Hybrid Representation Fusion for Twitter Hate Speech Identification

Wentao Yu, Benedikt Boenninghoff and Dorothea Kolossa

Institute of Communication Acoustics, Ruhr University Bochum, Germany

Abstract

The amount and impact of hate speech on social media is already alarming, causing harm for individuals and societies, yet it still keeps increasing. It has been reported that hate speech is strongly related to hate crimes and suicide rates. To stop hate speech on social media, online automatic approaches for hate speech detection in text have thus been drawing wide attention in the last decade.

In this work, we take up the HASOC-21 shared tasks, which focus on hate speech and offensive content identification in English. Two subtasks are addressed—one to identify the presence of any form of hate, offensive and profane content in a post, the other to discriminate between these three types of problematic content. We propose a hybrid representation fusion (HRF) structure, using both TF-IDF and BERTweet-based representations. Specifically, a Convolutional-BiLSTM (CBLSTM) network is used to extract semantic information from the BERTweet embeddings. Finally, we use the cosine similarity between different types of features as a reliability measure. We compare the performance between the proposed model *with* and *without* cosine similarity as an additional reliability measure. The former could reduce the standard deviation of the macro f1 score in a 10-fold cross-validation by over 27.27% for *subtask A* and 22.22% for *subtask B*, respectively. In the HASOC-21 ranking, our approach achieved an 80.13% macro F1 score for *subtask A* and 64.82% for *subtask B* on the official HASOC-21 test set.

Keywords

Hate Speech, BERTweet, CBLSTM, TF-IDF, Cosine Similarity, Feature Fusion

1. Introduction

In recent years, digital services like social media platforms or instant messaging systems have been struggling with hateful comments and abusive content, where users—typically anonymous—justify or instigate violence and discrimination against a person or a group. Hate speech permeates the vast majority of social media platforms and instant messaging systems, and has been increasing in volume and severity [1].

Nowadays, the impact of online hate speech (and misinformation) has entered all protagonists' consciousness, as hate speech strongly affects violent crimes in the real world. Research studies show a strong connection between the increased amount of shared and distributed hate speech messages and a higher occurrence of hate-related offenses against minorities [2]. And even

FIRE '21, Forum for Information Retrieval Evaluation, December 13-17, 2021, India

wentao.yu@rub.de (W. Yu); benedikt.boenninghoff@rub.de (B. Boenninghoff); dorothea.kolossa@rub.de (D. Kolossa)

https://cognitive-signal-processing.de/index.php/team/ (W. Yu);
https://cognitive-signal-processing.de/index.php/team/ (B. Boenninghoff);
https://cognitive-signal-processing.de/index.php/team/ (D. Kolossa)

Table 1
Class distribution of HASOC-21 dataset

	NOT	HOF			total
		HATE	OFFN	PRFN	
training	1342	683	622	1196	3843
test	483	224	195	379	1281

though not all cases of criminal acts directly follow from abusive online messages, its impact on, e.g., individual votes during democratic elections is vast [3, 4]. Additionally, victims are often strongly affected by the floods of hateful comments, and suffer from feelings of hopelessness or isolation, eating or sleep disorders, depression, and thoughts of suicide[5, 6, 7].

Consequently, there is an urgent need to find suitable tools and instruments to at least reduce the prevalence and impact of hate speech. However, due to the rising amount of shared and distributed poisonous content on social media platforms or instant messaging systems, it is infeasible to identify all, or even a significant part, manually.

The HASOC-21 shared task [8] encourages the participants working on automated solutions to counter this negative trend. The shared task offers annotated messages to develop machine learning methods for identifying hate speech and differentiating between different types of hate speech. More precisely, the dataset contains Twitter posts for both, English and Indo-Aryan languages [9]. In this work, we decided to work only with the annotated English tweets.

The paper is organized as follows: The dataset and our feature extraction are described in Section 2. Section 3 gives an overview of the proposed models, followed by the experimental setup, which is presented in Section 4. Finally, the results are discussed in Section 5 and conclusion are drawn in Section 6.

