RACAI at ClimateActivism 2024: Improving Detection of Hate Speech by Extending LLM Predictions with Handcrafted Features

Vasile Păiș
Research Institute for Artificial Intelligence "Mihai Drăgănescu", Romanian Academy
Bucharest, Romania
vasile@racai.ro

Abstract

This paper describes the system that participated in the Climate Activism Stance and Hate Event Detection shared task organized at The 7th Workshop on Challenges and Applications of Automated Extraction of Sociopolitical Events from Text (CASE 2024). The system tackles the important task of hate speech detection by combining large language model predictions with manually designed features, while trying to explain where the LLM approach fails to predict the correct results.

1 Introduction

Hate speech identification is an important task when analyzing climate change activism events. The shared task (Thapa et al., 2024) organized at the CASE 2024 workshop provided a place to test different approaches for detecting hate speech in short messages specific to social media platforms, such as X (previously known as Twitter). Hate speech can be defined as any message that denigrates individuals or groups based on some characteristic such as race, color, ethnicity, gender, sexual orientation, nationality, or religion (Nockleby, 1994). Messages of interest for the task are exchanged during or related to climate change activism events.

Since many recent works focus on the application of Large Language Models (LLMs) for classifying messages as hateful or not, this work investigated the possibility of improving LLM predictions using handcrafted features. A decision tree was trained in the hope that the resulting decisions could explain the failure of LLM predictions in certain cases.

The rest of this paper is organized as follows: Section 2 provides related work, Section 3 briefly introduces the task and describes the dataset, Section 4 gives an overview of the participating system, including pre-processing and architecture, Section 5 presents the results, and Section 6 gives conclusions and future work.

2 Related work

The survey of Schmidt and Wiegand (2017) presents a number of methods and features useful for hate speech classification, including simple surface features, word generalization, sentiment analysis, lexical resources, linguistic features, knowledge-based features, and meta-information. Further analysis is provided by Parihar et al. (2021). Poletto et al. (2021) provides a review of existing resources and benchmark corpora for hate speech detection. The survey of Jahan and Oussalah (2023) presents different methods employing word embedding representations (both static and contextualized) for hate speech detection.

The recent HaSpeeDe3 shared task (Lai et al., 2023) provided another place for evaluating hate speech detection systems. The system of Grotti and Quick (2021) employed two pre-trained cased BERT-based (Devlin et al., 2019) LLMs, with initial pre-processing by turning hashtags into words to reduce noise. The system of Di Bonaventura et al. (2023) made use of ALBERTo (Polignano et al., 2019) LLM, combined with the Ontology of Dangerous Speech (Stranisci et al., 2022).

Apart from general hate speech detection, specific lexical phenomena have been studied. Dinu et al. (2021) studied the usage of pejorative language in social media. Davidson et al. (2017) acknowledges the distinctions between hate speech and offensive language, which makes the task of hate speech detection more challenging.

3 Dataset and task

The goal of the hate speech detection task is to identify for a given message if it contains hate speech or not. This is a binary label associated with each provided message in the dataset. Dataset files (with

Proceedings of the 7th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2024), pages 67–72
March 22, 2024 ©2024 Association for Computational Linguistics

splits for training, validation and testing) were provided in CSV format, containing three columns: *index* is a numeric value identifying the message; *tweet* is the actual message; *label* is a numeric value, 1 if the message contains hate speech and 0 otherwise. The dataset is described in detail by Shiwakoti et al. (2024).

The training file contains 7,284 messages of various sizes. The shortest message has only 29 characters, while the largest has 985 characters. The validation file has 1,561 messages with sizes from 29 characters to 940 characters. The test file has 1,562 messages with sizes from 1 character to 960 characters. The size distribution is given in Figure 1. Overall there are 10,407 messages in the entire dataset, 1,277 marked as containing hate speech (12.27%).

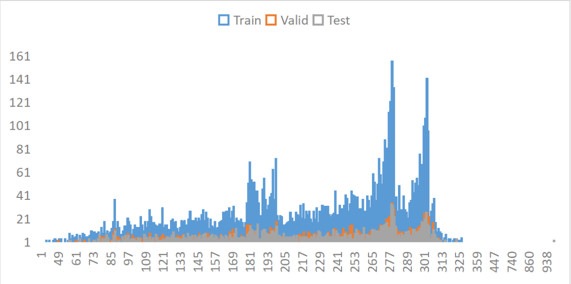

Figure 1: Raw message size distribution.

