A Web News Classification Method: Fusion Noise Filtering and Convolutional Neural Network

Chunhui He
Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha 410073, China
xtuhch@163.com

Yanli Hu*
Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha 410073, China
huyanli@nudt.edu.cn

Aixia Zhou
College of Meteorology and Oceanography, National University of Defense Technology, Nanjing 211101, Jiangsu Province, China
zhouaixia2011@qq.com

Zhen Tan*
Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha 410073, China
tanzhen08a@nudt.edu.cn

Chong Zhang
Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha 410073, China
chongzhang@nudt.edu.cn

Bin Ge
Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha 410073, China
gebin@nudt.edu.cn

ABSTRACT

As the way of Internet information transfer, web news plays a significant role in information sharing. Considering that web news usually contains a lot of content, after in-depth analysis, we found that not all content is related to the news topic, and a lot of web news contains some noise content, and these noises content have serious interference to the text classification task. So, how to filter noise and purify web news content to improve the accuracy of web news classification has become a challenging problem. In this paper, we proposed a web news classification method via fusing noise detection, BERT-based semantic similarity noise filtering and convolutional neural network (NF-CNN) to solve the problem. In order to comprehensively evaluate the performance of the method, we use the Chinese public news classification dataset to evaluate it. The experimental results demonstrate that our method can effectively detect and filter a lot of noise text and the average F_1 score can reach 95.61% on web news classification task.

CCS Concepts

• **Computing methodologies→Machine learning→Learning paradigms→Supervised learning→Supervised learning by classification.**

Keywords

Web news classification; Noise filtering; NF-CNN; Semantic similarity.

Conference'10, Month 1–2, 2010, City, State, Country.

DOI: http://dx.doi.org/10.1145/12345.67890

1. INTRODUCTION

With the development of the Internet, web news has been widely used to spread important information in our daily lives. It contains both the text and figures. Figure 1 displays a sample of English web news.

Figure 1. A sample of English web news[1].

According to Figure 1, we can find that web news usually contains title (inside the red box), release time (inside the yellow box), body content (inside the green box) and other relevant recommended content (inside the blue box on the right).

[1] http://www.chinadaily.com.cn/a/202005/18/WS5ec1e8ffa310a8b2411565fb.html

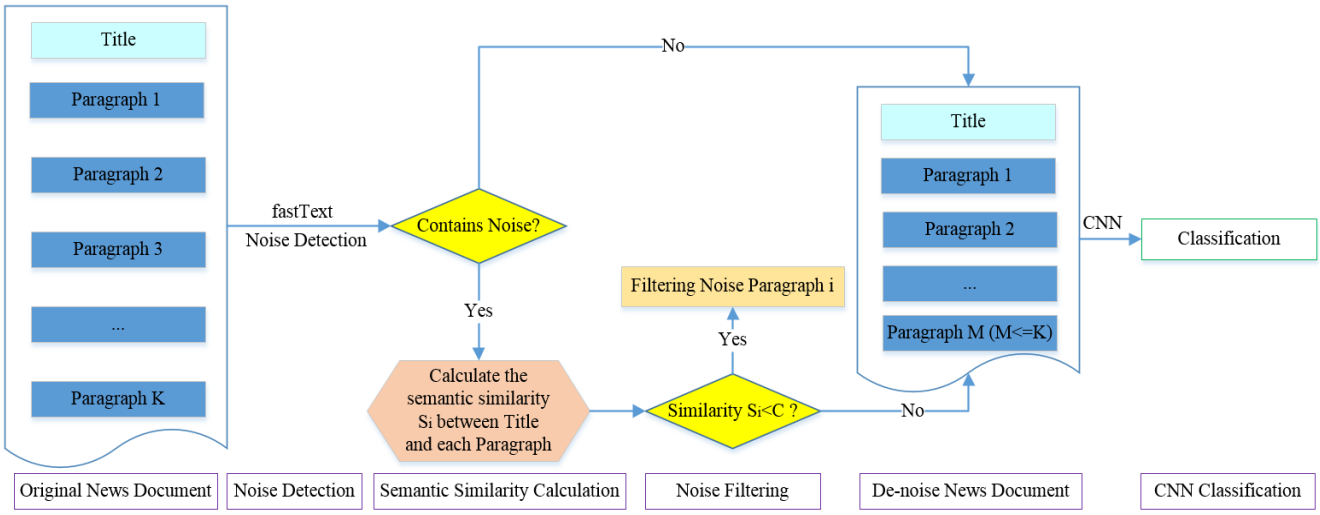

Figure 2. The framework of NF-CNN model.