2. Dataset

The HASOC-21 dataset consists of annotated Twitter posts, with training and test set comprising 3843 and 1281 tweets, respectively. Each tweet has been sorted into the binary categories NOT (Non-Hate-Offensive) or HOF (Hate-Offensive). The HOF tweets are further discriminated into the three classes HATE (Hate speech), OFFN (Offensive), or PRFN (Profane). Table 1 shows the class distribution of the dataset.

2.1. Feature Extraction

In this work, we employ two types of features, the statistical term frequency-inverse document frequency (TF-IDF) features, and semantic text embeddings extracted from the pre-trained BERTweet model [10].

2.1.1. Text Preprocessing

Before extracting the TF-IDF features, we perform text normalization on the raw tweets to remove redundant and noisy information. Concretely, we decided to conduct the following steps:

1. Replacing all user names and hashtags with *USER* and *HASH*.
2. Removing all URLs.
3. Converting all Emojis into corresponding text representations[1].

To see the effect of this normalization, consider this raw tweet:

> @maiysha Death threats to Actor siddharth and his family for exposing BJP What can we expect from the party which files FIR for asking Oxygen 😡 #IStandWithSiddharth https://t.co/4Fwpw4y8Bf #ModiKaVaccineJumla.

After text normalization, we obtain:

> USER Death threats to Actor siddharth and his family for exposing BJP What can we expect from the party which files FIR for asking Oxygen :pouting_face: HASH HASH.

Apart from these steps, we also tested the effect of converting all upper-case tokens into lower case. However, results on the PAN18 English dataset [11], did not show any notable improvement.
The BERTweet model also provides a built-in text normalization stage, resulting in the following text-normalized tweet:

> @USER Death threats to Actor siddharth and his family for exposing BJP What can we expect from the party which files FIR for asking Oxygen :pouting_face: #IStandWithSiddharth HTTPURL #ModiKaVaccineJumla .

2.1.2. TF-IDF Features

TF-IDF features are based purely on term frequencies in collections of documents. Similar to [12], the TF-IDF features are based on n-gram models, where we perform Latent Semantic Analysis (LSA) [13] to reduce the dimension of the features. We use both, token-based 1- to 3-gram and character-based 3- to 5-gram models to extract the TF-IDF features, by the implementation provided in the scikit-learn library [14]. The TF-IDF features are extracted by the `TfidfVectorizer` function, which transforms each tweet into a vector[2]. The minimum document frequency is set to 2. The maximum document frequency is set to 1.0 (100%), which means terms occurring in all documents are ignored. Then, LSA is applied through the truncated singular value decomposition (SVD) [15]. The truncated SVD transforms the TF-IDF features to a 500-dimensional vector, which is done separately for the token- and character-based n-gram features [3]. Finally, the token- and the character-based transformed n-gram TF-IDF features are concatenated to obtain a 1000-dimensional feature vector.

2.1.3. BERTweet Features

In addition, we also extract semantic text embeddings from the pre-trained contextualized BERTweet language model [4]. As described in [10], this model was pre-trained on a large corpus

[1]https://github.com/carpedm20/emoji
[2]https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html
[3]https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html
[4]https://github.com/VinAIResearch/BERTweet.git

of English tweets.

Specifically, we chose the BERTweet-large model, where the embedding dimension for a single token is 1024, so that we obtain a sequence of 1024-dimensional token embeddings for each tweet.

3. System Overview

In this section, we present the model architecture of our final submission.

