A Intelligent Detection Method for Irony and Stereotype Based on Hybird Neural Networks

Notebook for PAN at CLEF 2022

Zexian Yang[1] , Li Ma[1] , Wenyin Yang[1] , Qidi Lao [1] , Zhenlin Tan[1]

[1] *Foshan University, Foshan, China*

Abstract

For the task of Profiling Irony and Stereotype Spreaders on Twitter[1,2], a deep learning model based on a combination of RNN and CNN is proposed in this paper. A special RNN is used to solve the context's long-term dependency, and CNN is used to further extract relational features. The task involves classifying authors as Ironic or non-Ironic based on the number of their tweets, and the task is a judgment for those authors who use irony to spread a stereotype (ISS), that the task does as a binary classification task. After training and predicting on the task datasets given by PAN 22, the accuracy of the model announced by the organizer is about 0.9056.

Keywords

Author Profiling, Irony and Stereotype Spreaders, Bi-LSTM

1. Introduction

Today, with the birth of various new technologies such as big data and cloud computing, the technology of online social platforms is becoming more and more mature. Freely express personal remarks, so that people can express their personal remarks more freely on the online communication platform. Nowadays, because people wantonly publish such inflammatory remarks that are not conducive to the stable development of the country, social stability, and the physical and mental health of others, such remarks will cause serious harm to indiviuuals or the entire society [3]. Therefore, the social platform designs a corresponding algorithm to identify whether the speech sent by the user is excessive, incitement, hatred, or other speech to be restricted [4]. However, people's expressions today are also improving with the advancement of technology. After social platforms have restricted excessive speech, people use language in a metaphorical and subtle way to express the opposite of the literal meaning. That is, the language is ironic and negative. However, this type of language is offensive irony, used to ridicule and despise victims, causing certain psychological trauma to users. Considering the huge amount of daily information on social platforms, it is time-consuming, expensive, and inefficient to manually detect such ironic remarks. Therefore, it is necessary to develop an algorithm that can automatically identify ironic speech [5].

Therefore, the task of Profiling Irony and Stereotype Spreaders on Twitter at PAN 2022 is to verify whether the authors are likely to spread ironic remarks. Based on preprocessing the datasets with a custom function, this paper proposes a Bidirectional Long Short-Term Memory network (Bi-LSTM) and a Convolutional Neural Network (CNN) [6] composition. The spaces in the text are segmented through Textvectorization, and the segmented words are generated one by one corresponding to numerical values, thereby constructing a dictionary. Each word segmented from the training set will be used as a value, and each word is mapped to the value of the key in the dictionary.

[1] CLEF 2022 – Conference and Labs of the Evaluation Forum, September 5–8, 2022, Bologna, Italy
EMAIL: roiv@qq.com (A. 1); molly_917@fosu.edu.cn (A. 2)(*corresponding author); cswyyang@163.com (A. 3)
ORCID: 0000-0001-6060-7603 (A. 1); 0000-0002-5013-052X (A. 2); 0000-0003-4842-9060 (A. 3)

Then the positive integer sequence obtained by the Textvectorization text preprocessing function, and then the keys in the dictionary are mapped to the 120-dimensional word embedding layer. Put the data into the designed model to get the final desired result.

2. Datasets

Profiling Irony and Stereotype Spreaders on Twitter provided a training set and a test set. The data sets are shown in **Table 1**. The datasets are all composed of XML files. Each XML file corresponds to an author, and there are 200 tweets in the XML file corresponding to each author. In the official training data set, there is also a real value file, giving each author the corresponding XML file tag of N or NI.

Table 1
Statistics of datasets

Datasets	Number of texts	Number of author	Number of datas
Training set	420	420	84000
Test set	180	180	36000

3. Irony and Stereotype Evangelist Identification Model Structure

The neural network model proposed in this paper is to realize the discrimination task. The model consists of a textvectorization layer, an embedding layer, a Bi-LSTM layer,a convolutional layer and a fully connected layer. The neural network structure is shown in **Figure 1**.

