Leveraging Text Generated from Emojis for Hate Speech and Offensive Content Identification

Nkwebi Peace Motlogelwa, Edwin Thuma, Monkgigi Mudongo,
Tebo Leburu-Dingalo and Gontlafetse Mosweunyane

Department of Computer Science, University of Botswana

Abstract

In this paper, team University of Botswana Computer Science (UBCS) investigate whether enriching social media data with text generated from emojis can help in the identification of Hate Speech and Offensive Content. In particular, we build three different binary text classifiers that can detect Hate and Offensive content (HOF) or Not Hate-Offensive content (NOT) on data sampled from Twitter. In building our first classifier, we used pre-processed text from twitter only without emojis. In the second classifier, we enrich our preprocessed text from Twitter with text generated from emojis within the Tweets. Our result suggests that enriching Tweets with text generated from emojis within the Tweets improves the classification accuracy of our hate and offensive content classier.

Keywords

Hate Speech, Binary Classification, fastText, Emojis

1. Introduction

There is a considerable increase on the use of hate speech and offensive content on social media. Since such content can often cause instability to a democracy or even hurt a person who it is directed to, there is now a lot of pressure on social media companies to be able to automatically detect such content. In recent years, there has been an increasing amount of literature on the identification of such content on social media. This is primarily influenced by evaluation campaigns such as HASOC [1], GermEval [2] and OffensEval [3]. Several teams have participated at the aforementioned evaluation campaigns and several studies have confirmed the effectiveness of the latest Bidirectional Encoder Representations from Transformers (BERT) model in the classification of hate speech on social media [4]. In particular, Andrie et al. [5] used the BERT model with a pre-training phase based on German Wikipedia and German Twitter corpora. They then used this to fine-tune on the GermEval competition data. In their evaluation, they reported an F1 score of 76.95. Their BERT based model outperformed other systems that participated at the GermEval-2019 task [2]. In the same vein Bashar et al. [6] proposed a custom Convolutional Neural Network (CNN) architecture built on word embeddings

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

motlogel@ub.ac.ub.bw (N. P. Motlogelwa); thumae@ub.ac.bw (E. Thuma); mudongom@ub.ac.bw (M. Mudongo); leburut@ub.ac.bw (T. Leburu-Dingalo); mosweuny@ub.ac.bw (G. Mosweunyane)
https://www.ub.bw/connect/staff/830 (N. P. Motlogelwa); https://www.ub.bw/connect/staff/1966 (E. Thuma); https://www.ub.bw/user/11462 (M. Mudongo); https://www.ub.bw/connect/staff/202 (T. Leburu-Dingalo); https://www.ub.bw/connect/staff/1379 (G. Mosweunyane)

pre-trained on relevant social media corpus. In their experimental results, they suggest that transfer learning of word embeddings can significantly improve the classification accuracy of hate speech and offensive content. Mishra et al. [7] also used BERT pre - trained transformer based neural network models to fine tune their model. In their work, they utilized BERT implementation present in pytorch-transformers library. Their proposed solution outperformed other participants in the HASOC 2019 shared task [1]. In this paper, we present our proposed solution to the HASOC 2021 shared task English Sub-task A, which is binary classification task [8, 9]. In the aforementioned task, participating system are required to classify Tweets into two classes, namely: Hate and Offensive (HOF) and Non- Hate and offensive (NOT). In our participation, we investigate whether enriching social media text with text generated from emojis can improve the classification accuracy of our binary classifier. Our proposed solution is motivated by the fact that people usually include emojis to accompany the text in order to fill in emotional cues that are missing in the typed messages. For example, one may use an angry face emoji only in their message to depict that they are disgusted and outraged or they can use this message to accompany the typed conversation.

2. Methodology

In this Section, we present our binary text classification approaches for classifying tweets into two classes, namely: Hate and Offensive (HOF) and Non- Hate and offensive (NOT). HOF class signifies that the tweet contains Hate, offensive and profane content. NOT signifies that the tweet does not contain any Hate speech, profane, offensive content. Our proposed binary text classifier used fastText [10]. fastText [1], contributed by Facebook AI Research (FAIR), is an open-source library for efficient text classification and word representation.

2.1. Training Dataset

The training dataset was pre-processed to make it compatible with fastText by moving the labels (HOF or NOT) to the beginning of each sentence and adding __label__ as prefix to each label. Additional pre-processing was then performed on the dataset. In particular, we used the Natural Language Toolkit (NLTK)[2], a suite of libraries and programs for symbolic and statistical natural language processing to stem the text and for stop words removal. The Porter stemming algorithm was used for stemming [11]. In addition, the following pre-processing steps were applied to the dataset mainly to clean the text:

- Removing HTML tags
- Removing URLs
- Converting all cases to lower case
- Hashtags and mentions not removed, as well as punctuations not removed.