The messages are included as they were collected from the Internet, without any preprocessing. Therefore, they contain elements such as hashtags, emojis, user references, new lines, spelling errors, inconsistent casing (including all uppercase letters for the entire message or message parts), new lines, URLs.

Some messages contain a large number of hashtags (the maximum number in the training set is 26 in a single message), URLs (maximum 6) or user mentions (maximum 50 in a single message). This is sometimes used to make a message easily discoverable by people looking for a certain hashtag. However, hashtags (sometimes comprised of multiple words, such as "#stopfakegreen") are used to convey a message, which could be hateful. An example message is: *This is why UK politicians are so reluctant to divest from fossil fuels: 1/7 GOVUK #Corruption #ToryCorruption #ExtinctionRebelliom #XR #KeepItInTheGround #ClimateJustice #FridaysForFuture #GreenNewDeal #UKPolitics #TalkingClimate Lets_Discuss_CC*. In this message, simply ignoring the hashtags provide no clues as to why it was marked as hate speech. However, considering the hashtags (especially "#Corruption" and "#ToryCorruption") clarifies the labeling.

URLs present in the dataset are shortened, always starting with "https://t.co" and followed by a code. Therefore URL itself does not add information useful for hate speech detection.

Shiwakoti et al. (2024) mention that rigorous measures were taken to anonymize all usernames and identifiable user information within the dataset. Therefore, the text associated with the user references was not considered relevant for this work.

4 Methodology

4.1 Pre-Processing

The pre-processing operation aimed to transform the raw messages into regular text. All blank characters, including new lines, tabs and other UTF-8 characters, were transformed to regular spaces. Multiple space characters were replaced with a single space. Different UTF-8 characters representing quotation marks were removed. URLs and user mentions were removed as well. Hashtags were split into words when possible, using an algorithm similar to the one described by Micu et al. (2022).

Special characters, including emojis, were removed from text. Even though the use of emojis was shown to improve the results on certain tasks, such as sentiment polarity classification (Gupta et al., 2023), for this work emojis were not considered, primarily because they were not properly handled by the LLMs used.

Due to the inconsistent use of casing in messages, the text was transformed to all lowercase characters.

The resulting pre-processed message size distribution is given in Figure 2. The distribution is more even compared to the original distribution. A large number of messages now have 36 characters, the smallest message having 0 characters (initially had 1 character) and the largest message has 266 characters. Given the relative shortness of the messages, no special considerations are needed when tokenizing and encoding using a LLM.

User mentions are sometimes used as a forwarding mechanism (also known as "retweet") where a user repeats a message to make specific users aware of its content. By using the pre-processing steps above, a number of 1,249 messages were identified as duplicates, thus from the total of 7,284 training examples, only 6,035 were unique.

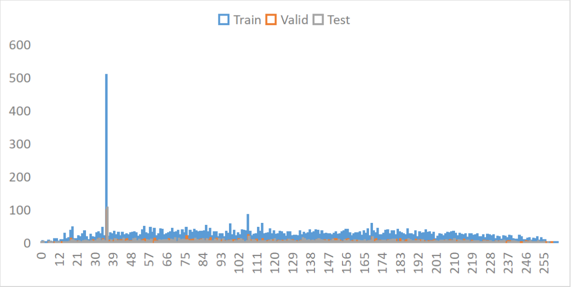

Figure 2: Raw message size distribution.

4.2 System architecture

The system is developed around a text classifier employing a BERT LLM. It has two additional linear layers, with 2,048 and 1,024 cells respectively, employing ReLU and tanh activation functions respectively. These are followed by a final class prediction head.

In order to potentially improve on the LLM predictions and to explore the cases where the LLM gets the result wrong, a set of handcrafted features were produced. The initial set of features that were considered comprises: number of raw hashtags, remaining hashtags after pre-processing, hashtags that were split during pre-processing, user mentions, URLs, raw size, pre-processed size, size difference, TF-IDF prediction. Out of these the raw size, pre-processed size, size difference and raw hashtags were removed from the final system, their influence being limited. Initial experiments showed they had no contribution towards increasing the decision tree accuracy. Furthermore, their usage as leafs on the tree may lead to the model overfitting on potentially less relevant features. On the other hand, there is a difference between the average number of hashtags per message (4.9 for non-hate vs 6.89 for hate speech), the average number of user mentions per message (1.06 for non-hate vs 0.59 for hate), and the average number of URLs per message (0.83 for non-hate vs 0.26 for hate). The numbers were computed on the training set.

For TF-IDF predictions only, the text was further lemmatized using the WordNet (Fellbaum, 1998) lemmatizer available in the NLTK library[1]. Common English words were removed using the stop words set provided by the same NLTK library.