In recent years, a lot of web news has been published on the Internet. So, when we face massive amounts of web news, how to classify them effectively has become a hot task in the field of natural language processing. Through analysis of the related references, it is found that the current popular method is used by machine learning algorithms, such as SVM [1], Naive Bayes [2] and Text-CNN [3] to directly classify the web news. Unfortunately, by this way, we will face two problems: (a) the web news content usually contains some text that is not directly related to the topic (such as the blue box and yellow box in Figure 1), in fact, they can be regarded as noise in the text classification task. (b) The content of body text is generally longer, for example, the content length exceeds 5000. Some studies have shown that these noise and extra-long text will reduce the performance of the classifier [4, 5]. Thus, before a classification task, how to effectively filtering the noise and purify the original content to improve the classification performance is a very meaningful and challenging task.

In order to solve the problems (a) and (b), in this paper, we proposed a web news classification method via fusing noise filtering and convolutional neural network (NF-CNN). Our method is implementing effective noise filtering via two-stage filtering method with the fastText classification [6] and BERT-based semantic similarity calculations. Finally, we use the de-noised text as corpus to train the CNN web news classification model. Experimental results show that our method achieves better classification performance than the baseline methods on one public Chinese web news classification dataset. In this article, our contributions are as follows:

1) We used a two-stage noise filtering method for detecting and filtering the noise text.
2) We proposed a web news classification method via fusing noise filtering and CNN (NF-CNN).
3) We completed a large-scale noise and non-noise Chinese web news annotation dataset, which can be used to carry out further research in the field of Chinese text noise recognition.

2. RELATED WORK

Noise Filtering. For noise filtering, the noise in the text can be defined as any kind of difference in the surface form of an electronic text from the intended, correct or original text [7]. Kaur S. [8] has done in-depth research on removing noise from web pages. Wang L. [9] proposed a noise recognition method based on the analysis of microblog text stream, which has good performance in filtering the noise of microblog text. Although these methods can solve the problem of partial noise filtering, it should be pointed out that they cannot be directly used for the classification task of web news. In addition, Tan, Z. [10] proposed combining web news title and character-based similarity algorithms to filter noise, and this method has achieved good performance in the news body content extraction task.

Text Classification. For text classification task, we introduce from the two dimensions of text vector representation and classification algorithm. For text representation, VSM [11] was the most elementary text transformation model. In recent years, with the development of word embedding, distributed vector representation methods based on large-scale corpus pre-training have been widely used, such as word2vec [12] and BERT [13] have achieved the start-of-art performance on many natural language processing tasks. For text classification, the early methods were based primarily on traditional machine learning algorithms such as SVM [1], Naive Bayes [2], Random Forest [14], and AdaBoost [15]. At the current stage, with the development of algorithms, fastText [6], CNN [3,16-18] and LSTM [19] and GAN [20] have been extensively used in different applications to solve text classification task. Inspired by the above methods, in this paper, we proposed a web news classification method via fusing noise filtering and convolutional neural network.

3. METHODOLOGY

Web news is generally automatically collected on the internet through web crawlers. In information extraction, we will parse the text according to specific tags (such as <Title></Title>, <h_i></h_i>,) in the html web page to obtain the title and body content of the news (the detailed process is described in reference [10]). Unfortunately, in information extraction, we often encounter noise as part of the body content. In fact, the noise may interfere with the performance of the web news classification. Therefore, we need to filter out noise before classification to improve the performance. Depending on the above analysis, we proposed a web news classification method that fuses noise filtering and CNN (NF-CNN). The framework of the method is shown in Figure 2.