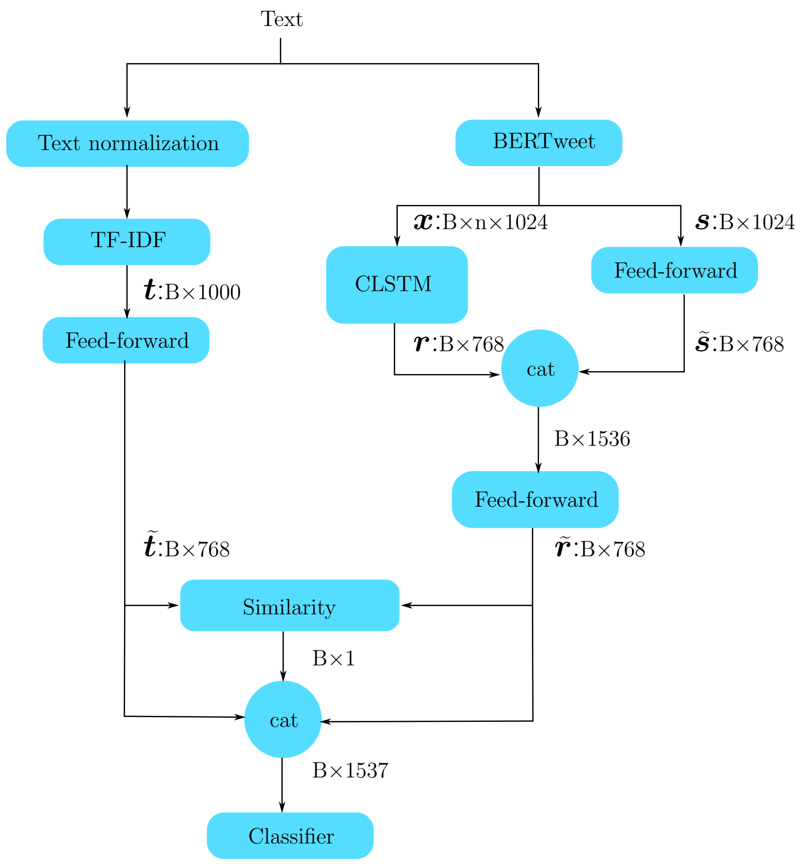


Figure 1: The hybrid representation fusion (HRF) model. *B* indicates the batch size, while *n* defines the maximum text length of the current batch.

Figure 1 shows the overall topology of our two-stage approach. In the first stage, the TF-IDF and BERTweet feature vectors are fed into two separate neural networks. In the second stage, we combine the outputs of both networks, also measuring the similarity between both network outputs. The similarity of two representations are considered as a reliability measure, which helps the TF-IDF model and BERTweet models make the same decision.

To reduce the computational complexity, a feed-forward network (FF) maps the features into a lower dimension.

$$\tilde{\boldsymbol{t}} = \mathrm{FF}(\boldsymbol{t}), \tag{1}$$

where $\boldsymbol{t}$ represents the TF-IDF features.

The pre-trained BERTweet model maps each token onto a 1024-dimensional vector. As described in [16], the input text of the BERTweet model is started with a special classification token [CLS] and ended with [SEP], e.g. "[CLS] @dxnprivv Stop being a twat then. [SEP]". After the BERTweet embedding, the first token [CLS] representation $\boldsymbol{s}$ is always used as a special classification token for text classification tasks.

Hence, we use the representation vector of the first token $\boldsymbol{s}$ (i.e. [B, 0, 1024]) as BERTweet classification feature. A feed-forward network is used to map $\boldsymbol{s}$ into a lower dimension.

$$\tilde{\boldsymbol{s}} = \mathrm{FF}(\boldsymbol{s}). \tag{2}$$

Additionally, the complete embedding sequence is fed into a Convolutional-BiLSTM (CBLSTM) network [17] to extract semantic information. We denote the input embedding sequence for a tweet as $\boldsymbol{x} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots \boldsymbol{x}_n]$, where $\boldsymbol{x}_i$, $i \in \{1, \ldots, n\}$ is a 1024-dimensional embedding vector of the i-th token. In the CBLSTM network, the sequence $\boldsymbol{x}$ is first fed into 4 BiLSTM layers,

$$\boldsymbol{h}_i^{(1)} = \mathrm{BiLSTM}(\boldsymbol{x}_i) \tag{3}$$

$$\boldsymbol{h}_i^{(2)} = \mathrm{BiLSTM}(\boldsymbol{h}_i^{(1)}) \tag{4}$$

$$\vdots \tag{5}$$

$$\boldsymbol{h}_i^{(4)} = \mathrm{BiLSTM}(\boldsymbol{h}_i^{(3)}) \tag{6}$$

where the cell dimension of each layer is $d = 768$, and $\boldsymbol{h}^{(4)} = [\boldsymbol{h}_1^{(4)}, \boldsymbol{h}_2^{(4)}, \cdots, \boldsymbol{h}_n^{(4)}]$ with $\boldsymbol{h}^{(4)} \in \mathrm{R}^{n \times 2d}$ represents the stacked matrix of all hidden layer states.

