

Research Article

Front-Page News Classification Model Based on the Stacking of Textual Context and Attribute Information

Kan Wang,¹ Kaichen Cao,¹ Mingren Chen,² Zhenzi Yan,³ Lingfeng Zhong ,⁴ Hui Yang,⁵ and Shimin Cai ²

¹Southwest Institute of Electronic Technology, Chengdu 610036, China

²School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China

³School of Foreign Languages, University of Electronic Science and Technology of China, Chengdu 611731, China

⁴Civil Aviation Flight University of China, Guanghan 618307, China

⁵Institute of Southwestern Communication, Chengdu 610041, China

Correspondence should be addressed to Shimin Cai; shimin.cai81@gmail.com

Received 23 May 2022; Accepted 22 September 2022; Published 5 October 2022

Academic Editor: Dongpo Xu

Copyright © 2022 Kan Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The front-page news in authoritative newspapers usually represents extremely significant national policies. Accurately classifying the front-page news from an amount of news helps us to quickly acquire and deeply understand the changing political and economic situations. In this paper, we propose a front-page news classification model StackText based on stacking the textual context and attribute information of news. In the proposed model, we first balance the classification size in the training set through the weighted random sampling algorithm, then construct the context and attribute feature vectors through the textual embedding and statistical analysis of news, and finally pretrain a classifier StackNet based on a neural network to realize the front-page news classification in the testing set. Taking People's Daily as the experimental example, we compare the proposed model with benchmark methods based on statistical learning and deep learning. Based on the news sets in four stages of People's Daily, the experimental results of the front-page news classification show that the proposed model achieves the highest average accuracy and a better balance between precision and recall, which is verified by the corresponding F_1 -score.

1. Introduction

News in authoritative newspapers reports influential social phenomenon and mass media information. In them, these pieces of news that usually represent extremely significant national policies are printed on the front page of authoritative newspapers, which is the so-called front-page news. For example, People's Daily, one of the most authoritative newspapers in China, is a propaganda tool for the Chinese government to announce political and economic views to the outside world [1, 2]. It has played a decisive role in guiding Chinese mainland politics at different times by artificially composing the front-page news. There are many ways from different perspectives to study People's Daily. For the news category in social science, accurately and automatically classifying the front-page news from an amount of news

apparently helps us to quickly acquire and deeply understand the changing political and economic situations [3–5].

With the great technological advances in computer science, many text classification models have been consecutively developed based on machine learning and natural language processing [6–11]. There are two kinds of text classification models. One is based on statistical learning and the other is based on deep learning. They are widely applied for the news classification task, where news is mainly represented by a large number of textual information [12–17]. In the early stage, these text classification models based on statistical learning are used to categorize news into different polarization or topics. For example, the text classification models based on the support vector machine (SVM) classify news in each group of Twitter into positive and negative categories [18, 19]. In further, more feature extraction

methods are introduced to construct the text classification models based on SVM and its variants, which are applied for multitopic news classification [20, 21]. In recent years, the quick development of deep neural networks promotes to study of these text classification models based on deep learning, which has become the mainstream in the field of text classification. A lot of benchmark methods combining word embedding and deep neural networks, such as FastText, TextCNN, TextRNN, and TextRCNN, have been proposed one after another [22–26]. They are frequently applied for news classification [27–29].

In this paper, we aim to propose a novel text classification model for authoritative newspapers (e.g., People’s Daily) to accurately and automatically categorize candidate news into the front-page news or the other news. Apparently, the front-page news classification task is an imbalanced classification problem because the number of the front-page news is much less than that of the other news. The proposed text classification model should consider the feature, which is the unbalance of the number of the front-page news and the other news. For that, we study a framework involving three important modules to construct the front-news classification model StackText based on stacking the textual context and attribute information of news. Especially, the news sampling module in StackText outputs the balanced training set, solving the imbalanced number between the front-page news and the other news, and promoting the classification performance. A more detailed flowchart of StackText will be illustrated in the following section.

In all, we make the following contributions:

- (1) We propose a novel text classification model, StackText, that is suitable for front-page news classification. StackText is constructed based on the stacking of the textual content and attribute information of news, and thus blends more features. In addition, it also introduces the new sampling module to balance the classification size in the training set, promoting the classification performance of the pretrained classifier StackNet based on a neural network.
- (2) We construct diverse experimental subsets from the datasets of news in People’s Daily. Each subset shows the varied styles of different leaderships during the ruling period. These subsets make us evaluate StackText more accurately and adequately.
- (3) We carry out extensive testing experiments and verify the better classification performance of StackText based on the diverse indexes by comparing it with other benchmark methods according to the experimental results.

The reminders of this paper are organized as follows: In Section 2, we present the problem statement of the front-page news classification. Section 3 detailedly illustrates the whole framework of StackText, including three critical modules. In Section 4, the experiment design gives the construction of the dataset and the division of the training

set, the testing set, and the validation set. Section 5 explores the necessity of the news sampling module by contrastively analyzing the experimental results and verifying the effectiveness of StackText. We finally conclude the paper in Section 6.

2. Problem Statement

The news classification task can be described as obtaining such a function mapping,

$$\Phi: D \times C \longrightarrow \{T, F\}, \quad (1)$$

where $D = \{d_1, d_2, \dots, d_n\}$ represents a collection of n news that need to be classified, and $C = \{c_1, c_2, \dots, c_f\}$ represents a collection of f categories under a predefined classification model. Note that the output result of Φ has only two possible values. If $\Phi(d_i, c_j) = T$, d_i belongs to c_j , otherwise it does not belong to c_j . In other words, the aim is to find a valuable function mapping to classify news accurately into categories.

