BERT-based ironic authors profiling

Notebook for PAN at CLEF 2022

Wentao Yu[1], Benedikt Boenninghoff[1] and Dorothea Kolossa[1]

[1]*Institute of Communication Acoustics, Ruhr University Bochum, Germany*

Abstract

The present paper addresses the PAN at CLEF 2022 challenge "Profiling Irony and Stereotype Spreaders on Twitter" (IROSTEREO). The challenge strives to identify whether an author spreads sarcasm through their tweets. In general, author profiling tasks, whether mechanical or manual, are based on extensive author text. Many machine learning-based author profiling studies indicate that author-by-author classification could benefit the performance compared with a text-by-text way.

We address the challenge through fine-tuning BERT model. BERT has shown satisfactory results on many natural language processing tasks. However, BERT model cannot exert its advantages in some specific tasks, like handling long documents, due to its limitation on the maximum input token length. Our author profiling task is one of these specific tasks. The present work addresses this dilemma through a re-segmentation approach: We first concatenate all tweets of an author into one document representation. We then split the document in such a way that the split text lengths do not exceed the maximum input token length of the BERT model and that we still retain the advantages of continuous text through an appropriate choice of overlap. Ultimately, the BERT model uses the hard voting method made the final decision.

Our work first compares the performance of two pre-trained BERT models, i.e., the RoBERTa and BERTweet models, trained with external datasets. Then, we fine-tune BERT models with three different loss functions. In addition, we also demonstrate and evaluate a BERT feature-based CNN model. The winning models of the PAN author profiling task in recent years are re-implemented as baselines. Finally, the BERTweet model trained with the cross-entropy weighted focal loss function achieves an accuracy of 98.89% on the official test set. Adding a further soft voting ensemble method, which integrates BERTweet models with different loss functions as well as the BERT feature-based CNN model, we placed first in the challenge and improved our model performance to 99.44%. [1]

Keywords
BERT, Long text, Irony detection

1. Introduction

Despite great effort being exerted by researchers and developers to detect and filter toxic language, the amount and impact of hate speech on social media are still posing serious threats to the mental health and well-being of users as well as to the possibility of democratic discourse

on such platforms. According to the latest statistics [1], Twitter receives more than 5 million tweets per day. Hate speech is not in the minority in these tweets. Although Twitter has enacted a hateful conduct policy[1], hate speech is still rampant. As an implicit way to express hatred, irony makes detection more challenging. In this work, we address the PAN at CLEF 2022 challenge: profiling the irony spreaders on Twitter. The challenge provides a dataset containing authors with a set of their tweets to identify whether the author spreads irony within their tweets.

The previous PAN author profiling tasks reveal that combining all documents from an author into one document representation and using it for author profiling has better performance than tweet-by-tweet classification. It is easy to figure out that not every tweet from an irony spreader may be ironic. Also, tweet-wise author profiling may cause much noise for classification. In accordance with this reasoning, in the PAN 2021 challenge for the detection of hate speech spreaders, the best performance was achieved by training a convolutional neural network (CNN) on the combined author document representations [2].

The Bidirectional Encoder Representations from Transformers (BERT) model [3] was proposed in 2018 and is widely used in many natural language processing (NLP) tasks due to its outstanding performance. Training a BERT model from scratch requires large amounts of training data, and the number of parameters of the BERT model is also considerable. Therefore, it requires specific hardware, and the training process is also time-consuming. Transfer learning [4] overcomes these disadvantages and makes the Transformer model more attractive compared with other approaches. However, the BERT model has a maximum token length limitation. For our task, the combined document representation could exceed this limitation. Thus, this work improves the BERT model performance with long documents by adopting a re-segmentation strategy, i.e., the combined document representation is split with overlap to fit BERT's length limit.

The remainder of the paper is structured as follows: To establish the context of this work, Section 2 looks back at the author profiling tasks of PAN and their winning models in the last decade. Section 3 describes the dataset of this year's PAN challenge and the employed text preprocessing strategies. All models are detailed in Section 4, with their training setup introduced in Section 5. Section 6 shows and analyzes the training results of each model. Finally, Section 7 summarizes the strategies, conclusion and experience of this work.

2. Related Works

The PAN organizing committee launched a series of author profiling tasks during the past decade. Until 2021, all challenges were multilingual tasks. Table 1 gives an overview of all PAN author profiling challenges.