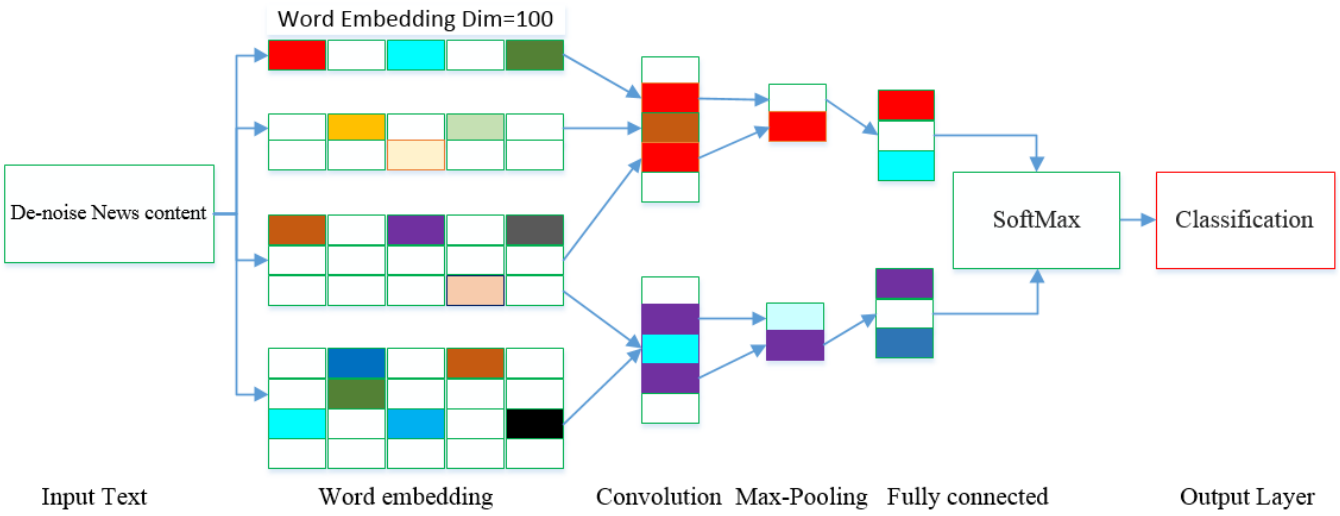

Figure 3. The network design framework of CNN model.

3.1 Two-stage Noise Filtering

According to Figure 2, We can know that noise filtering is divided into two stages. For a web news, it is easy to get its title and all text paragraphs. Here, we used a trained fastText binary classification model (noise, non-noise) to implement noise detection for original content. It should be noted that the corpus used to train the binary fastText classification model was a Chinese text dataset annotated by humans, which contain 30278 normal samples and 21024 noise samples. The experimental results show that the average F_1 score of the binary fastText classification algorithm has reached 93.07%. After using the fastText algorithm to detect the original news content, if it is found that the original content does not contain noise, and the original content will be directly input into the CNN model to complete the classification task. On the contrary, if the fastText algorithm detects that the original content contains noise, it will use the BERT-based semantic similarity algorithm to filter all the noise paragraphs in the original content, and finally input the de-noise content into the CNN to conduct classification task. The BERT-based semantic similarity filtering algorithm is as follows:

1) Get the title T and all paragraphs P in the original content, and add them into a list in order.
2) Use bert-as-service [2] tool to convert the title T and all paragraph content P_i into a fixed-length vectors T_e and P_{ie}, and then use the cosine similarity algorithm to calculate the semantic similarity between T_e and each paragraph P_{ie}, the detailed calculation formula is as follows:

$$S_i(T_e, P_{ie}) = cosine(T_e, V_i) = \frac{T_e^T V_i}{\|T_e\| \|V_i\|} \quad (1)$$

where, T_e and V_i are the fixed-length vectors representation of the title and Paragraph i.
3) According to the calculation results of $S_i(T_e, P_{ie})$, we filter out noise paragraphs with similarity <C (C is set to 0.1 in our experiment).
4) Finally, we input the de-noised title and all paragraph text into the CNN classification model.