Concretely speaking, the front-page news classification is a binary classification task. If making $c_1 = 1$ indicates the front-page news and $c_2 = 0$ indicates the other news, C is simplified as $C = \{0, 1\}$. Assuming that there is n news in total, each one has the possibility of being classified into two different categories. We obtain the possibility matrix $\text{Prob} = \{\text{prob}_{i,j}\} \in \mathbb{R}^{n \times 2}$. $j = 1$ denotes the category of the front-page news, and $j = 0$ denotes the category of the other news. Note that for each news, the sum of $\text{prob}_{i,1}$ and $\text{prob}_{i,0}$ equals to 1. If $\text{prob}_{i,1} > \text{prob}_{i,0}$, it means that i -th news is the front-page news; otherwise, it is the other news.

3. Model

In this section, we introduce the front-page news classification model, StackText. As shown in Figure 1, the schematic flowchart of StackText is composed of three core modules that are news sampling, feature extraction, and StackNet. Specifically, the news sampling module uses the weighted random sampling algorithm to solve the imbalance of training data labels. The feature extraction module uses the Doc2Vec embedding algorithm to obtain the context feature vector and captures attribute information to construct the attribute feature vector, respectively. The StackNet module is a classifier based on a neural network by stacking the context and attribute feature vectors. To better understand the framework of StackText, each module will be illustrated in the following subsections.

3.1. News Sampling Module. When dealing with the front-page news classification problem of People’s Daily, we find that the number of front-page news is much less than that of the other news. The unbalance of the number of the front-page news and the other news easily makes the classification result biased [30]. That is, candidate news will be classified into the other news in a biased way. To solve such a problem, we introduce a weighted random sampling algorithm that balances the samples of the front-page news, and the other news in training set to acquire a better classification model.

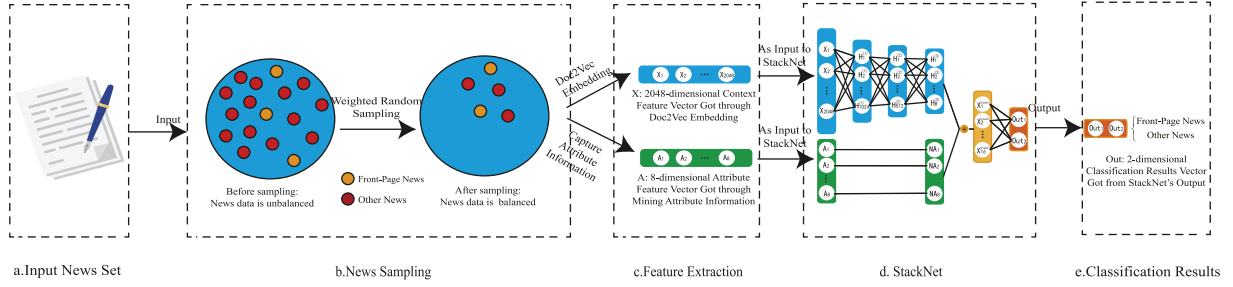


FIGURE 1: The schematic flowchart of StackText.

Let the training set $D = \{d_i\}$ where $i = 1, 2, \dots, n$. Firstly, we quantify the weights of samples (i.e., the front-page news and the other news) in D . For the category c_i with the size S_{c_i} , the weight w_i of the sample d_i belonging to it can be expressed by

$$w_i = \frac{1}{|C| \cdot S_{c_i}} \text{ if } d_i \in c_i, \quad (2)$$

$|C|$ denotes the number of categories, and it is $|C| = 2$ according to the description of the front-page news classification task. Note that each sample in the same category has an identical weight, so that w_i involves two values, w_1 and w_0 . w_1 denotes the weight of the samples of the front-page news and w_0 denotes the weight of the samples of the other news. Then, we use the weighted random sampling algorithm to select m samples from the training set according to their weights. Specifically, the selecting procedure is illustrated as follows:

- (1) Initially, we randomly select n samples from D to construct the reservoir $R = \{r_i\}$ where $i = 1, 2, \dots, n$. Note that the original R includes the imbalanced number of the front-page news and the other news.
- (2) Given a sample $r_i \in R$ with $w_i \in \{w_1, w_0\}$, we generate a uniformly distributed random number $u_i \in (0, 1)$, and set a rank-score $k_i = u_i^{1/w_i}$. Across R , we obtain the rank-score set of n samples.
- (3) We sort the rank-score set and obtain the minimum rank-score k_{\min} .
- (4) We randomly select a sample $d_i \in D$, and compute its rank-score k_j . If k_j is larger than k_{\min} , d_i replaces the sample with the rank-score k_{\min} .
- (5) Repeat steps 3 and 4 in m ($m \geq n$) iterations, and finally, obtain the updated.

It can be guaranteed that the updated R includes the balanced number of the front-page news and the other news, where the ratio between the front-page news and the other news tends to be 1:1[31]. Note that a sample in D can be a repetitive occurrence in the updated R . We use the updated R for the balanced training set D' .

3.2. Feature Extraction Module. The news of People's Daily not only contains the textual context information but also includes the attribute information. For the textual context information, it cannot be directly described by the news

text because of the diversity of news content. Alternatively, the textual context information can be represented by the context feature vector extracted from the news text. Compared with a range of textual feature representation algorithms, Doc2Vec is an unsupervised learning algorithm that can effectively represent the context feature vector of the news text of any [32] length. Figure 2 shows the framework for extracting the context feature vector based on Doc2Vec.

