Triplet Embedding Convolutional Recurrent Neural Network for Long Text Semantic Analysis

Jingxuan Liu[1], Ming Zhu[2], Huajiang Ouyang[3], Guozi Sun[4], and Huakang Li[5⋆]

[1] [5]School of AI and Advanced Computing, Xi'an Jiaotong-Liverpool University,
111 Ren'ai Road, Suzhou Industrial Park, Suzhou,
215123, Jiangsu Province, China
{Huakang.Li}@xjtlu.edu.cn
[2] [4]School of Computer Science, Nanjing University of Posts and Telecommuncations,
Nanjing 210003, China
{sun}@njupt.edu.cn
[3] School of Engineering, University of Liverpool,
Liverpool, L69 3GH, United Kingdom
{H.Ouyang}@liverpool.ac.uk

Abstract. Deep Recurrent Neural Network has an excellent performance in sentence semantic analysis. However, due to the curse of the computational dimensionality, the application in the long text is minimal. Therefore, we propose a Triplet Embedding Convolutional Recurrent Neural Network for long text analysis. Firstly, a triplet from each sentence of the long text. Then the most crucial head entity into the CRNN network, composed of CNN and Bi-GRU networks. Both relation and tail entities are input to a CNN network through three splicing layers. Finally, the output results into the global pooling layer to get the final results. Entity fusion and entity replacement are also used to retain the text's structural and semantic information before triplet extraction in sentences. We have conducted experiments on a large-scale criminal case dataset. The results show our model significantly improves the judgment prediction task.

Keywords: Long text analysis · CRNN · Triplet embedding · Entity fusion and entity replacement

1 Introduction

Text analysis refers to various methods to learn and understand the meaning of the text, including analysis of long and short texts [1]. Although long text analysis has been quite effective in some domains, like social and financial, there are specific domains, like the judicial domain, where the application of artificial intelligence is lacking. So we propose a model for triplet embedding Convolutional Recurrent Neural Network (CRNN), which can be applied in crime and relevant

⋆ Corresponding author.

legal long text prediction. Long text analysis can assist professionals in making judgments and can facilitate domain intelligence and has specific research implications.

To solve the problem of long text identification, researchers have adopted a variety of approaches to achieve more accurate results in long text semantic analysis, which include: statistical rules [2], machine learning [3] and deep learning using neural networks [4]. Still, statistical rules lack practicality and robustness. Meanwhile, machine learning is still based on shallow text features and cannot judge complex texts.

To sum up, this paper uses multi-entity fusion and entity replacement (EFR) from the characteristics of long text data to reduce the dimensionality of the word vector space. Use the specified person's name to replace the corresponding victim and perpetrator's name in different texts, and the experimental results show that the EFR is better than the remaining approaches. Meanwhile, this paper uses semantic dependency analysis to obtain the semantic structure tree information of text and combines the sequences of its dependency triples as features and proposes a TeCRNN model based on dependency relations. The experimental results show that the TeCRNN model has better robustness. In summary, the main contributions of this paper are the following:

1. The use of entity recognition techniques to identify entities in text and the use of predefined entities to replace the recognition results, which to some extent acts as a dimensionality reduction and preserves the semantic structure of the text.

2. The use of dependency triples to obtain the semantic as well as structural relationships of factual text. The input text sequence is changed into a sequence of dependent triples, and then the semantic and structural features of the sequence are automatically obtained.

The rest of the paper is organized as follows. Section 2 includes a short review of related work. Section 3 describes the details of the proposed model. Section 4 presents the experimental design. Finally, we conclude the paper and discuss future work in Section 5.

2 Related Work

The literature related to this paper is reviewed regarding two aspects: core sentence matching of long text and full text matching of long text.

2.1 Core sentence matching

Long text analysis can be studied in some ways, one of which is to determine the degree of similarity of text content by looking at the core sentences of the text. Kim et al. proposed a word-level representation and fed it into a Convolutional Neural Network (CNN) [5] for classification [6]. Long et al. extracted the textual features of the different contents using a pairwise mutual attention mechanism to capture the interactive features between different texts [7]. In addition, Yang et al. proposed to use Long-Short Term Memory (LSTM) [8] and a self-attentive mechanism [9] to embed the law definition and fact description into a low-dimensional space and then capture multiple repetitive interaction features between the fact description and the law [10].

2.2 Full text matching

Due to the complexity and diversity of long texts, covering many different topics and domains, full text matching is required to determine the degree of similarity of text content.. Yang et al. modeled the text-level and introduced a hierarchical attention mechanism into the model for computation [11]. Wei et al. use deep learning models, including CNN, Gated Recurrent Unit (GRU) [12], and attention mechanisms to obtain feature vectors and combine the predicted number of labels of the dataset with the output probability to get the result of label classification [13]. Besides, Li et al. proposed a multi-channel attention neural network model, which learns semantic representations of different features from long text and performs integrated tasks in a unified framework. The model further improves the performance of long text analysis [14].