The objective of the Author Profiling Tasks from PAN 2013 to 2016 was to identify authors' gender and age group through their documents [5, 6, 7, 8]. In 2015, in addition to identifying gender and age, participants also needed to score five personality traits [7]. The 2017 PAN challenge focused not only on authors' gender detection but also on the language variety identification[9]. The task in 2018 was still to detect gender, but in a multi-modal way. The organizer provided text and image data for the model training [10]. With the rapid rise in social

---

Table 1
PAN author profiling task timeline.

2013	multilingual, predicting authors' age and gender
2014	multilingual, predicting authors' age and gender
2015	multilingual, predicting authors' age and gender, scoring five personalities
2016	multilingual, predicting authors' age and gender
2017	multilingual, predicting authors' gender and language variety
2018	multilingual, multi-modal, identification authors' gender
2019	multilingual, bot detection, human authors' gender identification
2020	multilingual, fake news spreader identification
2021	multilingual, hate speech spreader identification
2022	monolingual, irony spreader identification

network users, people have also become aware of the problems that arise in such virtual spaces. Hence, the PAN challenge author profiling task has a new focus on Twitter from 2019. The goal in 2019 was bot detection, meanwhile identifying the gender of the human authors [11]. The tasks for 2020 and 2021 were Twitter fake news and hate speech spreader identification, respectively [12, 13].

The best result papers in the past show the development of author profiling over the last few years. Most contestants achieved the best results using conventional classifiers like support vector machines (SVMs), decision trees, Expectation Maximization Clustering (EMC), and LibLINEAR [14, 15, 16, 17, 18, 19, 20, 21]. Among these, SVM combined with n-gram features is the most frequently used. Before 2018, researchers mainly discussed text preprocessing and feature extraction with the conventional classifiers. Since 2018, some new deep learning algorithms have gained advantages–the winning group in 2018 adopted representation fusion: text and image features are extracted by RNN and CNN, respectively [22]. Features are fused by using direct-product, column- and row-wise pooling. The fused representation of the texts and images is fed to the fully connected layers for classification. In 2021, deep learning methods still outperformed conventional classifiers. The optimal model utilizes CNN for classification; a self-trained embedding layer extracts the features [2] (detailed in Section 4.1).

Since the transformer model was proposed in 2017 [23], attention mechanisms have attracted much attention and discussion. In 2018, the BERT model, proposed by the Google team, has achieved remarkable results in many natural language processing (NLP) tasks. Due to the attractiveness of the BERT model, more and more teams choose the BERT model to handle author profiling tasks. Only one team used the BERT model in 2019 [24], in 2020, there were three teams [25, 26, 27], and most recently, the BERT model was widely used in the 2021 PAN author profiling task.

The author profiling tasks are usually based on many documents of that author. It is a sensible and effective strategy to profile by combining all of the author's manuscripts, as can be confirmed in some previous winning models [2]. Although the BERT model is attractive, it has a bound on the maximum input token sequence length, thus limiting the ability of the BERT model to handle long texts. To overcome this drawback, some researchers have proposed a document re-segmentation strategy, dividing long documents into sub-documents that match the maximum sequence length of BERT [28, 29]. Last year, one team achieved the best accuracy

for the English author profiling task by fine-tuning the BERT model with a similar strategy [30], concatenating 20 tweets of each author into one sample.

3. Dataset

This year's PAN challenge [31] author profiling subtask is a monolingual task that aims to identify English-language sarcasm spreaders on Twitter [32]. The balanced training set contains 420 author samples, each of which has 200 tweets by this author. The official test set has 180 author samples, again with 200 tweets each. Tagged users, hashtags, and URLs are already normalized as "#USER#", "#HASHTAG#" and "#URL#", respectively. To train the model and test its effectiveness, a 4-fold cross-validation is adopted during the training stage. Thus, the official training set is split into an inner training set and a test set in each fold, with 315 and 105 author samples, respectively.

This work experiments with two BERT models. The first is the `twitter-roberta-base -irony` [33] [2], and the other is `bertweet-large` [34] [3]. Two distinct text preprocessing schemes are adopted for these two BERT models. In addition, there is one scheme for the CNN model and the TF-IDF (term frequency-inverse document frequency) features. The TF-IDF features are used to train conventional classification models like SVMs.