The training dataset contains 3843 tweets. Of this, 1342 are not hate speech and 2501 are hate speech. During training, the training dataset was subdivided such that 3043 tweets train our

[1] https://fasttext.cc/

[2] http://www.nltk.org/

classification model and 800 tweets are used for validation. The subdivision was done such that the first 3043 tweets are for training the model, and the last 800 tweets are validation. This was done using standard Linux head and tail commands.

- head -n 3043 en_hasoc_clean.csv > en_hasoc_clean.train
- tail -n 800 en_hasoc_clean.csv > en_hasoc_clean.valid

2.2. Testing Dataset

The same pre-processing done in the training dataset was performed on the test data set, except for pre-processing that deals with labelling the tweets as hate speech or none hate speech.

3. Description of Runs

We submit 3 runs for: Subtask 1A: Identifying Hate, offensive and profane content from the post. Below is a brief description of each run:

3.1. Run 1 - UBCS

This is our baseline run. We used fastText to build a binary classifier for the identification of Hate and Offensive (HOF) and Non - Hate and Offensive (NOT). When building our binary classifier, fastText automatically generated a Tweet vector by averaging the word embeddings for each tweet in the pre-processed training set as features. To train and test our classification model, fastText used multinomial logistic regression [12], which is a linear learner. Before making predictions of the labels for the test dataset using the trained model, fastText also generates feature vectors for the Tweets in the test set using the same techniques used in generating feature vectors for the training set. Both the training and test dataset underwent the same pre-processing steps as described in Section 2.1.

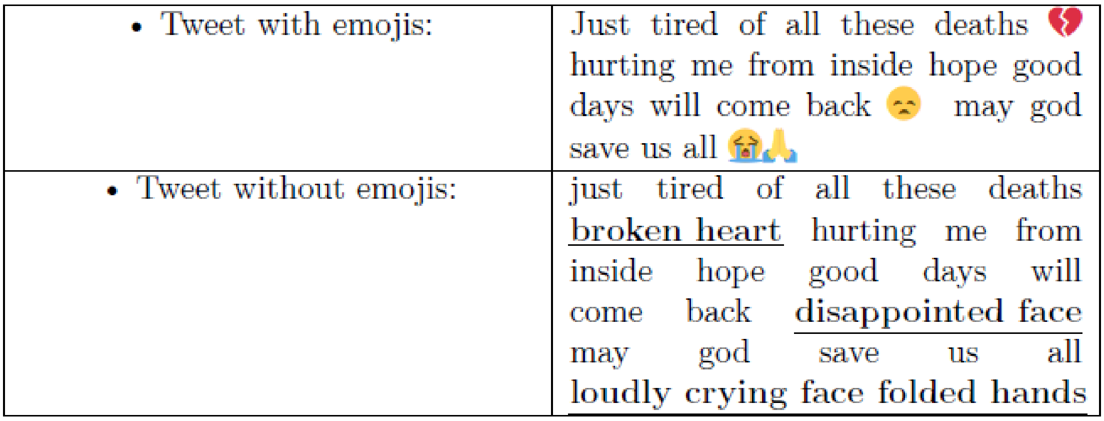
3.2. Run 2 - UBCS

In this run, our aim is to improve the classification accuracy of our binary classifier in our baseline run (**Run 1 - UBCS**) by replacing emojis with text. For our emoji replacement, we used emojis 1.6.1 [3], which is a Python package for converting emoticons to words and vice versa. In particular, we used the *demojize()* function to convert the emojis to text. The pre-processing and emoji removal was applied to both the training dataset and the test dataset. Both the training data and test data were pre-processed as described in Section 2.1. Figure 1 shows emoji replacement.

3.3. Run 3 - UBCS

In this run, our aim was to improve the classification accuracy of our binary classifier after for **Run 2 - UBCS** where both the training data and test data were pre-processed and emojis replaced

[3] https://pypi.org/project/emoji/



• Tweet with emojis:	Just tired of all these deaths 💔 hurting me from inside hope good days will come back 😞 may god save us all 😭🙏
• Tweet without emojis:	just tired of all these deaths broken heart hurting me from inside hope good days will come back disappointed face may god save us all loudly crying face folded hands

Figure 1: Replacing emojis with text

with corresponding text. In particular, we fine tuned the parameters of our classifier in order to improve the performance of our model. Specifically, we explored the following: Learning rate (-lr), number of epochs (-epoch), and maximum length of word ngrams (-wordNgrams). The model that improved on performance was then used to predict labels of the pre-processed test data. This was achieved using this command: *./fasttext supervised -input en_hasoc_clean.train -output model_hasoc_clean_epoch -lr 0.5 -epoch 50 -wordNgrams 2.*

