Electronic Engineering and Informatics
G. Izat Rashed (Ed.)

doi:10.3233/ATDE240092

A Study on an Improved Word Extraction Algorithm Combined with Deep Learning for Recognising Bad Text

Shuchen BAI, Na LIU[1], Tao CAO
School of Information Science and Engineering, Dalian Polytechnic University, Dalian, 116034, China

Abstract. With the popularity of social media networks, the emergence of undesirable texts seriously affects the online environment. Although the existing recognition methods can filter out most of the sensitive information, the recognition of sensitive words with variants is still deficient, on the one hand, an improved word extraction algorithm is designed, which adequately constructs and extends the sensitive word database. On the other hand for the recognition of bad text, in order to improve the recognition accuracy of bad text, this paper proposes TCMHA bad text detection model, which incorporates MultiHeadAttention on the basis of TextCNN model, which improves the recognition rate of sensitive words to a certain extent, and the experimental results prove the effectiveness of the deep learning method.

Keywords. MultiHeadAttention; BERT; TextCNN; deep learning

1. Introduction

Most of the early sensitive word recognition methods are simple matching algorithms, such as the BF (Brute Force) algorithm, the WM algorithm for multi-pattern matching, the KMP (Knuth Mor-ris Pratt) pattern matching algorithm based on the BF algorithm [1-2], and also the decision tree based Deterministic Finite State Automata (DFA) algorithm. is currently a more desirable algorithm for Chinese sensitive word recognition. Guan *et al.* [3] proposed an improved DFA algorithm to optimize the sensitive word pre-processing and recognition methods, which improved the recognition rate of sensitive words. With the development of machine learning technology, algorithms such as plain Bayes and support vector machine (SVM) have also been continuously applied in sensitive word recognition [4].

Deep learning continues to be used in the field of machine learning for its strong learning ability and unsupervised advantages [5-6]. Convolutional neural networks CNNs were initially used in the field of image processing [7] and were later applied in the direction of word processing. Recurrent neural networks RNNs have made great progress in text sequence processing by considering not only the current input [8-10] but also the previous input.

[1] Corresponding author: Na LIU, School of Information Science and Engineering, Dalian Polytechnic University, e-mail: liuna@dlpu.edu.cn

In this paper, on the basis of the common sensitive lexicon, the sensitive lexicon is fully constructed and extended. It also proposes a deep learning model of TextCNN combined with MultiHeadAttention, which improves the recognition rate of undesirable text, and the related research done in this paper is as follows:

(1) Based on the relevant features of sensitive words, this paper uses an improved word frequency algorithm to extract sensitive words, and secondly, summarises and classifies the sensitive words with variants and proposes the corresponding processing methods, and finally completes the expansion of the sensitive word database.

(2) TextCNN model is proposed to extract deeper feature vectors to further improve the bad text recognition accuracy.

(3) MultiHeadAttention is added on the basis of the proposed model to finally obtain excellent recognition results.

2. TCHMA model structure

Traditional deep learning methods still have a big problem for the accurate recognition of bad text, this paper proposes a bad text recognition model based on TextCNN and MultiHeadAttention. The overall structure of the model is shown in Figure 1.

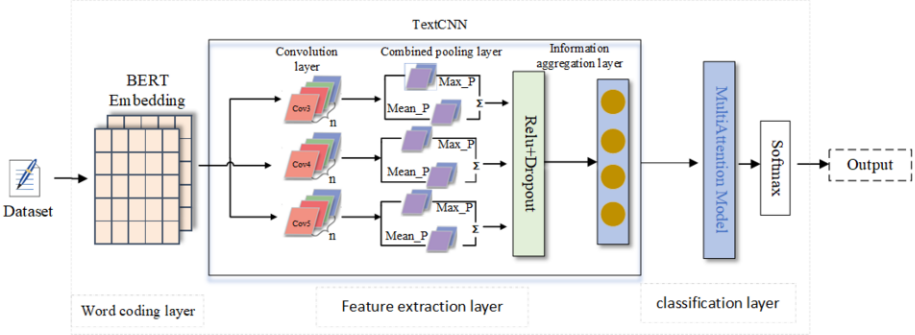

Figure 1. TCHMA model structure

This model fuses TextCNN with MultiHeadAttention mainly includes input layer, convolutional layer, pooling layer, and fully connected layer.