3.2 Convolutional Neural Network Model

With the development of deep learning algorithms, Convolutional Neural Network (CNN) has been widely used in the text classification task. In our experiment, we construct the CNN model based on Tensorflow [21] platforms. The CNN model contains input layer, word embedding layer, convolution layer, pooling layer, fully connected layer and output layer. The framework of network design is shown in Figure 3.

In our model, it should be noted that for the word embedding layer, we are divided into four regions, they represent the different word embedding with unigram, bigram, trigram, and 4-gram sequences. n the word embedding layer, we used four vectors to represent the four different features. Finally, a weighted function is used to obtain a fixed-length vector as the vector representation of the entire input corpus. For the detail theoretical introduction to the N-Gram language model, please go to the literature [22]. If the input sentence contains M effective words in unigram, bigram, trigram, and 4-gram sequences, then the vector corresponding to the input sentence is:

$$x = x_1 \oplus x_2 \oplus x_3 \oplus x_4 \quad (2)$$

Where $\oplus$ is the connection operator. The convolutional layer is composed of several units, and the parameters of each convolution unit are obtained by the back-propagation process. The pooling layer is used to reduce the dimension of the feature, and improve the fault tolerance of the model. In our experiments, we using the max sampling method as the pooling method. Through the convolution layer and pooling layer, the obtained feature maps are sequentially expanded in rows and connected into vectors, which are then transferred to the fully connected layer. Then, let $x_{i:j}$ be the connection of the word vector $x_i, x_{i+1}, \cdots, x_{i+j}$. The convolution kernel is $w \in R^{K*d}$, where K is the convolution window size, and it's setting to K=2, 3, and 4 in our method (3 different convolutional layers). Here, d is a word vector dimension. The feature F_i generated by the convolution is:

$$F_i = f(w \cdot x_{i:i+K} + b) \quad (3)$$

Where b is a bias vector, and f is the activated function, the Relu function is used in our experiments. After that, the convolution kernel is applied to every possible window $\{x_{1:K}, x_{2:K+1}, \cdots x_{M-K+1:M}\}$, and finally a feature map is generated:

$$F = [F_1, F_2, \cdots, F_{M-K+1}] \quad (4)$$

The mapping F is given by a maximum pooling operation to obtain the feature $\hat{F} = Max(F)$. Finally, the generated result will be input into the fully connected layer, and the category and probability values will be return in the output layer.

[2] https://github.com/hanxiao/bert-as-service

Table 1. Summary of two experiment datasets.

Dataset	Training Samples	Val Samples	Test Samples	Total Samples	Categories
Sub-THUCNews	96000	12000	12000	120000	12
Noise-Detection	40000	5651	5651	51302	2

Table 2: The experimental results of fastText and baseline models on Noise-Detection dataset.

Model	Avg (Precision)	Avg (Recall)	Avg (F_1)	Time-consuming of Training (S)
Naïve Bayes	85.16%	81.37%	83.22%	386
SVM	87.49%	84.54%	85.99%	1672
fastText	94.57%	91.61%	93.07%	**293**
CNN	**95.02%**	**94.7%**	**94.9%**	1808

4. EXPERIMENTS

4.1 Dataset and Metrics

Dataset. In our experiment, we just select one public Chinese news classification sub-dataset of THUCNews[3] to train and test the CNN news classification model. In addition, we used a private Chinese noise classification dataset to train the fastText noise detection model. The statistical information of all the datasets are shown in Table 1.

According to Table 1, for the sub-THUCNews dataset, which have 12 categories (Finance, Real Estate, Stocks, Home Furnishing, Education, Technology, Society, Fashion, Politics, Sports, Games, Entertainment) and each category contains 10,000 news documents. Although, the original THUCNews dataset contains 14 categories. But, considering that the sample data under category Lottery Ticket and Constellation is less than 10,000. To balance the dataset, we removed all samples under the two categories. The Noise-Detection dataset have 2 categories, and it consists of 51,302 Chinese news text. All division of training, validation and test set is close to 8: 1: 1.

