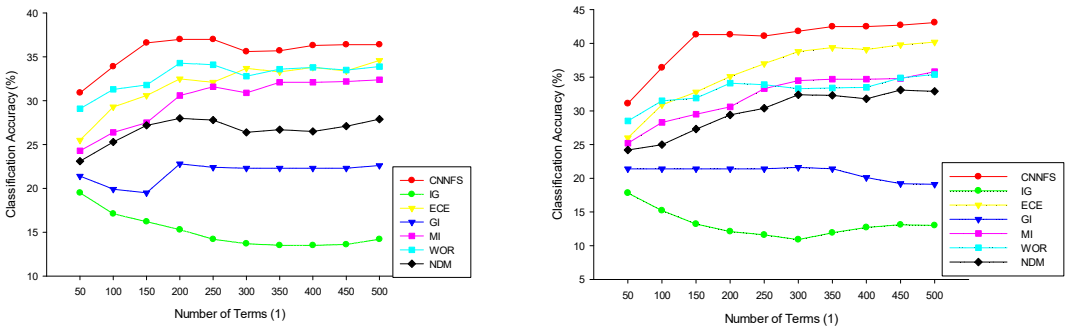
Method	Description
Information Gain (IG)	$IG(t_j) = P(t_j)\sum_{c_k \in CL} P(c_k \mid t_j) log \frac{P(c_k \mid t_j)}{P(c_k)} + P(\bar{t}_j)\sum_{c_k \in CL} P(c_k \mid \bar{t}_j) log \frac{P(c_k \mid \bar{t}_j)}{P(c_k)}$
Expected Cross Entropy (ECE)	$ECE(t_j) = P(t_j)\sum_{c_k \in CL} P(c_k \mid t_j) log \frac{P(c_k \mid t_j)}{P(c_k)}$
Mutual Information (MI)	$MI(t_j) = \sum_{c_k \in CL} P(c_k) log \frac{P(c_k \mid t_j)}{P(c_k)}$
Gini Index (GI)	$GI(t_j) = \sum_{c_k \in CL} P(c_k \mid t_j)^2$
Weighted Odd Ratio (WOR)	$WOR(t_j) = \sum_{c_k \in CL} P(c_k) log \frac{P(t_j \mid c_k)[1 - P(t_j \mid \bar{c}_k)]}{[1 - P(t_j \mid c_k)]P(t_j \mid \bar{c}_k)}$
Normalized Difference Measure (NDM)	$NDM(t_j) = \sum_{c_k \in CL} P(c_k) \frac{\lvert P(t_j \mid c_k) - P(t_j \mid \bar{c}_k) \rvert}{\min(P(t_j \mid c_k), P(t_j \mid \bar{c}_k))}$

4.3. Results and Discussions

This section presents experimental results and discusses these results in three different aspects: classification accuracy analysis, semantics analysis, and sparsity analysis. Classification accuracy analysis compares our CNNFS with six other methods on improving classification accuracy of SVM and MNB classifiers. Semantic analysis compares the terms selected by our method and comparative methods on the aspect of semantic meanings. Sparsity analysis compares sparsity of document–term matrices under T' selected by our method and other methods.

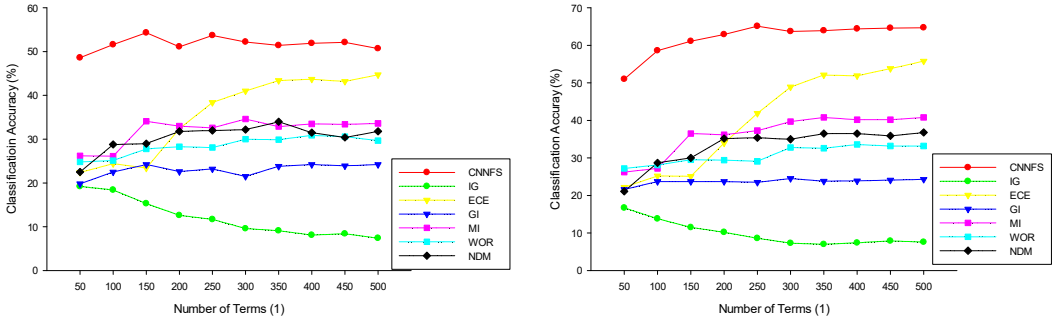
4.3.1. Classification Accuracy Analysis

Figures 6–14 show the classification accuracy of our method CNNFS and six other comparative methods. The specific classification accuracy is presented in Table A1 in Appendix A. Let N denote the number of terms. All the methods can be grouped into three levels.



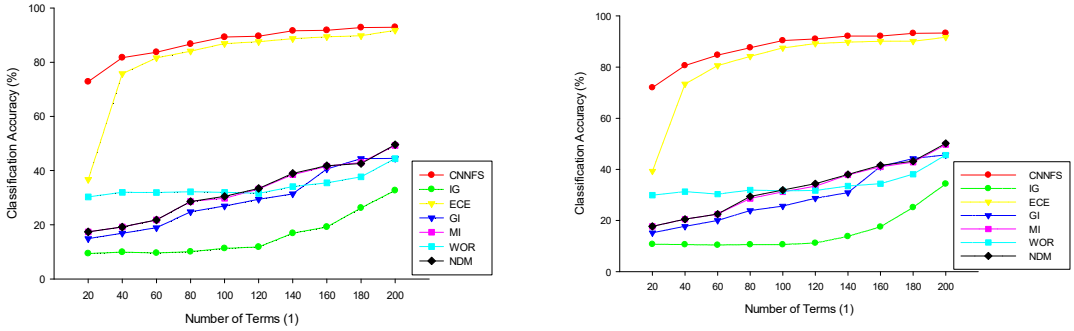
(**a**) SVM (**b**) MNB

Figure 6. Classification accuracy of the CF dataset.



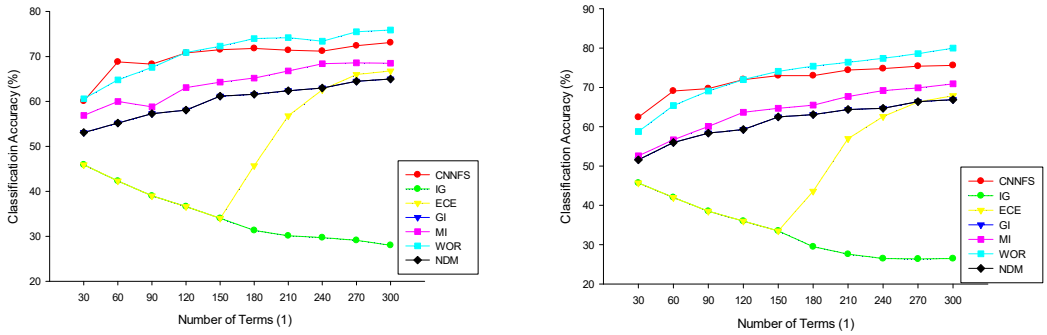
(**a**) SVM (**b**) MNB

Figure 7. Classification accuracy of the CR dataset.



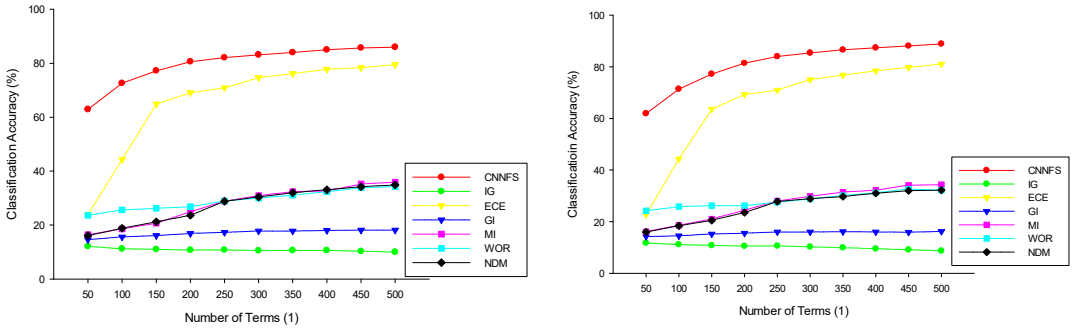
(**a**) SVM (**b**) MNB

Figure 8. Classification accuracy of the CNAE dataset.



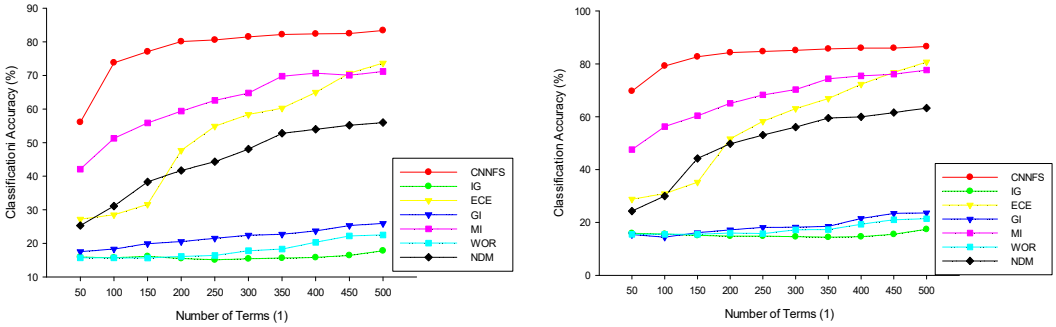
(**a**) SVM (**b**) MNB

Figure 9. Classification accuracy of the IMDB dataset.



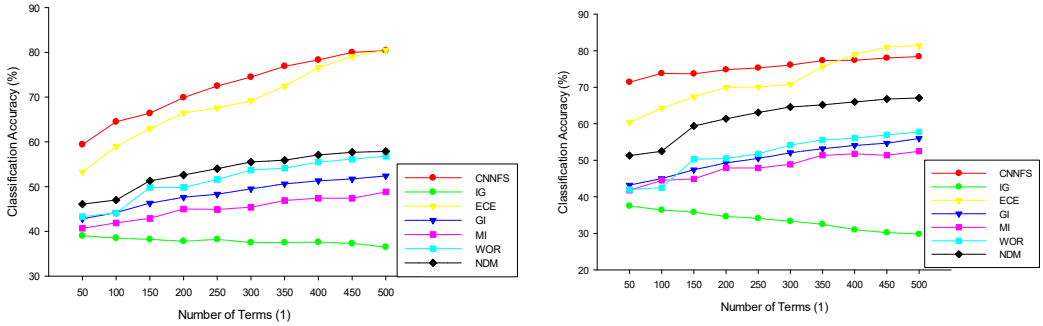
(**a**) SVM (**b**) MNB

Figure 10. Classification accuracy of the KDC dataset.