The CBLSTM now calculates a tweet representation, $\boldsymbol{r}$, using the attention mechanism. Firstly, we compute the dot product,

$$\tilde{h}_i = (\boldsymbol{h}_i^{(4)})^T \cdot \boldsymbol{h}_n^{(4)}, \quad i \in \{1, \ldots, n\} \tag{7}$$

where the last hidden layer state, $\boldsymbol{h}_n^{(4)}$, is used as a query and $\boldsymbol{h}_i^{(4)}$ is the current hidden layer state. Secondly, a convolutional layer is incorporated to extract relevant semantic information, yielding

$$c_i^m = \mathrm{ReLU}\left(\boldsymbol{w} \cdot \boldsymbol{h}_{i:i-m+1}^{(4)} + b\right). \tag{8}$$

Here, $\boldsymbol{w} \in \mathrm{R}^{1 \times m \cdot 2d}$ represents the convolution filter and b is the corresponding bias. The window size is chosen as $m \in \{3, 4, 5\}$. The convolution is performed on each possible window of the matrix $\boldsymbol{h}^{(4)}$ to produce a feature map $\boldsymbol{c}_m = [c_1^m, c_2^m, \cdots c_{n-m+1}^m]$. It is obvious that not all tokens contribute equally to the final decision. Consequently, a maximum-pooling layer is used to find the most informative features in the document for the task,

$$v_m = \max\{c_1^m, c_2^m, \cdots, c_{n-m+1}^m\}. \tag{9}$$

After concatenating the pooling results, $\boldsymbol{v} = [v_3, v_4, v_5]$, the attention weights are obtained by

$$\alpha_i = \operatorname{softmax}\left(\tilde{h}_i \cdot \boldsymbol{W}_\alpha^T \cdot \boldsymbol{v}\right), \tag{10}$$

where $\boldsymbol{W}_\alpha \in \mathrm{R}^{3\times 1}$ is the trainable parameter vector. The final hidden state representation is given by:

$$\boldsymbol{s} = \sum_{i=1}^{n} \alpha_i \cdot \boldsymbol{h}_i^{(4)}. \tag{11}$$

Now, the hidden state representation is concatenated with the last state of the last hidden layer, $\boldsymbol{h}_n^{(4)}$, and fed into a feed-forward network to obtain the final tweet representation $\boldsymbol{r} \in \mathrm{R}^{1\times d}$.

$$\boldsymbol{r} = \mathrm{FF}\left(\left[\boldsymbol{h}_n^{(4)}, \boldsymbol{s}\right]\right). \tag{12}$$

Subsequently, the BERTweet classification vector $\tilde{\boldsymbol{s}}$ in Eq (2) and the hidden state representation $\boldsymbol{r}$ in Eq. (13) are concatenated. Again, a feed-forward network projects the obtained vector into a lower dimension.

$$\tilde{\boldsymbol{r}} = \mathrm{FF}\left([\boldsymbol{r}, \tilde{\boldsymbol{s}}]\right). \tag{13}$$

Now, we can use the TF-IDF- and the BERTweet-based representations to make the decision. If the representations for these two feature types are similar, the TF-IDF model and the BERT model are tending towards the same outcome and vice versa. To help the model towards unified decisions, we calculate the cosine similarity of these two representations. Finally, the two feature representations and the cosine similarity are concatenated, and a feed-forward classifier makes the final decision based on this vector.

4. Experimental Setup

We evaluated our proposed model on two subtasks. *subtask A* is a binary task, to identify each tweet as 'NOT' (non-offensive) or as 'HOF' (hate/offensive speech). For this task, the network output dimension is thus 1 and we used a sigmoid function as the output layer activation function. The network was trained with the binary cross-entropy as the loss function. *subtask B* is to classify tweets into four different categories, namely 'HATE' (Hate speech), 'OFFN' (Offensive), 'PRFN' (Profane) or 'None' (No hate speech). The output dimension was thus set to 4, with a softmax function as the output layer activation function and the cross-entropy loss. Each feed-forward network contains two hidden layers of 768 units each, with ReLU activation functions.