The final stage of the system is represented by a decision tree which combines the LLM predictions with the features. The overall system architecture is presented in Figure 3. At this stage, the different features were written as numerical columns in a CSV file, each row representing a message. Predictions from BERT and TF-IDF were added as two new columns. Only the actual predicted label (0 or 1) was added, without any probabilities. Finally, the resulting file was fed into a decision tree classifier.

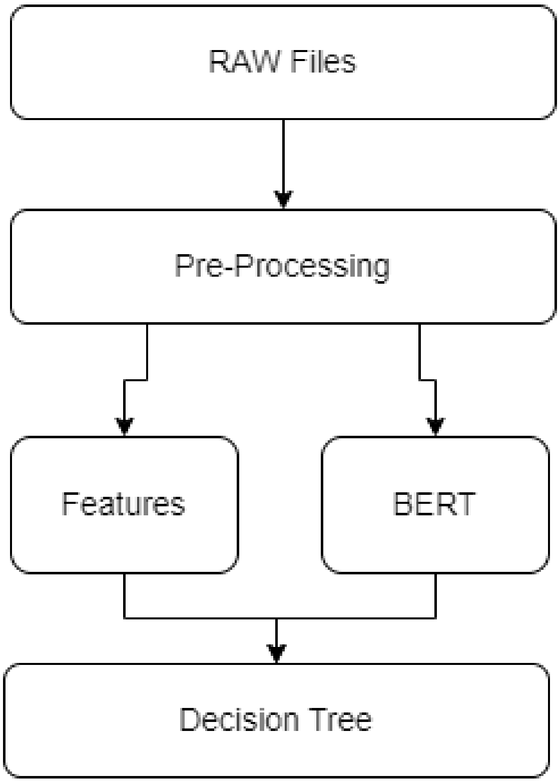

Figure 3: System architecture.

5 Results and discussion

The LLM used for training the system was BERT-large-uncased. The choice of an uncased model version is justified by the pre-processing step that removes capitalization and transforms the text into lowercase characters. The model was trained for at least 5 epochs and a maximum of 20 epochs, with early stopping, when there was no improvement for 3 epochs. During the first 3 epochs, the LLM was frozen and only the last linear layers were actually trained. A batch size of 128 was used. The learning rates for the LLM and the other layers were kept separated. The best hyper-parameters were determined by performing a grid search, with the encoder learning rate possible values of 1e-05, 2e-05, 3e-05, 5e-05, 9e-06, and the learning rate for the linear layers with values of 5e-05, 4e-05, 3e-05, 2e-05, 8e-05. The choice for these specific values is justified by previous experience as well

[1] https://www.nltk.org/

System	P	R	F1	Acc
BERT	89.07	**82.79**	**85.55**	94.37
TF-IDF	**96.79**	78.69	84.93	**94.81**
DT	91.17	81.53	85.48	94.56
Baseline	-	-	70.80	90.10

Table 1: Results on the test dataset.

as the time constraints associated with the shared task, further exploration was not possible within the allocated time.

During training, a 10-fold cross validation approach was used. For each hyper-parameter values 10 experiments were performed and the best values were selected. This resulted in the final system using 3e-05 for the encoder learning rate and 2e-05 as the learning rate for the linear layers. The final model training lasted 11 epochs.

Results are given in Table 1. "Baseline" represents the results reported in the dataset description paper (Shiwakoti et al., 2024), based on a BERT model. "BERT" is the system trained in this paper, using bert-large-uncased with the classification head and parameters described above. "TF-IDF" is the application of a TF-IDF algorithm, as implemented by the Sci-Kit[2] learn library, on the pre-processed text. "DT" is the application of a decision tree based on the results of "BERT", "TF-IDF" and the rest of the features described in Section 4.2. Results were computed using the official evaluation script, available in the shared task's CodaLab environment.

Interestingly, all three systems, including the basic TF-IDF were able to surpass the F1 and Accuracy scores reported by Shiwakoti et al. (2024), using a BERT model. This is probably due to the pre-processing described in Section 4.1. Each system has its strong points, "TF-IDF" provides the best precision and accuracy, "BERT" provides the best recall and F1, while the combination of the two systems, as well as additional features, using the decision tree "DT" provides good values for all metrics. However, since the shared task evaluation was conducted based on F1 score only, the results of the fine-tuned BERT model were submitted for the final evaluation.

