An Ensemble Machine Learning Classifier for Profiling Irony and Stereotype Spreaders on Twitter

Notebook for PAN at CLEF 2022

Zengyao Li, Zhongyuan Han*, Mingjie Huang, Leilei Kong

Foshan University, Foshan, China

Abstract

The Profiling Irony and Stereotype Spreaders on Twitter (profiling IROSTEREO) task is to judge which author can be considered ironic based on the author's comments. We treat this task as a text binary classification task. This paper proposes a feature extraction method based on a pre-trained language model and a classifier based on an ensemble machine learning model. Our proposed method and model achieve 0.9222 accuracy on the test set for this task.

Keywords

Pre-trained model, Classification, Irony and Stereotype Spreaders.

1. Introduction

With irony, language is employed in a figurative and subtle way to mean the opposite of what is literally stated. In the case of sarcasm, a more aggressive type of irony, the intent is to mock or scorn a victim without excluding the possibility of being hurt [1]. Irony as a literary technique is widely used in online texts such as Twitter tweets.

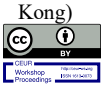
The task of Profiling Irony and Stereotype Spreaders on Twitter (profiling IROSTEREO [1, 2]) is presented in this background. Typically, in author analysis tasks ,such as profiling Hate Speech Spreaders(HSSs) on Twitter task [3], the context representing the positive or negative intent of the text is consistent. However, in the profiling IROSTEREO task, a text's ironic intent is defined by its context incongruity. For example, in the phrase "I love being ignored", irony is defined by the incongruity between the positive word "love" and the negative context of "being ignored" [4]. This task aims to find those authors that can be considered ironic through their tweets on the Twitter author. We use the pre-trained language model BERT [5] for this task to extract the features and propose a classifier model based on an ensemble machine learning model.

The rest of this paper is organized as follows. The related work and methodology are discussed in Section 2 and 3. The experimental setup and results are subsequently reported in Section 4 and eventually concludes with summarize in Section 5.

2. Related Work

The profiling Hate Speech Spreaders on Twitter task [3] proposed by Pan last year is similar to this task. We explore it as a text classification task. A previous method for profiling HSSs, like classifying an author as HSS (Hate Speech Spreader) or not , takes advantage of a CNN based on a single convolutional layer [6]. In addition, hate speech spreader detection using n-grams and voting classifier

CLEF 2022 – Conference and Labs of the Evaluation Forum, September 5–8, 2022, Bologna, Italy
EMAIL: lzy1512192979@gmail.com (Z. Li); hanzhongyuan@gmail.com (Z. Han)(*corresponding author); mingjiehuang007@163.com (M. Huang); kongleilei@fosu.edu.cn (L. Kong)
ORCID: 0000-0001-8472-4150 ;(Z. Li)0000-0001-8960-9872 (Z. Han); 0000-0002-0889-5027 (M. Huang); 0000-0002-4636-3507(L. Kong)



also achieved good results [7]. And it also works well for deep modeling of latent representations based on Transformer [8] model for profiling HSSs task [9].

Furthermore, we will introduce some state-of-the-art methods used in the paper for profiling IROSTEREO tasks. In the paper by Shiwei Zhang, the authors formulate irony detection instead as a transfer learning task where supervised learning on irony labeled text is enriched with knowledge transferred from external sentiment analysis resources. Importantly, they focus on identifying the hidden, implicit incongruity without relying on explicit incongruity expressions [4]. In the ref[10], the authors use two different interpretable methods to identify stereotypes about immigration: Transformer-based deep learning models and text masking techniques .

Presently, pre-trained language models are the mainstream models, and they have been tested to outperform other models in evaluation metrics on most tasks. At the same time, traditional machine learning classifiers are simple and effective for binary classification tasks. In the last year, participants in the profiling HSSs task chose only one of the models to complete the task. So, is it feasible to combine pre-trained languages and traditional machine learning models? In addition to this, we also looked up some related research on text classification. Hybrid methods exist in the literature, such as CNNs for extracting text features and SVMs for performing classification and prediction [6, 11]. Finally, similar results can be obtained with CNN [6]. So, combining pre-trained language models and machine learning classifiers to train and predict data is worth trying.

3. Method

This section gives a brief overview of our model, training process, and prediction process. Our proposed model mainly consists of the following two parts:

1. Fine-tuning Bert for feature extraction
2. Ensemble Machine Learning Classifier

We describe these two parts in Sections 3.1 and 3.2. Sections 3.3 and 3.4 will explain our training and prediction process in detail.