Doc2Vec trains the text vector and word vector at the same time. Let the one-hot encoding vector corresponding to the text of d_i be p_i , and the one-hot encoding vector corresponding to the word t in the text be v_t . The vector for the occurrence of the word t in the text can be constructed as follows:

$$X_{i,t} = h(v_t | p_i) = \sum_{s=t-t'}^{t+t'} v_s + p_i. \quad (3)$$

Substituting the vector $X_{i,t}$ of the word t into the neural network model of Doc2Vec, the output of the hidden layer vector can be expressed as follows:

$$h_{i,t} = \text{ReLU}(X_{i,t} \cdot W_1 + b_1), \quad (4)$$

W_1 is the parameter matrix between the input layer and hidden layer, b_1 is the bias in the neural network model of Doc2Vec, and $\text{ReLU}(X)$ is the activation function.

Then, we use the output of the hidden layer and the softmax function to predict the vector $y_{i,t}$ of the word t , which is expressed as follows:

$$y_{i,t} = \text{Softmax}(h_{i,t} \cdot W_2 + b_2), \quad (5)$$

W_2 is the parameter matrix between the hidden layer and the softmax layer, and b_2 is the bias in the neural network model of Doc2Vec.

Finally, we can use the vector $y_{i,t}$ to construct the loss function of the word t :

$$\text{Loss} = \sum_{i=1}^n \text{distance}(y_{i,t}, v_t), \quad (6)$$

The distance function measures the differences between $y_{i,t}$ and v_t , which is the Euclidean distance in general. By optimizing the loss function, $W_{1,\text{best}}$ and $b_{1,\text{best}}$ can be obtained. Taking p_i as input, the 2048-dimension context feature vector X^{2048} can be obtained:

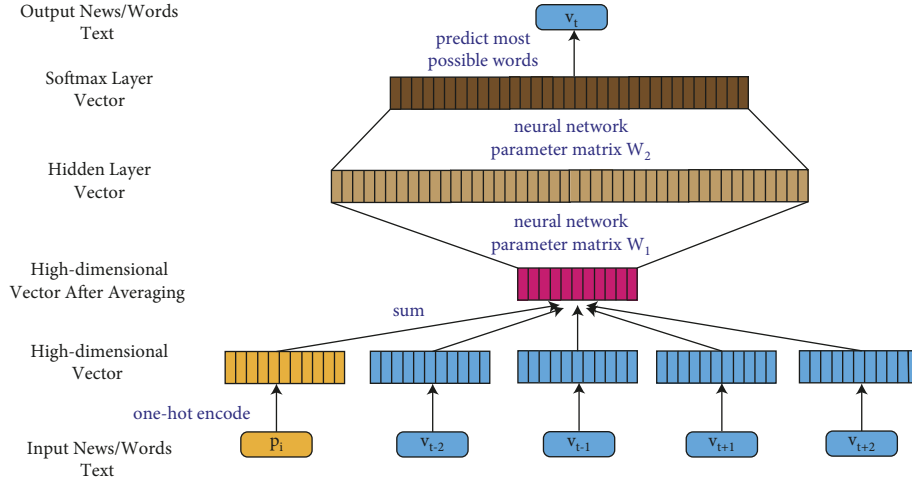


FIGURE 2: The framework of context feature extraction based on Doc2Vec.

$$X^{2048} = p_i \cdot W_{1,\text{best}} + b_{1,\text{best}}. \quad (7)$$

In addition, the candidate news in People's Daily determined as the front-page news, mainly relates to its text content. Nevertheless, due to the limitations of the front-page layout and posttypesetting, some candidate news with a too long or too short length of text content cannot be the front-page news. The context feature vector is not used to characterize the explicit features of the candidate news. Additionally, we define the attribute information that describes the explicit features of the candidate news. The attribute information includes the title length, the maximum and minimum title length in People's Daily of the day, the text length, the maximum and minimum text length in People's Daily of the day, the page number of People's Daily of the day, and the label of time stage (see the definition in the section of the experimental data). They are used to construct the 8-dimension attribute feature vector A^8 .

3.3. StackNet Module. The StackNet module, a classifier based on a neural network, is to stack the context feature vector and attribute feature vector into the refined feature vector and then classifies the candidate news. In this section, we mainly illustrate the framework of StackNet. Denote the feature vector X_{s_i} , the classification label $Y_{s_i} \in \{0, 1\}$ of a sample s_i in D' , and the classification model $F(X)$. The loss function is defined as

$$\text{Loss} = \sum_{i=1}^n \delta(F(X_{s_i}), Y_{s_i}). \quad (8)$$

The δ function is used to measure the consistency of the classification result and the label across the training set. That is, the output of the δ function is 0 when $f(X_{s_i})$ equals to Y_{s_i} . As mentioned above, the feature of s_i is represented by the context feature vector $X_{s_i}^{2048}$ and attribute feature vector $A_{s_i}^8$. We rewrite (8) as

$$\text{Loss} = \sum_{i=1}^n \delta(F(X_{s_i}^{2048}, A_{s_i}^8), Y_{s_i}). \quad (9)$$

We have obtained these feature vectors in the feature extraction module. However, the dimension of the context feature vector is larger than that of the attribute feature vector. When stacking these feature vectors, we need to align their dimensions. As shown in Figure 1, the dimension reduction of $X_{s_i}^{2048}$ is realized by designing a three-layer fully connected neural network. In the first layer of the fully connected neural network, the input is $X_{s_i}^{2048}$, and the output is the 1024-dimension vector, which is expressed as

$$H_{s_i}^{1024} = \text{ReLu}(W_1^{1024} \cdot X_{s_i}^{2048} + b_1^{1024}). \quad (10)$$