- **Scheme 1**: for the `twitter-roberta-base-irony` model
    - remove "#USER#", "#HASHTAG#" and "#URL#"
    - replace multiple spaces by one single space
    - convert emojis to text with the Python `emoji` [4] package
    - normalize all text into lowercase
    - remove punctuation and numbers

- **Scheme 2**: for the `bertweet-large` model. The BERTweet model has an embedded text normalization. Therefore, we only change the text to fit the BERTweet text style, then process the input text with BERTweet's text normalization.
    - replace "#USER#" with "@USER"
    - replace "#HASHTAG#" with "#HASHTA"
    - replace "#URL#" with "HTTPURL"
    - text normalization with the embedded normalization processes

- **Scheme 3**: for the CNN model and the conventional classification approaches like SVMs, linear regression (LR), and random forest (RF) classifiers.
    - remove "#USER#", "#HASHTAG#" and "#URL#"
    - replace multiple spaces by a single space
    - convert emojis to text with the Python `emoji` package
    - normalize all text into lowercase

---

[2]https://github.com/cardiffnlp/tweeteval
[3]https://github.com/VinAIResearch/BERTweet
[4]https://github.com/carpedm20/emoji

- remove punctuation and numbers
- remove stop words

The TF-IDF features are obtained by word-based 1- to 3-gram and character-based 3- to 5-gram models. Then the truncated singular value decomposition (SVD) [35] [5] reduces the feature dimension to 1000 to reduce the computational complexity. Specifically, the TF-IDF features are extracted by the `TfidfVectorizer` function from the scikit-learn library [6]. The minimum document frequency is 2, and the maximum is set to 100%, i.e., the terms occurring in all documents or in less than two documents are ignored. The word-based and character-based models are obtained separately, each producing a 1000-dimensional vector of every input tweet. Finally, the two vectors are concatenated as a representation of the tweet.

As introduced in Section 2, the BERT model is limited in handling long documents. However, profiling authors based on long manuscripts can benefit the accuracy. To address this conflict, a re-segmentation strategy is adopted to make the sequence length of the sub-document fit the maximum input token length of the BERT model. The continuity of the segmented sentences is guaranteed by overlapping segmentation. The specific steps are as follows:

- concatenate all 200 tweets of each author
- the new sub-document is segmented with the same text length N and with an overlap of O. To simplify the program, only the number of words in the text is considered here instead of the number of tokens. Therefore, N should be smaller than the maximum token length of the BERT model to guarantee that there are no text segments that are too long for BERT.
- the author's label is assigned to every sub-document of that author

Two external irony detection datasets were utilized to pre-train both BERT models for better performance. One is the Ironic Corpus [36], the other is the SemEval-2018 irony detection dataset [37]. The datasets were labeled on each document. Only 0.168% document exceeds the BERT maximum token length limitation. Therefore, we have not applied the sub-segmentation strategy for these two corpora.

4. Models

Besides two BERT-based models, we also consider a CNN model that builds on the BERT embeddings. In addition, three traditional classifiers are also trained as baselines.

4.1. Baselines

The models below are used as baselines. Among those, you can also find the winning models of the PAN author profiling task in previous years, in our respective re-implementations.

- *SVM, LR, and RF* models: All tweets of an author are concatenated and preprocessed according to **Scheme 3**. The TF-IDF features are extracted from the processed document. Finally, the scikit-learn library is utilized to train the models.

---

[5] https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html
[6] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

- *CNN*: We re-implemented the winning model of the PAN 2021 author profiling task [2]. Again, all the author's tweets are concatenated, and **Scheme 3** is adopted for text normalization. The model structure is the same as in [2]. The embedding layer projects each input token into a 100-dimensional vector. A 1D-convolution layer with 64 filters of size 36 was applied to the embedding tensors. Then an average pooling with a size of 8 reduces the features' complexity, and the global average pooling decreases the dimension of the features. Finally, a fully connected layer outputs the results in the desired size.
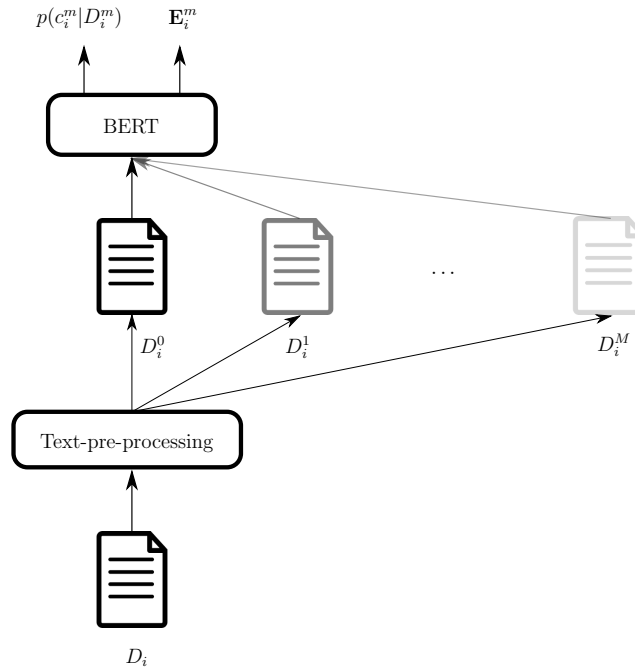
4.2. Fine-tuning BERT models