Metrics. In our experiment, we select average Precision, Recall, and F_1 score to measure our model. The Precision and Recall are widely used in the field of information retrieval. The average of Precision, Recall and F_1 score is calculated as following:

$$Avg(Precision) = \frac{1}{N}\sum_{i=1}^{N} Precision_i \quad (5)$$

$$Avg(Recall) = \frac{1}{N}\sum_{i=1}^{N} Recall_i \quad (6)$$

$$Avg(\mathrm{F_1}) = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{2*Precision_i*Recall_i}{Precision_i+Recall_i}\right) \quad (7)$$

where, $Avg(F_1)$ is the average F_1 score for all categories, $\mathrm{Precision_i}$ and $\mathrm{Recall_i}$ are the Precision and Recall for i-th category.

4.2 Experimental Results and Analysis

For noise detection, in our experiment, core parameter settings of fastText model, we used dim = 100, wordNgram = 2, minCount = 1, thread=12. Experimental results of the fastText and baseline models on Noise-Detection dataset are shown in Table 2.

The results in Table 2 show that the average F_1 score of the fastText model has reached 93.07% that is about 7% and 9% higher than SVM and Naïve Bayes. The results show that fastText model can detect most of the noise text. In addition, it should be noted that the classification accuracy of the CNN has reached 94.9% that is higher than the fastText model. But, the time-consuming of training is 1808 seconds for CNN, which is about 6.2 times more than the fastText model. Considering the time cost, in the end we chose the fastText model to finish the noise detection task. In our experiment, we using the CNN model to finish the classification task of de-noise news content. The core parameter settings of the CNN model are presented in Table 3.

Table 3: The core parameter settings of the CNN model.

Parameter Name	Value	Parameter Name	Value
Batch-size	16	Max-length	2048
Embedding-dim (w2v)	100	Epochs	12
Kernel-size	[2,3,4]	dropout	0.5

In order to evaluate the performance of the CNN model fused with noise filtering (NF-CNN). Comparative experimental results on Sub-THUCNews dataset with some benchmark classification models are shown in Table 4.

Table 4: The comparative experimental results on the Sub-THUCNews dataset.

Model	Avg (Precision)	Avg (Recall)	Avg (F_1)
SVM	89.22%	83.45%	86.23%
fastText	92.51%	89.29	90.87%
CNN	94.68%	92.27%	93.46%
NF-CNN (Ours)	**96.84%**	**94.42%**	**95.61%**

The experimental results in Table 4 show that the CNN model has better classification performance on the Sub-THUCNews dataset compared to SVM and fastText, with the highest average F_1 score reaching 93.46%. However, after fusing the noise filtering, the classification performance of CNN model is further improved, and the average F_1 score reaches 95.61% that is about 2.1% increase than no noise filtering CNN model, which fully shows that noise filtering can effectively improve the performance of the classification.

5. DISCUSSION

Here, we share some phenomena discovered in our experiment. First, why we chose fastText as the noise detection model instead of CNN. Mainly considering the time consumption of the training model, the fastText model can achieve an average F_1 score of 93.07% in only 293 seconds. This is an important evaluation principle. Second, in our experiments, we used the Chinese news classification data set to evaluate the performance of the model. In fact, our method is also applicable to Chinese blogs or other semi-structured text classification tasks that include the title.

In addition, in the process of semantic similarity calculation, running the BERT-based vector conversion and similarity

[3] http://thuctc.thunlp.org/#中文文本分类数据集 THUCNews

calculation services is a basic requirement. This is a point on which can be improved. So, we recommended that users set the number of service threads reasonably according to the performance of their machines, which can enhance the overall performance.

6. CONCLUSION and FUTURE WORK

This article proposed a web news classification method via fusing noise filtering and convolutional neural network (NF-CNN). It builds an integrated method to solve noise filtering and web news classification task that can accurately detect noise text and classify the web news content. Experimental results show that the NF-CNN model can effectively improve the performance of web news classification.