4. Results and Analysis

In this paper, we investigate whether enriching social media tweets with text generated from emojis that accompany the text can improve the classification accuracy of our classifier. Table 1 presents the results of our investigation. Run 1 - UBCS is our baseline run, which does not include text generate from emojis. This baseline run performed poorly compared to the other runs in terms of Macro F1, which was used as the official evaluation measure for the HASOC 2021 binary classification task. Run 2 - UBCS is our best run, with a Micro F1 score of **0.7070**. For this run, we fixed all the parameters used in our baseline run (RUN 1 - UBCS) and then enriched the tweets in the training and testing set with emojis. The results of our investigation suggest incorporating emotions as text from emojis can improve the classification accuracy of hate speech or offensive content on social media. In our third run, we attempted to improve the classification accuracy of our second run (Run 2 - UBCS) using the optimal parameters that gave the best classification accuracy on our training set. In particular, we varied the epoch and the learning rate. However, this resulted in the degradation in the classification accuracy on the test set.

5. Discussion and Conclusion

The most obvious finding to emerge from this study is that we can improve the classification accuracy of our binary classifier for identification of hate speech or offensive content in social

Table 1
English Subtask A - Evaluation Results

Run	Macro F1	Macro Precision	Macro Recall	Accuracy
Run 1 - UBCS	0.6962	0.7282	0.6887	73.54
Run 2 - UBCS	**0.7070**	**0.7292**	0.6998	**74.01**
Run 3 - UBCS	0.7059	0.7087	**0.7022**	72.69

media tweets by enriching the tweets with text generated from emojis. You will recall that evidence from previous studies suggest that BERT based models produce better performance. This is also evidenced by the overall performance of teams that participated in this years task. Further studies need to be carried out in order to validate whether emojis can significantly improve the classification accuracy of a binary classier which is built to identify hate speech or offensive content in social media tweets using BERT based models.

References

[1] S. Modha, T. Mandl, P. Majumder, D. Patel, Overview of the HASOC track at FIRE 2019: Hate speech and offensive content identification in indo-european languages, in: P. Mehta, P. Rosso, P. Majumder, M. Mitra (Eds.), Working Notes of FIRE 2019 - Forum for Information Retrieval Evaluation, Kolkata, India, December 12-15, 2019, volume 2517 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2019, pp. 167–190. URL: http://ceur-ws.org/Vol-2517/T3-1.pdf.

[2] J. M. Struß, M. Siegel, J. Ruppenhofer, M. Wiegand, M. Klenner, Overview of germeval task 2, 2019 shared task on the identification of offensive language, in: Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019), German Society for Computational Linguistics & Language Technology, Erlangen, Germany, 2019, pp. 354–365.

[3] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, R. Kumar, SemEval-2019 task 6: Identifying and categorizing offensive language in social media (OffensEval), in: Proceedings of the 13th International Workshop on Semantic Evaluation, Association for Computational Linguistics, Minneapolis, Minnesota, USA, 2019, pp. 75–86. URL: https://aclanthology.org/S19-2010. doi:10.18653/v1/S19-2010.

[4] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, 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), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. URL: https://aclanthology.org/N19-1423. doi:10.18653/v1/N19-1423.

[5] A. Paraschiv, D.-C. Cercel, Upb at germeval-2019 task 2: Bert-based offensive language classification of german tweets, in: Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019), German Society for Computational Linguistics & Language Technology, Erlangen, Germany, 2019, pp. 398–404.

[6] M. A. Bashar, R. Nayak, Qutnocturnal@hasoc'19: CNN for hate speech and offensive content identification in hindi language, in: P. Mehta, P. Rosso, P. Majumder, M. Mitra

(Eds.), Working Notes of FIRE 2019 - Forum for Information Retrieval Evaluation, Kolkata, India, December 12-15, 2019, volume 2517 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2019, pp. 237–245. URL: http://ceur-ws.org/Vol-2517/T3-8.pdf.

[7] S. Mishra, S. Mishra, 3idiots at HASOC 2019: Fine-tuning transformer neural networks for hate speech identification in indo-european languages, in: P. Mehta, P. Rosso, P. Majumder, M. Mitra (Eds.), Working Notes of FIRE 2019 - Forum for Information Retrieval Evaluation, Kolkata, India, December 12-15, 2019, volume 2517 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2019, pp. 208–213. URL: http://ceur-ws.org/Vol-2517/T3-4.pdf.

[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. K. 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] A. Joulin, E. Grave, P. Bojanowski, T. Mikolov, Bag of tricks for efficient text classification, arXiv preprint arXiv:1607.01759 (2016).

[11] M. F. Porter, An algorithm for suffix stripping, Program 14 (1980) 130–137.

[12] D. Böhning, Multinomial logistic regression algorithm, Annals of the Institute of Statistical Mathematics 44 (1992) 197–200. URL: https://ideas.repec.org/a/spr/aistmt/v44y1992i1p197-200.html. doi:10.1007/BF00048682.