Input layer: the input model to be recognised, after preprocessing, is word vectorised using Bert to facilitate subsequent convolutional operations.

Convolutional layer: the Bert word embedding is processed by TextCNN, the convolutional kernel sizes chosen here are 2, 3 and 4, and the text features are jointly extracted by the convolutional operation.

Pooling layer: the dimension of the feature vector extracted by the convolutional layer is very large, and needs to be processed for dimensionality reduction, here the combination of maximum pooling method and average pooling is used for dimensionality reduction.

Fully connected layer: the MultiHeadAttention layer assigns weights to the dimensionality-reduced word vectors, and the softmax function is used to obtain the recognition results and output them.

3. Sensitive Thesaurus Design Process with Variants

Firstly, the relevant corpus is crawled, and on the one hand, the L-CPBL word extraction algorithm is used to construct the seed word set, and then the seed word set is extended to obtain the sensitive word library; on the other hand, the sensitive word library is further extended by identifying the sensitive word variants. The process is shown in Figure 2:

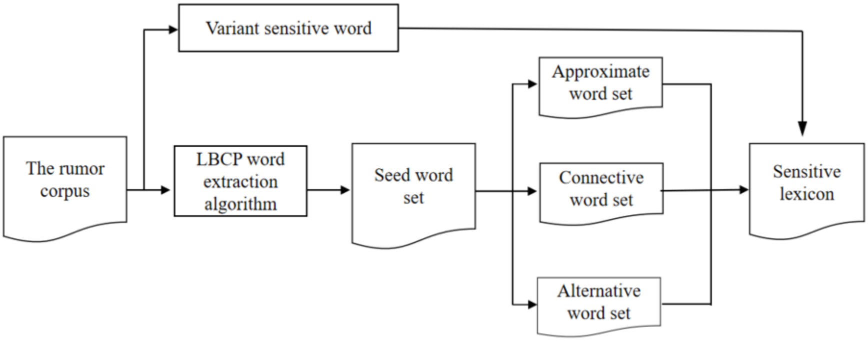

Figure 2. Sensitive thesaurus construction process

3.1 The L-CPBL word extraction algorithm extracts the seed word set

Since most of the current word separation tools are universal in nature and the identification of sensitive words is often domain-specific, previous word separation methods can cause poor results for sensitive word identification. Therefore, this study adopts the LBCP word extraction algorithm using a combination of weights and positions, which first calculates the internal cohesion and external cohesion of words, then adds the improved TF-IDF weights and fuses the position weights to obtain the seed word set; the seed word set is then expanded in terms of approximations, associations, and substitutions, and finally merged into a sensitive word bank.

If the field in the text exceeds a certain threshold, it will be judged as a word output, but the output may not be a complete word, such as “Search Public” and “Public”. “Public” is easier to output as a complete word, the field is represented by XYZ, and P(*) is the probability of the occurrence of a word. Dividing the field into two parts, P(XY)* p(Z) or P(X)*P(YZ) calculates the probability product of the occurrence of the two parts in the corpus (that is, p(left)*p(right)), and the maximum product is the degree of cohesion of the segmentation. The definition formula of internal cohesion is as follows:

$$f(x)=\min\left\{\frac{p(x)}{p(x_1)*p(x_2...x_n)},\frac{p(x)}{p(x_1x_2)*p(x_3...x_n)}...,\frac{p(x)}{p(x_1...x_{n-1})*p(x_n)}\right\} \quad (1)$$

where x is the candidate field and n is the length of the field.

Looking only at the internal cohesion may change the “Search...” It is considered as a word, and the cohesion degree is used to represent the context relationship. If a word can be collocated with other words in different contexts, it is indicated that it is a complete word. For example, “Friends” can be composed of “Delete Friends” and “Add Friends”. Here, the degree of out-cohesion is measured by the left and right information entropy.

$$g(x) = \min\{-\sum p(x_i x_l)\log p(x_i x_l), -\sum p(x_i x_r)\log p(x_i x_r)\} \tag{2}$$

where $x_i x_l$ is x_i and the left word $x_i x_r$ is the right word of x_i.

