Use pre-trained models and multi-classifier voting methods to identify the ironic authors on Twitter

Notebook for PAN at CLEF 2022

Jian Qin, Leilei Kong*, Zhaoqian Huang, Jialin Huang, Yansheng Guo, Mingjie Huang, Zeyang Peng

Foshan University, Foshan, China

Abstract

This paper focuses on the task published on PAN at CLEF 2022 of profiling the author's tweets to determine whether the author is ironic. This research is aimed identifying those authors that employ irony to spread stereotypes. In this work, we use a pre-trained model to extract the textual features and train multiple classifiers to make the final decision. We divided the training data of 2022 into 80% as the training dataset and 20% as the validation dataset. The best result for a single classifier on the validation dataset is 0.92857. The best result of multi-classifier voting on the verification dataset is 0.98805. In the final results, the accuracy of our method reached 0.95000. This experiment shows that the multiple-classifier voting method can effectively improve prediction accuracy.

Keywords

Pre-trained model, Multi-classifier voting, Ironic authors on Twitter, Bert

1. Introduction

Nowadays, people's lives are inseparable from the Internet and social media, and satirical language permeates social platforms and everyday speech. Irony and stereotypes often hurt more than general irony. Analyzing the author's tweets, identifying sarcastic sentences from their language and filtering them can help protect victims from discrimination and victimization. Accordingly, it has become one of the staple sharing tasks at PAN [1]. The work presented in this paper was developed as a solution to the Profiling Irony and Stereotype Spreaders task for the competition PAN @ CLEF 2022 [2]. Our task is to determine whether the author spreads Irony and Stereotypes through Twitter in English.

Approaches for irony detection on Twitter can be roughly classified into three classes: rule-based approaches, classical feature-based machine learning methods and deep neural network models. Deep neural network models have recently been applied for irony detection [3, 4, 5, 6, 7, 8] and show better performance than classical feature-based machine learning models.

The profiling Irony and Stereotype Spreaders task belongs to the text classification task. In order to better solve the problem of identifying Irony and Stereotype Spreaders on Twitter, we use the pre-training model such as BERT to encode the text and then make the final decisions by integrating five classifiers. Our model is divided into three parts. The first is an encoder used to encode the input text. The second is a classifier used to classify the text. The third is to use the voting mechanism to make the final decision.`

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

EMAIL:qinjian0516@163.com (A. 1); kongleilei@fosu.edu.cn (A. 2)(*corresponding author); zhaoqian543@163.com (A. 3); sytjl5@163.com (A. 4); guoyansheng2021@163.com (A. 5); mingjiehuang007@163.com (A. 6); pengzeyang008@163.com (A. 7);
ORCID: 0000-0001-5411-1513 (A. 1); 0000-0002-4636-3507 (A. 2); 0000-0002-0623-9050 (A. 3); 0000-0003-4726-951X (A. 4); 0000-0003-2625-9101 (A. 5) ; 0000-0002-0889-5027 (A. 6); 0000-0002-8605-4426 (A. 7);

The article is organized as follows: Section 2 presents the latest relevant work for this task. Section 3 describes our proposed method and introduces our network architecture, Section 4 shows trials and results, and we make a conclusion about this work in the last section.

2. Related works

To address identifying whether authors use irony to spread stereotypes on Twitter, we first start with state-of-the-art text classification [9, 10, 11] techniques and research. Last year, a team consisting of Marco Siino and others won the 2021 PAN competition on profiling HSSs. The team used a deep learning model based on a Convolutional Neural Network (CNN). They used a CNN based on a single convolutional layer to classify authors as Hate Speech Spreader (HSS) or not-Hate Speech Spreader (nHSS) and used 5-fold cross-validation in testing. On that binary classification task, their proposed model achieved a maximum accuracy of 0.80 on a multilingual (i.e. English and Spanish) training set.

However, to date, most research on hate speech in Natural Language Processing (NLP) has focused on detecting hate speech in a single message [12]. A Twitter sharing task team participating in PAN@CLEF2021 proposed a method using contextualized word embeddings and statistical feature extraction to find words used by haters and non-haters in different contexts and compared these words to as features to train a classifier. They also used the BERT sequence representation dataset, using the intermediate sequence representations of all the user's tweets as a feature to train a model to classify users as haters and non-haters.In the last SemEval task for detecting offensive language, the best team reached an F1 score of 0.9204, and the other teams mostly achieved very similar performances in a tight competition [12]. So we considered using this method to explore further the task of identifying stereotype communicators.