The second and third layers of fully connected neural networks are similar to the first layer, which are also the hidden layers that further reduce the dimension of $X_{s_i}^{2048}$. Their outputs are expressed as,

$$\begin{aligned} H_{s_i}^{512} &= \text{ReLu}(W_2^{512} \cdot H_{s_i}^{1024} + b_2^{512}), \\ H_{s_i}^8 &= \text{ReLu}(W_3^8 \cdot H_{s_i}^{512} + b_3^8). \end{aligned} \quad (11)$$

We finally obtain the 8-dimension context feature vector $H_{s_i}^8$ via a three-layer fully connected neural network. The activation function is $\text{ReLu}(X)$. Note that although the more hidden layers can characterize the context feature vector more subtly, they also bring more parameters that result in a higher computational cost. In addition, $A_{s_i}^8$ is normalized by the maximum of each attribute, respectively. The normalized attribute feature vector is denoted as $NA_{s_i}^8$.

Stacking the 8-dimension context feature vector and the 8-dimension attribute feature vector is directly concatenated. The stacking feature vector $X_{s_i}^{\text{stack}}$ is expressed as

$$X_{s_i}^{\text{stack}} = H_{s_i}^8 \oplus NA_{s_i}^8. \quad (12)$$

The classification model is specifically realized by a forward neural network. The input is the stacking feature vector, and the output F_{s_i} is the 2-dimension vector of the classification result via the softmax function, which is expressed as

$$F_{s_i} = \text{Softmax}(W_4^2 \cdot X_{s_i}^{\text{stack}} + b_4^2). \quad (13)$$

F_{s_i} actually includes two probabilities that s_i is classified into the front-page news or the other news. We set s_i with the classification label having the larger probability and minimize the loss function across all samples to obtain StackNet.

4. Experiment Design

4.1. Dataset. In the experiment, we extracted all news published in People’s Daily from 1946 to 2008 to construct the dataset. These pieces of news printed on the front-page are automatically labelled as the front-page news, and the reminder ones are labelled as the other news. Because of the varied styles of different leadership during the ruling period, we artificially divide the dataset into four subsets according to the time scale (see in Table 1).

More concretely, the time scale of stage 1 is from May 1, 1946, to July 15, 1977, which is considered as Mao Zedong’s period. The subset has 445,749 news, including 77,021 front-page news (17.28%) and 368,728 (82.72%) other news. In stage 2, the time scale is from July 16, 1977, to June 24, 1987, which is considered as Deng Xiaoping’s period. The subset has 237,351 news, including 30,542 (12.87%) front-page news and 206,809 (87.13%) other news. In stage 3, the time scale is from June 25, 1987, to November 14, 2002, which is considered as Jiang Zeming’s period. The subset has 476,690 news, including 48,142 (10.10%) front-page news and 428,548 (89.90%) other news. In stage 4, the time scale is from November 15, 2002, to December 31, 2008, which is considered as Hu Jintao’s period. The subset has 207,400 news, including 15,697 (7.57%) front-page news and 191,703 (92.43%) other news. In the experimental design, we divide each subset into the training set, testing set, and validation set according to the ratio 6 : 2:2. The proportions of two kinds of news in each subset are kept in the training set, the testing set, and the validation set.

4.2. Evaluation Index. The unbalance of the number of the front-page news and the other news remains in the testing set, which makes us choose imbalanced classification indexes. Thus, four indexes, specificity, precision, recall (i.e., sensitivity) and F_β -score, are introduced to evaluate the effectiveness of StackText. We first compute the confusion matrix of classification results. It includes four values according to the predicted class and actual class, the true positive rate (TP), which describes the consistency of positive examples (i. e., the front-page news), and the true negative rate (TN), which describes the consistency of negative examples (i.e., the other news), the false positive rate (FP) that describes the positive examples are incorrectly labelled as the negative examples, and the false negative rate (FN) that describes the negative examples are incorrectly labelled as the positive examples. Then, to keep our description as self-contained as possible, we present these indexes briefly.

4.2.1. Specificity. Specificity measures the prediction performance as the percentage of the negative examples correctly predicted among all predicated negative examples. Its specific formula is as follows:

TABLE 1: Attribute information of subset in each stage.

Period	Total news	The front-page news	The other news
Stage1	445749	77021 (17.28%)	368728 (82.72%)
Stage2	237351	30542 (2.87%)	206809 (87.13%)
Stage3	476690	48142 (10.10)%	428548 (89.90%)
Stage4	207400	15697 (7.57%)	191703 (92.43%)

$$\text{specificity} = \frac{TN}{TN + FP}. \quad (14)$$

4.2.2. Precision. Precision measures the prediction performance, which is the percentage of the positive examples among the predicted results. Its specific formula is as follows:

$$\text{precision} = \frac{TP}{TP + FP}. \quad (15)$$

4.2.3. Recall. Recall measures the prediction performance, which is the percentage of the positive examples correctly predicted among all predicated positive examples. Its specific formula is as follows:

$$\text{recall} = \frac{TP}{TP + FN}. \quad (16)$$

Note that the indexes of recall and sensitivity equal in the formula.