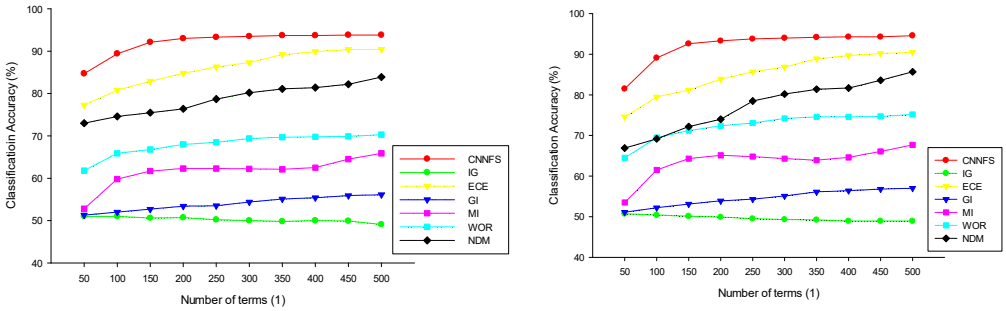
(**a**) SVM (**b**) MNB

Figure 11. Classification accuracy of the TTC dataset.



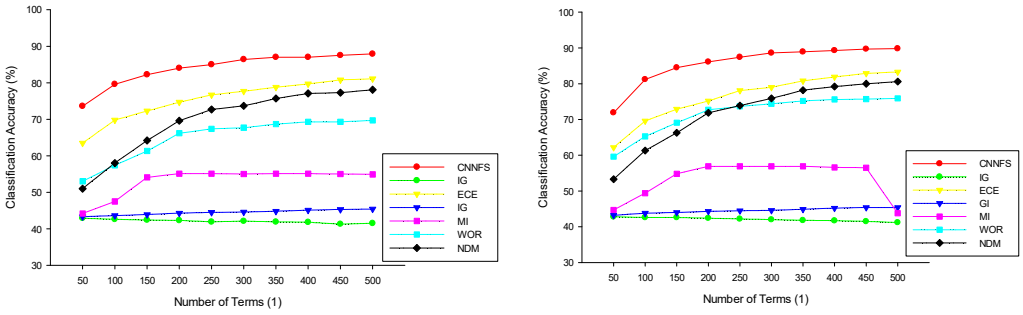
(**a**) SVM (**b**) MNB

Figure 12. Classification accuracy of the WEKBE dataset.



(**a**) SVM (**b**) MNB

Figure 13. Classification accuracy of the R8 dataset.



(**a**) SVM (**b**) MNB

Figure 14. Classification accuracy of the R52 dataset.

In Figure 6, Level 1 contains CNNFS. Level 2 contains ECE, MI, WOR, and NDM. ECE outperforms MI. ECE outperforms WOR when $N \geq 150$ for MNB. However, WOR outperforms ECE when $N \leq 250$ for SVM. WOR outperforms MI when $N \geq 50$ for SVM and $N \leq 250$ for MNB. NDM performs worst at Level 2, while GI surpasses IG at Level 3.

In Figure 7, Level 1 contains CNNFS. Level 2 contains ECE, MI, GI, WOR, and NDM. ECE achieves the highest accuracy in Level 2 when $N \geq 200$. MI outperforms WOR when $N \geq 50$ for SVM and $N \geq 100$ for MNB. MI outperforms NDM when $N \in [150, 200, 250, 300, 350, 400, 450, 500]$ for SVM and $N \geq 150$ for MNB. NDM outperforms WOR when $N \in [100, 150, 200, 250, 300, 350, 500]$ for SVM and $N \geq 200$ for MNB. GI achieves the lowest accuracy in Level 2. Level 3 contains IG.

In Figure 8, Level 1 contains CNNFS and ECE. CNNFS outperforms ECE. Level 2 contains MI, GI, WOR, and NDM. MI achieves similar accuracy to NDM. WOR achieves the highest accuracy in Level 2 when $N \leq 100$. GI achieves the lowest accuracy in Level 2 when $N \leq 160$. Level 3 contains IG.

In Figure 9, Level 1 contains CNNFS and WOR. WOR outperforms CNNFS when $N \geq 150$. Level 2 contains MI, GI, and NDM. MI outperforms GI and NDM. GI achieves similar accuracy to NDM. Level 3 contains ECE and IG. ECE outperforms IG when $N \geq 150$. ECE achieves similar accuracy to GI and NDM when $N \geq 240$.

In Figure 10, Level 1 contains CNNFS and ECE. CNNFS outperforms ECE. Level 2 contains MI, WOR, and NDM. WOR outperforms MI and NDM when $N \leq 200$. MI, WOR, and NDM achieve similar accuracy when $N \geq 200$. Level 3 contains IG and GI. GI outperforms IG.

In Figure 11, Level 1 contains CNNFS, and Level 2 contains ECE, MI, and NDM. ECE outperforms NDM when N $\geq$ 200. MI outperforms ECE and NDM. ECE outperforms NDM when $N \geq 200$. Level 3 contains IG, GI, and WOR. IG achieves the lowest accuracy in Level 3. IG outperforms WOR when $N \geq 50$ for SVM and when $N \geq 200$ for MNB. IG achieves similar accuracy to WOR when $N \leq 200$.

In Figure 12, Level 1 contains CNNFS and ECE. CNNFS outperforms ECE when $N \leq 450$ for SVM and $N \leq 350$ for MNB. Level 2 contains MI, GI, WOR, and NDM. NDM achieves the highest accuracy in Level 2. WOR achieves the second-highest accuracy in Level 2 when $N \geq 150$. GI outperforms MI when $N \geq 50$ for SVM and $N \geq 150$ for MNB. Level 3 contains IG.

In Figure 13, Level 1 contains CNNFS and ECE. CNNFS outperforms ECE. Level 2 contains MI, WOR, and NDM. Generally, NDM outperforms WOR, while WOR outperforms MI. Level 3 contains IG and GI. GI outperforms IG.

In Figure 14, Level 1 contains CNNFS. Level 2 contains ECE, WOR, and NDM. ECE achieves the highest accuracy in Level 2. NDM outperforms WOR when $N \geq 150$ for SVM and $N \geq 300$ for MNB. Level 3 contains IG, GI, and MI. Generally, MI outperforms GI, while GI outperforms IG.

Furthermore, *t*-tests are conducted to statistically validate the performance of CNNFS. The *p*-values of the *t*-tests are given in Table A2 in Appendix A. In most cases (except the four cases of bold *p*-values in Table A2 (3) and (4)), the *p*-values are smaller than 0.05, indicating a significant difference between the performance of CNNFS and the performance of comparative methods.

In summary, CNNFS outperforms six other comparative methods in most of the classification cases. IG generally achieves the lowest accuracy among all the methods. The performance ECE, GI, MI, WOR, and NDM is not stable in experiments.

4.3.2. Semantics Analysis

This section presents discussions on the semantics of terms selected by CNNFS and comparative methods. For better illustration, we select semantic terms which are related to the topics of the corresponding datasets out of the top 20 terms from CF, CR, IMDB, WEBKB, R8, and R52. Terms in other datasets are unable to be recognized because they are replaced by variable names (e.g., "feature X"). The semantic terms in the top 20 terms of the datasets are summarized in Figure 15. The bold terms are the manually selected semantic terms of datasets (all the terms are stemmed, and all the uppercase letters are transformed to lowercase). The percentages in circles indicate the percentage of semantic terms in the top 20 terms which are selected by feature selection methods.

The CF dataset relates to topic of Toyota cars. The semantic terms are considered as car features or sentiment words. Among the top 20 terms selected by CNNFS, 95% of them are semantic terms. Following CNNFS, ECE also achieves relatively good performance because 45% of its selected top 20 terms are semantic terms. Other comparative methods select low percentages of semantic terms for their lists of the top 20 terms. The case of the CR dataset is similar to that of the CF dataset.