3.2 Seed word set expansion

The expansion of the seed word set mainly includes approximate words, associated words and alternative words. The seed word vectors are extracted by Word2Vec, and the new word vectors are obtained by adjusting the vector space dimensional weights of the sensitive word seed word sets to obtain the new word vectors after feature space optimization and then by k-means clustering to obtain the similar word sets. An improved mutual information algorithm is used to find the associated word set of the seed word set, and word frequency information is added to the mutual information algorithm to improve the recognition rate of associated words. The process of expanding the seed word set is shown in Figure 3.

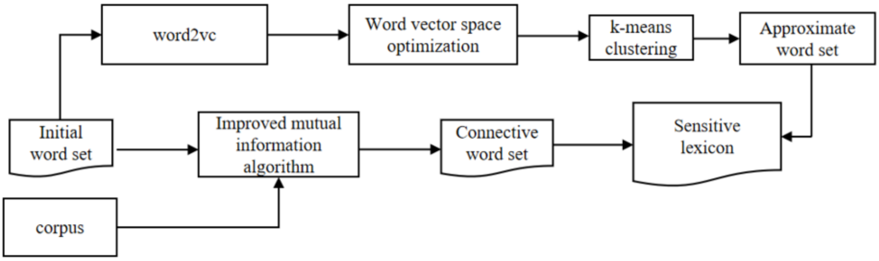

Figure 3. Seed word set extension process

The experiment crawls the microblog text data by crawler, and then retains about 110,000 texts containing bad information (political, pornographic, abusive, gambling, advertisement, and normal information) through manual screening, pre-processes the data, cleanses the data, removes deactivated words, etc., and uses an improved extraction algorithm to extract 310 seed words, and expands the seed word set, which is expanded to about 2,270 words. Part of the seed word set is shown in table 1.

Table 1. Example of Seed Word Set Part

Types	Some examples
Seed word set	abduction, free, winning, forward, sale, missing, urgent, serious, honorarium, underground illegal, religious, dead, explosion ......

4. Bad text recognition based on TCHMA

The experiments in this paper are mainly completed in the Windows environment, the CPU is Inter(R)Core i7-9700, the GPU is GeForce GTX 1080/8 GB, and the programming language is Python 3.7. Hyperparameter selection: A sensitive word recognition algorithm based on TextCNN and the Attention mechanism is designed in this paper. The number of convolution kernels of the TextCNN model is set to 256, the size of the convolution kernel is set to 2, 3, and 4, the learning rate is 0.0001, and the

Adam optimizer is selected as the optimization algorithm. And a Dropout layer is added to prevent overfitting.

4.1 Datasets

For about 90,000 undesirable text messages crawled, 20,000 texts containing undesirable messages and 20,000 normal messages are randomly selected and divided into training set, testing set and validation set in the ratio of 8:1:1. to fully validate the effectiveness of the proposed TCHMA model.

4.2 Evaluation Metrics

This paper uses indicators including *precision, recall,* and *F1* value to evaluate the recognition effect of sensitive words. The calculation formula is as follows.

$$Precision = \frac{T_p}{T_P + F_P} \tag{3}$$

$$Recall = \frac{T_P}{T_P + F_N} \tag{4}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{5}$$

4.3 Comparative experiments and analyses

Table 2. Experimental Data Results

Encoding mode	Model	Index	Precision	Recall	F1	Accuracy
Word2Vec	TextCNN	Normal	0.82776	0.78734	0.80704	83.58%
		Abnormal	0.84624	0.85432	0.85026	
		Macro avg	0.83700	0.82083	0.82865	
		Weighted avg	0.84007	0.83197	0.83584	
	TCMHA	Normal	0.80175	0.82079	0.81116	80.53%
		Abnormal	0.80334	0.80127	0.80230	
		Macro avg	0.80255	0.81103	0.80673	
		Weighted avg	0.80281	0.80778	0.80526	
BERT	TextCNN	Normal	0.83122	0.85276	0.84185	84.15%
		Abnormal	0.84256	0.84435	0.84345	
		Macro avg	0.83689	0.84856	0.84265	
		Weighted avg	0.83878	0.84716	0.84292	
	TCHMA	Normal	0.85741	0.82780	0.84235	86.73%
		Abnormal	0.87338	0.88641	0.87985	
		Macro avg	0.86540	0.85711	0.86110	
		Weighted avg	0.86805	0.86685	0.86734	