To compare the model performance in different setups, we applied 10-fold cross-validation on the training set. Five different model versions are implemented:

- *TF-IDF*: the TF-IDF stream in the *HRF* model (Figure 1) only uses TF-IDF $\boldsymbol{t}$ as features.
- *BERT-C*: the BERTweet classification stream in the *HRF* model, based solely on the BERTweet classification features $\tilde{\boldsymbol{s}}$.
- *BERT-E*: the BERTweet embedding stream in *HRF* model, based on the BERTweet embedding features $\boldsymbol{r}$.

Table 2
Cross-validation results on the validation set. Bold print indicates the maximum mean value and the minimum standard deviation.

subtask	model	Macro F1	Macro Precision	Macro Recall	Accuracy
subtask A	*TF-IDF*	0.739±0.051	0.730±0.047	0.778±0.033	0.784±0.028
	BERT-C	**0.816**±0.015	**0.808**±**0.016**	0.827±0.019	**0.837**±0.015
	BERT-E	0.807±0.015	0.796±**0.016**	0.827±**0.012**	0.833±0.012
	HRF-NC	0.808±0.019	0.798±0.023	0.825±0.013	0.833±0.012
	HRF	0.809±**0.012**	0.799±0.017	**0.830**±**0.012**	0.834±**0.007**
subtask B	*TF-IDF*	0.441±0.075	0.490±0.045	0.458±0.151	0.607±0.036
	BERT-C	0.653±0.028	0.657±0.026	0.673±0.017	0.699±**0.014**
	BERT-E	0.656±**0.015**	0.656±0.015	**0.676**±0.017	0.700±**0.014**
	HRF-NC	0.651±0.019	0.653±0.018	0.673±**0.013**	0.700±0.015
	HRF	**0.660**±0.017	**0.660**±**0.014**	0.672±0.020	**0.701**±0.016

- *HRF*: our proposed model combining both feature streams as shown in Figure 1.
- *HRF-NC*: our proposed model, without the cosine similarity as a reliability measure.

All models were implemented using the PyTorch library [18]. We trained the models for 50 epochs, choosing a batch size of 48, and using the AdamW optimizer [19]. To avoid overfitting, we incorporated early stopping to monitor the degradation on the held-out set. More precisely, the training was stopped when the accuracy on the validation set did not increase over 9 epochs. We set the initial learning rate to $2 \cdot 10^{-5}$ and we further installed a linear learning rate scheduler with warm-up to change the learning rate during the first four epochs of the training stage.

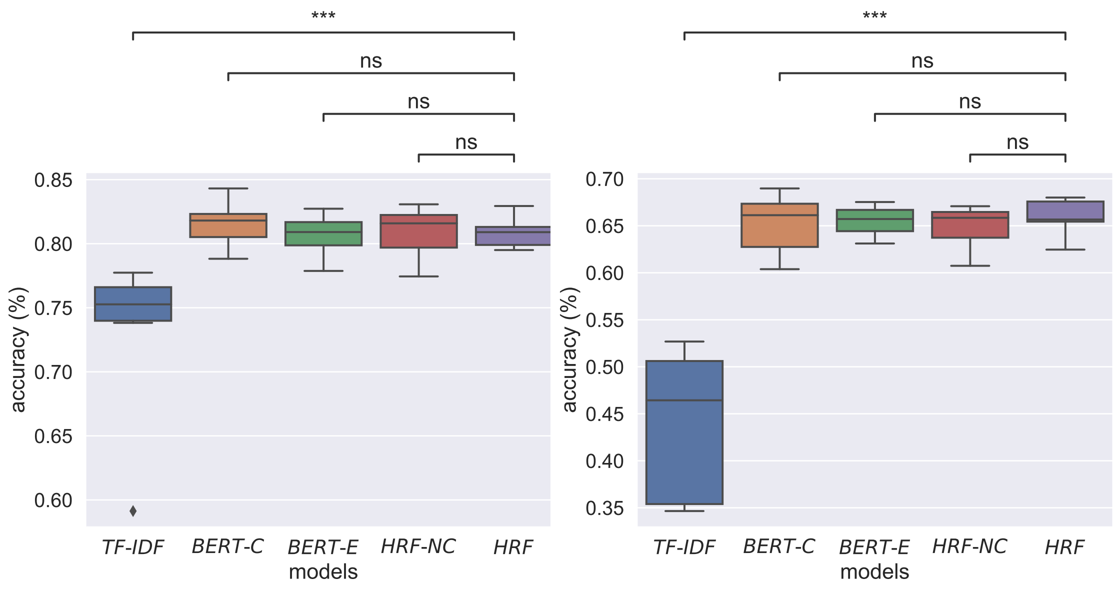
During training, all trainable parameters of the pre-trained BERTweet models were also fine-tuned. Inspired by [20], we finally averaged the weights of those two epoch-wise models, which returned the highest validation accuracy during the training stage.