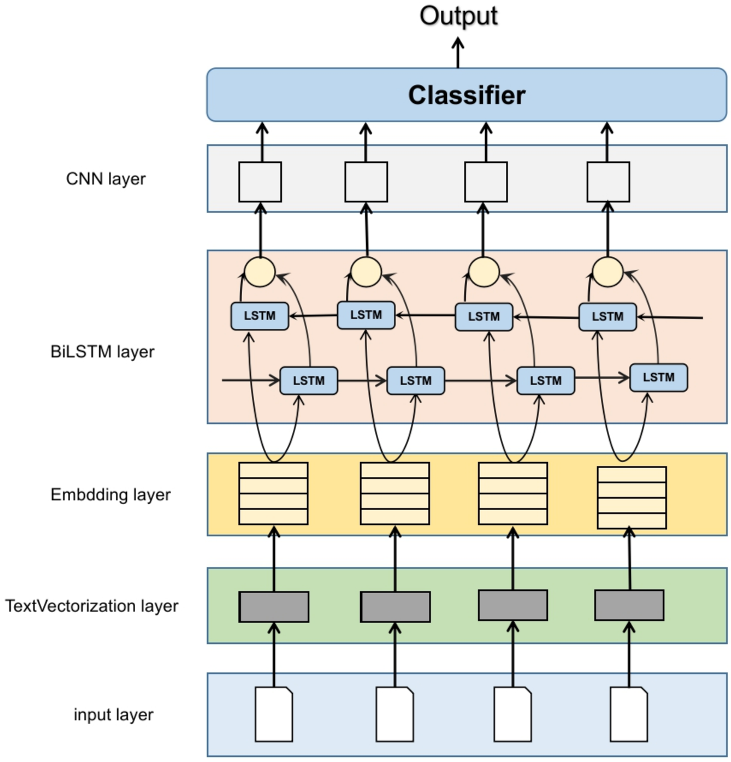

Figure 1: Architecture diagram for model

3.1. Textvectorization Layer

The data after preprocessing is passed into the Textvectorization layer of the model. This layer mainly passes in the processed XML file data, divides the words according to spaces, and maps the words into the required integer sequence. In the dictionary learned by the Textvectorization function, in addition to the learned content, it also includes an empty character as padding (filling if the sentence length is not enough), and Unknown (UNK) represents that the character does not exist. It will be performed using UNK. This layer will further process the data for the word embedding layer mapping.

3.2. Word Embedding Layer

Embedding layer [7] as a dictionary, that is, map integer indices (specific words) into dense vectors, will receive integers as input, look up these integers in the internal dictionary, and return the associated vector. And we will use the tensor input composed of the previous layer of integers to map to a 120-dimensional vector, and use the vector to solve the disadvantage that the integer encoding cannot express the relationship between words.

3.3. Bi-LSTM Layer

A special model in RNN (Recurrent neural network) is called LSTM [8] which is used to solve the context dependence problem in RNN and is suitable for processing time series data. The structure is shown in **Figure 2**.