4.2.4. F_β -Score. F_β -score is generally defined in mathematics to reconcile the results of precision and recall. Its specific formula is as follows:

$$F_\beta - \text{score} = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}. \quad (17)$$

In the process of merging, the weight of recall is β times precision. In Ref. [33], it illustrates the effect of β on the derivation of F_β -score. In most cases of evaluating classification models, setting $\beta = 1$ thinks that precision and recall are equally important [34]. Herein, we follow the common application of F_β -score and use F_1 -score to evaluate StackText. F_1 -score is defined as follows:

$$F_1 - \text{score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}. \quad (18)$$

5. Experimental Results

The experimental results are obtained by classifying the above-mentioned subsets based on StackText and other benchmark methods. We use diverse benchmark methods, which include the algorithms based on statistical learning, extreme gradient boosting (XGBoost), and random forest (RF), and the algorithms based on deep learning, FastText, TextCNN, TextRNN, and TextRCNN. For these algorithms based on statistical learning, they only use the 2048-dimensional context feature vector in the feature engineering

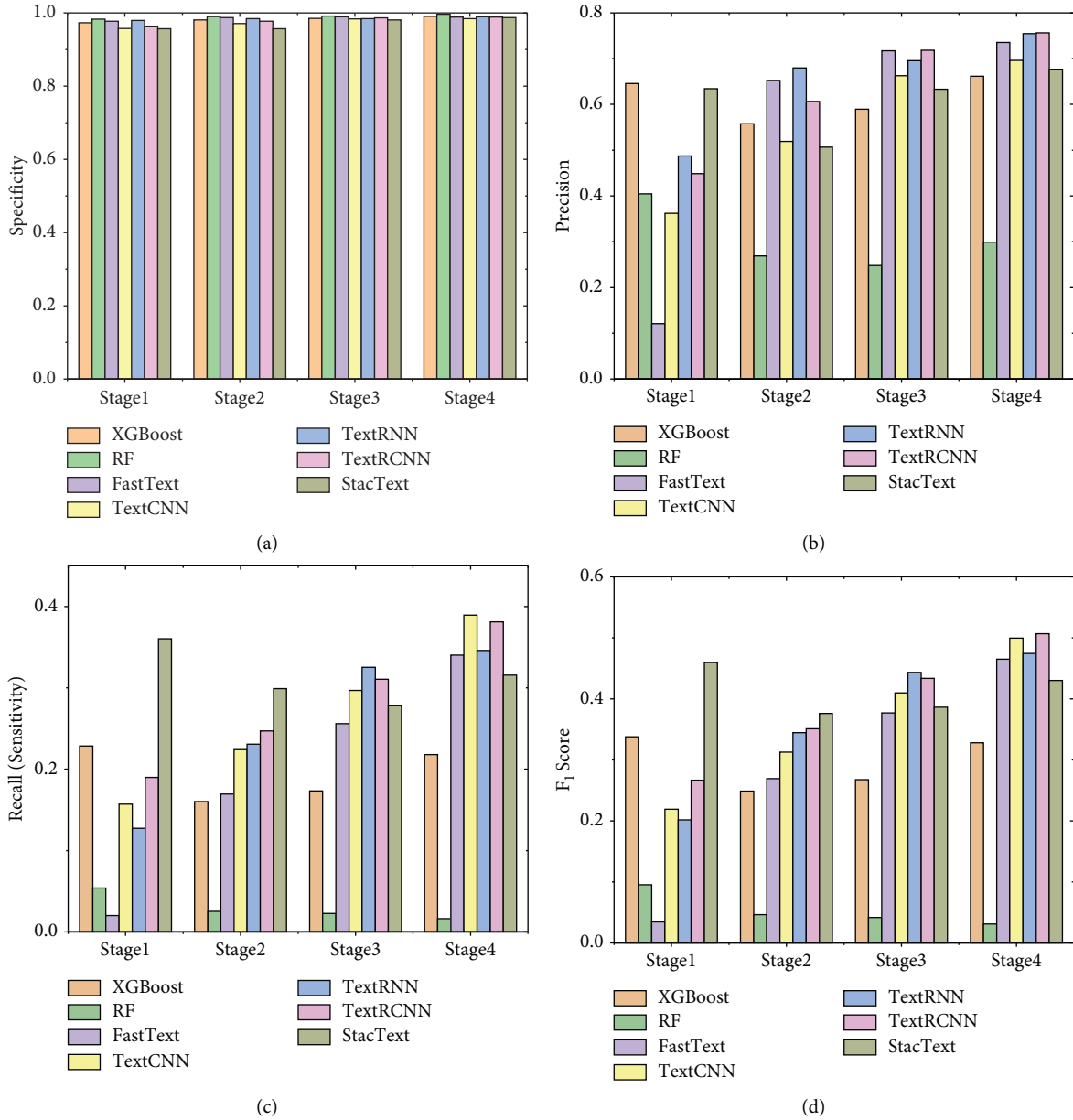


FIGURE 3: The effect of the new sampling module on the classification results of StacText and other benchmark methods based on four subsets.

to train the classification models. For the algorithms based on deep learning, directly input textual context information of news into their corresponding pretrain models to refine the classification models.

Before verifying StackText, we explore the necessity of the new sampling module firstly. By neglecting the new sampling module in the flowchart of StacText, news in the (imbalanced) training set is directly translated into the context feature vector and attribute feature vector based on the feature extraction module. After that, we train StackNet via the input of these two feature vectors and use StackNet to classify candidate news into the front-page ones and others. The same training set is used for these benchmark methods. Figure 3 shows the classification results based on four subsets.

