“Feature Selection” with Pretrained-BERT for Hate Speech and Offensive Content Identification in English and Hindi Languages

Surya Agustian[1], Reski Saputra [1] and Aidil Fadhilah[1]

[1] *UIN Sultan Syarif Kasim, Jl. H.R. Soeberantas km 11.5 Panam, Pekanbaru, Riau, Indonesia*

Abstract

The intensive use of social media has led people to express non-formal spoken language, in interactions with others on the internet through text posts. Often, people spill out their annoyance without concern about the use of hate speech, profanity, and abusive language, when is meant to attack and even oppress someone. HASOC 2021 is a shared task that aims to identify hate and abusive content in tweets. In this event, we proposed BERT (and FastText) based transfer learning approach to solve this classification problem. The results obtained by our team UINSUSKA, for English task 1A and 1B, and Hindi task 1A are in the rank 8, 5 and 12 respectively. As for the Hindi task 1B, due to time constraints, our team could not have enough time to develop experiments with BERT, and was ranked 18th for the result using FastText.

Keywords

Hate speech, abusive content, profane words, BERT, FastText, transfer learning

1. Introduction

The differences of personal preference in political, religious, gender, social, cultural and economic backgrounds, often become the source of contention on social media. Abusive and hateful expressions could be made in attacking the interlocutor on social media like Twitter, Facebook, YouTube comments and Instagram. Bullying in a group can also occur against certain person, which is sometime harmful to the person being attacked, so that he/she becomes stressed and depressed, and in some cases lead to suicide [1].

Hate speech, abusive language, profane words, and verbal violence that attack ethnicity, nation, religion, race, or gender are the main factors that are very damaging in social life [2]. They are the cause of hostility to severe bullying on social media [3]. Therefore, these harmful messages must be minimized, filtered and even blocked from social media posts.

Detection of hate speech contents, profane words, and abusive languages in social media has attracted the interest of many researchers around the world in recent years [2, 3, 4, 5, 6]. Various studies and shared tasks show significant progress in English and other languages which have similar language structures [7, 8, 9] in [10].

Some of the most promising detection methods are language models using word embeddings that can recognize word contexts, such as word2vec [11], Glove [12], and FastText [13]. In recent years, language models that have been previously trained on a very large corpus [14], have shown effective results for various NLP tasks, such as question answering, machine translation, automatic summarization, text classification and so on [15]. There are several pre-trained language models such as Universal Language Model Tine-Tuning (ULMFiT) [16], Embeddings from Language Models (ELMo) [17], OpenAI Generative Pre-trained Transformer (GPT) [18], and Google BERT [15].

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

surya.agustian@uin-suska.ac.id (S. Agustian); 11651101881@students.uin-suska.ac.id (R. Saputra);
11651103464@students.uin-suska.ac.id (A. Fadhilah)

With transfer learning, the deep learning model can be used and modified for different NLP tasks [19]. This transfer learning can produce a good result without requiring a training on a large corpus for the new tasks. It often works well by using a small dataset, depends on the task we handle [20].

There are numbers of studies which utilized the pre-trained language model for classification tasks. Among the models that have been studied, BERT and its variants have been reported to produce state-of-the-art performance [4, 10, 17, 18] to be applied on various languages around the globe [20, 21, 22]. For this event, we developed a method implementing transfer learning with BERT [15] and use of FastText language model [14] for classification/detection [23] of hate speech and offensive content (HASOC) in English and Hindi.

The next section of this paper describes the classification task in HASOC 2021, the available data provide by organizer, and then followed by the method we developed to solve it. In the fourth section, the results obtained and analysis are discussed. The last section is the conclusion of this study regarding the results among other participants in HASOC 2021.

2. HASOC Task Description

HASOC 2021 offers two types of classification tasks. The first is hate, offensive, and profane content identification in English, Hindi and Marathi tweets. While the second task is to identify hate and offensive content in tweet conversations in mixed language (English and Hindi). In this event, we only focus on task 1, which is further divided into 2 subtasks [24] specifically as follows.

