Profiling Irony and Stereotype Spreaders on Twitter Using TF-IDF and Neural Network

Haolong Ma[1], Dingjia Li[1] and Yutong Sun[1*]

[1]Heilongjiang Institute of Technology, Harbin , China

Abstract

In this paper, we describe our participation in the author profiling task at PAN 2022.This task is mainly to profile the irony and stereotype spreaders on Twitter (IROSTEREO).We regard this task as a binary classification problem.Our proposed methods adopt TF-IDF, Bi-GRU and Text CNN models to extract the word frequency statistical features and deep semantic features of text respectively. Based on this series of features, the fully connected network layer is used to complete the classification prediction.Our final submitted system has an accuracy of 93.33% in the test set.This result verified the idea that word frequency statistical features and deep semantic features obtained by neural network jointly predict irony recognition.

Keywords

Irony and Stereotype, Bi-GRU ,Text CNN,TF-IDF, Embedding

1. Introduction

With the rapid development of the Internet, social media such as Facebook, Twitter, and Weibo have emerged in large numbers.While shrinking the communication distance between people, controversial remarks such as immigration or sexism and misogyny frequently appear.It has greatly affected the human rights and security of special groups, and has had a bad impact on the society[1,2].Therefore, identifying possible spreaders of irony and stereotype on Twitter can effectively prevent the large-scale dissemination of these controversial remarks among Twitter online users. Research on how to distinguish authors who have published irony and stereotype remarks in the past from authors who have never made irony and stereotype remarks as far as we know has important implications for regulating the legal compliance of social media information and protecting the purity of online speech dissemination.

The IROSTEREO task announced by PAN@CLEF in 2022 refers to given a Twitter feed in English, determine whether its author spreads irony and stereotypes [3].The data set provided in the task consisted of a set of users who shared some ironic and stereotypical remarks, such as women or the LGTB community.The goal will be to classify authors as ironic or not depending on their number of tweets with ironic content. For the IROSTEREO task, this paper proposed a deep learning method based on the TF-IDF+Bi-GRU+Text CNN ensemble model to extract the n-gram statistical features and deep semantic features in the text,which achieved 93.33% of accuracy in the provided test set.

In Section 2, we present some related work on profiling irony and stereotype spreaders. In Section 3, we mainly describe the method proposed in this paper, including the extracted feature form and the overall model based on deep learning. In Section 4 ,we introduce the specific work in the experiment and the comparison of experimental results. Finally, in Section 5, we present the conclusions and future work.

[1] CLEF 2022 – Conference and Labs of the Evaluation Forum, September 5–8, 2022, Bologna, Italy

EMAIL: email1@mail.com (A. 1); email2@mail.com (A. 2); email3@mail.com (A. 3)
ORCID: XXXX-XXXX-XXXX-XXXX (A. 1); XXXX-XXXX-XXXX-XXXX (A. 2); XXXX-XXXX-XXXX-XXXX (A. 3)

2. Related work

Detection and recognition of hateful and irony speech is a hot topic in natural language processing research in recent years.For example, Ibereveal, PAN@CLEF and other academic activities have successively released their related tasks, attracting the participation of many universities and research institutes around the world. In the ‘Ibereveal 2018 Automatic Misogyny Recognition’ task [4], the method ranking first in accuracy rate uses a combination of multiple statistical features such as style, structure and n-gram vocabulary, and is based on SVM for prediction. In 2021 PAN@CLEF,the shared task [5] is to profiling hate speech Spreaders on Twitter. There are many methods for classification, preprocessing and feature selection.The best performing method is to use an ensemble classifier consisting of five different machine learning models for prediction. Four of them use word n-grams as features, while the fifth one was based on statistical features extracted from the Twitter feeds.The model achieved 75% accuracy in English and 80.5% in Spanish.Through the analysis, it can be found that the current research on the author profiling task usually adopts the traditional discrete statistical model. In terms of feature extraction, the N-gram model based on text vocabulary or word frequency is mainly used to extract features, and then TF-IDF is used to filter the features to capture the feature representation information of the text.However, such machine learning models usually lack context based semantic features.