Figure 1: Fine-tuning a BERT model

We initially chose the RoBERTa model (`twitter-roberta-base-irony`) [33] and pre-trained it on the SemEval2018 irony detection database [37]. However, the leaderboard of TweetEval on GitHub [7] indicates that the BERTweet model (`bertweet-large`) [34] outperforms other candidate models for the sarcasm identification task. Therefore, the BERTweet model is also considered. Both base models are first trained with external datasets, and these pre-trained models are marked as *RoBERTa-ext* and *BERTweet-ext*. Subsequently, these two pre-trained models are fine-tuned with the PAN challenge dataset to obtain the final models *RoBERTa* and *BERTweet*. Figure 1 depicts an overview of the fine-tuning process for the BERT model. $D_i^m, m \in M$ is the mth sub-document of author i; c_i^m is the predicted class, in our case $c_i^m \in [0, 1]$. The input text D_i is processed using the above text preprocessing and

---

[7]https://github.com/cardiffnlp/tweeteval

re-segmentation strategy. The BERT model takes one sub-document at a time and gives two outputs. One is the probability predicted based on the current sub-document $p(c_i^m|D_i^m)$, and the other is the corresponding word-embeddings $\mathbf{E}_i^m$.
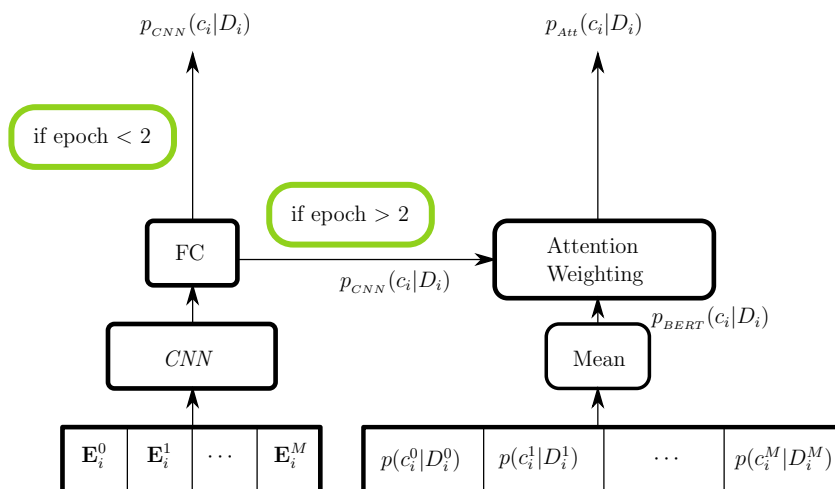
4.3. Feature-based approach

Figure 2: *BERTweet-CONV* model, with FC denoting a fully connected layer.

As stated in [3], there are many benefits to training a model using BERT embeddings as fixed features. On the one hand, the BERT model structure cannot suit all tasks, rather, sometimes it needs to add some task-specific design to increase flexibility. On the other hand, feature-based methods can speed up the computation because the text representation only needs to be computed once. This paper adopts the CNN layer for classification (*BERTweet-CONV*). The word embeddings of all sub-documents of an author are concatenated to train the CNN model, where the embeddings are extracted from the pre-trained *BERTweet* model. The CNN model structure here is principally the same as in the *CNN* described in Section 4.1, the only difference is the input dimension. The Bi-LSTM layer is not considered because the word-embeddings of all sub-documents are concatenated. The concatenated word-embeddings could be regarded as a word-embedding of a long document, and its token length could be as long as 4000. For such a lengthy document, BLSTM layers could face the problem of vanishing gradients and exploding gradients.

Figure 2 illustrates the details of the *BERTweet-CONV* model. On the left, the *CNN* model based on the *BERTweet* embedding is shown. The concatenated embedding is the input of the CNN model. On the right side, the predicted probabilities of the *BERTweet* model for each sub-document of that author are utilized. The average of the probabilities of each sub-document $p_{BERT}(c_i|D_i)$ and the probabilities predicted by the CNN model $p_{CNN}(c_i|D_i)$ are input to an attention weighting block. The probability $p_{Att}(c_i|D_i)$ on the right is obtained as follows:

$$p_{Att}(c_i|D_i) = w_1 \cdot p_{BERT}(c_i|D_i) + w_2 \cdot p_{CNN}(c_i|D_i), \tag{1}$$

where

$$\boldsymbol{w} = \text{softmax}(\text{FC}(p_{BERT}(c_i|D_i); p_{CNN}(c_i|D_i))). \tag{2}$$

During the first two epochs, only the CNN model parameters are trained because in comparison with the CNN, the initial *BERTweet* model predictions are too accurate early in the training, which could dominate the entire model. After two epochs, the final probability is predicted using attention-weighting for both model type outputs.