3. Our Method

3.1. Network Architecture

We propose a method based on a pre-trained model and a voting mechanism to solve the identification work. Figure 1 shows the network architecture.

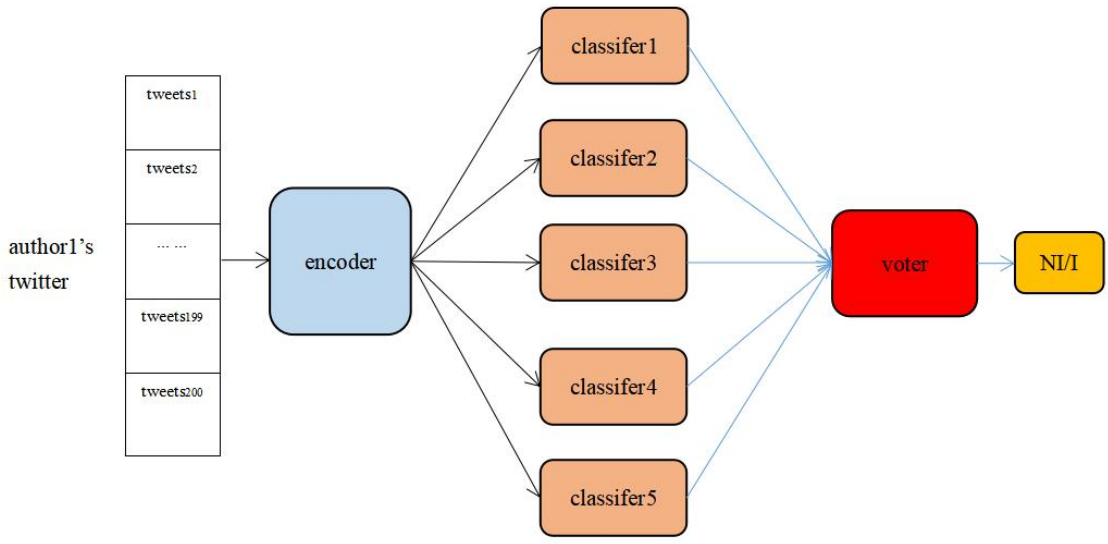

Figure 1: Architecture diagram for our model.

In the training dataset, each piece of data consists of an author, 200 tweets he wrote and a label. Suppose author1's twitter= {tweets$_1$, tweets$_2$, ……, tweets$_{200}$}, where tweets$_1$ is the first tweet of author1 and tweets$_{200}$ is the 200th tweet of author1. Pre-trained model BERT is used as the encoder to extract features of total of 200 tweets. All tweets were individually sent into the model for encoding. During this process, each tweet would be tokenized and sequence padded into a vector of 768 dimensions, which is regarded as the feature of this piece of the tweet. Then these features were fed to a fully connected layer for classification.

3.2. Voting Mechanism

The voting mechanism is divided into two steps. The first step is to determine whether the author of these tweets is ironic according to the distribution of ironic tweets. In the second step, we vote again according to the judgment results of multiple classifiers to obtain the final result.

3.2.1. Step One

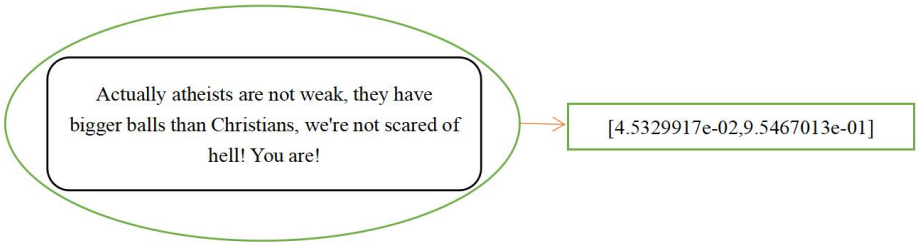

Figure 2: This is an example of getting the predictive value of a tweet. The first value of the array is the score indicating this tweet is not ironic, and the second value is the score indicating the tweet is ironic.