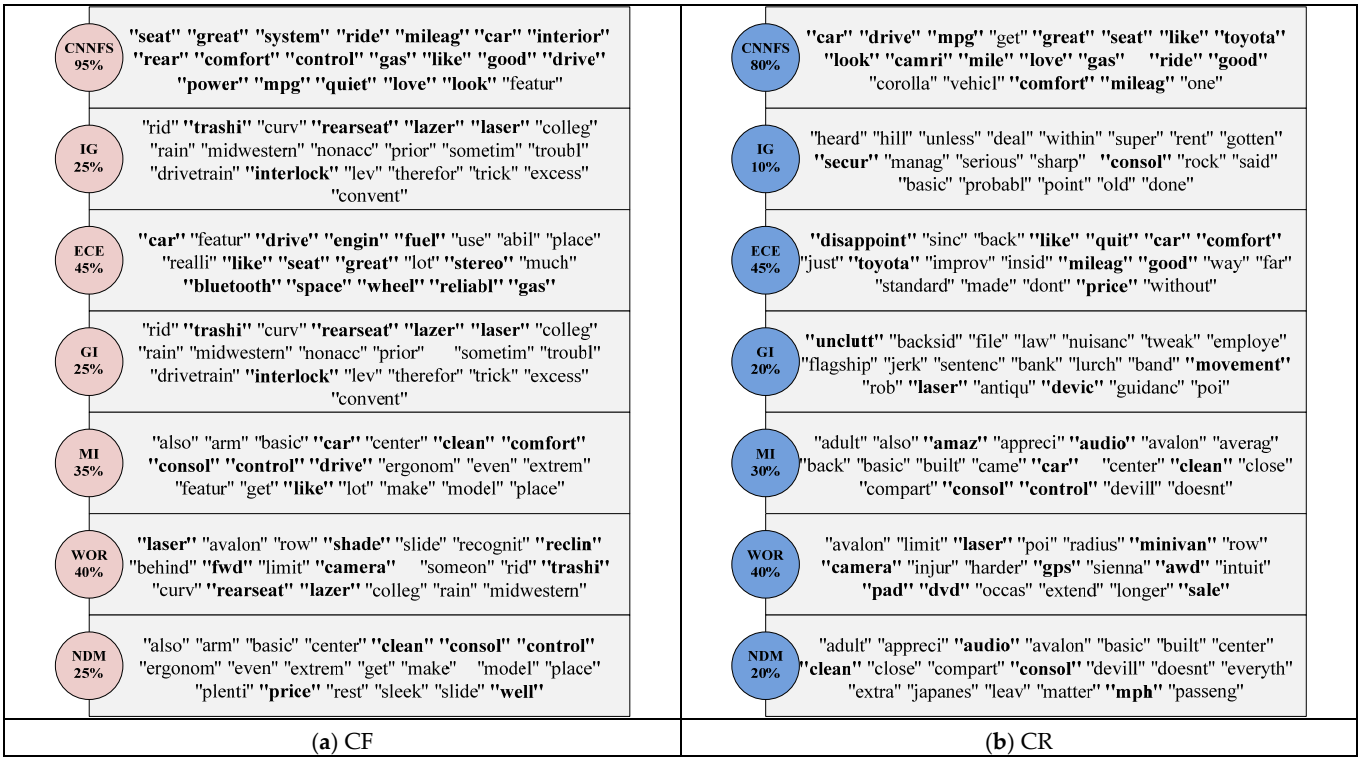


Figure 15. *Cont.*

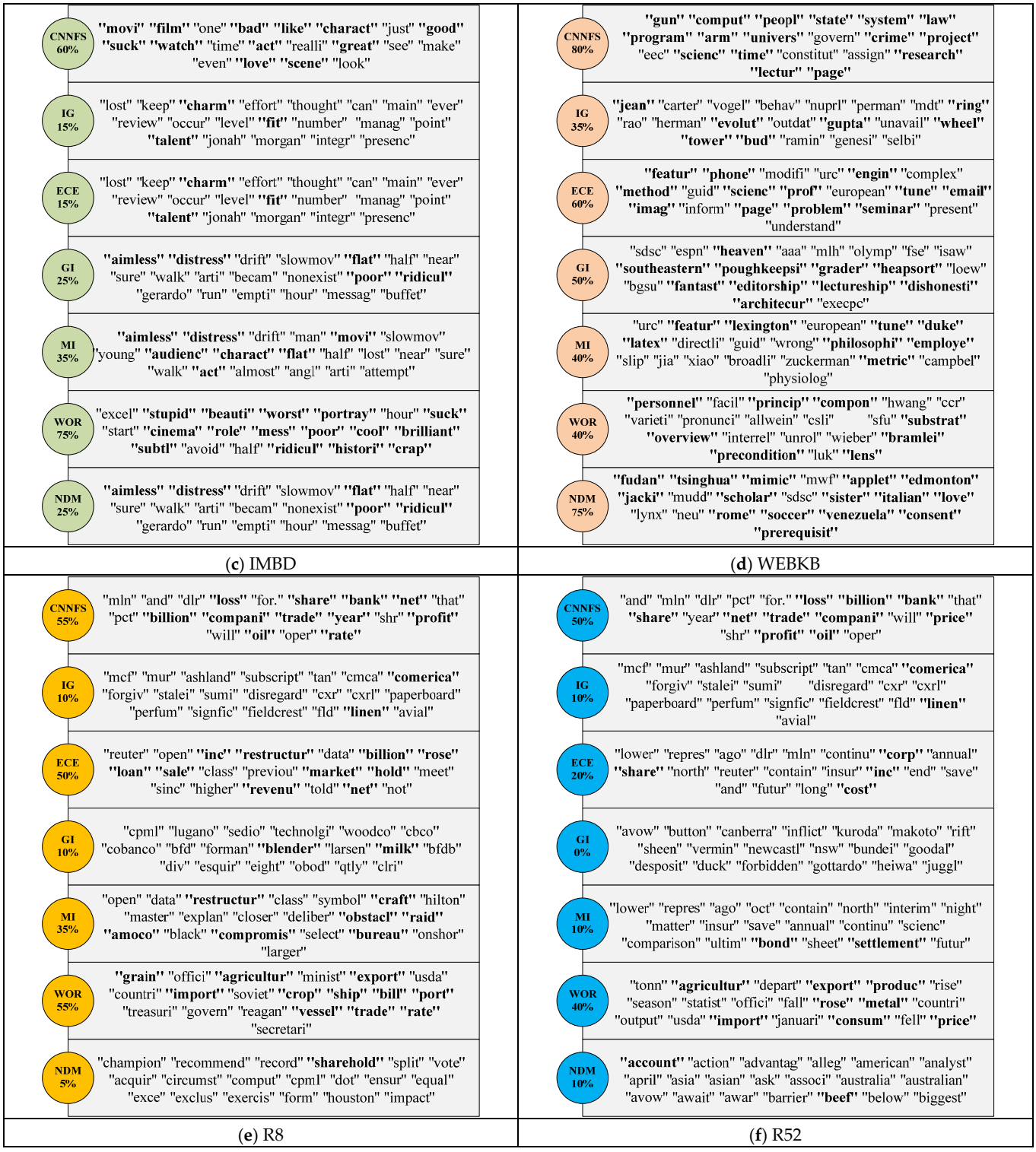


Figure 15. Terms related to topics of datasets in the top 20 terms.

The IMDB dataset contains positive or negative opinions on movies. The semantic terms are considered as movie features and sentiment words. As we can see, 60% of the selected top 20 terms of CNNFS are related to movie features and sentiment words. In addition, WOR performs better than CNNFS. Low percentages of semantic words are provided by other comparative methods in their selected top 20 terms.

The WEBKB dataset contains text from web pages from departments of computer science from four universities. The semantic terms are considered as nouns due to complexity

of the contents of the dataset. CNNFS offers 80% of the semantic terms in its selected top 20 terms. NDM selects 75% of its top 20 terms as semantic terms. The percentages offered by comparative methods are relatively low.

The R8 and R52 datasets contain economic news. The semantic terms are considered as terms related to economy. Among the selected top 20 terms, CNNFS provides 50% and 55% semantic terms, respectively, for the two datasets. ECE and WOR also perform well in the R8 dataset (50% and 55%, respectively). WOR offers a relatively high percentage (40%) of economy-related terms in the R52 dataset. Relatively, other comparative methods do not perform well in these two datasets.

In summary, the selected top 20 terms selected by CNNFS are more semantically related to the topics of datasets than comparative methods.

4.3.3. Sparsity Analysis

In this section, we evaluate the sparsity of document–term matrices, which are constructed using the terms selected by CNNFS and comparative methods. For better illustration, the number of terms is set as 500, that is, $T' = 500$. The results are shown in Figure 16. It is obvious that CNNFS achieves the best performance in reducing the sparsity of document–term matrices in eight of the nine datasets. Moreover, the effects of the sparsity reduction are significant. The sparsity reduction effect generally falls in the range of [2%, 8%].

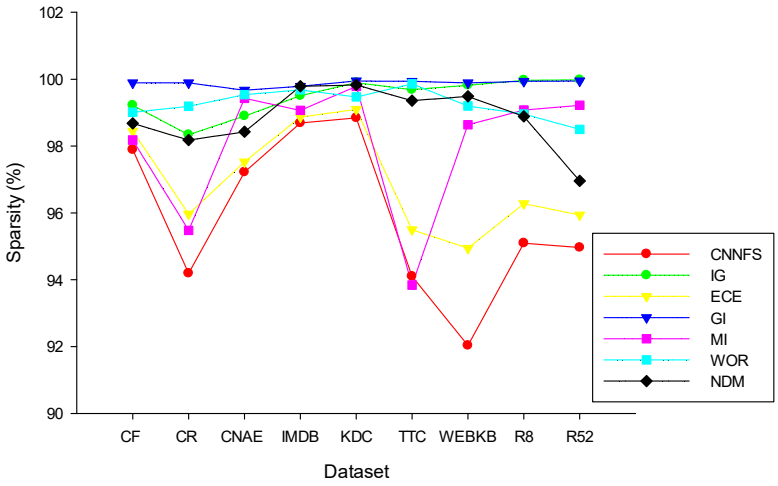


Figure 16. Sparsity of document–term matrices.

5. Conclusions

This paper proposes a CNN-based feature selection method, aiming to improve text classification accuracy. Our method constructs and trains a three-layer CNN to capture deep features of terms and then selects discriminative terms by the trained CNN. Terms with a lot of deep information are considered discriminative. The results in experiments demonstrate that CNNFS achieves more stable and better performance than comparative methods.

Our method, CNNFS, has three advantages: (1) CNNFS applies a deep learning model CNN to extract deep features of terms and selects discriminative terms by means of integrating term frequencies and deep features. Deep features are another view of information used as the supplementary information for the limited information contained in term frequencies. (2) Experimental results indicate that CNNFS outperforms some existing methods (as comparative methods) in the tasks of text classification, semantic terms selection, and sparsity reduction.

The research in this paper has limitations. (1) In CCNFS, the extracted deep information by the CNN is local information in text [83,84]. Local information is not that capable of reflecting global information in text. Future studies can focus on designing hybrid-deep-neural-network-based feature selection methods to leverage both local and global information, for example, a feature selection method which utilizes a CNN to extract local information and Bi-LSTM to extract global dependency information. (2) CNNFS ignores the relationship between words. The word relationship helps to evaluate the relative importance of a word among other words in the same document. Future studies can make efforts to integrate relationship information into feature selection by tools such as dependency parsing and topic modeling. (3) CNNFS is conducted based on bag-of-words frequency information which is a kind of statistical information. Pre-trained large language models (LLMs) are helpful to achieved semantic information about words. The semantic information is also useful for selecting discriminative words. Future studies can pay attention to integrating frequency information and semantic information, for more effective feature selection.

Author Contributions: Conceptualization, J.X. and M.H.; Methodology, J.X. and M.H.; Investigation, J.X. and M.H.; Writing—original draft, J.X. and M.H.; Writing—review & editing, J.X. and M.H.; Funding acquisition, J.X. and M.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Guangdong Philosophy and Social Sciences Fund, Grant No. GD25CTS08; Guangzhou Science and Technology Plan Project, Grant No. 2025A04J3937.