3.1. Fine-tuning Bert

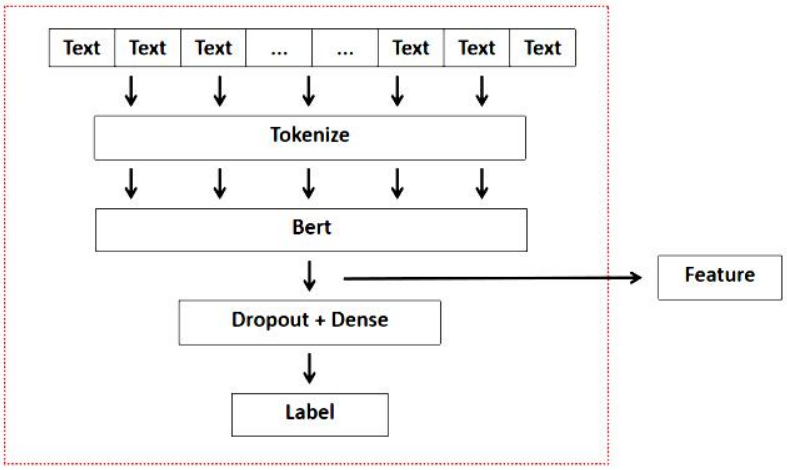


Figure 1: In the structure of the fine-tuned Bert(inside the red frame), we will use the hidden layer vector output from the last layer of Bert as a feature.

The raw text is preprocessed before training and prediction, as detailed below. The preprocessed text will be given the same label as the author of the text, and then fed into the Bert model one by one for fine-tuning. When the accuracy rate on the validation set no longer increased within two epochs, the training was stopped, and the model with the highest accuracy rate was saved. In the feature extraction stage, we can extract 768-dimensional cls tokens through the trained model. The structure of the fine-tune Bert model is shown in Figure 1.

3.2. Ensemble Machine Learning Classifier (EMLC)

We build Ensemble Machine Learning Classifier (EMLC), which integrates LR, RF, and SVM models, and the specific structure is shown in the red box in Figure 2. In Figure 2, x_i represents the input text, and the fine-tuned Bert is used to extract its feature representation and train it as the input of the EMLC. Then the probabilities output by the LR, RF, and SVM models are averaged, and the class with the highest probability is taken as the final prediction result y_i.

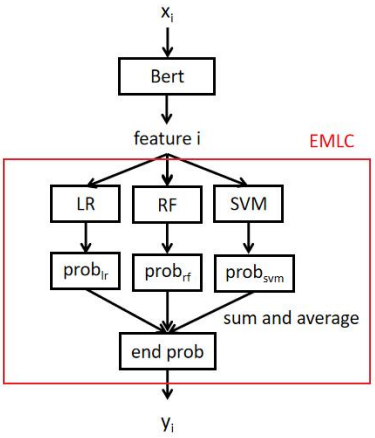


Figure 2: The structure of the EMLC.

The training process of our model is mainly divided into two parts: training of Bert model and training of EMLC. We feed the split 5 data (see section 4 for details of data) into five initial Bert models for training and end up with five fine-tuned Bert models, the first part of training. In the second part of the training process, we use the total training set to input these 5 fine-tuned Bert models for feature extraction, resulting in 5 feature representations. These feature representations are then fed into five EMLCs for training, which finally completes all training of the model. The training process of EMLC is shown in Figure 3. The x_i on the right side of the figure represents the text in the total training set.

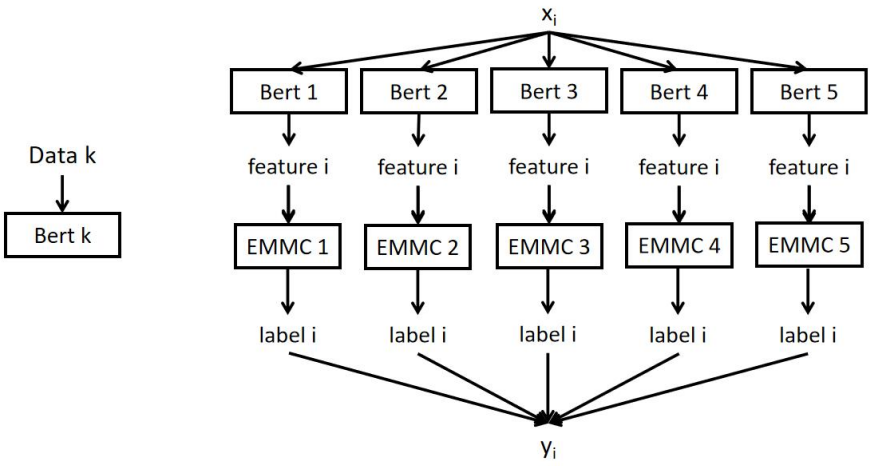


Figure 3: The left side of the figure is the Bert k pre-trained based on Data k. The right side of the figure shows the training process of EMLC. The features of x_i are extracted by the fine-tuned Bert k model and then trained in the corresponding EMLC k (k = 1, 2, 3, 4, 5).