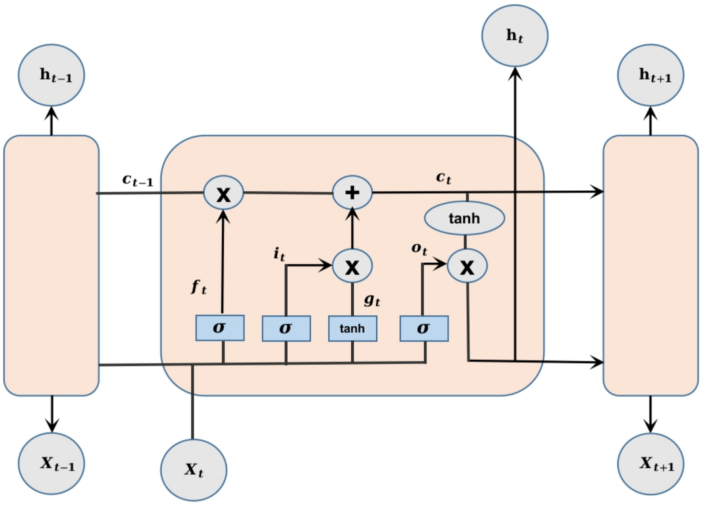

Figure 2: LSTM structure

Since LSTM can only use historical data, it cannot use future data information. Thus, the forward LSTM and the backward LSTM are combined to obtain a new Bi-LSTM structure [9]. Using Bi-LSTM is to insert the same input sequence into the forward and backward two LSTM, and then connect the hidden layers of the two networks together. computable information is improved so that the model can obtain historical and future information. Bi-LSTM [10] includes four parts: memory gate i, forgetting gate f, output gate o, and cell state c. The calculation process of LSTM is:

(A) Choose the forgotten information, enter the word at the current moment x_t through the hidden layer state at the previous moment h_{t-1}, and obtain the value of the forgetting gate f_t. The formula is as follows:

$$f_t = \delta(W_f x_t + u_f h_{t-1} + b_f) \quad (1)$$

(B) By selecting the information to be memorized by inputting the hidden layer state at the previous moment h_{t-1} and the word at the current moment x_t, the value of the memory gate i_t and the temporary cell state g_t are obtained. The formula is as follows:

$$i_t = \delta(W_i x_t + u_i h_{t-1} + b_i) \quad (2)$$

$$g_t = tanh(W_g x_t + u_g h_{t-1} + b_g) \quad (3)$$

(C) By inputting the value of the memory gate i_t, the value of the forgetting gate f_t and the temporary cell state g_t, the cell state at the current moment c_t is obtained. The formula is as follows:

$$c_t = f_t \times c_{t-1} + i_t \times g_t \quad (4)$$

(D) Through the hidden layer state at the previous moment h_{t-1}, the input word at the current moment x_t, and the cell state at the current moment c_t, the output gate o_t and the hidden layer state at the current moment h_t are obtained. The formula is as follows:

$$o_t = \delta(W_o x_t + u_o h_{t-1} + b_o) \quad (5)$$

$$h_t = o_t \times tanh(c_t) \quad (6)$$

(E) Finally, since Bi-LSTM has forward LSTM and reverse LSTM represented by h_n and h_m, respectively, represent the output context hidden layer state vector, and connect and get the output of Bi-LSTM [9] at time t as

$$h_t = o_t \times tanh(c_t) \quad (7)$$

In the formula, W and u represent the weight matrix, and b represents the offset.

3.4. CNN Layer

Because Bi-LSTM can extract the feature relationship of the bidirectional time series dimension of the text, the CNN layer [11] is utilized to further extract the associated features in order to improve semantic analysis on the association between neighboring features. The complexity and quantity of parameters used in neural network model training can be decreased while maintaining the essential characteristics. It can successfully prevent overfitting and enhance the model's capacity for generalization.

3.5. DNN Layer

In the fully connected layer of the last two layers, the first layer uses the nonlinear activation function "Relu" for classification, and the last layer uses a simple linear activation function for the final result classification, and obtains the final two-classification result definition. Positive value is NI and negative value is I.

4. Experiments and Results

4.1. Experimental setting

The word embedding layer included with Keras is used in this study to map words into 120-dimensional vectors. The activity of the model is then increased by adjusting the rate of SpatialDropout1D to 0.2. The Bi-LSTM was designed with 128 units. Relu is used as the activation function in Conv1D along with 64 convolution kernels, a convolution kernel stride of 1, a size of 4, GlobalMaxPooling1D for pooling calculation, and a rate of 0.3 dropout to avoid overfitting. The output unit of the first fully connected layer is 128 and the activation function is Relu. The weight matrix for classification is initialized by a unique kernel initializer in the final fully linked layer. During the training process, set the epoch to 5, and its optimization is Adam.

4.2. Results

The data given by the organizer is divided into 80% for training and 20% for verification, and the trained model is used to verify the model. The training sample uses 5 epochs (E1, E2, E3, E4, E5). The results are shown in **Table 2**.