Concretely, specificity and precision are shown in Figures 4(a) and 4(b), respectively. It can be seen that all classification models show high values of specificity and relatively low values of precision. We can easily understand that the unbalance of the number of the front page, and the other news makes the trained classification models prefer to predict the candidate news as the negative examples and results in high values of specificity. It is also verified through lower values of the sensitivity shown in Figure 4(c). However, the interest of the classification task is to correctly predict the candidate news as positive examples. We thus tend to comprehensively consider precision and recall, which is synthetically evaluated by F_1 -score. In Figure 4(d), we find that StackText and the algorithms based on deep learning show relatively high values of F_1 -score in

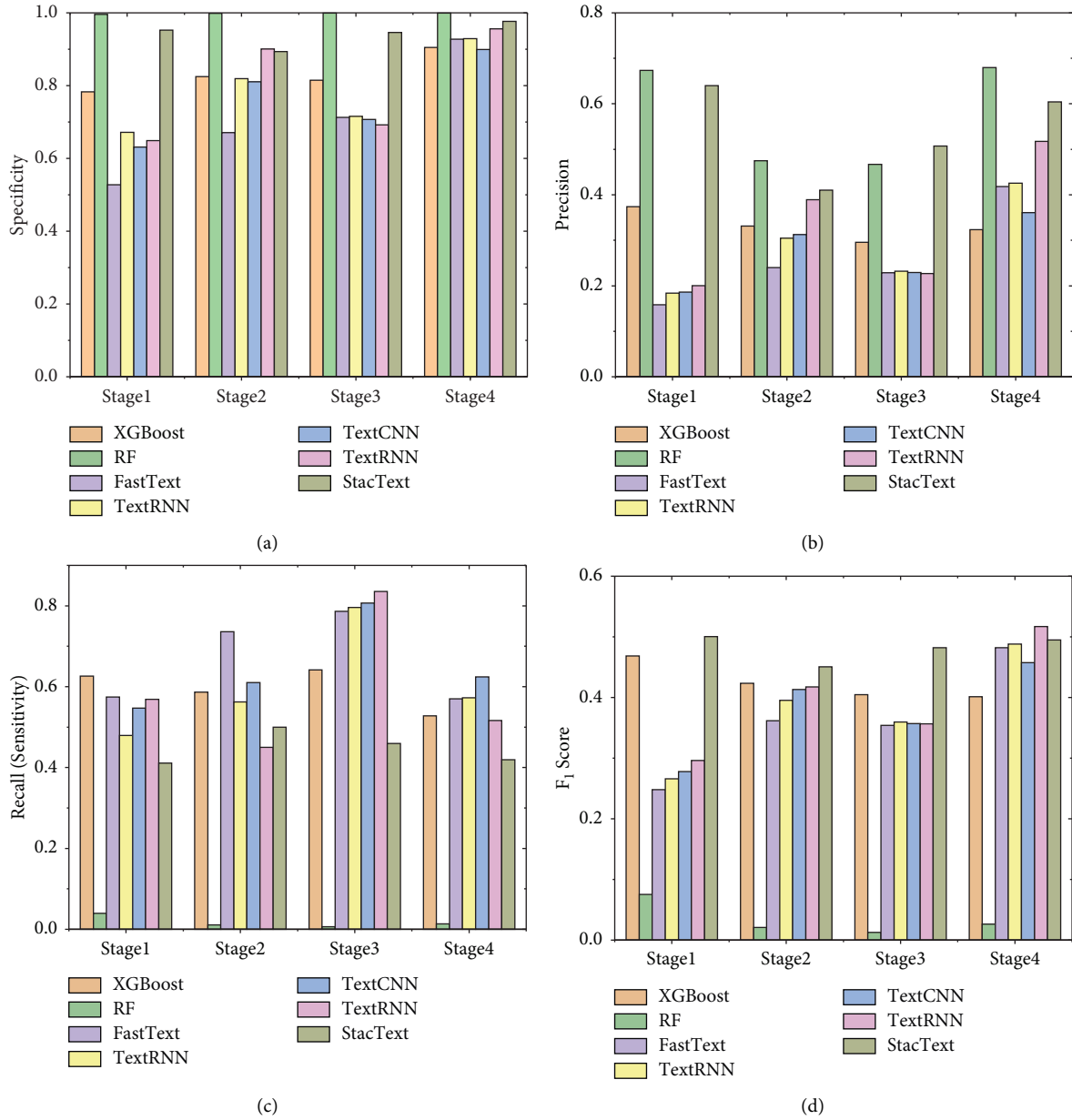


FIGURE 4: The classification results of StacText and other benchmark methods based on four subsets. (a) Specificity, (b) precision, (c) recall(sensitivity), and (d) F1-score.

TABLE 2: The average values (the top 2 performers are highlighted in bold) of four evaluation indexes in respect to 7 classification models when neglecting the new sampling module.

Index	XGBoost	RF	FastText	TextCNN	TextRNN	TextRCNN	StacText
Specificity	0.983	0.990	0.986	0.974	0.985	0.979	0.970
Precision	0.614	0.305	0.557	0.560	0.654	0.632	0.613
Recall	0.195	0.030	0.196	0.267	0.257	0.282	0.313
F_1 -score	0.296	0.054	0.287	0.360	0.366	0.390	0.413

comparison with the algorithms based on statistical learning. On average, we present four evaluation indexes in Table 2, and further find that StackText presents the highest value of F_1 -score.

According to the above-mentioned analysis, we know that the absence of the news sampling module greatly affects

the performance of classification models. Thus, it is expected that the performance (i.e., F_1 -score) of classification models can be improved by introducing the news sampling module to obtain a balanced training set. We train classification models of StackText and other benchmark methods based on the balanced training set and use them to obtain the

TABLE 3: The average values of four evaluation indexes in respect to 7 classification models. In the bracket, four evaluation indexes are increasing or declining degrees in comparison of the classification results in Table 2 (the top 2 performers are highlighted in bold).