Analyzing the decision tree diagram, shows that apart from the TF-IDF and BERT predictions (these are the top-level decision nodes in the tree), the most important features are the number of hashtags that were split during pre-processing, the number of remaining hashtags (without being split) and the number of URLs. Analyzing the message numbers, an average of 0.57 hashtags were split on non-hate messages, compared to an average of 0.11 in hate messages. This seem to indicate that the presence of a large number of these elements makes the text harder to classify by both BERT and TF-IDF. However, the results of the decision tree classifier indicate that relying solely on these numbers to adjust the predictions is not possible. Instead, research is needed into properly handling messages with a large number of hashtags and URLs. Furthermore, research is needed into handling difficult hashtags, containing multiple words or names that are harder to split using automated methods.

6 Conclusion

Results, as discussed above, indicate that simpler algorithms, such as TF-IDF, may provide good enough results for certain tasks within a reduced amount of time compared to deep neural networks. However, the result is clearly influenced by proper pre-processing operations, since TF-IDF when applied on pre-processed text provides improved results compared to the baseline BERT approach applied on raw text.

Explainable AI approaches try to improve our understanding of black-box neural models by explaining their predictions and thus contributing to our trust in such models (Dwivedi et al., 2023; Nauta et al., 2023; Xu et al., 2019). In this paper, by using a decision tree to combine LLM results with TF-IDF and other features, the final model tries to explain and improve upon failures of the LLM approach. This highlighted a need for further research into handling social media messages with a large number of hashtags, URLs, or complex hashtags that may not be easily split into words.

During the pre-processing operations, special characters, including emojis were discarded. However, the inclusion of emoji representations, such as Emoji2Vec (Eisner et al., 2016), may improve the system's results. Furthermore, the current work focused only on BERT-like LLM. Exploration of other model architectures for hate speech detection is needed. Inclusion of additional features, such as the usage of pejorative words, could better the explanation of when the LLM fails to provide correct results.

The dataset provided for this task provided

[2] https://scikit-learn.org/stable/

boolean indications of messages containing or not hate speech. Other tasks offered additional classification, such as the targets of hate speech (individual, organization, and community targets). For the purposes of this work only the task-specific dataset was considered, with no additional resources. However, further investigation may involve combining other datasets in order to better understand if a certain type of hate speech is less likely to be identified by the proposed system. Even more, other authors explore the intensity associated with hate speech (Geleta et al., 2023) or other classifications (Paz et al., 2020). Extending the dataset with additional indicators may allow future work to better explore a model's failures and provide clues that may aid in improving the model's performance.

In accordance with open science principles, the source code of the participating system is made open source in our GitHub repository[3]. A rendered diagram of the decision tree is available in the same place[4], while the image size prevents its inclusion directly in the paper.

Limitations

The current system implementation, models and resources are limited to the English language. The system architecture does not take into account long messages that surpass the direct capability of the LLMs used.

Ethics Statement

We do not foresee ethical concerns with the research presented in this paper. However, it is important to acknowledge that unintended bias might be present in the dataset, even considering the high level of agreement between annotators, and this could be reflected in the resulting models.

References

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. *Proceedings of the International AAAI Conference on Web and Social Media*, 11(1):512–515.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Chiara Di Bonaventura, Arianna Muti, Marco Antonio Stranisci, B McGillivray, and A Meroño-Peñuela. 2023. O-dang at hodi and haspeede3: A knowledge-enhanced approach to homotransphobia and hate speech detection in italian. *Proceedings of the Eighth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian (EVALITA 2023)*, 3473.

Liviu P. Dinu, Ioan-Bogdan Iordache, Ana Sabina Uban, and Marcos Zampieri. 2021. A computational exploration of pejorative language in social media. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3493–3498, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Rudresh Dwivedi, Devam Dave, Het Naik, Smiti Singhal, Rana Omer, Pankesh Patel, Bin Qian, Zhenyu Wen, Tejal Shah, Graham Morgan, and Rajiv Ranjan. 2023. Explainable ai (xai): Core ideas, techniques, and solutions. *ACM Comput. Surv.*, 55(9).

Ben Eisner, Tim Rocktäschel, Isabelle Augenstein, Matko Bošnjak, and Sebastian Riedel. 2016. emoji2vec: Learning emoji representations from their description. In *Proceedings of the Fourth International Workshop on Natural Language Processing for Social Media*, pages 48–54, Austin, TX, USA. Association for Computational Linguistics.

Christiane Fellbaum. 1998. *WordNet: An electronic lexical database*. MIT press.