3 Methodology and Model Design

3.1 Framework of EFR

Chinese is semantically focused, this paper mainly uses semantic dependency analysis [15] for long text analysis, which describes the dependency relationships between individual words. We use the dependency syntactic analysis of the Language Technology Platform (LTP) [16]. Specifically, segmentation of factual texts, different annotations are added to the relationships between words, and the dependency triad is obtained by LTP dependency analysis. The flow of data preprocessing is shown in Figure. 1.

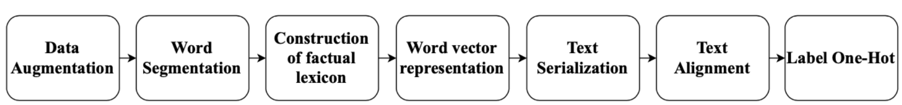

Fig. 1. Data processing flow

First of all, the resampling method [17] was selected for data augmentation. Then the results of word segmentation of authentic judicial texts are analyzed statistically, and each different word is numbered and mapped one by one. The results of word separation are input into the Word2vec tool, and the word vectors are trained using CBOW. After that, interconversion of the results of the judicial text segmentation with the numbers in the factual dictionary, transforming the word sequences into numerical sequences. Then select a threshold value of the text length, using forward truncation and backward completion. Finally, the label is initialized, whose length is the number of categories of the label, and the corresponding position value of the label is changed.

3.2 SCRNN

The long text has many entities, so the choice was made to fuse the names of non-key persons, normalize them into a unified entity, and then replace all place names with entity nouns, which reduces the errors due to the segmentation of words. The model architecture designed in this section is the SCRNN model.

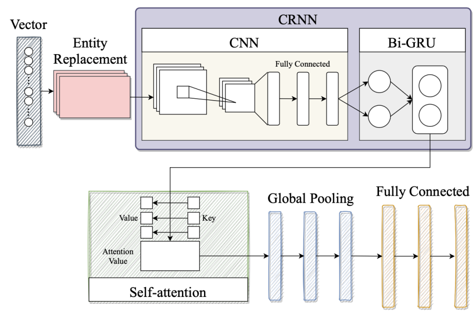

Fig. 2. Framework diagram of SCRNN model

As shown from Figure. 2, the model has three main parts: entity replacement and word vector representation, CRNN, and Self-attention.

Entity replacement and word vector representation: First, a regular expression replaces the determined result with a common entity name. Next, the new text is segmented into words using a custom dictionary with a word splitting tool. Finally, the new word vector is trained using the CBOW method in Word2vec.

CRNN: Use the CNN layer to extract the text surface features. The Bi-GRU is used to obtain the discourse order features.

Self-attention layer: It is less dependent on external data and better at capturing the internal relevance of data. In addition, there are also multiple events in the judicial fact text, and the inherent causal links between multiple events can also affect the final verdict.

3.3 TeCRNN

The TeCRNN model consists mainly of the CRNN model with auxiliary vectors based on the dependency triad, and the use of CRNN after the head vector allows for more sequential features to be obtained. The architecture is shown in Figure. 3.

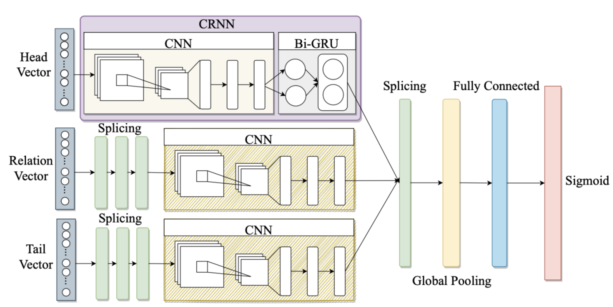

Fig. 3. Framework diagram of TeCRNN model

In the model in Figure. 3, the head vector (A vector) is the sequence union of the head in the input triplet sequence, the tail vector is the set of sequences of the tail, and the relation vector is the set of dependency vectors. In this paper, the A vector is spliced with the relation vector and denoted as the B vector, and the B vector is spliced with the tail vector and denoted as the C vector.

$$A = head \tag{1}$$

$$B = [head; relation] \tag{2}$$

$$C = [head; relation; tail] \tag{3}$$

In the first step, the A, B and C vectors are passed through the CNN layer respectively to extract the textual multivariate features $ConV_A$, $ConV_B$, $ConV_C$.

$$ConV_A = f(W_A \cdot A + b_A) \tag{4}$$

$$ConV_B = f(W_B \cdot B + b_B) \tag{5}$$

$$ConV_C = f(W_C \cdot C + b_C) \tag{6}$$

Where f is the activation function, W_A, W_B, W_C are the convolution kernels, resulting in the output of 3 classes of convolutional features.