Index	XGBoost	RF	FastText	TextCNN	TextRNN	TextRCNN	StackText
Specificity	0.832 (-15.4%)	0.998 (0.8%)	0.710 (-28.0%)	0.784 (-19.5%)	0.762 (-22.6%)	0.799 (-18.4%)	0.942 (-2.9%)
Precision	0.331 (-46.1%)	0.574 (88.2%)	0.261 (-53.1%)	0.287 (-48.8%)	0.272 (-43.1%)	0.333 (-47.3%)	0.541 (-11.7%)
Recall	0.595 (205.1%)	0.018 (-40.0%)	0.667 (240.3%)	0.602 (162.9%)	0.647 (151.8%)	0.593 (110.3%)	0.447 (42.8%)
F_1 -score	0.424 (43.2%)	0.034 (-37.0%)	0.362 (26.1%)	0.377 (4.7%)	0.375 (2.8%)	0.397 (1.8%)	0.482 (16.7%)

classification results. Figure 4 shows the performance of classification models indicated by four evaluation indexes.

Concretely, the values of specificity and precision are reduced except that of RF classification model (see in Figures 4(a) and 4(b)). Nevertheless, the values of sensitivity are improved except for that of the RF classification model (see in Figure 4(c)). The reason is that the balanced training set makes the classification models equally predicate candidate news as the front-page news or the other news. More candidate news is classified into the front-page news, which increases (reduces) the number of true and false positive (negative) samples in comparison with the previous classification results. Meanwhile, we find that the increasing number of true and false positive samples greatly improves the values of F_1 -score.

We average the values of four evaluation indexes in respect to 7 classification models (see in Table 3). In comparison with the classification results in Table 2, we explicitly see that although they uniformly reduce the values of specificity and precision, six classification models (except the RF classification model) increase the values of recall and F_1 -score. It verifies the good effects of the news sampling module. Meanwhile, the classification results also suggest that six classification models present diverse increasing or declining degrees of four evaluation indexes. In comparison with these benchmark methods, StackText shows much better increasing or declining degrees and the highest values of specificity, precision and F_1 -score. Thus, we deem that StackText is a better front-page news classification model, and it is necessary to introduce the news sampling module.

6. Conclusion

The task of categorizing the front-page news of national newspapers and periodicals correctly is of great significance for judging changes in national political and economic policies. In this paper, we propose a front-page news classification model (i.e., StackText) to solve such a task. StackText is constructed based on the stacking of the textual context information and attribute information of news, and its framework includes three important modules involving news sampling, feature extraction, and the classifier based on a neural network (i.e., StackNet).

Taking People’s Daily in the period 1946–2008 as an example, we testify the classification performance of StackText based on four evaluation indexes. The experimental results suggest that StackText outperforms the classification models built via both the algorithms based on statistical learning and the algorithms based on deep learning and is relatively insensitive to the news sampling

module. We know that the front-page news classification is an imbalanced classification task. This also suggests that StackText may be suitable for the imbalanced text classification task to a great extent.

In further, correctly categorizing candidate news into the front-page news via StackText can make us quickly and deeply mine the economic and political policy information. Such a task will be performed in our future works.

Data Availability

The experimental data used to support the findings of this study were supplied by mirror site of People’s Daily under license and so cannot be made freely available. Requests for access to these data should be made to S. C. (e-mail: shimin.cai81@gmail.com).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors’ Contributions

K. W, K. C, M. C, L. Z, H. Y, and S. C. proposed the motivation; K. W, K. C, and M. C. designed and performed the experiment; Z. Y, L. Z, H. Y, and S. C. prepared and wrote the manuscript.

Acknowledgments

This work was partially supported by the National Nature Science Foundation of China (grant nos. 61673086 and 61803352), the Security Construction Foundation of Civil Aviation Administration of China (no. AQ20200019), and the Foundation of CAFUC (no. J2021-072).

References

- [1] Y. Gao and L. Vaughan, “Web hyperlink profiles of news sites: a comparison of newspapers of USA, Canada, and China,” in *ASLIB Proceedings: New Information Perspectives*, vol. 57, pp. 398–411, Emerald Group Publishing Limited, 2005.
- [2] P. Teo and H. Xu, “A comparative analysis of Chinese and american newspaper reports on China’s belt and road initiative,” *Journalism Practice*, pp. 1–20, 2021.
- [3] W. Zhong and J. T. K. Chan, “Reading china: Predicting Policy Change with Machine Learning.” AEI Economics Working Paper Series (No. 2018-11), American Enterprise Institute, Washington, D.C, 2018.
- [4] S. Fan, L. Xue, and J. Xu, “What drives policy attention to climate change in China? an empirical analysis through the