The following is the method for judging the writer' s identity by a single classifier. The output of classification is a 2-dimention vector $\{v_1, v_2\}$ activated with softmax, where $v_1$is the score indicating this tweet is not ironic and v_2 the opposite. When the v_2 score is greater than 0.5, this piece of tweet is considered ironic.

Suppose Y_i is the number of tweets written by the i-th author that are judged to be ironic and Z_i is the average score of all v_2 scores of thses 200 tweets written by the i-th author. And then we set a threshold X to compare with the value of $(Y_i * Z_i)$. If $(Y_i * Z_i)$ is greater than X, the author is considered to be ironic. Otherwise, the author is considered not ironic

3.2.2. Step Two

We use 5-fold cross-validation to improve the accuracy of the model. Specifically, In cross-validation, we divided the 2022 training dataset into five parts equally, four of which were combined into a new training dataset and the other part was used as a validation dataset. According to different combination orders, we can get five different pairs of training set and validation set. These five datasets were used to train the model to obtain five classifiers. These classifiers form the structure of session3.1. The input test dataset was used to obtain five predicted authorship sets. Based on these five result sets, a hard voting method is used to vote on the authors'identity. Suppose there are H_i classifiers to determine the i-th author of the test set is ironic. A threshold K is set and compared with the value of H_i. If H_i is greater than K, the author is considered to be ironic. Otherwise, the author is considered not ironic.

4. Experiments and Results

4.1. Experimental setting

We chose pre-trained model BERT-base (L=12, H=768, A=12, Total Parameters=110M) as the encoder and used Keras to construct BERT and fully connected network. In the fine-tuning pre-trained model phase, we set batch_size=20, maxlen=60, epochs=10 and use sparse categorical cross-entropy as the loss function, and the optimization method is Adam with a 2e-5 learning rate.

The following are the loss function and activation function used during training of our model.

Loss Function： $Q=-\frac{1}{m}\sum_{i=1}^{m}\left(y_i log\left(f(x_i)\right)+(1-y_i) log\left(1-f(x_i)\right)\right)$, *(1)*

Activation Function: $\tanh x=\frac{\sinh x}{\cosh x}=\frac{e^x-e^{-x}}{e^x+e^{-x}}$, *(2)*

4.2. Data processing

Firstly, we replace all emojis with corresponding words by using emojiswitch. The reviewer provided the identity of each author in the training dataset for 2022. We added the labels NI or I after each tweet of the corresponding author according to their identity, where NI idicates the tweet is not ironic and I indicates the tweet is ironic. if the author is considered ironic, the label I was added after each of his tweets. If the author is considered not ironic, add the label NI to each of his tweets. And then the training data in 2022 is divided into 20% and 80% parts, 80% part is used as a training dataset, and 20% part is used as a validation dataset. We obtained five such training dataset and validation dataset pairs according to different arrangement order combinations.

4.3. Thresholds

To get the model we need, we fine-tune the pre-trained model by setting the first threshold *X* in step 1 in the range of 10 to 140. We set epoch=10. A total of 10 rounds are trained, and at the end of each round a validation dataset is used to check the accuracy of the model prediction, and if the accuracy is greater than that of the previous fine-tuned model, the new fine-tuning settings are saved. When 10 rounds of training are completed, let the model predict the validation dataset again and get the final accuracy, recorded as final_val_acc. The accuracy is calculated as the number of correctly predicted authors / total number of authors in the dataset. Table 1 records the final_val_acc obtained for one of the pairs of data sets when the threshold *X* takes different values.

Table 1

The result obtained in a validation dataset when *X* takes different values.

X	final_val_acc	*X*	final_val_acc
10	0.86904	80	0.90476
20	0.86904	90	0.91667
30	0.89286	100	0.86904
40	0.90467	110	0.85719
50	**0.92857**	120	0.76190
60	0.89286	130	0.71428
70	0.90476	140	0.71428

It can be observed that when *X* is set to 50, a better score is obtained on the validation dataset.So we set *X* to 50 and used our five test datasets and valid datasets to get five classifiers.

The threshold K in the second step is used to compare with the number of votes H_i. When H_i is greater than K, i-th author is considered as ironic. A random sample of 20% of the 2022 training dataset is used to test the accuracy of the multi-classifier when K takes different values. The experimental data are shown in Table 2.

Table 2

The accuracy achieved by the multi-classifier in the random 20% test dataset when K takes different values.