Data Availability Statement: The data used to support the findings of this study is available from the corresponding author upon request.

Conflicts of Interest: The authors declare that they have no conflicts of interest.

Appendix A

Table A1. Classification accuracy in datasets (unit: %).

(1) CF Dataset							
SVM	**CNN**	**IG**	**ECE**	**GI**	**MI_avg**	**WOR**	**NDM**
50	30.9	19.5	25.5	21.4	24.3	29.1	23.1
100	33.9	17.1	29.3	19.9	26.4	31.3	25.3
150	36.6	16.2	30.6	19.5	27.5	31.8	27.2
200	37	15.3	32.5	22.8	30.6	34.3	28
250	37	14.2	32.1	22.4	31.6	34.1	27.8
300	35.6	13.7	33.7	22.3	30.9	32.8	26.4
350	35.7	13.5	33.3	22.3	32.1	33.6	26.7
400	36.3	13.5	33.8	22.3	32.1	33.8	26.5
450	36.4	13.6	33.4	22.3	32.2	33.5	27.1
500	36.4	14.2	34.6	22.6	32.4	33.9	27.9

Table A1. *Cont.*

(1) CF Dataset							
MNB	**CNN**	**IG**	**ECE**	**GI**	**MI_avg**	**WOR**	**NDM**
50	31.08	17.8	26	21.4	25.2	28.5	24.2
100	36.4	15.2	30.9	21.4	28.3	31.5	25
150	41.3	13.2	32.8	21.4	29.5	31.9	27.3
200	41.3	12.1	35.1	21.4	30.6	34.1	29.4
250	41.1	11.6	37	21.4	33.3	33.9	30.4
300	41.8	10.9	38.8	21.6	34.5	33.3	32.4
350	42.5	11.9	39.4	21.4	34.7	33.4	32.3
400	42.5	12.7	39.1	20.1	34.7	33.5	31.8
450	42.7	13.1	39.8	19.2	34.8	34.9	33.1
500	43.1	13	40.2	19.1	35.8	35.4	32.9
(2) CR Dataset							
SVM	**CNN**	**IG**	**ECE**	**GI**	**MI_avg**	**WOR**	**NDM**
50	48.6	19.2	22.4	19.8	26.2	24.8	22.5
100	51.6	18.4	24.4	22.5	26.1	25.1	28.8
150	54.3	15.3	23.4	24.2	34.1	27.8	29
200	51.1	12.6	32.4	22.6	33	28.3	31.8
250	53.7	11.7	38.4	23.2	32.6	28.1	32
300	52.2	9.6	41	21.5	34.6	30	32.2
350	51.4	9.1	43.4	23.8	32.9	29.9	34
400	51.9	8.1	43.7	24.2	33.5	30.9	31.5
450	52.1	8.4	43.2	23.9	33.4	30.6	30.4
500	50.7	7.4	44.7	24.2	33.6	29.7	31.8
MNB	**CNN**	**IG**	**ECE**	**GI**	**MI_avg**	**WOR**	**NDM**
50	51	16.7	22.2	21.6	26.2	27.2	21.1
100	58.6	13.8	25.2	23.7	27.2	28.1	28.7
150	61.1	11.5	25.1	23.7	36.5	29.5	30
200	62.9	10.2	34	23.7	36.2	29.4	35.2
250	65.1	8.6	41.9	23.5	37.3	29.1	35.4
300	63.7	7.3	48.9	24.5	39.7	32.8	35
350	63.9	7	52.1	23.8	40.8	32.6	36.5
400	64.4	7.4	51.9	23.9	40.2	33.6	36.5
450	64.6	7.9	53.8	24.1	40.2	33.2	35.9
500	64.7	7.6	55.8	24.3	40.8	33.2	36.8

Table A1. *Cont.*

(3) CNAE Dataset							
SVM	**CNN**	**IG**	**ECE**	**GI**	**MI_avg**	**WOR**	**NDM**
20	72.8	9.4	36.7	14.9	17.4	30.3	17.4
40	81.7	9.9	75.8	16.9	19.2	32	19.2
60	83.7	9.6	81.6	18.9	21.8	31.9	21.8
80	86.7	10.1	84.1	24.8	28.6	32.2	28.6
100	89.3	11.3	86.9	26.9	29.8	32	30.5
120	89.6	11.8	87.6	29.4	33.2	31.6	33.4
140	91.6	16.9	88.7	31.4	38.5	34.1	39
160	91.8	19.2	89.4	40.6	41.6	35.5	41.8
180	92.8	26.2	89.8	44.4	43	37.7	42.6
200	92.9	32.7	91.8	44.5	49.3	44.4	49.6
MNB	**CNN**	**IG**	**ECE**	**GI**	**MI_avg**	**WOR**	**NDM**
20	72	10.7	39.3	15.2	17.7	29.9	17.7
40	80.6	10.6	73.4	17.7	20.5	31.3	20.5
60	84.7	10.4	80.6	20	22.5	30.3	22.5
80	87.6	10.6	84.2	23.9	28.6	31.9	29.4
100	90.4	10.6	87.5	25.6	31.2	31.6	31.9
120	91	11.2	89.2	28.7	33.4	31.7	34.4
140	92.1	13.8	89.8	30.9	37.8	33.5	38
160	92.1	17.5	90.1	41.1	41	34.4	41.6
180	93.2	25.1	90.1	44.2	42.8	38.1	43.2
200	93.3	34.3	91.7	45.7	49.7	45.6	50.2
(4) IMDB Dataset							
SVM	**CNN**	**IG**	**ECE**	**GI**	**MI_avg**	**WOR**	**NDM**
30	60.1	45.9	45.9	53.1	56.9	60.6	53.1
60	68.8	42.3	42.3	55.2	60	64.8	55.2
90	68.3	39	39	57.3	58.8	67.6	57.3
120	70.8	36.6	36.6	58.1	63.1	70.9	58.1
150	71.5	34	34	61.2	64.3	72.3	61.2
180	71.8	31.3	45.7	61.6	65.2	74	61.6
210	71.4	30.1	56.8	62.4	66.8	74.2	62.4
240	71.2	29.7	62.6	63	68.4	73.4	63
270	72.4	29.1	66	64.5	68.6	75.5	64.5
300	73.1	28	66.8	65	68.5	75.9	65

Table A1. *Cont.*

(4) IMDB Dataset							
MNB	**CNN**	**IG**	**ECE**	**GI**	**MI_avg**	**WOR**	**NDM**
30	62.4	45.7	45.7	51.6	52.6	58.8	51.6
60	69.1	42	42	56	56.7	65.4	56
90	69.7	38.5	38.5	58.4	60.1	69.1	58.4
120	72	36	36	59.3	63.7	72	59.3
150	73	33.5	33.5	62.5	64.7	74.1	62.5
180	73	29.5	43.6	63.1	65.5	75.4	63.1
210	74.4	27.6	57	64.4	67.7	76.4	64.4
240	74.8	26.5	62.6	64.7	69.2	77.4	64.7
270	75.4	26.4	66.3	66.4	69.9	78.6	66.4
300	75.6	26.5	67.9	66.9	70.9	80	66.9
(5) KDC Dataset							
SVM	**CNN**	**IG**	**ECE**	**GI**	**MI_avg**	**WOR**	**NDM**
50	62.9	12.1	24	14.6	16.4	23.6	16.1
100	72.6	11.2	44.3	15.6	18.6	25.6	18.8
150	77.2	11	64.9	16.1	20.7	26.2	21.2
200	80.6	10.7	69.1	16.9	24.9	26.8	23.6
250	82.1	10.8	70.9	17.3	28.9	28.9	28.8
300	83.1	10.5	74.7	17.8	30.9	30	30.4
350	84	10.6	76.2	17.8	32.4	31.1	32
400	85	10.6	77.8	18	32.6	32.4	33.1
450	85.7	10.3	78.4	18.1	35.2	33.8	34.2
500	86	10	79.5	18.1	35.9	34.3	34.9
MNB	**CNN**	**IG**	**ECE**	**GI**	**MI_avg**	**WOR**	**NDM**
50	61.9	11.7	22.7	14.2	16.1	24.2	16
100	71.4	11.1	44.3	14.5	18.6	25.8	18.4
150	77.2	10.8	63.6	15.2	21	26.2	20.5
200	81.4	10.5	69.3	15.5	24.4	26.2	23.5
250	84	10.6	71	16	28	27.5	27.8
300	85.4	10.2	75.1	16	29.8	28.8	28.9
350	86.6	9.9	76.8	16.1	31.42	30.1	29.7
400	87.4	9.5	78.5	16	32.2	31.2	31
450	88.1	9.1	79.8	15.9	34.1	32.6	32
500	88.9	8.7	81.1	16.2	34.3	32.2	32.2

Table A1. *Cont.*

(6) TTC Dataset							
SVM	**CNN**	**IG**	**ECE**	**GI**	**MI_avg**	**WOR**	**NDM**
50	56.1	15.9	27.2	17.5	42.1	15.7	25.3
100	73.8	15.7	28.5	18.3	51.3	15.7	31.1
150	77.1	16.1	31.6	19.9	55.9	15.6	38.3
200	80.1	15.5	47.7	20.5	59.4	16.1	41.7
250	80.6	15.1	54.9	21.5	62.6	16.4	44.3
300	81.5	15.4	58.4	22.4	64.8	17.8	48.1
350	82.2	15.6	60.3	22.7	69.8	18.3	52.8
400	82.4	15.8	65	23.7	70.7	20.3	54
450	82.5	16.4	70.7	25.3	70.1	22.2	55.2
500	83.4	17.8	73.7	25.9	71.2	22.5	56
MNB	**CNN**	**IG**	**ECE**	**GI**	**MI_avg**	**WOR**	**NDM**
50	69.7	15.9	28.8	15.4	47.6	15.5	24.3
100	79.2	15.6	30.9	14.4	56.3	15.5	30
150	82.7	15.1	35.2	16.1	60.4	15.5	44.2
200	84.3	14.8	51.7	17.2	65.1	16	49.8
250	84.7	14.8	58.3	18.1	68.3	15.75	53.1
300	85.1	14.6	63.1	18.2	70.3	17.2	56.1
350	85.7	14.4	66.9	18.5	74.4	17.3	59.5
400	86	14.6	72.3	21.5	75.5	19.3	60
450	86	15.5	76.8	23.4	76.1	21	61.6
500	86.6	17.4	80.75	23.6	77.7	21.4	63.3
(7) WEKBE Dataset							
SVM	**CNN**	**IG**	**ECE**	**GI**	**MI_avg**	**WOR**	**NDM**
50	59.4	39	53.3	42.8	40.7	43.3	46.1
100	64.5	38.5	59	44.2	41.9	44.1	47
150	66.4	38.2	63	46.3	42.9	49.8	51.3
200	69.9	37.8	66.5	47.6	45	49.8	52.6
250	72.5	38.2	67.6	48.3	44.9	51.6	54
300	74.5	37.5	69.2	49.5	45.4	53.7	55.5
350	76.9	37.5	72.5	50.6	46.9	54.1	55.9
400	78.3	37.6	76.6	51.3	47.4	55.5	57.1
450	80	37.3	79.1	51.7	47.4	56.1	57.7
500	80.4	36.5	80.5	52.4	48.8	56.8	57.9