As can be seen from Table 2, The Word2Vec-TextCNN model has an F1 score of 0.80704 for normal text recognition and 0.85026 for bad text recognition, indicating that the model has higher accuracy in bad text recognition.Compared to Word2Vec-TextCNN, the Word2Vec-TCMHA model shows a slight decrease in performance on all metrics, which may be due to the fact that the TCMHA model is not as effective as TextCNN for Word2Vec-generated word embeddings.BERT-TextCNN outperforms all Word2Vec configurations with F1 scores of 0.84185 and 0.84345 for normal and bad text recognition, respectively, with an accuracy of 84.15%.The most significant improvement occurs with the BERT-TCHMA configuration, which achieves an F1 score of 0.87985 for bad text recognition and 0.84235 for normal text, with an accuracy of 86.73%, which emphasises the power of the BERT model in combination with TCHMA in dealing with bad text recognition tasks.

In summary, the BERT model, especially when combined with TCHMA, shows superior performance on bad text recognition tasks. This may be attributed to the deep semantic comprehension capabilities of BERT as well as the fine-grained attentional marshalling of TCHMA. Furthermore, these findings suggest the importance of advanced attentional mechanisms in improving the model's ability to recognise complex text patterns.

5. Conclusion

Sensitive word recognition is inseparable from the construction of the sensitive word library, this paper through the word extraction algorithm to extract the seed word set fusion of positional weights to build a sensitive word library, and fusion of sensitive words with variants to expand the word library, and ultimately achieve better experimental results, in terms of deep learning-based recognition of undesirable text to improve the recognition accuracy. However, the sensitive word variants are real-time as well as complex and variable, and in the follow-up work, we continue to consider the optimisation of model parameters as well as the study of other variants to further improve the accuracy of sensitive word recognition.

References

[1] Liang, Y. Tohti, T. and Hamdulla, A. (2022) Multimodal false information detection method based on Text-CNN and SE module. *J. Plos One*, 11.https://dio.org/10.1371/journal.pone.0277463

[2] Wang, H. He, J. and Zhang, X. (2020) A Short Text Classification Method Based on N-Gram and CNN. J. *Chinese Journal of Electronics*, 29: 248-254. https://doi.org/10.1049/cje.2020.01.001

[3] Guan, X. Zhao, M. and Wu, W. (2023) A short text information filtering algorithm based on FA. J. *Software Guide*, 103-108.

[4] Li, Y. Wang, X. and Xu, P. (2019) Chinese Text Classification Model Based on Deep Learning. J. *Future Internet*, 10. https://doi.org/10.3390/fi10110113

[5] Xu, G. Yu, Z. Yao, H. Li, F. and Meng, Y. (2019) Chinese Text Sentiment Analysis Based on Extended Sentiment Dictionary. J. *IEEE Access*, 7: 43749-43762. https://doi.org/10.1109/ACCESS.2019.2907772

[6] Xu, G. Wu, X. Yao, H. Li, F. and Yu, Z. (2019) Research on Topic Recognition of Network Sensitive Information Based on SW-LDA Model. J. *IEEE Access*, 7: 21527-21538.https://doi.org/10.1109/ACCESS.2019.2897475

[7] Xu, G. and Yu, Z. (2018) Efficient Sensitive Information Classification and Topic Tracking Based on Tibetan Pages. J. *IEEE Access*, 6: 55643-55652. https://doi.org/10.1109/ACCESS.2018.2870122

[8] Sun, X. and Huo, X. (2022) Word-Level and Pinyin-Level Based Chinese Short Text Classification. J. *IEEE Access*, 10: 125552-125563. https://doi.org/10.1109/ACCESS.2022.3225659

[9] Gao, Z. Li, Z. Luo, J. and Li, X. (2022) Short Text Aspect-Based Sentiment Analysis Based on CNN plus BiGRU. j. *Applied Sciences-Basel*, 12. https://doi.org/10.3390/app12052707

[10] Dashtipour, K. Gogate, M. Adeel, A. Larijani, H. and Hussain, A. (2021) Sentiment Analysis of Persian Movie Reviews Using Deep Learning. J. *Entropy*, 23. /https://doi.org/10.3390/e23050596