	K				
Accuary	1	2	3	4	5
	0.988	0.988	0.964	0.964	0.500

When K is set to 1, 2 or 3, the accuracy rate is high. The data used for testing were randomly selected from the training set, and some of them overlapped with the training data, resulting in high scores. Based on accuracy and fault tolerance considerations, we chose to set K to 2 and then tested the 2022 test dataset.

4.4. result

Table 3
Accuracy achieved on the 2022 test dataset.

2022 test dataset	Accuary
	0.9500

We compressed the test results of the test dataset and uploaded them to TIRA[13]. based on the feedback from the organizers, we achieved an accuracy of 0.9500 for ours.

5. Conclusion

In this experiment, we use the method that utilizes a pre-trained model and voting mechanism to solve the Profiling Irony and Stereotype Spreaders in the PAN@CLEF 2022. The tweets of the authors are added with the corresponding labels and fed into the model one by one to get the model we need for training. Then we build multiple datasets to train and get multiple classifiers for voting to improve accuracy and fault tolerance. Finally, our method achieves an accuracy of 0.9500.

6. Acknowledgement

This research was supported by the Natural Science Foundation of Guangdong Province, China (No. 2022A1515011544).

7. References

[1] J. Bevendorff, B. Chulvi, E. Fersini, A. Heini, M. Kestemont, K. Kredens, M. Mayerl, R. Ortega-Bueno, P. Pezik, M. Potthast, F. Rangel, P. Rosso, E. Stamatatos, B. Stein, M. Wiegmann, M. Wolska, E. Zangerle. Overview of PAN 2022: Authorship Verification, Profiling Irony and Stereotype Spreaders, and Style Change Detection. Vol. 13390. Springer, 2022.

[2] O. Reynier, C. Berta, R. Francisco, R. Paolo, F. Elisabetta. Profiling Irony and Stereotype Spreaders on Twitter (IROSTEREO) at PAN 2022. CEUR-WS.org, 2022.

[3] S. Poria, E. Cambria, D. Hazarika, P. Vij, A deeper look into sarcastic tweets using deep convolutional neural networks, in: Proceedings of the 26th International Conference on Computational Linguistics, 2016, pp. 1601–1612.

[4] A. Joshi, V. Tripathi, K. Patel, P. Bhattacharyya, M. Carman, Are word embedding-based features useful for sarcasm detection?, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016, pp. 1006–1011.

[5] Y.-H. Huang, H.-H. Huang, H.-H. Chen, Irony detection with attentive recurrent neural networks, in: Proceedings of European Conference on Information Re trieval, Springer, 2017, pp. 534–540.

[6] S. Oraby, V. Harrison, A. Misra, E. Riloff, M. Walker, Are you serious?: Rhetor ical questions and sarcasm in social media dialog, in: Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue, 2017, pp. 310–319.

[7] D. Ghosh, A. R. Fabbri, S. Muresan, The role of conversation context for sar casm detection in online interactions, in: Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue, 2017, pp. 186–196.
[8] A. Ghosh, T. Veale, Magnets for sarcasm: Making sarcasm detection timely, con textual and very personal, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 482–491
[9] M. Thangaraj, M. Sivakami, Text classification techniques: A literature review., Interdisci plinary Journal of Information, Knowledge & Management 13 (2018).
[10] B. Altınel, M. C. Ganiz, Semantic text classification: A survey of past and recent advances, Information Processing & Management 54 (2018) 1129–1153.
[11] R. Oshikawa, J. Qian, W. Y. Wang, A survey on natural language processing for fake news detection, arXiv preprint arXiv:1811.00770 (2018).
[12] Tanise Ceron and Camilla Casula. Exploiting Contextualized Word Representations to Profile Haters on Twitter—Notebook for PAN at CLEF 2021. In Guglielmo Faggioli et al., editors, *CLEF 2021 Labs and Workshops, Notebook Papers*, September 2021. CEUR-WS.org. [bib] [copylink] [publisher]
[13] M. Potthast, T. Gollub, M. Wiegmann, and B. Stein. TIRA Integrated Research Architecture. In Nicola Ferro and Carol Peters, editors, Information Retrieval Evaluation in a Changing World, The Information Retrieval Series. Springer, Berlin Heidelberg New York, September 2019.