Table A1. *Cont.*

(7) WEKBE Dataset							
MNB	**CNN**	**IG**	**ECE**	**GI**	**MI_avg**	**WOR**	**NDM**
50	71.4	37.5	60.4	43.2	41.8	42	51.3
100	73.8	36.4	64.3	45	44.5	42.5	52.5
150	73.7	35.8	67.4	47.4	44.9	50.3	59.4
200	74.8	34.6	70	49.3	47.9	50.5	61.4
250	75.3	34.1	70.1	50.5	47.9	51.8	63.1
300	76.1	33.3	70.8	52.1	48.9	54.2	64.6
350	77.3	32.5	75.6	53.2	51.4	55.6	65.2
400	77.4	31	79	54.1	51.8	56.1	66
450	78	30.2	81	54.7	51.4	57	66.8
500	78.4	29.8	81.4	56	52.5	57.8	67.1
(8) R8 Dataset							
SVM	**CNN**	**IG**	**ECE**	**GI**	**MI_avg**	**WOR**	**NDM**
50	84.7	51	77.3	51.3	52.8	61.8	73
100	89.4	51	80.8	52	59.8	65.9	74.6
150	92.1	50.6	82.9	52.7	61.7	66.8	75.5
200	93	50.7	84.8	53.4	62.3	68	76.4
250	93.3	50.2	86.3	53.5	62.3	68.5	78.7
300	93.5	50	87.3	54.4	62.2	69.4	80.2
350	93.7	49.8	89.2	55.1	62.1	69.7	81.1
400	93.7	50	89.9	55.4	62.5	69.8	81.4
450	93.8	49.9	90.4	55.9	64.5	69.9	82.2
500	93.8	49.1	90.4	56.1	65.9	70.3	83.9
MNB	**CNN**	**IG**	**ECE**	**GI**	**MI_avg**	**WOR**	**NDM**
50	81.5	50.7	74.6	51.1	53.5	64.5	66.9
100	89.1	50.4	79.5	52.2	61.5	69.5	69.2
150	92.6	50.1	81.2	53.1	64.3	71.2	72.2
200	93.3	49.9	83.9	53.9	65.1	72.4	74
250	93.8	49.5	85.7	54.3	64.8	73.1	78.5
300	94	49.3	86.9	55.1	64.3	74.2	80.2
350	94.2	49.2	88.9	56.1	63.9	74.6	81.4
400	94.3	48.9	89.7	56.4	64.6	74.6	81.7
450	94.3	48.9	90.2	56.8	66.1	74.7	83.6
500	94.6	48.9	90.5	57	67.7	75.2	85.7

Table A1. *Cont.*

(9) R52 Dataset							
SVM	**CNN**	**IG**	**ECE**	**GI**	**MI_avg**	**WOR**	**NDM**
50	73.6	42.9	63.5	43.3	44.2	53	51
100	79.6	42.6	69.8	43.6	47.5	57.4	58
150	82.2	42.4	72.3	43.9	54.1	61.3	64.2
200	84	42.3	74.7	44.3	55.1	66.2	69.6
250	85	41.9	76.7	44.5	55.1	67.4	72.7
300	86.4	42.1	77.7	44.6	55	67.7	73.7
350	87	41.9	78.8	44.8	55.1	68.7	75.7
400	87	41.8	79.7	45.1	55.1	69.3	77.1
450	87.5	41.3	80.8	45.3	55	69.3	77.3
500	87.9	41.5	81.1	45.4	54.9	69.7	78.1
MNB	**CNN**	**IG**	**ECE**	**GI**	**MI_avg**	**WOR**	**NDM**
50	71.9	42.8	62.2	43.2	44.7	59.6	53.3
100	81.2	42.6	69.6	43.8	49.4	65.3	61.3
150	84.5	42.6	72.9	44	54.8	69.1	66.3
200	86.1	42.4	75.2	44.3	56.9	72.7	71.9
250	87.4	42.2	78.1	44.5	56.9	73.7	73.9
300	88.6	42	79	44.6	56.9	74.4	75.9
350	88.9	41.8	80.8	44.9	56.9	75.2	78.2
400	89.3	41.7	81.9	45.2	56.6	75.6	79.2
450	89.7	41.5	82.9	45.4	56.5	75.7	80
500	89.8	41.2	83.3	45.4	43.8	75.9	80.6

Table A2. The *p*-values of *t*-tests in datasets.

(1) CF Dataset						
	CNN vs. IG	**CNN vs. ECE**	**CNN vs. GI**	**CNN vs. MI_avg**	**CNN vs. WOR**	**CNN vs. NDM**
SVM	2.74×10^{-8}	3.46×10^{-5}	2.10×10^{-9}	4.08×10^{-6}	1.68×10^{-6}	2.24×10^{-12}
MNB	9.64×10^{-8}	3.25×10^{-5}	1.57×10^{-7}	1.06×10^{-7}	1.68×10^{-6}	2.23×10^{-8}
(2) CR Dataset						
	CNN vs. IG	**CNN vs. ECE**	**CNN vs. GI**	**CNN vs. MI_avg**	**CNN vs. WOR**	**CNN vs. NDM**
SVM	1.00×10^{-9}	0.00	1.89×10^{-13}	1.80×10^{-9}	1.05×10^{-10}	1.57×10^{-9}
MNB	3.95×10^{-9}	0.00	1.06×10^{-10}	1.31×10^{-10}	1.32×10^{-10}	6.50×10^{-14}
(3) CNAE Dataset						
	CNN vs. IG	**CNN vs. ECE**	**CNN vs. GI**	**CNN vs. MI_avg**	**CNN vs. WOR**	**CNN vs. NDM**
SVM	4.16×10^{-11}	**0.11**	3.66×10^{-10}	3.14×10^{-10}	8.86×10^{-11}	3.25×10^{-10}
MNB	2.07×10^{-10}	**0.07**	5.91×10^{-10}	1.86×10^{-10}	2.61×10^{-10}	2.16×10^{-10}

Table A2. *Cont.*

(4) IMDB Dataset						
	CNN vs. IG	**CNN vs. ECE**	**CNN vs. GI**	**CNN vs. MI_avg**	**CNN vs. WOR**	**CNN vs. NDM**
SVM	9.80×10^{-7}	0.00	1.70×10^{-7}	2.78×10^{-5}	**0.19**	1.70×10^{-7}
MNB	1.39×10^{-6}	0.00	2.25×10^{-9}	2.33×10^{-6}	**0.39**	2.25×10^{-9}
(5) KDC Dataset						
	CNN vs. IG	**CNN vs. ECE**	**CNN vs. GI**	**CNN vs. MI_avg**	**CNN vs. WOR**	**CNN vs. NDM**
SVM	4.80×10^{-10}	0.00	1.20×10^{-10}	7.67×10^{-13}	4.64×10^{-11}	5.55×10^{-13}
MNB	2.16×10^{-9}	0.00	9.36×10^{-10}	1.35×10^{-12}	8.29×10^{-10}	2.69×10^{-12}
(6) TTC Dataset						
	CNN vs. IG	**CNN vs. ECE**	**CNN vs. GI**	**CNN vs. MI_avg**	**CNN vs. WOR**	**CNN vs. NDM**
SVM	1.78×10^{-9}	8.74×10^{-5}	4.42×10^{-10}	6.71×10^{-7}	7.34×10^{-10}	1.29×10^{-8}
MNB	2.06×10^{-11}	0.00	1.84×10^{-12}	8.24×10^{-6}	3.87×10^{-12}	1.09×10^{-6}
(7) WEKBE Dataset						
	CNN vs. IG	**CNN vs. ECE**	**CNN vs. GI**	**CNN vs. MI_avg**	**CNN vs. WOR**	**CNN vs. NDM**
SVM	1.97×10^{-7}	0.00	1.20×10^{-8}	1.51×10^{-8}	1.61×10^{-9}	1.21×10^{-8}
MNB	5.56×10^{-10}	0.04	3.67×10^{-11}	5.44×10^{-13}	6.97×10^{-9}	9.28×10^{-7}
(8) R8 Dataset						
	CNN vs. IG	**CNN vs. ECE**	**CNN vs. GI**	**CNN vs. MI_avg**	**CNN vs. WOR**	**CNN vs. NDM**
SVM	2.29×10^{-11}	1.12×10^{-5}	2.42×10^{-13}	4.52×10^{-14}	4.13×10^{-15}	1.34×10^{-8}
MNB	3.31×10^{-10}	1.07×10^{-5}	7.65×10^{-12}	2.14×10^{-14}	1.62×10^{-12}	8.16×10^{-7}
(9) R52 Dataset						
	CNN vs. IG	**CNN vs. ECE**	**CNN vs. GI**	**CNN vs. MI_avg**	**CNN vs. WOR**	**CNN vs. NDM**
SVM	6.84×10^{-10}	5.57×10^{-9}	1.25×10^{-10}	7.01×10^{-13}	3.52×10^{-11}	5.99×10^{-6}
MNB	2.63×10^{-9}	9.65×10^{-8}	7.61×10^{-10}	8.86×10^{-9}	8.08×10^{-12}	1.76×10^{-6}