In the second step, the vector $ConV_A$ as the input to the Bi-GRU.

$$\overrightarrow{h} = \overrightarrow{GRU}(ConV_A) \tag{7}$$

$$\overleftarrow{h} = \overleftarrow{GRU}(ConV_A) \tag{8}$$

$$H = [\overrightarrow{h}; \overleftarrow{h}] \tag{9}$$

$\overrightarrow{h}$ represents the forward network and $\overleftarrow{h}$ represents the backward network. $GRU()$ represents the calculation of the GRU model to obtain the hidden layer output H.

In the third step, H, $ConV_B$, $ConV_C$ are stitched together through the splicing layer to obtain the fusion vector D.

$$D = [H; ConV_B; ConV_C] \tag{10}$$

In the fourth step, Global Max Pooling to obtain the most obvious features of the model and reduce the dimensionality of the data.

$$P = GlobalMaxPooling(D) \tag{11}$$

Finally, the first fully connected layer uses the ReLU function, which implements the non-linear feature transformation. Tthe second layer uses the Sigmoid function as the final classification function, setting the corresponding threshold — considering probabilities greater than this threshold as actual labels and those less than this threshold as non-labels, to achieve the prediction of the result.

4 Experimental Results and Analysis

4.1 Experimental environment

The model runs mainly in a Linux environment with 32G of RAM and 6G of video memory, using the Keras 2.1.2 deep learning framework based on TensorFlow 1.4.0. The programming language is Python 3.6.

4.2 Dataset statistic

In this paper, we use the dataset of CAIL2018 [18], which includes 154,592 train set, 17,131 valid set, and 32,508 test set.

4.3 Baselines

For comparison, we used several models as baselines, including the base CNN, Bi-LSTM, Bi-GRU, the CNN model with Self-attention added, and the main SCRNN and TeCRNN models proposed in this paper. Meanwhile, we remove the Bi-GRU layer from the TeCRNN model, denoted as the TeCNN model, and add Self- attention to the TeCNN model. In summary, this paper selected eight models for experiment.

4.4 Experiment parameters setting

The CNN model has 128 convolutional kernels, the convolution kernel size is 3, the stride is 1, the activation function is ReLU, the Pooling method is Global Max Pooling for CNN. Based on a combination of operational efficiency and effects, the number of neurons used is chosen to be 64 for Bi-GRU.

Due to the imbalance data nature of the dataset used in this paper. In view of this, we employ $F_1 - micro$, $F_1 - macro$ and $Accuracy$ [19], which are more suitable for imbalance datasets, as evaluation metrics.

4.5 The impact of the EFR method

To verify the effectiveness of the EFR method, this section uses a single-layer CNN, Bi-GRU, and Bi-LSTM model as the base model. The experimental results are shown in Table 1.

Table 1. Results for different text processing methods on different models

Pre-processing method	Type of model	$F_1 - micro$	$F_1 - macro$	$Accuracy$
Segmentation word	CNN	80.51	63.78	72.15
	Bi-LSTM	81.10	67.03	74.07
	Bi-GRU	81.09	67.01	74.05
Stop word list	CNN	80.49	63.90	72.20
	Bi-LSTM	81.04	66.93	73.99
	Bi-GRU	81.33	66.89	74.11
EFR	CNN	81.15	65.91	73.53
	Bi-LSTM	81.78	67.83	74.81
	Bi-GRU	**81.98**	**67.95**	**74.97**

As shown in the Table 1, the EFR method proposed in this paper also reduces the dimensionality of the data in the actual prediction process and retains the structural features of the text, laying a solid foundation for further analysis of the text. The EFR method shows some improvement compared to using only segmentation or the stop word list.

4.6 Experimental results of all models comparison

To verify the validity of dependencies on judicial decisions, this section conducts experiments on all models; the results are shown in Table 2.

Table 2. Comparison of all related models

Type of model	Related legal articles prediction			Crime prediction		
	$F_1 - micro$	$F_1 - macro$	$Accuracy$	$F_1 - micro$	$F_1 - macro$	$Accuracy$
CNN	81.15	65.91	73.53	83.71	73.73	78.72
Bi-LSTM	81.78	67.83	/	/	/	/
Bi-GRU	81.98	67.95	/	/	/	/
CNN+Self-attention	81.99	69.21	75.60	84.52	75.13	79.83
SCRNN	**83.41**	69.98	76.70	**86.13**	75.12	80.63
TeCNN	81.97	71.49	76.73	84.81	75.76	80.29
TeCNN+Self-attention	81.83	71.01	76.42	84.84	75.31	80.08
TeCRNN	81.51	**72.45**	**76.98**	85.65	**76.25**	**80.95**

As can be seen in the Table 2, in the related legal articles prediction task. The TeCRNN model outperformed the SCRNN model by 0.28% in the $Accuracy$ and improved by 2.47% in $F_1 - macro$, fully reflecting the role of dependencies in the small class. TeCRNN model enhanced by 0.25% of $Accuracy$ compared to the TeCNN model. TeCRNN improved all evaluation metrics in the crime prediction task compared to the TeCNN model.