5. Experimental Setup

Participants can access only the training set at the beginning. For each DNN model, our work first uses the Python RAY [8] package to find the best hyperparameters. The training set has a total of 420 author samples. To find the best set of hyperparameters, 100 authors were randomly selected as the internal test set and the remaining 320 authors were used for the internal training set. When the best set of hyperparameters is found, 4-fold cross-validation is applied to test the robustness of the model. For this purpose, 105 authors are used as the internal test set in each fold. We used the scikit-learn library `GridSearchCV` package to find the optimal hyperparameters for the SVM and LR models, while the `RandomizedSearchCV` package is applied for the RF model. Table 2 lists the optimal hyperparameters of different DNN models.

Table 2
Optimal hyperparameters of different mdoels

	lr	Batch size	Epochs	Optimizer	Scheduler
CNN	0.001	4	25	Adam	ReduceLROnPlateau
RoBERTa-ext	5e-5	16	1	AdamW	get_linear_schedule_with_warmup
BERTweet-ext	1e-5	16	3	AdamW	get_linear_schedule_with_warmup
RoBERTa	1e-5	16	3	AdamW	get_linear_schedule_with_warmup
BERTweet	1e-5	4	3	AdamW	get_linear_schedule_with_warmup
BERTweet-CONV	2e-2	4	4	Adam	ReduceLROnPlateau

As previously described, we use the re-segmentation strategy to improve the performance of the BERT model. However, the re-segmentation leads to a problem: the originally balanced dataset is now imbalanced. The cross-entropy (CE) loss function for binary classification

$$\text{CE} = -\log(p_k) \tag{3}$$

generally does not perform well on imbalanced data [38]. In contrast to the cross-entropy, the focal loss function [39] for binary classification

$$\text{F} = -(1 - p_k)^{\gamma}\log(p_k) \tag{4}$$

assigns a larger weight to poorly estimated training samples.

---

[8]https://github.com/ray-project/ray

This work adopts the Cross-Entropy Weighted Focal (CEWF) loss function [38] to deal with this problem during the model training. It is defined as

$$\text{CEWF} = -\frac{e^{(1-p_k)t} + e^{p_k t}(1 - p_k)^{\gamma}}{e^{p_k t} + e^{(1-p_k)t}} \log(p_k), \tag{5}$$

where p_k is the estimated target probability. The CEWF is a compromise between the CE and focal loss functions. When the classifier is very confident about the classification probability, the CEWF is close to CE; otherwise, the CEWF is close to the focal loss function. In our work, we set $t = 4$ and $\gamma = 5$ in Equation 5.

The *RoBERTa* and *BERTweet* models make predictions c_i^m for each segment D_i^m because of the re-segmentation strategy. The prediction of one author's class c_i is obtained by applying hard voting (HV) over all of the sub-segments of this author:

$$c_i = \text{mode}(c_i^0, c_i^1 \ldots c_i^M). \tag{6}$$

To implement the ensemble approach, we also use the soft voting method (SV) to obtain the class probability of each author:

$$p(c_i|D_i) = \text{mean}(p(c_i^0|D_i^0), p(c_i^1|D_i^1) \ldots p(c_i^M|D_i^M)). \tag{7}$$

All our models are trained by the PyTorch library [9]. Early stopping prevents overfitting. Specifically, the training is terminated if the evaluation accuracy does not improve within four epochs. The AdamW optimizer optimizes the parameters of the BERT model; the `get_linear_schedule_with_warmup` from the transformer model is used to schedule the learning rate. The learning rate has a warm-up process in the first four epochs, its upper limit is given in Table 2. Other models are optimized by Adam; the learning rate is scheduled by `ReduceLROnPlateau`, i.e., the learning rate is reduced by 50% if the evaluation accuracy does not improve. For the BERT fine-tuning models, the segment length N is 500 with an overlap of $O = 128$. Since the *BERT-CONV* model is based on the trained *BERTweet* model, the order of the author's tweets is shuffled to avoid overfitting during the *BERT-CONV* model training phase; the segment length N is still 500, while the overlap length O is 64. The output dimension in all models is 2 with a softmax output function. The training process is carried out on NVIDIA's Volta-based DGX-1 multi-GPU system, using 2 TeslaV100 GPUs with 32 GB memory each.