References

1. Joachims, T. Transductive Inference for Text Classification Using Support Vector Machines. In Proceedings of the 16th International Conference on Machine Learning, Bled, Slovenia, 27–30 June 1999; pp. 200–209.
2. Tan, A.H. Text Mining: The State of the Art and the Challenges. In Proceedings of the Pakdd Workshop on Knowledge Disocovery from Advanced Databases, Beijing, China, 26–28 April 1999; pp. 65–70.
3. Mujtaba, G.; Shuib, L.; Raj, R.G. Detection of Suspicious Terrorist Emails Using Text Classification: A Review. *Malays. J. Comput. Sci.* **2019**, *31*, 271–299. [CrossRef]
4. Srivastava, S.K.; Singh, S.K.; Suri, J.S. Effect of Incremental Feature Enrichment on Healthcare Text Classification System: A Machine Learning Paradigm. *Comput. Methods Programs Biomed.* **2019**, *172*, 35–51. [CrossRef] [PubMed]
5. Wang, P.; Yang, Y.; Si, Y.D.; Zhu, G.C.; Zhan, X.P.; Wang, J.; Pan, R.S. Classification of Proactive Personality: Text Mining Based on Weibo Text and Short-Answer Questions Text. *IEEE Access* **2020**, *8*, 97370–97382. [CrossRef]
6. Ostrogonac, S.J.; Rastovic, B.S.; Popovic, B. Automatic Job Ads Classification, Based on Unstructured Text Analysis. *ACTA Polytech. Hung.* **2021**, *18*, 191–204. [CrossRef]

7. Li, R.; Yu, W.K.; Liu, Y.Y. Patent Text Classification Based on Deep Learning and Vocabulary Network. *Int. J. Adv. Comput. Sci. Appl.* **2023**, *14*, 54–61. [CrossRef]
8. Kim, J.; Jang, S.; Park, E.; Choi, S. Text Classification Using Capsules. *Neurocomputing* **2020**, *376*, 214–221. [CrossRef]
9. Dai, Y.; Shou, L.J.; Gong, M.; Xia, X.L.; Kang, Z.; Xu, Z.L.; Jiang, D.X. Graph Fusion Network for Text Classification. *Knowl.-Based Syst.* **2022**, *236*, 107659. [CrossRef]
10. Zhan, Z.Q.; Hou, Z.F.; Yang, Q.C.; Zhao, J.Y.; Zhang, Y.; Hu, C.J. Knowledge Attention Sandwich Neural Network for Text Classification. *Neurocomputing* **2020**, *406*, 1–11. [CrossRef]
11. Tezgider, M.; Yildiz, B.; Aydin, G. Text Classification Using Improved Bidirectional Transformer. *Concurr. Comput.-Pract. Exp.* **2022**, *34*, e6486. [CrossRef]
12. Xu, Y.H.; Yu, Z.W.; Cao, W.M.; Chen, C.L.P. Adaptive Dense Ensemble Model for Text Classification. *IEEE Trans. Cybern.* **2022**, *52*, 7513–7526. [CrossRef]
13. Wang, Y.Z.; Wang, C.X.; Zhan, J.Y.; Ma, W.J.; Jiang, Y.C. Text FCG: Fusing Contextual Information via Graph Learning for text classification. *Expert Syst. Appl.* **2023**, *219*, 119658. [CrossRef]
14. Gan, S.F.; Shao, S.Q.; Chen, L.; Yu, L.J.; Jiang, L.X. Adapting Hidden Naive Bayes for Text Classification. *Mathematics* **2021**, *9*, 2378. [CrossRef]
15. Shi, Z.; Fan, C.J. Short Text Sentiment Classification Using Bayesian and Deep Neural Networks. *Electronics* **2023**, *12*, 1589. [CrossRef]
16. Rijcken, E.; Kaymak, U.; Scheepers, F.; Mosteiro, P.; Zervanou, K.; Spruit, M. Topic Modeling for Interpretable Text Classification from EHRs. *Front. Big Data* **2022**, *5*, 846930. [CrossRef]
17. Sinoara, R.A.; Camacho-Collados, J.; Rossi, R.G.; Navigli, R.; Rezende, S.O. Knowledge-enhanced Document Embeddings for Text Classification. *Knowl.-Based Syst.* **2019**, *163*, 955–971. [CrossRef]
18. Wang, H.Y.; Zeng, D.H. Fusing Logical Relationship Information of Text in Neural Network for Text Classification. *Math. Probl. Eng.* **2020**, *2020*, 5426795. [CrossRef]
19. Shanavas, N.; Wang, H.; Lin, Z.W.; Hawe, G. Knowledge-driven Graph Similarity for Text Classification. *Int. J. Mach. Learn. Cybern.* **2021**, *12*, 1067–1081. [CrossRef]
20. Sulaimani, S.A.; Starkey, A. Short Text Classification Using Contextual Analysis. *IEEE Access* **2021**, *9*, 149619–149629. [CrossRef]
21. Tan, C.G.; Ren, Y.; Wang, C. An Adaptive Convolution with Label Embedding for Text Classification. *Appl. Intell.* **2023**, *53*, 804–812. [CrossRef]
22. Witten, I.H.; Frank, E.; Hall, M.A.; Pal, C.J. *Data Mining: Practical Machine Learning Tools and Techniques*; Green, T., Ed.; Morgan Kaufmann: Burlington, NJ, USA, 2017.
23. Bahassine, S.; Madani, A.; Al-Sarem, M.; Kissi, M. Feature Selection Using an Improved Chi-square for Arabic Text Classification. *J. King Saud Univ.-Comput. Inf. Sci.* **2020**, *32*, 225–231. [CrossRef]
24. Cekik, R.; Uysal, A.K. A Novel Filter Feature Selection Method Using Rough Set for Short Text Data. *Expert Syst. Appl.* **2020**, *160*, 113691. [CrossRef]
25. Zhou, H.F.; Ma, Y.M.; Li, X. Feature Selection Based on Term Frequency Deviation Rate for Text Classification. *Appl. Intell.* **2020**, *51*, 3255–3274. [CrossRef]
26. Amazal, H.; Kissi, M. A New Big Data Feature Selection Approach for Text Classification. *Sci. Program.* **2021**, *2021*, 6645345. [CrossRef]
27. Parlak, B.; Uysal, A.K. A Novel Filter Feature Selection Method for Text Classification: Extensive Feature Selector. *J. Inf. Sci.* **2021**, *49*, 59–78. [CrossRef]
28. Cekik, R.; Uysal, A.K. A New Metric for Feature Selection on Short Text Datasets. *Concurr. Comput.-Pract. Exp.* **2022**, *34*, e6909. [CrossRef]
29. Parlak, B. Class-index Corpus-index Measure: A Novel Feature Selection Method for Imbalanced Text Data. *Concurr. Comput.-Pract. Exp.* **2022**, *34*, e7140. [CrossRef]
30. Jin, L.B.; Zhang, L.; Zhao, L. Feature Selection Based on Absolute Deviation Factor for Text Classification. *Inf. Process. Manag.* **2023**, *60*, 103251. [CrossRef]
31. Verma, G.; Sahu, T.P. Deep Label Relevance and Label Ambiguity Based Multi-label Feature Selection for Text Classification. *Eng. Appl. Artif. Intell.* **2025**, *148*, 110403. [CrossRef]
32. Ige, O.P.; Gan, K.H. Ensemble Filter-Wrapper Text Feature Selection Methods for Text Classification. *CMES-Comput. Model. Eng. Sci.* **2025**, *141*, 1847–1865. [CrossRef]
33. Mohanrasu, S.S.; Janani, K.; Rakkiyappan, R. A COPRAS-based Approach to Multi-Label Feature Selection for Text Classification. *Math. Comput. Simul.* **2024**, *222*, 3–23. [CrossRef]
34. Cekik, R. A New Filter Feature Selection Method for Text Classification. *IEEE Access* **2024**, *12*, 139316–139335. [CrossRef]
35. Farek, L.; Benaidja, A. A Non-redundant Feature Selection Method for Text Categorization Based on Term Co-occurrence Frequency and Mutual Information. *Multimed. Tools Appl.* **2024**, *83*, 20193–20214. [CrossRef]