Recently, based on the shortcomings of the above machine learning methods, deep learning methods have gradually attracted people's attention. For example, Siino et al.[6] used a Convolutional Neural Network (CNN) as a classifier to classify authors as HSS or nHSS, and its prediction accuracy on Spanish was 0.85. In addition, some teams use recurrent neural network (RNN), or BERT pre-training language model has also achieved good results.

3. Model

The model proposed in this paper consists of a total of four components, each of which has its own specific function. Three of these components, TF-IDF, Bi-GRU+Attention, and TextCNN, act as feature extractors in the model, while the fully connected layer is used as a classification prediction.

In the feature extraction, for each component, we extract a specific set of features separately: (i) use TF-IDF to extract feature vector T1; (ii) use Bi-GRU and attention mechanism to learn feature vector o'; (iii) Extract feature vectors V1 and V2 using two Text CNN models of uni-gram and bi-gram. In general, the feature vectors extracted from different components are aggregated using the integration method, and the label prediction results are output based on the softmax operation in the fully connected network layer.

The proposed model is shown in Figure 1.The following sections describe the details of the different components.

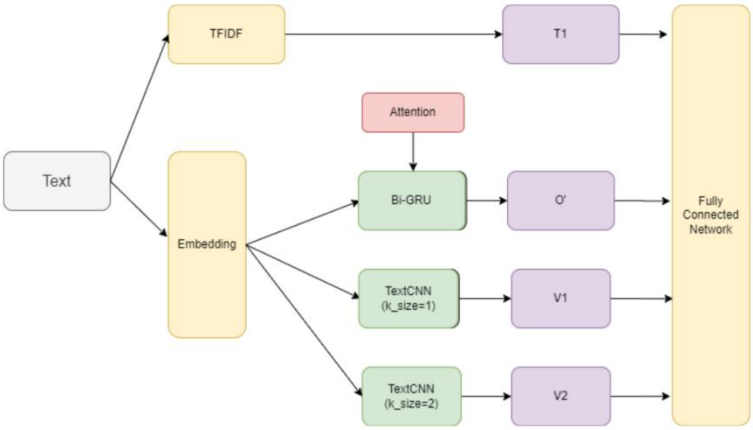

Figure 1: Model architecture based on TF-IDF and neural network

3.1. TF-IDF model

In the experiment, after preprocessing the text data,we directly use the TF-IDF algorithm of sklearn for training to create a set of statistical features based on word frequency for each author. In particular, the output of TF-IDF is subjected to dimension reduction processing through the PCA algorithm, and a feature vector T1 with an dimension of 800 is obtained.

3.2. Bi-GRU+Attention model

For the Embedding information, the Bi-GRU neural network mode [7] is used in the experiment to extract the semantic information of words based on the text context. Moreover, the attention calculation matrices Matrix1 and Matrix2 are used as the attention calculation matrices, and the attention weighting is done for the output of each word, which is finally combined into a vector o. The specific steps are as follows:

- Step 1: Splicing the output of the top layer of Bi-GRU model to get the vector $c = [h1; h'_n, h2; h'_{n-1}, \ldots\ldots, hn; h'_1]$. Its shape is $(n, h * 2)$.
- Step 2: Multiply the vector c with matrix1, and then multiply with matrix2. Finally obtain the weight vector w through the softmax function, the shape of w is $(1, n)$. The calculation method of w is shown in Equation 1.

$$w = SoftMax((c * Matrix1) * Matrix2) \tag{1}$$

- Step 3: Along the direction of n, use w to make hadmard product of c to weight each word, get $o = w \odot c$.The shape of vector o is $(n, h * 2)$.
- Step 4: Along the direction of n, sum the vector o to obtain the vector o' whose size is $h*2$.

The Bi-GRU+Attention Model is shown in Figure 2.