4. Experiment and Results

4.1. Dataset

The English dataset provided by the task organizer consists of two parts: a training set and a test set. The training set consists of 840,000 tweets: the training set has 420 authors, each author assigns a label, and each author has 200 tweets. The test set consists of 360,000 tweets: the test set contains 180 authors with 200 tweets per author. Table 1 shows the data analysis of the dataset.

Table 1
The detail of profiling IROSTEREO datasets

Dataset	Author	Tweets	Labels
Train dataset	420	84000	420
Test dataset	180	36000	None

4.2. Text Preprocessing

For each author's 200 tweets, we first remove some unusual symbols and strings, then convert all text to lowercase, and merge every eight tweets into a new tweet in sequence so that An author gets 25 new tweets. In addition, we also divided the training set into five parts according to the idea of 5fold, named Data1~5 respectively (as shown in Figure 4). Each data is independently trained to obtain an independent Bert model.

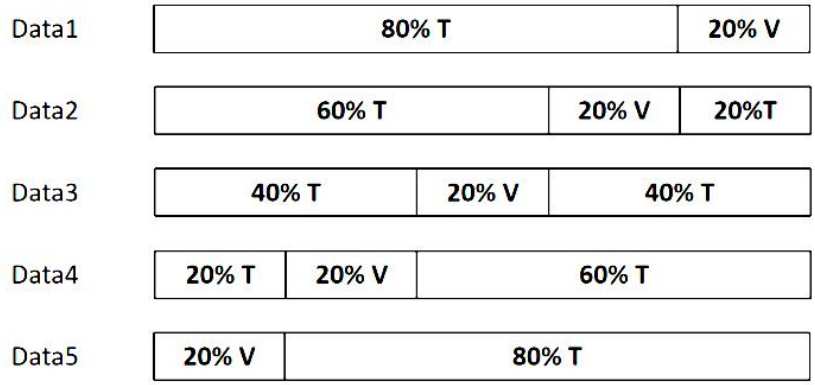


Figure 4: Construction diagram of data1~5. T stands for the training set, and V stands for the validation set.

4.3. Experimental setting

In this work, the pre-trained language model chosen for the first part of our model is $Bert_{base}$ [1](L=12, H=768, A=12, Total Parameters=110M). Specifically, the implementation of HuggingFace[2] called BertForSequenceClassification is used. During the fine-tuning stage of the pre-trained model, we set batch_size=25 and used cross-entropy as Bert's loss function. As the optimizer, we choose AdamW, and the learning rate is set to 1e-5. For the second part of the model, we employ an integrated machine learning classifier of LR, RF, and SVM. For these three machine learning models, we chose to use the default settings and set the "voting" parameter of the VotingClassifier to "soft". We use the PyTorch framework for the whole model to conduct our experiments. Our source code is publicly available at https://github.com/Zero-Lzy/Pan_2022_Twitter.

4.4. Model Prediction Process

[1] https://github.com/google-research/bert
[2] https://huggingface.co/

When making predictions, we first preprocess the text of the dataset, use five fine-tuned Bert models to extract features from the text, and then input the extracted features into the corresponding EMLC. Five EMLCs will get five labels about the text and cast hard votes

[1] on these five labels to get the final label of the text. The acquisition process of text labels is consistent with the training process, as shown on the right side of Figure 3, but at this time, x_i represents the preprocessed text in the test set. And for the task, what we need to predict is the author label. After the dataset is preprocessed, one author will have 25 texts. That is, 25 text labels will be predicted. Based on these 25 text labels, we choose the classification with the most votes as the final author label. The prediction process is shown in Figure 5.

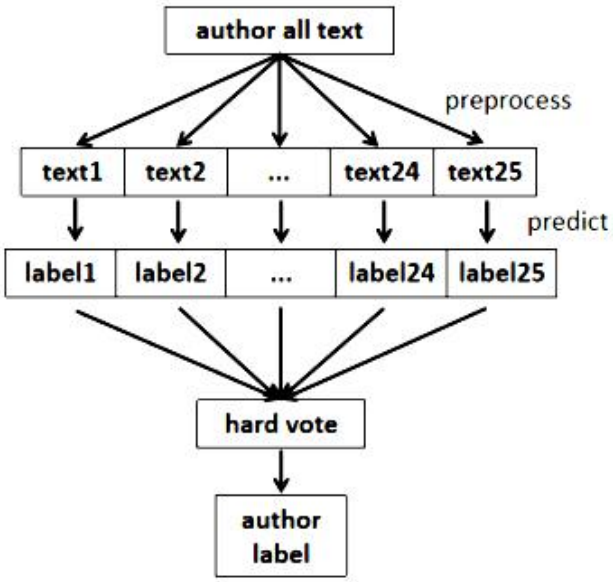


Figure 5: Predict author label based on author's preprocessed tweets.