6. Results

Our experimental results are presented in this section. Table 3 below lists the results of the two selected BERT models, which are trained on the external training set. Our training results are similar to those listed on TweetEval's leaderboard. The *BERTweet-ext* model clearly outperforms the *RoBERTa-ext* model on sarcasm detection.

Table 4 lists the experimental results of the previously described models on the PAN database. The final prediction of an author is obtained by hard voting. All models are trained under the same 4-fold cross-validation, i.e., each model's internal training and test sets are identical in each

---

[9]https://github.com/pytorch/pytorch

Table 3
The accuracy of the *BERTweet-ext* and *RoBERTa-ext* models, trained on external datasets with Cross Entropy (CE) as the loss function.

	Loss	Acc.
RoBERTa-ext + CE	0.599	0.670
BERTweet-ext + CE	**0.409**	**0.872**

fold. The *CNN* model achieves better results than the *RoBERTa* model. The advantage of the *CNN* model is that its training time is much faster than fine-tuning a BERT model. Comparing the two BERT models, the *BERTweet* model is again much more effective. Under the same loss function, the *BERTweet* + CE model performs a relative error rate reduction by 7.091% compared to the *RoBERTa* + CE model on average. One possible reason is that the *BERTweet* model was pre-trained with a vast amount of Twitter data, and the PAN database was also collected from Twitter.

Having established this, we focus on the BERTweet model with different loss functions. However, the results of the three loss functions are not significantly different. The best average accuracy is the *BERTweet* model with the CEWF loss function. Due to the outstanding performance of the *BERTweet* + CEWF model, the feature-based *BERTweet-CONV* model utilizes the *BERTweet* + CEWF to extract the embeddings as the fixed features. However, the feature-based strategy did not improve model accuracy on average. This conclusion is also in line with what was claimed in [3], although, interestingly, the *BERTweet-CONV* model achieves the best accuracy in one of the folds.

Table 4
Accuracy comparison between different models and setups.

	0	1	2	3	mean±std
CNN	0.933	0.895	0.924	0.895	0.912±0.020
RoBERTa + CE	0.905	0.933	0.895	0.829	0.891±0.044
BERTweet + CE (1)	0.952	0.981	0.971	**0.933**	0.959±0.021
BERTweet + F (2)	**0.971**	**0.981**	0.962	0.924	0.960±0.025
BERTweet + CEWF (3)	0.962	0.971	0.990	**0.933**	0.964±0.024
BERTweet-CONV + CE (4)	0.952	0.962	**1.000**	0.923	0.959±0.031
ensemble$_{1,2,3}$	**0.971**	**0.981**	0.981	**0.933**	0.966±0.020
ensemble$_{1,2,3,4}$	**0.971**	**0.981**	**1.000**	0.923	**0.969±0.028**

Ultimately, the soft voting ensemble learning method is implemented to boost the final performance. The author class probability $p(c_i|D_i)$ of the BERTweet model is obtained through Equation 7. We integrate the *BERTweet* models with different loss functions, as well as the BERT feature-based CNN model *BERTweet-CONV*. The ensemble model achieves the best accuracy on average (*ensemble*$_{1,2,3,4}$ in Table 4).

For comparison and completeness, we also give the results of the 'classical' baseline models— the *SVM*, *LR*, and *RF* models—in Table 5. These three models are trained on the same internal training and test sets, i.e., 320 author samples are used for training; 100 author samples form the test set. Comparing Table 4 and Table 5, all results of the *BERTweet* model are much better

Table 5
Accuracy of SVM, LR, and RF, evaluated on 100 author samples.

	SVM	LR	RF
Acc.	0.900	0.890	0.860

than those of the *SVM*, *LR*, and *RF* models. Among these three models, the *LR*, and *RF* models are inferior to the *SVM* model.

The evaluation on the official test set was performed on the TIRA platform [40]. The hard voting of the 4-fold *BERTweet* + CEWF model achieves an accuracy of 98.89%. The soft voting of the 4-fold $ensemble_{1,2,3,4}$ model improves the performance to 99.44% on the official PAN test set.