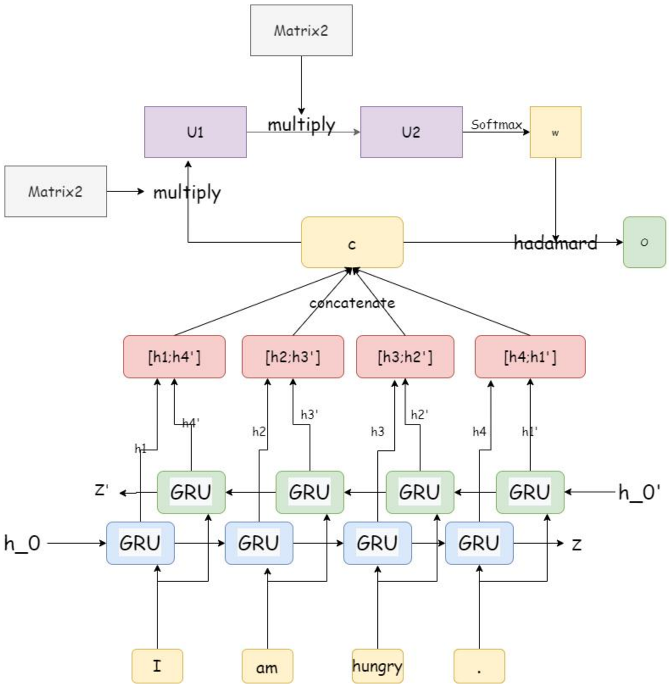

Figure 2: Model architecture based on Bi-GRU+Attention

3.3. Text-CNN model

In this paper, the TextCNN, a convolutional neural network based on text classification, is also used to extract the Embedding information of words. A TextCNN can be used to obtain ngram-like contextual information. 1D TextCNN, which means that the unigram feature is extracted when the convolution kernel is 1, is the bigram feature when the convolution kernel is 2.Taking 1D Text CNN as an example, the steps to extract features are as follows:

- Step 1: Passing the word vector through the 1D convolution kernel to get the output $C = [c_1, c_2, \ldots\ldots, c_m]$,where m is the number of channels.
- Step 2: By K-max-pooling, the top K values in each channel are extracted to get $C' = k - max - pooling(C) = [c'_1, c'_2, \ldots\ldots, c'_m]$, c'_i is a vector of size K.
- Step 3: Flatten all channels to get $V1$, $V1 = flatten(C') = [v_1, v_2, \ldots\ldots, v_{m*k}]$.

Based on the above method, the eigenvector V2 based on bi-gram can be obtained by changing the size of convolution kernel to 2.Splicing V1 and V2 to get the output vector of the Text CNN model.The Text CNN model is shown in Figure 3.

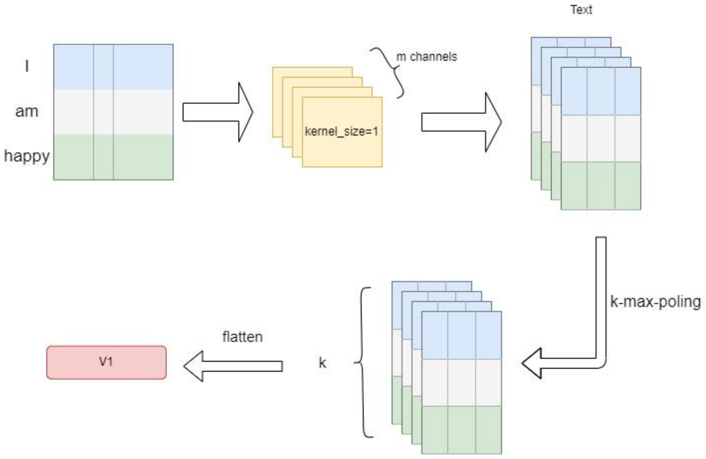

Figure 3: Model architecture based on Text-CNN

3.4. Fully connected layer

The model proposed in this paper uses two fully connected layers.Input the spliced vector T into layer1 to obtain $Layer1(T) = p = [p_1, p_2, \ldots, p_{m/2}]$.

Then passing the output of the first layer through the function LeakyReLu to get $\mathrm{LeakReLu(p)} = \mathrm{p'} = [\mathrm{p'}_1, \mathrm{p'}_2, \ldots, \mathrm{p'}_{\mathrm{m}/2}]$.

Finally, passing the output of the activation function through the Layer2 to get the final output $Leyar2(p') = p'' = [p''_1, p''_2]$.

4. Experiment

The following sections mainly describe the experimental setup and the comparative analysis of the experimental results.