All models were trained using NVIDIA's Volta-based DGX-1 multi-GPU system, using 2 Tesla V100 GPUs with 32 GB memory each.

5. Results

The two box plots in Figure 2 provide the cross-validation results of the baseline models with our proposed hybrid model on the training set. The validation set is used for early stopping. As it can be seen, the F1 score of the *TF-IDF*-based model that only relies on traditional features is significantly lower than that of the other models. We use the Mann-Whitney-U test [21] to indicate statistically significant differences in the F1 score between the proposed *HRF* model and other baselines. However, from Plots 2a and 2b, we do not notice an improvement from the combination of both, traditional and neural features.

Table 2 shows all obtained F1 scores in greater detail. Starting with *subtask A*, the *BERT*-based models again significantly outperform the *TF-IDF* system, while *BERT-C* is on par with *HRF*. For *subtask B*, the proposed hybrid model slightly outperforms the *BERT-C* model. However, overall, for both *subtask A* and *subtask B*, there is no statistically significant difference between the different *BERT*-based models.



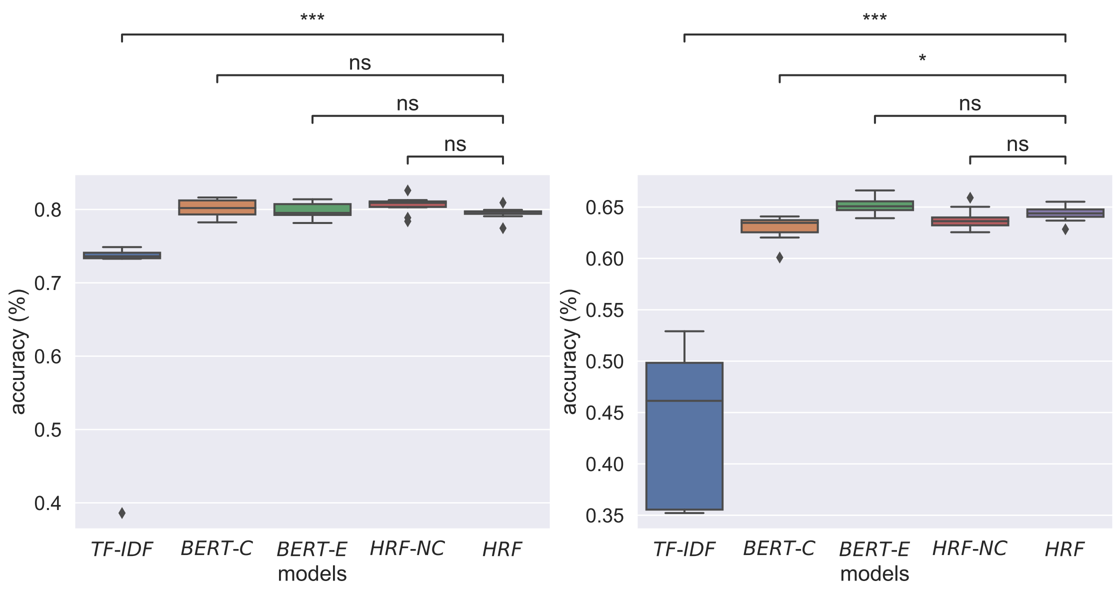
(a) *subtask A* (b) *subtask B*

Figure 2: Macro F1 scores for 10-fold cross-validation on the validation set. Asterisks indicate a statistically significant difference, where *** denotes 1e-04 < p <= 1e-03, ns denotes p > 5e-02.