4.5. Result

We conducted experiments with 5-fold cross-validation on the training set. The dataset was folded five times as described in subsection 4.2. Table 2 reports the accuracy obtained on the validation set used at each fold, along with the arithmetic mean and standard deviation. As can be seen from Table 2, our cross-validation experiments achieved an average accuracy of 0.9976. We believe that our model is relatively reliable.

Table 2

Results were achieved by the model on a 5-fold cross-validation on the complete training set.

	Fold					
Accuracy	1	2	3	4	5	Avg.
	0.988	1.000	1.000	1.000	1.000	0.9976

Table 3

Results achieved by our model on the test set

Rank	Accuracy
40	0.9222

Finally, we compress the results predicted on the test set and upload them to TIRA [12]. As reported on the PAN website, in the test set given by the organizer, our evaluation metrics accuracy can reach 0.9222 as shown in Table 3. The accuracy differs from the result on the validation set by 0.0754. It may be because the validation set is too small or the correlation of the split data is high,

[1] minority obeys the principle of majority. For example, three labels out of 5 labels are 0, 2 labels are 1, and The final label is 0

resulting in the accuracy of the validation set being too high. At the same time, there may be some unsolvable overfitting problems in the model.

5. Conclusion

To address the profiling IROSTEREO task proposed by PAN2022, we offer a feature extraction method based on pre-trained language models and a classifier based on ensemble machine learning models in this paper. At the same time, to solve the problem of data underfitting, we constructed multiple datasets for multiple training and voting. To solve the problem of data overfitting, we use early stopping. In the end, we reached 90% on the accuracy score. Therefore, our proposed method is still effective for this task.

6. Acknowledgments

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

7. References

[1] 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.

[2] J. Bevendorff, B. Chulvi, E. Fersini, et al. Overview of PAN 2022: Authorship Verification, Profiling Irony and Stereotype Spreaders, and Style Change Detection. In: Proceedings of the Thirteenth International Conference of the CLEF Association (CLEF 2022), volume 13390 of Lecture Notes in Computer Science. Springer, 2022.

[3] F. Rangel, G. L. D. L. P. Sarracén, B. Chulvi, E. Fersini, P. Rosso, Profiling Hate Speech Spreaders on Twitter Task at PAN 2021, in: A. J. M. M. F. P. Guglielmo Faggioli, Nicola Ferro (Ed.), CLEF 2021 Labs and Workshops, Notebook Papers, CEUR-WS.org, 2021.

[4] Zhang S., Zhang X., Chan J., Rosso P. Irony Detection via Sentiment-based Transfer Learning. In: Information Processing & Management, vol. 56, issue 5, pp. 1633-1644, 2019.

[5] Devlin J., Chang M.W., Lee K., et al. Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 4171-4186, 2019.

[6] M. Siino, E. D. Nuovo, I. Tinnirello, M. L. Cascia, Detection of Hate Speech Spreaders using Convolutional Neural Networks, in: G. Faggioli, N. Ferro, A. Joly, M. Maistro, F. Piroi (Eds.), CLEF 2021 Labs and Workshops, Notebook Papers, CEUR-WS.org, 2021.

[7] F. Balouchzahi, S. H. L., G. Sidorov, HSSD: Hate Speech Spreader Detection using N-grams and Voting Classifier, in: G. Faggioli, N. Ferro, A. Joly, M. Maistro, F. Piroi (Eds.), CLEF 2021 Labs and Workshops, Notebook Papers, CEUR-WS.org, 2021.

[8] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. In: 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, pp. 6000–6010, 2017.

[9] R. L. Tamayo, D. Castro-Castro, R. O. Bueno, Deep Modeling of Latent Representations for Twitter Profiles on Hate Speech Spreaders Identification. Notebook for PAN at CLEF 2021, in: G. Faggioli, N. Ferro, A. Joly, M. Maistro, F. Piroi (Eds.), CLEF 2021 Labs and Workshops, Notebook Papers, CEUR-WS.org, 2021.

[10] Sánchez-Junquera J., Rosso P., Montes-y-Gómez M., Chulvi B. Masking and BERT-based Models for Stereotype Identification. In: Procesamiento del Lenguaje Natural (SEPLN), num. 67, pp. 83-94, 2021.

[11] Z. Wang, Z. Qu, Research on web text classification algorithm based on improved cnn and svm, in: 2017 IEEE 17th International Conference on Communication Technology (ICCT), IEEE, pp. 1958–1961, 2017.

[12] 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.