Subtask 1A: Identifying hate, offensive and profane content from the posts.

Sub-task A is to identify hate speech and offensive language in English, Hindi, and Marathi tweets. It is a coarse-grained binary classification which classify tweets into two classes, namely: Hate and Offensive (HOF) and Non-Hate and offensive (NOT).

Subtask 1B: Discrimination between hate, profane and offensive posts

The goal of this sub-task is a fine-grained identification in English, and Hindi tweets. If the tweets are classified as HOF from the sub-task A, then further classification is conducted to determine if the tweets fall into one of these three categories:

- (HATE) Hate speech: The posts under this class contain hate speech content.
- (OFFN) Offensive: The posts under this class contain offensive content.
- (PRFN) Profane: These posts contain profane words.

Tabel 1. The label distribution of HASOC 2020 [4] and HASOC 2021 [24]

Subtask	Language (year)	Label	Train
1A	English (2021)	NOT	1342
		HOF	2501
	English (2020)	NOT	1852
		HOF	1856
	Hindi (2021)	NOT	3161
		HOF	1433
	Hindi (2020)	NOT	2116
		HOF	847
	Marathi (2021)	NOT	1205
		HOF	669
1B	English (2021)	HATE	683
		PRFN	1196
		OFFN	622
		NONE	1342
	English (2020)	HATE	158
		PRFN	1377
		OFFN	321
		NONE	1852

	Hindi (2021)	HATE	566
		PRFN	213
		OFFN	654
		NONE	3161
	Hindi (2020)	HATE	234
		PRFN	148
		OFFN	465
		NONE	2116

The data available for each sub-task is as shown in Table 1. In developing our system, we also considered using 2020 datasets and utilize them as training and validation data. We combine 2020 and 2021 train-set for training, and using 2020 test-set as validation data to obtain a model which perform the best.

3. System Method

We developed two types of classification system based on word embedding, namely BERT [15] and FastText [13]. The reason behind BERT is that this model has been widely reported to provide state-of-the-art results in various NLP tasks. Since we don't understand the basics of Hindi at all, the FastText model was also developed based on the assumption, that Hindi has some different Language structures from English. For example, like German or Arabic, there are so many phrases (tokens) constructed on several words which are glued altogether without the use of a space as a separator.

In processing the tweet texts, we perform several stages of text preprocessing as follows:

1. Case folding: normalize all tweet texts into lower case
2. Mention handling: transform all mentions into token “@USER”
3. Hyperlink removal: remove all hyperlinks in tweets
4. Emoticon conversion: transform some selected popular emoticons into their text descriptions[2]
5. Punctuation removal: remove all punctuation and special characters in texts
6. Number removal: remove all numbers in texts

3.1. BERT-based method

The architecture of the BERT-based method is shown in Figure 1 for binary classification (Task 1A) and Figure 2 for multi-label classification (Task 1B) as adapted from [25]. We use pre-trained English language model namely BERT-base-uncased [26] with a maximum length (N) of input tweet is 150. While for Hindi we use the pre-trained RoBERTa-hindi-guj-san[3], which was trained on Wikipedia articles in Hindi, Sanskrit and Gujarati.

Before becoming an input sequence in the BERT block as seen in Figure 1 and 2, the text is preprocessed according to the experiment scenario. Then, the output of the BERT is fed to neural network, with the number of input nodes is corresponding to the dimensions of the BERT, which is 768. The sigmoid activation function is applied to produce output in binary classification. As for Task 1B, the outputs of 4 neurons with a sigmoid activation function in each, are converted back into a single class with 4 labels option, i.e., HATE, PRFN, OFFN, and NONE.

The optimizer used for learning is Adam, with the lost function used is L1-norm regularization. Since combination variations of the text preprocessing act as the feature selection for BERT input, the best model is chosen if it has the highest classification accuracy on validation dataset.