- lens of people's daily," *Sustainability*, vol. 10, no. 9, p. 2977, 2018.
- [5] J. Grimmer, M. E. Roberts, and B. M. Stewart, "Machine learning for social science: an agnostic approach," *Annual Review of Political Science*, vol. 24, no. 1, pp. 395–419, 2021.
 - [6] C. C. Aggarwal and C. X. Zhai, "A survey of text classification algorithms," in *Mining Text Data*, pp. 163–222, Springer, Berlin, Germany, 2012.
 - [7] V. Korde and C. N. Mahender, "Text classification and classifiers: a survey," *International Journal of Artificial Intelligence & Applications*, vol. 3, no. 2, pp. 85–99, 2012.
 - [8] Li Shen, Z. Zhao, T. Liu, R. Hu, and X. Du, "Initializing convolutional filters with semantic features for text classification," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1884–1889, Copenhagen, Denmark, September 2017.
 - [9] K. Kowsari, K. Jafari Meimandi, M. Heidarysafa, Mendu, Barnes, and Brown, "Text classification algorithms: a survey," *Information*, vol. 10, no. 4, 2019.
 - [10] J. Hartmann, J. Huppertz, C. Schamp, and M. Heitmann, "Comparing automated text classification methods," *International Journal of Research in Marketing*, vol. 36, no. 1, pp. 20–38, 2019.
 - [11] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, "Deep learning-based text classification: a comprehensive review," *ACM Computing Surveys*, vol. 54, no. 3, pp. 1–40, 2022.
 - [12] M. I. Rana, S. Khalid, and M. U. Akbar, "News classification based on their headlines: a review," in *Proceedings of the 17th IEEE International Multi Topic Conference*, pp. 211–216, Karachi, Pakistan, December 2014.
 - [13] L. Q. Trieu, H. Q. Tran, and M. T. Tran, "News classification from social media using twitter-based doc2vec model and automatic query expansion," in *Proceedings of the Eighth International Symposium on Information and Communication Technology*, pp. 460–467, New York, NY, USA, December 2017.
 - [14] F. Miao, Pu Zhang, L. Jin, and H. Wu, "Chinese news text classification based on machine learning algorithm," in *Proceedings of the 2018 10th International Conference on Intelligent Human-Machine Systems and Cybernetics*, vol. 2, pp. 48–51, Hangzhou, China, August 2018.
 - [15] B. Büyükköz, H. Ali, and A. Özgür, "Analyzing elmo and distilbert on socio-political news classification," *Proceedings of the Workshop on Automated Extraction of Socio-political Events from News*, pp. 9–18, Marseille, France, May 2020.
 - [16] P. Barberá, A. E. Boydston, S. Linn, R. McMahon, and J. Nagler, "Automated text classification of news articles: a practical guide," *Political Analysis*, vol. 29, no. 1, pp. 19–42, 2021.
 - [17] M. Choudhary, S. S. Chouhan, E. S. Pilli, and S. K. Vipparthi, "Berconvonet: a deep learning framework for fake news classification," *Applied Soft Computing*, vol. 110, Article ID 107614, 2021.
 - [18] I. Dilrukshi, K. De Zoysa, and A. Caldera, "Twitter News Classification Using Svm," in *Proceedings of the 2013 8th International Conference on Computer Science & Education*, pp. 287–291, Colombo, Sri Lanka, April 2013.
 - [19] O. Demirsoz and R. Ozcan, "Classification of news-related tweets," *Journal of Information Science*, vol. 43, no. 4, pp. 509–524, 2017.
 - [20] I. Dilrukshi and K. d Zoysa, "A feature selection method for twitter news classification," *International Journal of Machine Learning and Computing*, vol. 4, no. 4, pp. 365–370, 2014.
 - [21] P. Saigal and V. Khanna, "Multi-category news classification using support vector machine based classifiers," *SN Applied Sciences*, vol. 2, no. 3, p. 458, 2020.
 - [22] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification," in *Twenty-ninth AAAI Conference on Artificial Intelligence*, pp. 2267–2273, AAAI Press, Palo Alto, California, 2015.
 - [23] J. Armand, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov, "Fasttext. zip: compressing text classification models," 2016, <https://arxiv.org/abs/1612.03651>.
 - [24] B. Guo, C. Zhang, J. Liu, and X. Ma, "Improving text classification with weighted word embeddings via a multi-channel textcnn model," *Neurocomputing*, vol. 363, pp. 366–374, 2019.
 - [25] R. A. Stein, P. A. Jaques, and J. F. Valiati, "An analysis of hierarchical text classification using word embeddings," *Information Sciences*, vol. 471, pp. 216–232, 2019.
 - [26] W. Hu, Z. Gu, Y. Xie, Le Wang, and K. Tang, "Chinese text classification based on neural networks and word2vec," in *Proceedings of the IEEE Fourth International Conference on Data Science in Cyberspace*, pp. 284–291, Hangzhou, China, June 2019.
 - [27] K. Shu, L. Cui, S. Wang, D. Lee, and H. Liu, "defend: explainable fake news detection," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, New York, NY, USA, July 2019.
 - [28] W. Wang, F. Ding, B. Li, and J. Tian, "Atextcnn model: a new multi-classification method for police situation," in *Proceedings of the International Conference on Advanced Data Mining and Applications*, Foshan, China, January 2020.
 - [29] C. Raj and P. Meel, "Convnet frameworks for multi-modal fake news detection," *Applied Intelligence*, vol. 51, 2021.
 - [30] A. K. Uysal and S. Gunal, "The impact of preprocessing on text classification," *Information Processing & Management*, vol. 50, no. 1, pp. 104–112, 2014.
 - [31] P. S. Efraimidis and P. G. Spirakis, "Weighted random sampling with a reservoir," *Information Processing Letters*, vol. 97, no. 5, pp. 181–185, 2006.
 - [32] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," *International Conference on Machine Learning*, vol. 32, pp. 1188–1196, 2014.
 - [33] Y. Sasaki, "The truth of the f-measure," *Teach Tutor Mater*, vol. 1, no. 5, 2007.
 - [34] D. Chicco and G. Jurman, "The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, no. 1, p. 6, 2020.