4.1. Data set

The data set used by the IROSTEREO task contains more than 400 XML files and a truth txt. A XML file per author (Twitter user) with 200 tweets. The name of the XML file correspond to the unique author id. The truth.txt file with the list of authors and the ground truth,and the first column corresponds to the author id. The second column contains the truth label.

Additionally, the performance on the IROSTEREO task will be evaluated by accuracy. Accuracy is defined as the ratio of the number of correct predictions to the total number of predictions.

4.2. Preprocessing

The steps for preprocessing are as follows:

- Step 1: First, all tweets of each given author are extracted and stored in the corresponding list auth through regular matching.And extract all the authors' tweets and save the resulting ‘auth’ to a new list $\text{authors} = [\text{auth1}[\text{twitter1}, \ldots, \text{twitter200}], \text{auth2}[\text{twitter1}, \ldots, \text{teitter}] \ldots \ldots]$.
- Step 2: Perform lemmatization and word segmentation on the authors list file. Among them, lemmatization is performed through regular expressions, and each sentence is directly segmented through NLTK to separate words from special symbols such as punctuation, tabs, and expressions.
- Step 3: Build a dictionary and word vector information.Download the Glove word vector (300d) from the Stanford official website, import and load about 400,000 tokens and word vectors.And merge all the texts in the authors, select the 95% tokens with the highest frequency, make a difference set with the 400,000 tokens in the above word vector, and get the token difference set named Rest. Create word vectors for each token in Rest set, and then merge them into the 400000 tokens and word vectors formed above.The specific process of building a dictionary is shown in Figure 4.

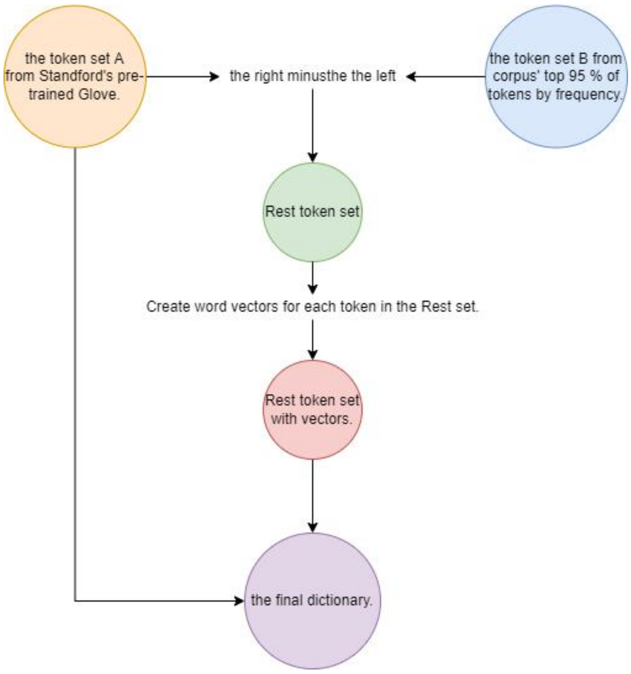
the token set A from Standford's pre-trained Glove.
the right minusthe the left
the token set B from corpus' top 95 % of tokens by frequency.
Rest token set
Create word vectors for each token in the Rest set.
Rest token set with vectors.
the final dictionary.

Figure 4: Dictionary building process

4.3. Experimental parameters

The experiment uses train_test_split function to shuffle the authors list file to obtain train_authors and validation_authors, of which the training set accounts for 70% and the validation set accounts for 30%.Table 4.1 lists the parameters of Bi-GRU, Text CNN models, and other parameters choose default values.