Overall, the SCRNN model outperformed the other models in terms of $F_1 - micro$, while the $F_1 - macro$ and $Accuracy$ of the TeCRNN model are higher than other models. As we can see, our model has better generalization performance on the judicial decision prediction task of long text semantic analysis.

5 Conclusion

This paper proposes using EFR as an alternative to stop word lists to retain the structural and semantic information of the text to the greatest extent possible and reduce the dimensionality of the data. The model's results using standard stop word lists for the data are similar to those without processing, further illustrating the disadvantages of non-proprietary stop word lists. In addition, the model that uses dependencies has a significant advantage in acquiring features, even when compared to models that incorporate Recurrent Neural Networks or Self-attention, and has a competitive advantage.

Acknowledgements This work was supported by the "Six talent peaks" High Level Talents of Jiangsu Province (XYDXX-204), Province Key R&D Program of Jiangsu (BE2020026), XJTLU Research Development Funding (RDF-20-02-10), Suzhou Science and Technology Development Planning Programme-Key Industrial Technology Innovation-Prospective Applied Basic Research Project (SGC2021086), Special Patent Research Project of China National Intellectual Property Office (Y220702).

References

1. Hassan, A., Mahmood, A.: Deep learning approach for sentiment analysis of short texts. In: 2017 3rd international conference on control, automation and robotics (ICCAR), pp. 705-710. (2022)
2. Lauderdale, B. E., Clark, T. S.: The Supreme Court's many median justices. In: American Political Science Review, 106(4), pp. 847-866. (2012)
3. Gonçalves, T., Quaresma, P.: Is linguistic information relevant for the classification of legal texts?. In: Proceedings of the 10th international conference on Artificial intelligence and law, pp. 168-176. (2005)
4. Liu, P., Qiu, X., Huang, X.: Recurrent neural network for text classification with multi-task learning. arXiv preprint arXiv:1605.05101. (2016)
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778. (2016)
6. Kim, Y.: Convolutional neural networks for sentence classification. arXiv: 14085882. (2014)
7. Long, S., Tu, C., Liu, Z., Sun, M.: Automatic judgment prediction via legal reading comprehension. In: China National Conference on Chinese Computational Linguistics, pp. 558-572. Springer, Cham (2019)
8. Ma, X., Tao, Z., Wang, Y., Yu, H., Wang, Y.: Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. In: Transportation Research Part C: Emerging Technologies, 54, 187-197. (2015)
9. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555. (2014)
10. Yang, Z., Wang, P., Zhang, L., Shou, L., Xu, W.: A recurrent attention network for judgment prediction. In: International Conference on Artificial Neural Networks, pp. 253-266. Springer, Cham (2019)
11. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E.: Hierarchical attention networks for document classification. In: Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies, pp. 1480-1489. (2016)
12. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078. (2014)
13. Wei, D., Lin, L.: An external knowledge enhanced multi-label charge prediction approach with label number learning. arXiv preprint arXiv:1907.02205. (2019)
14. Li, S., Zhang, H., Ye, L., Guo, X., Fang, B.: Mann: A multichannel attentive neural network for legal judgment prediction. IEEE Access, 7, 151144-151155. (2019)
15. Koopman, H., Sportiche, D., Stabler, E.: An introduction to syntactic analysis and theory. John Wiley & Sons. (2013).
16. Che, W., Li, Z., Liu, T.: Ltp: A chinese language technology platform. In: Coling 2010: Demonstrations, pp. 13-16. (2010)
17. Li, K. W., Yang, L., Liu, W. Y., Liu, L., LIU, H. T.: Classification method of imbalanced data based on RSBoost. In: Computer Science, 42(9), 249-252. (2015)
18. Xiao, C., Zhong, H., Guo, Z., Tu, C., Liu, Z., Sun, M., ... Xu, J.: Cail2018: A large-scale legal dataset for judgment prediction. arXiv preprint arXiv:1807.02478. (2018)
19. Li, Q., Peng, H., Li, J., Xia, C., Yang, R., Sun, L., ... He, L.: A text classification survey: from shallow to deep learning. arXiv preprint. arXiv:2008.00364. (2020)