In the future, it will be a direction to improve the accuracy of semantic similarity calculation. In addition, implementing this method to other semi-structured text classification task is also a very important work.

7. ACKNOWLEDGMENTS

This research was supported by the National Natural Science Foundation of China (NSFC) via grant No.61872446 and No.61902417. In addition, it is funded by Basic Foundation via grant No. 2019-JCJQ-JJ-231. Yanli Hu and Zhen Tan is the co-corresponding author. Here, we are thanks all of the reviewers for their comments.

8. REFERENCES

[1] Goudjil, M., Koudil, M., Bedda, M., etc. 2018. A novel active learning method using SVM for text classification. International Journal of Automation and Computing, 15(3), 290-298.

[2] Xu, S. 2018. Bayesian Naïve Bayes classifiers to text classification. Journal of Information Science, 44(1), 48-59.

[3] Kim, Y. 2014. Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882.

[4] Yi, L., Liu, B., & Li, X. 2003. Eliminating noisy information in web pages for data mining. In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 296-305).

[5] Guru, D. S., Suhil, M., Raju, L. N., etc. 2018. An alternative framework for univariate filter based feature selection for text categorization. Pattern Recognition Letters, 103, 23-31.

[6] Joulin, A., Grave, E., Bojanowski, P., etc. 2016. Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759.

[7] L. Venkata Subramaniam, Shourya Roy, Tanveer A. Faruquie, etc. 2009. A survey of types of text noise and techniques to handle noisy text. In Proceedings of The Third Workshop on Analytics for Noisy Unstructured Text Data (AND '09). Association for Computing Machinery, New York, NY, USA, 115–122.

[8] Kaur, S., & Singh, I. 2017. An Algorithm to Eliminate Noisy Content from Web Pages (Doctoral dissertation, Lovely Professional University).

[9] Wang, L., Feng, S., & Xu, W. L. 2012. A filtering approach for spam discrimination and content similarity double detection for microblog text stream. computer applications and software, 2, 25-29.

[10] Tan, Z., He, C., Fang, Y., etc. 2018. Title-Based Extraction of News Contents for Text Mining. IEEE Access, 6, 64085-64095.

[11] Salton, G., Wong, A., & Yang, C. S. 1975. A vector space model for automatic indexing. Communications of the ACM, 18(11), 613-620.

[12] Mikolov, T., Chen, K., Corrado, G., & Dean, J. 2013. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

[13] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[14] Xu, B., Guo, X., Ye, Y., etc. 2012. An Improved Random Forest Classifier for Text Categorization. JCP, 7(12), 2913-2920.

[15] Hastie, T., Rosset, S., Zhu, J., etc. 2009. Multi-class adaboost. Statistics and its Interface, 2(3), 349-360.

[16] HE, C. H., ZHANG, C., HU, S. Z., etc. 2019. Chinese News Text Classification Algorithm Based on Online Knowledge Extension and Convolutional Neural Network. In 2019 16th International Computer Conference on Wavelet Active Media Technology and Information Processing (pp. 204-211). IEEE.

[17] Zhang, X., Zhao, J., & Lecun, Y. 2015. Character-level convolutional networks for text classification. neural information processing systems.

[18] Zhu, H., He, C., Fang, Y., etc. 2020. Patent Automatic Classification Based on Symmetric Hierarchical Convolution Neural Network. Symmetry, 12(2).

[19] Bin, G., Chunhui, H., Chong, Z., & Yanli, H. 2018. Classification Algorithm of Chinese Sentiment Orientation Based on Dictionary and LSTM. In Proceedings of the 2nd International Conference on Big Data Research (pp. 119-126).

[20] Li, Y., & Ye, J. 2018. Learning Adversarial Networks for Semi-Supervised Text Classification via Policy Gradient. knowledge discovery and data mining.

[21] Abadi, Martín, et al. 2016. "Tensorflow: A system for large-scale machine learning." 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16).

[22] Brown, P. F., Desouza, P. V., Mercer, R. L., etc. 1992. Class-based n-gram models of natural language. Computational linguistics, 18(4), 467-479.