Table 3
Results on the official test set

subtask	model	Macro F1	Macro Precision	Macro Recall	Accuracy
subtask A	*TF-IDF*	0.703±0.106	0.706±0.069	0.773±0.016	0.755±0.044
	BERT-C	0.801±0.011	0.792±0.013	0.820±0.009	0.821±0.009
	BERT-E	0.798±0.009	0.787±0.011	0.827±**0.005**	0.821±0.006
	HRF-NC	**0.806**±0.011	**0.795**±0.012	**0.831**±0.008	**0.826**±0.009
	HRF	0.795±**0.008**	0.784±**0.010**	0.822±0.006	0.817±**0.005**
subtask B	*TF-IDF*	0.438±0.072	0.482±0.041	0.442±0.129	0.603±0.018
	BERT-C	0.630±0.012	0.639±0.010	0.640±0.012	0.674±0.013
	BERT-E	**0.652**±0.008	**0.658±0.007**	**0.661±0.008**	**0.693±0.008**
	HRF-NC	0.638±0.009	0.646±**0.007**	0.654±0.012	0.685±**0.008**
	HRF	0.644±**0.007**	0.650±0.010	0.655±0.012	0.686±**0.008**

After getting access to the official groundtruth labels, we repeated the cross-validation procedure on the HASOC-21 test set. The results are shown in Figure 3 and Table 3. Altogether, we do not observe significant differences between the training and the test sets, which shows the robustness of our approaches. Having a closer look at Table 3, the *HRF* model reduces the standard deviation of the macro F1 score by over 27.27% for *subtask A* and 22.22% for *subtask B* compared to the *HRF-NC* model. However, on *subtask B*, the *BERT-E* model has the better performance. Figure 3a still shows no statistically significant difference between the *BERT*-based models, while we can see a slight difference between *BERT-C* and the proposed *HRF* model in Figure 3b.



(a) *subtask A* (b) *subtask B*

Figure 3: Macro F1 scores for 10-fold cross-validation on the official test set. Asterisks indicate a statistically significant difference, where *** denotes 1e-04 < p <= 1e-03, * denotes 1e-02 < p <= 5e-02, ns denotes p > 5e-02.

Finally we submitted different versions of the *HRF* model to participate in the HASOC-21 ranking. We randomly selected 128 Twitter posts from the training set as the validation set. The model performance is tested on the HASOC-21 official test set.

For the final raking, all participants were allowed a maximum of five submissions. Firstly, we submitted two different versions of the *HRF* system, *with* and *without* fine-tuned BERTweet parameters. The first two rows in Table 4 show these results for *subtask A* and emphasise the advantage of task-specific fine-tuning of the standard *BERTweet* model. Next, we submitted different *HRF* versions (for both tasks) which are based on the standard or large *BERTweet* version. It is observable that the large *BERTweet* model tends towards a slightly better performance, especially for *subtask B*.

In summary, our best models in Table 4 achieved rank 11 out of 56 participants for *subtask A* and placed 3rd out of 37 participants for *subtask B* in the HASOC-21 ranking [8].

6. Conclusion

We are presenting our approach for the HASOC-21 shared tasks of hate speech identification and classification, detailing its stages from the pre-processing steps, via the design of the neural model architecture, to the training procedure. The core of our approach is a novel neural fusion strategy to combine classical TF-IDF features with fine-tuned contextualized BERTweet embeddings. We incorporated the cosine similarity between the final TF-IDF- and BERT-based representations to learn a suitable fusion network. The results of our 10-fold cross-validation

Table 4
Performance of all versions of the proposed HRF model that participate in the HASOC-21 ranking. **type** indicates which BERTweet model is used in the HRF network; **finetuning** shows if the HRF network updates the BERTweet parameters during the training stage.

subtask	type	finetuning	Macro F1	Macro Precision	Macro Recall	Acc. (%)
subtask A	BERTweet	*without*	0.747	0.762	0.740	77.127%
		with	0.795	0.817	0.784	81.577%
	BERTweet-large	*with*	**0.801**	**0.817**	**0.793**	**81.967**%
subtask B	BERTweet	*with*	0.625	0.639	0.625	68.384%
	BERTweet-large	*with*	**0.648**	**0.651**	**0.652**	**68.931**%

are consistent with the test set results. In the challenge, our submissions ranked 11th out of 56 participants for *subtask A* and 3rd out of 37 participants for *subtask B*.