7. Conclusion

This work describes our models for the PAN 2022 challenge, which poses the task of identifying whether an author is spreading sarcasm in their tweets. By using a re-segmentation strategy for lengthy documents, we can overcome the text length limitation of the BERT model. We compare two BERT models, specifically looking at the Roberta model in comparison to the BERTweet model. Our experiments show that the BERTweet model is clearly more suitable for the sarcasm discrimination task in twitter data. Based on the BERTweet model, a feature-based model is also designed. However, the feature-based model can not improve the accuracy compared to fine-tuning the BERT model on average. Nevertheless, the advantage of the feature-based model cannot be neglected: Compared to fine-tuning a BERT model, training a feature-based model is faster, while the results are also comparable to the fine-tuned BERT model. In this work, we consider three different loss functions, seeing that the cross-entropy weighted focal loss function as a compromise between the cross-entropy and the focal loss function yields slightly better results. Three conventional classifiers are also evaluated, namely, SVM, LR, and RF models, but the BERTweet model far outperforms these traditional classifiers. Finally, our experiments demonstrate that an ensemble approach based on soft voting of BERTweet models with different loss functions and the feature-based CNN model can further boost performance on the PAN challenge task.

Acknowledgments

References

[1] R. Krikorian, New Tweets per second record, and how!, in: Twitter Official Blog, August 16, 2013. URL: https://blog.twitter.com/engineering/en_us/a/2013/new-tweets-per-second-record-and-how.

[2] M. Siino, E. Di Nuovo, I. Tinnirello, M. La Cascia, Detection of hate speech spreaders using convolutional neural networks, in: CLEF, 2021.

[3] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[4] L. Torrey, J. Shavlik, Transfer learning, in: Handbook of research on machine learning applications and trends: algorithms, methods, and techniques, IGI global, 2010, pp. 242–264.

[5] F. Rangel, P. Rosso, M. Koppel, E. Stamatatos, G. Inches, Overview of the author profiling task at PAN 2013, in: CLEF Conference on Multilingual and Multimodal Information Access Evaluation, CELCT, 2013, pp. 352–365.

[6] F. Rangel, P. Rosso, I. Chugur, M. Potthast, M. Trenkmann, B. Stein, B. Verhoeven, W. Daelemans, Overview of the 2nd author profiling task at PAN 2014, in: CLEF 2014 Evaluation Labs and Workshop Working Notes Papers, Sheffield, UK, 2014, 2014, pp. 1–30.

[7] F. M. Rangel Pardo, F. Celli, P. Rosso, M. Potthast, B. Stein, W. Daelemans, Overview of the 3rd Author Profiling Task at PAN 2015, in: CLEF 2015 Evaluation Labs and Workshop Working Notes Papers, 2015, pp. 1–8.

[8] F. Rangel, P. Rosso, B. Verhoeven, W. Daelemans, M. Potthast, B. Stein, Overview of the 4th author profiling task at PAN 2016: cross-genre evaluations, in: Working Notes Papers of the CLEF 2016 Evaluation Labs. CEUR Workshop Proceedings/Balog, Krisztian [edit.]; et al., 2016, pp. 750–784.

[9] F. Rangel, P. Rosso, M. Potthast, B. Stein, Overview of the 5th author profiling task at PAN 2017: gender and language variety identification in twitter, Working notes papers of the CLEF (2017) 1613–0073.

[10] F. Rangel, P. Rosso, M. Montes-y Gómez, M. Potthast, B. Stein, Overview of the 6th author profiling task at PAN 2018: multimodal gender identification in twitter, Working Notes Papers of the CLEF (2018) 1–38.

[11] F. Rangel, P. Rosso, Overview of the 7th author profiling task at PAN 2019: bots and gender profiling in twitter, in: Working Notes Papers of the CLEF 2019 Evaluation Labs Volume 2380 of CEUR Workshop, 2019.

[12] F. Rangel, A. Giachanou, B. H. H. Ghanem, P. Rosso, Overview of the 8th author profiling task at PAN 2020: Profiling fake news spreaders on twitter, in: CEUR Workshop Proceedings, volume 2696, Sun SITE Central Europe, 2020, pp. 1–18.

[13] F. Rangel, G. Sarracén, B. Chulvi, E. Fersini, P. Rosso, Profiling hate speech spreaders on twitter task at PAN 2021, in: CLEF, 2021.

[14] M. C. M. M. K. Brodziľska, B. Celmer, M. Patera, J. Pezacki, M. Wilk, Ensemble-based classification for author profiling using various features (2013).

[15] K. Santosh, R. Bansal, M. Shekhar, V. Varma, Author profiling: Predicting age and gender from blogs, Notebook for PAN at CLEF 2013 (2013).

[16] A. P. López-Monroy, M. Montes-y Gómez, H. J. Escalante, L. V. Pineda, Using intra-profile information for author profiling., in: CLEF (Working Notes), 2014, pp. 1116–1120.