Table 4.1
The model parameters

Model	Parameters
Bi-GRU	Layer:1,Hidden layer size:300
Text CNN	convolution kernel:1 or 2，channel:100，k=5
Training	Optimizer：Adam，Loss Function: Cross Entropy Learning Rate：0.0005，Batch:40

In the experiments, two baseline methods are used to compare the performance of the model proposed in this paper.The two baseline methods are as follows:

1. TF-IDF+SVM: This method adopts the idea of machine learning, and firstly uses the TF-IDF model to extract the statistical features based on word frequency in the text data. Secondly, the extracted statistical feature vector is sent to the LSA model to extract the latent semantic representation features between words in the text.Finally, based on the SVM classifier, the classification prediction is performed.
2. Bi-GRU+Fully Connected Network: This method is based on the idea of deep learning, and the text data is first sent to the Embedding layer for processing to obtain the corresponding word vector $v = [v_1, v_2, \dots, v_n]^{T}$.After the vector v is input to GRU, the output $g = [Z, Z']^{T}$ is obtained.Flatten the vector g to obtain a one-dimensional vector $g' = [g_1, g_2, \dots, g_m]$, where m is $2 * h$ and h is the size of the hidden layer.Finally, the full connection layer is approximately regarded as a classifier. The feature vector g' extracted by Bi-GRU is sent to the full connection layer, and the final output result is obtained through softmax function.

4.4. Experimental results

In the experiment, based on the idea of deep learning, we choose multiple groups of model structures to explore the influence of different feature extraction methods on the experimental results.Table 4.2 reports the results between the proposed method and the baseline method in the training data set. Table 4.3 shows the results of the proposed method on the TIRA platform[9].

Table 4.2
The results of the proposed model and the baseline methods

No.	Model	Accuracy
1	TF-IDF+SVM	0.928
2	Bi-GRU+Fully Connected Network	0.928
3	TF-IDF+Bi-GRU+Fully Connected Network	0.934
4	TF-IDF+Bi-GRU+Text CNN+Fully Connected Network	0.944

Table 4.3
The results of the proposed method in the test set

Model	Accuracy
longma22(Our Team)	0.933

The model 4 proposed in this paper adopts the shuffle separation method in data processing, which can change long text into short text, so as to solve the problem caused by the insufficient memory ability of GRU. In addition, the model adopts the method of multiple shuffle separation and voting, which makes the value of the judgment probability around 0.5 less and the prediction effect more stable.

Overall, the accuracy of the irony recognition algorithm based on Model 4 is 1.6%, 1.6%, and 1% higher than the baseline method, respectively.Model 4 combines the word frequency statistical features and the N-Gram features extracted by Text CNN. Compared with models 2 and 3, model 4 can better obtain various information of the text. It also proves that the traditional word frequency statistical features and the deep semantic features play a common role in the prediction of ironic spreaders.

5. Conclusion

In this paper, we summarize the model submitted through the TIRA system.The model includes three feature extraction components and one classification component.In terms of feature extraction, we use TF-IDF model to extract word frequency features of text, capture semantic features of text based on Bi-GRU, and use Text CNN model to extract uni-gram and bi-gram statistical features,and use these feature vectors to perform classification operations at the fully connected neural network layers. From the results of the training set, the proposed method is significantly better than the baseline method, and the accuracy of the test set is 93.33%.

For future work, we will consider using more features, such as implicit features and non text features, to further improve the accuracy of prediction.

6. Acknowledgments

This work is supported by the the scientific research project (2021QJ08) of Heilongjiang Province.

7. References

[1] Basile V., Bosco C., Fersini E., Nozza D., Patti V., Rangel F., Rosso P. and Sanguinetti M. SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter(2019).

[2] Zhang S., Zhang X., Chan J., Rosso P. Irony Detection via Sentiment-based Transfer Learning. In: Information Processing & Management, pp. 1633-1644(2019).

[3] 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).

[4] Fersini, Rosso P., Anzovino M. Overview of the task on automatic misogyny identification at ibereval 2018. In: IberEval@SEPLN(2018).

[5] Rangel F., Rosso P., Fersini F, et al. Profiling Hate Speech Spreaders on Twitter Task at PAN 2021. In: CLEF 2021 Labs and Workshops, Notebook Papers(2021).

[6] Siino M., Nuovo E. D., Tinnirello I., and Cascia M. L. Detection of hate speech spreaders using convolutional neural networks—Notebook for PAN at CLEF 2021. In: Guglielmo Faggioli et al., editors, CLEF 2021 Labs and Workshops, Notebook Papers(2021).

[7] Chung et al. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling(2014).

[8] Kim Y. Convolutional neural networks for sentence classification[J](2014).

[9] 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.