The vast amounts of information on social media cannot realistically be monitored for hate speech by linguists. To address the demand of this situation, we would like to design a model that focuses not only on accuracy but also on delivering reliable confidence values. In this context, and moving one step towards responsible AI, future work will focus on calibration methods such as Dirichlet calibration [22]. These are important for obtaining reliable output predictions, in which confidence is well-matched with accuracy, so that the system output provides a sound basis for real-world decision-making.

Acknowledgments

The work was supported by the PhD School "SecHuman - Security for Humans in Cyberspace" by the federal state of NRW, and partially funded by the Deutsche Forschungsgemeinschaft (DFG – German Research Foundation) - Project-ID 429873205.

References

[1] B. Mathew, R. Dutt, P. Goyal, A. Mukherjee, Spread of hate speech in online social media, ACM Press, USA, 2019.

[2] M. Williams, P. Burnap, A. Javed, H. Liu, S. Ozalp, Hate in the machine: Anti-black and anti-muslim social media posts as predictors of offline racially and religiously aggravated crime, The British Journal of Criminology (2020).

[3] C. Ezeibe, Hate speech and election violence in Nigeria, Journal of Asian and African Studies 56 (2021).

[4] J. V. Spanje, C. D. Vreese, The good, the bad and the voter: The impact of hate speech prosecution of a politician on electoral support for his party, Party Politics (2015).

[5] L. Leets, Experiencing hate speech: Perceptions and responses to anti-semitism and antigay speech, Journal of social issues (2002).

[6] M. Hsueh, K. Yogeeswaran, S. Malinen, "leave your comment below": Can biased online comments influence our own prejudicial attitudes and behaviors?, Human communication research (2015).

[7] N. Kteily, E. Bruneau, Backlash: The politics and real-world consequences of minority group dehumanization, Personality and Social Psychology Bulletin (2017).

[8] S. Modha, T. Mandl, G. K. Shahi, H. Madhu, S. Satapara, T. Ranasinghe, M. Zampieri, Overview of the HASOC Subtrack at FIRE 2021: Hate Speech and Offensive Content Identification in English and Indo-Aryan Languages and Conversational Hate Speech, in: FIRE 2021: Forum for Information Retrieval Evaluation, Virtual Event, 13th-17th December 2021, ACM, 2021.

[9] T. Mandl, S. Modha, G. K. Shahi, H. Madhu, S. Satapara, P. Majumder, J. Schäfer, T. Ranasinghe, M. Zampieri, D. Nandini, A. Jaiswal, Overview of the HASOC subtrack at FIRE 2021: Hate Speech and Offensive Content Identification in English and Indo-Aryan Languages, in: Working Notes of FIRE 2021 - Forum for Information Retrieval Evaluation, CEUR, 2021. URL: http://ceur-ws.org/.

[10] D. Nguyen, T. Vu, A. Nguyen, BERTweet: A pre-trained language model for English Tweets, 2020.

[11] F. Rangel, P. Rosso, M. M. 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.

[12] S. Daneshvar, D. Inkpen, Gender identification in twitter using n-grams and LSA, in: Proc. CLEF 2018), 2018.

[13] S. T. Dumais, Latent semantic analysis, Annual review of information science and technology (2004).

[14] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., Scikit-learn: Machine learning in Python, the Journal of machine Learning research 12 (2011).

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

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

[17] J. Zheng, L. Zheng, A hybrid bidirectional recurrent convolutional neural network attention-based model for text classification (2019).

[18] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, Advances in neural information processing systems (2019).

[19] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, arXiv preprint arXiv:1711.05101 (2017).

[20] G. Huang, Y. Li, G. Pleiss, Z. Liu, J. E. Hopcroft, K. Q. Weinberger, Snapshot ensembles: Train 1, get M for free, in: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, 2017.

[21] H. Mann, D. Whitney, On a test of whether one of two random variables is stochastically larger than the other, The annals of mathematical statistics (1947) 50–60.

[22] M. Kull, M. Perello-Nieto, M. Kängsepp, H. Song, P. Flach, et al., Beyond temperature scaling: Obtaining well-calibrated multiclass probabilities with dirichlet calibration, arXiv preprint arXiv:1910.12656 (2019).