Table 2
The result of training set

Epoch	Accuracy	Loss	val_accuracy	val_loss
E1	0.6310	0.6393	0.6220	0.6786
E2	0.6518	0.6038	0.8452	0.4877
E3	0.8363	0.4370	0.8571	0.3242
E4	0.9524	0.1464	0.8810	0.3010
E5	0.9851	0.0219	0.8810	0.3124

Task organizers invited participants to deploy their model on TIRA[12]. Through the five epochs, it can be clearly seen that the model is continuously trained, the accuracy is continuously improved, and the loss is reduced. After the accuracy of the validation set reaches the fifth time, the accuracy does not change, and the loss starts to increase. The organizer's test set is used to verify the model, and the accuracy attained is 0.9056.

5. Conclusion

In this paper, we describe the ironic speech task at PAN 22, in which we propose a deep learning-based model to detect Twitter users who spread ironic speech. By fine-tuning the hyperparameters during the training process of our proposed model, the model achieves the best accuracy of 0.9056. As the organizers announced, the model achieved accuracy on the English training set and on the final test set. At the same time, the experiment shows that the task is more challenging. Twitter is more than just text; it also has numerous emojis, which can be used sarcastically, and some people intentionally misspell the text. Errors never go away to avoid machine detection. This type of more complex detection remains a huge challenge, and people must think about it in order to devise better solutions to these problems.

6. Acknowledgments

This work was supported by grants from the Basic and Applied Basic Research Fund of Guangdong Province No.2019A1515111080.

7. References

[1] Bevendorff J, Chulvi B, Fersini E, et al. Overview of PAN 2022: Authorship Verification, Profiling Irony and Stereotype Spreaders, Style Change Detection, and Trigger Detection[C]//European Conference on Information Retrieval. Springer, Cham, 2022: 331-338.

[2] Ortega-Bueno R., Chulvi B., Rangel F., Rosso P. and Fersini E., "Profiling Irony and Stereotype Spreaders on Twitter (IROSTEREO) at PAN 2022," in CLEF 2022 Labs and Workshops, Notebook Papers, CEUR-WS.org, 2022.

[3] Bevendorff J, Chulvi B, Peña Sarracén G L D L, et al. Overview of PAN 2021: Authorship Verification, Profiling Hate Speech Spreaders on Twitter, and Style Change Detection[C]//International Conference of the Cross-Language Evaluation Forum for European Languages. Springer, Cham, 2021: 419-431.

[4] Rangel F, Giachanou A, Ghanem B H H, et al. Overview of the 8th author profiling task at pan 2020: Profiling fake news spreaders on twitter[C]//CEUR Workshop Proceedings. Sun SITE Central Europe, 2020, 2696: 1-18.

[5] Rangel F, Sarracén G, Chulvi B, et al. Profiling hate speech spreaders on twitter task at PAN 2021[C]//CLEF. 2021.

[6] Siino M, Di Nuovo E, Tinnirello I, et al. Detection of hate speech spreaders using convolutional neural networks[C]//CLEF. 2021.

[7] Wang B, Wang A, Chen F, et al. Evaluating word embedding models: Methods and experimental results[J]. APSIPA transactions on signal and information processing, 2019, 8.

[8] Yu Y, Si X, Hu C, et al. A review of recurrent neural networks: LSTM cells and network architectures[J]. Neural computation, 2019, 31(7): 1235-1270.

[9] Liu G, Guo J. Bidirectional LSTM with attention mechanism and convolutional layer for text classification[J]. Neurocomputing, 2019, 337: 325-338.

[10] Zhang Y, Rao Z. n-BiLSTM: BiLSTM with n-gram Features for Text Classification[C]//2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC). IEEE, 2020: 1056-1059.

[11] Yamashita R, Nishio M, Do R K G, et al. Convolutional neural networks: an overview and application in radiology[J]. Insights into imaging, 2018, 9(4): 611-629.

[12] Potthast M, Gollub T, Wiegmann M, et al. TIRA integrated research architecture[M]//Information Retrieval Evaluation in a Changing World. Springer, Cham, 2019: 123-160.