36. Liu, X.; Wang, S.; Lu, S.Y.; Yin, Z.T.; Li, X.L.; Yin, L.R.; Tian, J.W.; Zheng, W.F. Adapting Feature Selection Algorithms for the Classification of Chinese Texts. *Systems* **2023**, *11*, 483. [CrossRef]
37. Ashokkumar, P.; Shankar, G.S.; Srivastava, G.; Maddikunta, P.K.R.; Gadekallu, T.R. A Two-stage Text Feature Selection Algorithm for Improving Text Classification. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* **2021**, *20*, 49.
38. Garg, M. UBIS: Unigram Bigram Importance Score for Feature Selection from Short Text. *Expert Syst. Appl.* **2022**, *195*, 116563. [CrossRef]
39. Okkalioglu, M. A Novel Redistribution-based Feature Selection for Text Classification. *Expert Syst. Appl.* **2024**, *246*, 123119. [CrossRef]
40. Lazhar, F.; Amira, B. Semantic Similarity-aware Feature selection and Redundancy Removal for Text Classification using joint mutual information. *Knowl. Inf. Syst.* **2024**, *66*, 6187–6212. [CrossRef]
41. Rehman, S.U.; Jat, R.; Rafi, M.; Frnda, J. TransFINN "Transparent Feature Integrated Neural Network for Text Feature Selection and Classification". *IEEE Access* **2025**, *13*, 118821–118829. [CrossRef]
42. Chantar, H.; Mafarja, M.; Alsawalqah, H.; Heidari, A.A.; Aljarah, I.; Faris, H. Feature Selection Using Binary Grey Wolf Optimizer with Elite-based Crossover for Arabic Text Classification. *Neural Comput. Appl.* **2020**, *32*, 12201–12220. [CrossRef]
43. Alsaleh, D.; Larabi-Marie-Sainte, S. Arabic Text Classification Using Convolutional Neural Network and Genetic Algorithms. *IEEE Access* **2021**, *9*, 91670–91685. [CrossRef]
44. Thirumoorthy, K.; Muneeswaran, K. Feature Selection Using Hybrid Poor and Rich Optimization Algorithm for Text Classification. *Pattern Recognit. Lett.* **2021**, *147*, 63–70. [CrossRef]
45. Hosseinalipour, A.; Gharehchopogh, F.S.; Masdari, M.; Khademi, A. A Novel Binary Farmland Fertility Algorithm for Feature Selection in Analysis of the Text Psychology. *Appl. Intell.* **2021**, *51*, 4824–4859. [CrossRef]
46. Adel, A.; Omar, N.; Abdullah, S.; Al-Shabi, A. Co-Operative Binary Bat Optimizer with Rough Set Reducts for Text Feature Selection. *Appl. Sci.* **2022**, *12*, 1296. [CrossRef]
47. Priya, S.K.; Karthika, K.P. An Embedded Feature Selection Approach for Depression Classification Using Short Text Sequences. *Appl. Soft Comput.* **2023**, *147*, 110828. [CrossRef]
48. Singh, G.; Nagpal, A.; Singh, V. Optimal Feature Selection and Invasive Weed Tunicate Swarm Algorithm-based Hierarchical Attention Network for Text Classification. *Connect. Sci.* **2023**, *35*, 2231171. [CrossRef]
49. Wu, X.; Fei, M.R.; Wu, D.K. Enhanced Binary Black Hole algorithm for text feature selection on resources classification. *Knowl.-Based Syst.* **2023**, *274*, 110635. [CrossRef]
50. Kaya, C.; Kilimci, Z.H.; Uysal, M.; Kaya, M. Migrating Birds Optimization-based Feature Selection for Text Classification Fi cation. *PeerJ Comput. Sci.* **2024**, *10*, e2263. [CrossRef]
51. Msallam, M.M.; Bin Idris, S.A. Unsupervised Text Feature Selection by Binary Fire Hawk Optimizer for Text Clustering. *Clust. Comput.—J. Netw. Softw. Tools Appl.* **2024**, *27*, 7721–7740. [CrossRef]
52. Nachaoui, M.; Lakouam, I.; Hafidi, I. Hybrid Particle Swarm Optimization Algorithm for Text Feature Selection Problems. *Neural Comput. Appl.* **2024**, *36*, 7471–7489. [CrossRef]
53. Dhal, P.; Azad, C. A Fine-tuning Deep Learning with Multi-objective-based Feature Selection Approach for the Classification of Text. *Neural Comput. Appl.* **2024**, *36*, 3525–3553. [CrossRef]
54. Farek, L.; Benaidja, A. An Adaptive Binary Particle Swarm Optimization Algorithm with Filtration and Local Search for Feature Selection in Text Classification. *Memetic Comput.* **2025**, *17*, 45. [CrossRef]
55. El-Hajj, W.; Hajj, H. An Optimal Approach for Text Feature Selection. *Comput. Speech Lang.* **2022**, *74*, 101364. [CrossRef]
56. Saeed, M.M.; Al Aghbari, Z. ARTC: Feature Selection Using Association Rules for Text Classification. *Neural Comput. Appl.* **2022**, *34*, 22519–22529. [CrossRef]
57. Farghaly, H.M.; Abd El-Hafeez, T. A High-quality Feature Selection Method Based on Frequent and Correlated Items for Text Classification. *Soft Comput.* **2023**, *27*, 11259–11274. [CrossRef]
58. Sagbas, E.A. A Novel Two-stage Wrapper Feature Selection Approach Based on Greedy Search for Text Sentiment Classification. *Neurocomputing* **2024**, *590*, 127729. [CrossRef]
59. Liu, Y.; Cheng, X.; Stephen, L.S.; Wei, S.S. Advancing Text Classification: A Novel Two-stage Multi-objective Feature Selection Framework. *Inf. Technol. Manag.* **2025**. [CrossRef]
60. Jalilian, E.; Hofbauer, H.; Uhl, A. Iris Image Compression Using Deep Convolutional Neural Networks. *Sensors* **2022**, *22*, 2698. [CrossRef]
61. Xiao, Q.G.; Li, G.Y.; Chen, Q.C. Complex image classification by feature inference. *Neurocomputing* **2023**, *544*, 126231. [CrossRef]
62. Tyagi, S.; Yadav, D. ForensicNet: Modern convolutional neural network-based image forgery detection network. *J. Forensic Sci.* **2023**, *68*, 461–469. [CrossRef]
63. Wang, J.R.; Li, J.; Zhang, Y.R. Text3D: 3D Convolutional Neural Networks for Text Classification. *Electronics* **2023**, *12*, 3087. [CrossRef]

64. Xu, S.J.; Guo, C.Y.; Zhu, Y.H.; Liu, G.G.; Xiong, N.L. CNN-VAE: An Intelligent Text Representation Algorithm. *J. Supercomput.* **2023**, *79*, 12266–12291. [CrossRef]
65. Gasmi, K.; Ayadi, H.; Torjmen, M. Enhancing Medical Image Retrieval with UMLS-Integrated CNN-Based Text Indexing. *Diagnostics* **2024**, *14*, 1204. [CrossRef] [PubMed]
66. Liu, X.Z.; Wang, Y.Y.; Niu, N.N.; Zhang, B.Y.; Li, J.S. A Hybrid Architecture for Enhancing Chinese Text Processing Using CNN and LLaMA2. *Sci. Rep.* **2025**, *15*, 24735. [CrossRef] [PubMed]
67. Wahyu Trisna, K.; Huang, J.J.; Chen, Y.J.; Putra, I.G.J.E. Dynamic Text Augmentation for Robust Sentiment Analysis: Enhancing Model Performance with EDA and Multi-Channel CNN. *IEEE Access* **2025**, *13*, 31978–31991. [CrossRef]
68. Wu, X.; Cai, Y.; Li, Q.; Xu, J.Y.; Leung, H.F. Combining Weighted Category-aware Contextual Information in Convolutional Neural Networks for Text Classification. *World Wide Web-Internet Web Inf. Syst.* **2020**, *23*, 2815–2834. [CrossRef]
69. Xu, J.Y.; Cai, Y.; Wu, X.; Lei, X.; Huang, Q.B.; Leung, H.F.; Li, Q. Incorporating Context-relevant Concepts into Convolutional Neural Networks for Short Text Classification. *Neurocomputing* **2020**, *386*, 42–53. [CrossRef]
70. Butt, M.A.; Khatak, A.M.; Shafique, S.; Hayat, B.; Abid, S.; Kim, K.I.; Ayub, M.W.; Sajid, A.; Adnan, A. Convolutional Neural Network Based Vehicle Classification in Adverse Illuminous Conditions for Intelligent Transportation Systems. *Complexity* **2021**, *2021*, 6644861. [CrossRef]
71. Liang, Y.J.; Li, H.H.; Guo, B.; Yu, Z.W.; Zheng, X.L.; Samtani, S.; Zeng, D.D. Fusion of Heterogeneous Attention Mechanisms in Multi-view Convolutional Neural Network for Text Classification. *Inf. Sci.* **2021**, *548*, 295–312. [CrossRef]
72. Liu, X.Y.; Tang, T.; Ding, N. Social Network Sentiment Classification Method Combined Chinese Text Syntax with Graph Convolutional Neural Network. *Egypt. Inform. J.* **2022**, *23*, 1–12. [CrossRef]
73. Zhao, W.D.; Zhu, L.; Wang, M.; Zhang, X.L.; Zhang, J.M. WTL-CNN: A News Text Classification Method of Convolutional Neural Network Based on Weighted Word Embedding. *Connect. Sci.* **2022**, *34*, 2291–2312. [CrossRef]
74. Qorich, M.; El Ouazzani, R. Text Sentiment Classification of Amazon Reviews Using Word Embeddings and Convolutional Neural Networks. *J. Supercomput.* **2023**, *79*, 11029–11054. [CrossRef]
75. Thekkekara, J.P.; Yongchareon, S.; Liesaputra, V. An Attention-based CNN-BiLSTM Model for Depression Detection on Social Media Text. *Expert Syst. Appl.* **2024**, *249*, 123834. [CrossRef]
76. Guo, X.L.; Liu, Q.Y.; Hu, Y.R.; Liu, H.J. MDCNN: Multi-Teacher Distillation-Based CNN for News Text Classification. *IEEE Access* **2025**, *13*, 56631–56641. [CrossRef]
77. Liu, B.; Zhou, Y.; Sun, W. Character-level Text Classification via Convolutional Neural Network and Gated Recurrent Unit. *Int. J. Mach. Learn. Cybern.* **2020**, *11*, 1939–1949. [CrossRef]
78. Zeng, S.F.; Ma, Y.; Zhang, X.Y.; Du, X.F. Term-Based Pooling in Convolutional Neural Networks for Text Classification. *China Commun.* **2020**, *17*, 109–124. [CrossRef]
79. Lyu, S.F.; Liu, J.Q. Convolutional Recurrent Neural Networks for Text Classification. *J. Databased Manag.* **2021**, *32*, 65–82. [CrossRef]
80. Wang, H.T.; Tian, K.K.; Wu, Z.J.; Wang, L. A Short Text Classification Method Based on Convolutional Neural Network and Semantic Extension. *Int. J. Comput. Intell. Syst.* **2021**, *14*, 367–375. [CrossRef]
81. Liu, J.F.; Ma, H.Z.; Xie, X.L.; Cheng, J. Short Text Classification for Faults Information of Secondary Equipment Based on Convolutional Neural Networks. *Energies* **2022**, *15*, 2400. [CrossRef]
82. Zhou, Y.J.; Li, J.L.; Chi, J.H.; Tang, W.; Zheng, Y.Q. Set-CNN: A Text Convolutional Neural Network Based on Semantic Extension for Short Text Classification. *Knowl.-Based Syst.* **2022**, *257*, 109948. [CrossRef]
83. Xiong, Y.P.; Chen, G.L.; Cao, J.K. Research on Public Service Request Text Classification Based on BERT-BiLSTM-CNN Feature Fusion. *Appl. Sci.* **2024**, *14*, 6282. [CrossRef]
84. Li, J.H.; Yang, Y.; Sun, J.; Wang, F.; Chen, S.L. DT-GCNN: Dynamic Triplet Network with GRU-CNN for Enhanced Text Classification. *Int. J. Mach. Learn. Cybern.* **2025**, *16*, 9555–9567. [CrossRef]
85. Huang, M.H.; Xie, H.R.; Rao, Y.H.; Feng, J.R.; Wang, F.L. Sentiment Strength Detection with a Context-dependent Lexicon-based Convolutional Neural Network. *Inf. Sci.* **2020**, *520*, 389–399. [CrossRef]
86. Krishnan, H.; Elayidom, M.S.; Santhanakrishnan, T. Optimization Assisted Convolutional Neural Network for Sentiment Analysis with Weighted Holoentropy-based Features. *Int. J. Inf. Technol. Decis. Mak.* **2021**, *20*, 1261–1297. [CrossRef]
87. Usama, M.; Ahmad, B.; Song, E.; Hossain, M.S.; Alrashoud, M.; Muhammad, G. Attention-based Sentiment Analysis Using Convolutional and Recurrent Neural Network. *Future Gener. Comput. Syst.* **2020**, *113*, 571–578. [CrossRef]
88. Wang, X.Y.; Li, F.; Zhang, Z.Q.; Xu, G.L.; Zhang, J.Y.; Sun, X. A Unified Position-aware Convolutional Neural Network for Aspect Based Sentiment Analysis. *Neurocomputing* **2021**, *450*, 91–103. [CrossRef]
89. Ghorbanali, A.; Sohrabi, M.K.; Yaghmaee, F. Ensemble Transfer Learning-based Multimodal Sentiment Analysis Using Weighted Convolutional Neural Networks. *Inf. Process. Manag.* **2022**, *59*, 102929. [CrossRef]
90. Huang, M.H.; Xie, H.R.; Rao, Y.H.; Liu, Y.W.; Poon, L.K.M.; Wang, F.L. Lexicon-Based Sentiment Convolutional Neural Networks for Online Review Analysis. *IEEE Trans. Affect. Comput.* **2022**, *13*, 1337–1348. [CrossRef]