[17] M. A. Alvarez-Carmona, A. P. López-Monroy, M. Montes-y Gómez, L. Villasenor-Pineda, H. Jair-Escalante, INAOE's participation at PAN'15: Author profiling task, Working Notes Papers of the CLEF (2015) 103.

[18] M. B. op Vollenbroek, T. Carlotto, T. Kreutz, M. Medvedeva, C. Pool, J. Bjerva, H. Haagsma, M. Nissim, Gronup: Groningen user profiling, Notebook for PAN at CLEF (2016).

[19] A. Basile, G. Dwyer, M. Medvedeva, J. Rawee, H. Haagsma, M. Nissim, N-gram: new groningen author-profiling model, arXiv preprint arXiv:1707.03764 (2017).

[20] J. Pizarro, Using n-grams to detect bots on twitter., in: CLEF (Working Notes), 2019.

[21] J. Buda, F. Bolonyai, An ensemble model using n-grams and statistical features to identify fake news spreaders on twitter., in: CLEF (Working Notes), 2020.

[22] T. Takahashi, T. Tahara, K. Nagatani, Y. Miura, T. Taniguchi, T. Ohkuma, Text and image synergy with feature cross technique for gender identification, Working Notes Papers of the CLEF (2018).

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in neural information processing systems, 2017, pp. 5998–6008.

[24] Y. Joo, I. Hwang, L. Cappellato, N. Ferro, D. Losada, H. Müller, Author profiling on social media: An ensemble learning model using various features, Notebook for PAN at CLEF (2019).

[25] A. Baruah, K. A. Das, F. A. Barbhuiya, K. Dey, Automatic detection of fake news spreaders using BERT., in: CLEF (Working Notes), 2020.

[26] K. A. Das, A. Baruah, F. A. Barbhuiya, K. Dey, Ensemble of ELECTRA for profiling fake news spreaders., in: CLEF (Working Notes), 2020.

[27] S.-H. Wu, S.-L. Chien, A BERT based two-stage fake news spreader profiling system., in: CLEF (Working Notes), 2020.

[28] R. Zhang, Z. Wei, Y. Shi, Y. Chen, BERT-AL: BERT for arbitrarily long document understanding (2019).

[29] Z. Wang, P. Ng, X. Ma, R. Nallapati, B. Xiang, Multi-passage BERT: A globally normalized BERT model for open-domain question answering, arXiv preprint arXiv:1908.08167 (2019).

[30] D. Dukic, A. S. Kržic, Detection of hate speech spreaders with BERT (2021).

[31] 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, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Thirteenth International Conference of the CLEF Association (CLEF 2022), volume 13390 of *Lecture Notes in Computer Science*, Springer, 2022.

[32] O.-B. Reynier, C. Berta, R. Francisco, R. Paolo, F. Elisabetta, Profiling Irony and Stereotype Spreaders on Twitter (IROSTEREO) at PAN 2022, in: CLEF 2022 Labs and Workshops, Notebook Papers, CEUR-WS.org, 2022.

[33] F. Barbieri, J. Camacho-Collados, L. Neves, L. Espinosa-Anke, Tweeteval: Unified benchmark and comparative evaluation for tweet classification, arXiv preprint arXiv:2010.12421 (2020).

[34] D. Q. Nguyen, T. Vu, A. T. Nguyen, BERTweet: A pre-trained language model for english

tweets, arXiv preprint arXiv:2005.10200 (2020).

[35] N. Halko, P. Martinsson, J. Tropp, Finding structure with randomness: Stochastic algorithms for constructing approximate matrix decompositions (2009).

[36] B. C. Wallace, L. Kertz, E. Charniak, et al., Humans require context to infer ironic intent (so computers probably do, too), in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2014, pp. 512–516.

[37] C. Van Hee, E. Lefever, V. Hoste, SemEval-2018 task 3: Irony detection in English tweets, in: Proceedings of The 12th International Workshop on Semantic Evaluation, 2018. URL: https://aclanthology.org/S18-1005.

[38] Z. Wang, L. Wang, C. Huang, X. Luo, BERT-based chinese text classification for emergency domain with a novel loss function, arXiv preprint arXiv:2104.04197 (2021).

[39] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2980–2988.

[40] M. Potthast, T. Gollub, M. Wiegmann, B. Stein, TIRA Integrated Research Architecture, in: N. Ferro, C. Peters (Eds.), Information Retrieval Evaluation in a Changing World, The Information Retrieval Series, Springer, Berlin Heidelberg New York, 2019. doi:10.1007/978-3-030-22948-1_5.