Raisa Romanov Geleta, Klaus Eckelt, Emilia Parada-Cabaleiro, and Markus Schedl. 2023. Exploring intensities of hate speech on social media: A case study on explaining multilingual models with XAI. In *Proceedings of the 4th Conference on Language, Data and Knowledge*, pages 532–537, Vienna, Austria. NOVA CLUNL, Portugal.

Leonardo Grotti and Patrick Quick. 2021. Berticelli at haspeede 3: Fine-tuning and cross-validating large language models for hate speech detection. *world*, 2(3):4.

Shelley Gupta, Archana Singh, and Vivek Kumar. 2023. Emoji, text, and sentiment polarity detection using natural language processing. *Information*, 14(4).

Md Saroar Jahan and Mourad Oussalah. 2023. A systematic review of hate speech automatic detection using natural language processing. *Neurocomputing*, 546:126232.

[3] https://github.com/racai-ai/CASE2024_HateSpeech/

[4] https://github.com/racai-ai/CASE2024_HateSpeech/blob/master/tree.png

Mirko Lai, Fabio Celli, Alan Ramponi, Sara Tonelli, Cristina Bosco, and Viviana Patti. 2023. Haspeede3 at evalita 2023: Overview of the political and religious hate speech detection task. In *Proceedings of the Eighth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2023), CEUR. org, Parma, Italy*, pages 1–8.

Roxana Micu, Carol Luca Gasan, and Vasile Păiș. 2022. Splitting hashtags in romanian micro-blogging texts. In *Proceedings of the 17th Edition of the International Conference on Linguistic Resources and Tools for Natural Language Processing (CONSILR 2022)*, Chișinău, Moldova.

Meike Nauta, Jan Trienes, Shreyasi Pathak, Elisa Nguyen, Michelle Peters, Yasmin Schmitt, Jörg Schlötterer, Maurice van Keulen, and Christin Seifert. 2023. From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable ai. *ACM Comput. Surv.*, 55(13s).

John T Nockleby. 1994. Hate speech in context: The case of verbal threats. *Buff. L. Rev.*, 42:653.

Anil Singh Parihar, Surendrabikram Thapa, and Sushruti Mishra. 2021. Hate speech detection using natural language processing: Applications and challenges. In *2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)*, pages 1302–1308.

María Antonia Paz, Julio Montero-Díaz, and Alicia Moreno-Delgado. 2020. Hate speech: A systematized review. *SAGE Open*, 10(4):2158244020973022.

Fabio Poletto, Valerio Basile, Manuela Sanguinetti, Cristina Bosco, and Viviana Patti. 2021. Resources and benchmark corpora for hate speech detection: a systematic review. *Language Resources and Evaluation*, 55(2):477–523.

Marco Polignano, Pierpaolo Basile, Marco De Gemmis, Giovanni Semeraro, Valerio Basile, et al. 2019. Alberto: Italian bert language understanding model for nlp challenging tasks based on tweets. In *CEUR Workshop Proceedings*, volume 2481, pages 1–6. CEUR.

Anna Schmidt and Michael Wiegand. 2017. A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, pages 1–10, Valencia, Spain. Association for Computational Linguistics.

Shuvam Shiwakoti, Surendrabikram Thapa, Kritesh Rauniyar, Akshyat Shah, Aashish Bhandari, and Usman Naseem. 2024. Analyzing the dynamics of climate change discourse on twitter: A new annotated corpus and multi-aspect classification. *Preprint*.

Marco Antonio Stranisci, Simona Frenda, Mirko Lai, Oscar Araque, Alessandra Teresa Cignarella, Valerio Basile, Cristina Bosco, and Viviana Patti. 2022. O-dang! the ontology of dangerous speech messages. In *Proceedings of the 2nd Workshop on Sentiment Analysis and Linguistic Linked Data*, pages 2–8, Marseille, France. European Language Resources Association.

Surendrabikram Thapa, Kritesh Rauniyar, Farhan Ahmad Jafri, Shuvam Shiwakoti, Hari Veeramani, Raghav Jain, Guneet Singh Kohli, Ali Hürriyetoğlu, and Usman Naseem. 2024. Stance and hate event detection in tweets related to climate activism - shared task at case 2024. In *Proceedings of the 7th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE)*.

Feiyu Xu, Hans Uszkoreit, Yangzhou Du, Wei Fan, Dongyan Zhao, and Jun Zhu. 2019. Explainable ai: A brief survey on history, research areas, approaches and challenges. In *Natural Language Processing and Chinese Computing: 8th CCF International Conference, NLPCC 2019, Dunhuang, China, October 9–14, 2019, Proceedings, Part II 8*, pages 563–574. Springer.