91. Murugaiyan, S.; Uyyala, S.R. Aspect-Based Sentiment Analysis of Customer Speech Data Using Deep Convolutional Neural Network and BiLSTM. *Cogn. Comput.* **2023**, *15*, 914–931. [CrossRef]
92. Mutinda, J.; Mwangi, W.; Okeyo, G. Sentiment Analysis of Text Reviews Using Lexicon-Enhanced Bert Embedding (LeBERT) Model with Convolutional Neural Network. *Appl. Sci.* **2023**, *13*, 1445. [CrossRef]
93. Alnowaiser, K. Scientific Text Citation Analysis Using CNN Features and Ensemble Learning Model. *PLoS ONE* **2024**, *19*, e0302304. [CrossRef]
94. He, A.X.; Abisado, M. Text Sentiment Analysis of Douban Film Short Comments Based on BERT-CNN-BiLSTM-Att Model. *IEEE Access* **2024**, *12*, 45229–45237. [CrossRef]
95. Chen, J.Y.; Yan, S.K.; Wong, K.C. Verbal Aggression Detection on Twitter Comments: Convolutional Neural Network for Short-text Sentiment Analysis. *Neural Comput. Appl.* **2020**, *32*, 10809–10818. [CrossRef]
96. Qiu, M.; Zhang, Y.R.; Ma, T.Q.; Wu, Q.F.; Jin, F.Z. Convolutional-neural-network-based Multilabel Text Classification for Automatic Discrimination of Legal Documents. *Sens. Mater.* **2020**, *32*, 2659–2672. [CrossRef]
97. Heo, T.S.; Kim, C.; Kim, J.D.; Park, C.Y.; Kim, Y.S. Prediction of Atrial Fibrillation Cases: Convolutional Neural Networks Using the Output Texts of Electrocardiography. *Sens. Mater.* **2021**, *33*, 393–404. [CrossRef]
98. Li, Y.; Yin, C.B. Application of Dual-Channel Convolutional Neural Network Algorithm in Semantic Feature Analysis of English Text Big Data. *Comput. Intell. Neurosci.* **2021**, *2021*, 7085412. [CrossRef]
99. Jian, L.H.; Xiang, H.Q.; Le, G.B. English Text Readability Measurement Based on Convolutional Neural Network: A Hybrid Network Model. *Comput. Intell. Neurosci.* **2022**, *2022*, 6984586. [CrossRef]
100. Qiu, Q.J.; Xie, Z.; Ma, K.; Chen, Z.L.; Tao, L.F. Spatially Oriented Convolutional Neural Network for Spatial Relation Extraction from Natural Language Texts. *Trans. GIS* **2022**, *26*, 839–866. [CrossRef]
101. Boukhers, Z.; Goswami, P.; Jurjens, J. Knowledge Guided Multi-filter Residual Convolutional Neural Network for ICD Coding from Clinical Text. *Neural Comput. Appl.* **2023**, *35*, 17633–17644. [CrossRef]
102. Muppudathi, S.S.; Krishnasamy, V. Anomaly Detection in Social Media Texts Using Optimal Convolutional Neural Network. *Intell. Autom. Soft Comput.* **2023**, *36*, 1027–1042. [CrossRef]
103. Fan, W.T. BiLSTM-Attention-CNN Model Based on ISSA Optimization for Cyberbullying Detection in Chinese Text. *Inf. Technol. Control.* **2024**, *53*, 659–674. [CrossRef]
104. Zeng, Q.X. Enhanced Analysis of Large-scale News Text Data Using the Bidirectional-Kmeans-LSTM-CNN Model. *Peerj Comput. Sci.* **2024**, *10*, e2213. [CrossRef] [PubMed]
105. Faseeh, M.; Khan, M.A.; Iqbal, N.; Qayyum, F.; Mehmood, A.; Kim, J. Enhancing User Experience on Q&A Platforms: Measuring Text Similarity Based on Hybrid CNN-LSTM Model for Efficient Duplicate Question Detection. *IEEE Access* **2024**, *12*, 34512–34526.
106. Wu, H.C.; Li, H.X.; Fang, X.C.; Tang, M.Q.; Yu, H.Z.; Yu, B.; Li, M.; Jing, Z.R.; Meng, Y.H.; Chen, W.; et al. MDCNN: A Multimodal Dual-CNN Recursive Model for Fake News Detection via Audio- and Text-based Speech Emotion Recognition. *Speech Commun.* **2025**, *175*, 103313. [CrossRef]
107. Hong, M.; Wang, H.Y. Filter feature selection methods for text classification: A review. *Multimed. Tools Appl.* **2024**, *83*, 2053–2091.
108. Ganesan, K.; Zhai, C.X. Opinion-based Entity Ranking. *Inf. Retr.* **2012**, *15*, 116–150. [CrossRef]
109. Ciarelli, P.M.; Oliveira, E. Agglomeration and Elimination of Terms for Dimensionality Reduction. In Proceedings of the 2009 Ninth International Conference on Intelligent Systems Design and Applications, Pisa, Italy, 30 November–2 December 2009; pp. 547–552.
110. Ciarelli, P.M.; Salles, E.O.T.; Oliveira, E. An Evolving System Based on Probabilistic Neural Network. In Proceedings of the 2010 Eleventh Brazilian Symposium on Neural Networks, Sao Paulo, Brazil, 23–28 October 2010; Volume 1, pp. 182–187.
111. Kotzias, D.; Denil, M.; De Freitas, N.; Smyth, P. From Group to Individual Labels Using Deep Features. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, 10–13 August 2015; pp. 597–606.
112. Mustafa, A.M.; Rashid, T.A. Kurdish Stemmer Pre-processing Steps for Improving Information Retrieval. *J. Inf. Sci.* **2018**, *44*, 15–27. [CrossRef]
113. Rashid, T.A.; Mustafa, A.M.; Saeed, A.M. A Robust Categorization System for Kurdish Sorani Text Documents. *Inf. Technol. J.* **2017**, *16*, 27–34. [CrossRef]
114. Rashid, T.A.; Mustafa, A.M.; Saeed, A.M. Automatic Kurdish Text Classification Using KDC 4007 Dataset. In Proceedings of the 5th International Conference on Emerging Internetworking, Data & Web Technologies, Wuhan, China, 10–11 June 2017; Volume 6, pp. 187–198.
115. Kilinç, D.; Özçift, A.; Bozyiğit, F.; Yildirim, P.; Yucalar, F.; Borandağ, E. Ttc-3600: A New Benchmark Dataset for Turkish Text Categorization. *J. Inf. Sci.* **2015**, *43*, 174–185. [CrossRef]
116. Wang, G.; Lochovsky, F.H. Feature Selection with Conditional Mutual Information MaxiMin in Text Categorization. In Proceedings of the 2004 ACM CIKM International Conference on Information and Knowledge Management, Washington, DC, USA, 8–13 November 2004; pp. 342–349.

117. Gao, Z.; Xu, Y.; Meng, F.; Qi, F.; Lin, Z.Q. Improved Information Gain-Based Feature Selection for Text Categorization. In Proceedings of the International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE), Aalborg, Denmark, 11–14 May 2014; pp. 1–5.
118. Zheng, Z.; Wu, X.; Srihari, R. Feature Selection for Text Categorization on Imbalanced Data. *ACM SIGKDD Explor. Newsl.* **2004**, *6*, 80–89. [CrossRef]
119. Shang, W.; Huang, H.; Zhu, H.; Lin, Y.; Qu, Y.; Wang, Z. A Novel Feature Selection Algorithm for Text Categorization. *Expert Syst. Appl.* **2007**, *33*, 1–5. [CrossRef]
120. Uğuz, H. A Two-stage Feature Selection Method for Text Categorization by Using Information Gain, Principal Component Analysis and Genetic Algorithm. *Knowl.-Based Syst.* **2011**, *24*, 1024–1032. [CrossRef]
121. Azam, N.; Yao, J.T. Comparison of Term Frequency and Document Frequency Based Feature Selection Metrics in Text Categorization. *Expert Syst. Appl.* **2012**, *39*, 4760–4768. [CrossRef]
122. Rehman, A.; Javed, K.; Babri, H.A. Feature Selection Based on a Normalized Difference Measure for Text Classification. *Inf. Process. Manag.* **2017**, *53*, 473–489. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.