[2] https://unicode.org/emoji/charts/full-emoji-list.html#1f4aa
[3] https://huggingface.co/surajp/RoBERTa-hindi-guj-san

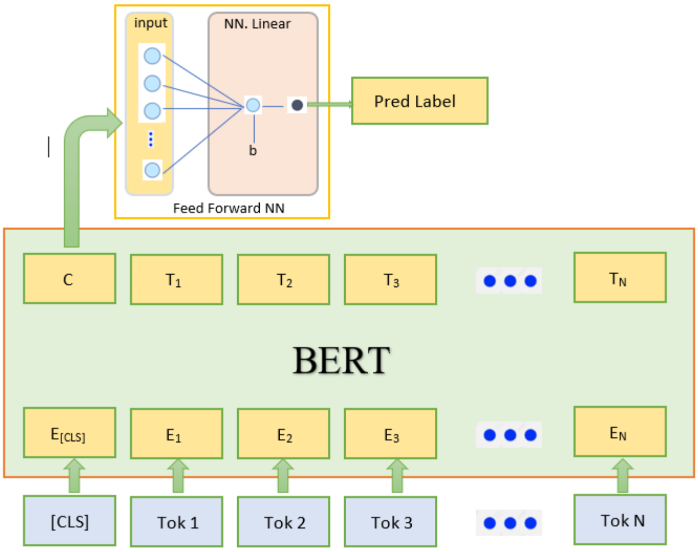

Figure 1. BERT-NN Architecture for Task 1A

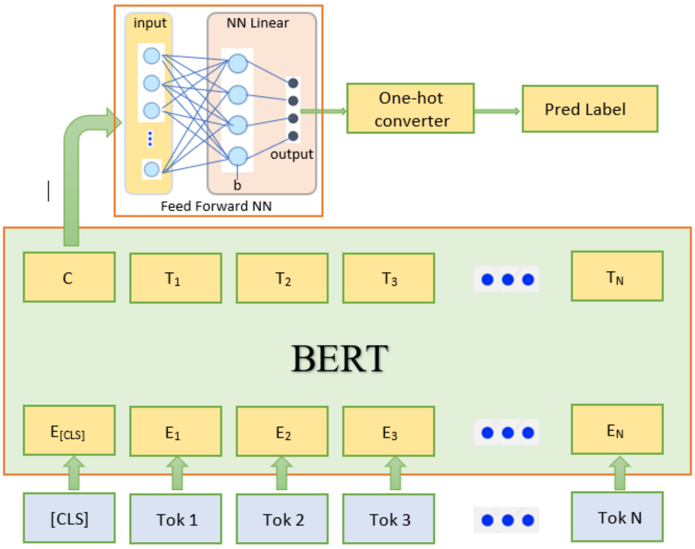

Figure 2. BERT-NN Architecture for Task 1B

3.2. FastText-based method

An alternative model developed for this task is also based on deep learning, namely FastText [13]. It is combined with conventional machine learning: KNN, Logistic Regression and Random Forrest. The classification process for tasks 1A and 1B is carried out in 2 phases, namely the phase of constructing the language model, and the phase of classification.

In phase 1, a separate training process is carried out to generate 128-dimensional word embeddings from sentences in the corpus. Each word should at least have 3 occurrences. Training on English and Hindi with 1000 iterations and window size of 4 is performed to produce FastText word embeddings. The corpus used in these trainings are merged of the 2020 and 2021 HASOC train datasets. The rationale is that the word vector produced should be better than using the 2021 dataset alone, in regards of the size of corpus source. For Hindi, we specifically implement stopword removal, which the stoplist is collected from github[4]. Because the language model is trained within this condition, we also remove stopwords in tweet inputs during classification. Since we utilize Google Colab[5] for all task computations, we did not use a FastText pre-trained model due to resource usage limitation.

The phase 2 is the classification process, which is actually carried out by a conventional machine learning method. It takes input from the FastText language model. A sentence embedding is generated by the vectorizer block, by calculating the resultant norm of the vector of tweet words. For training the ML module, we use the 2021 train-set, or the merge of 2020 and 2021 train-set. As for validation, we use the 2020 testing dataset. The experimental diagram for this FastText-based method can be found in Figure 3.

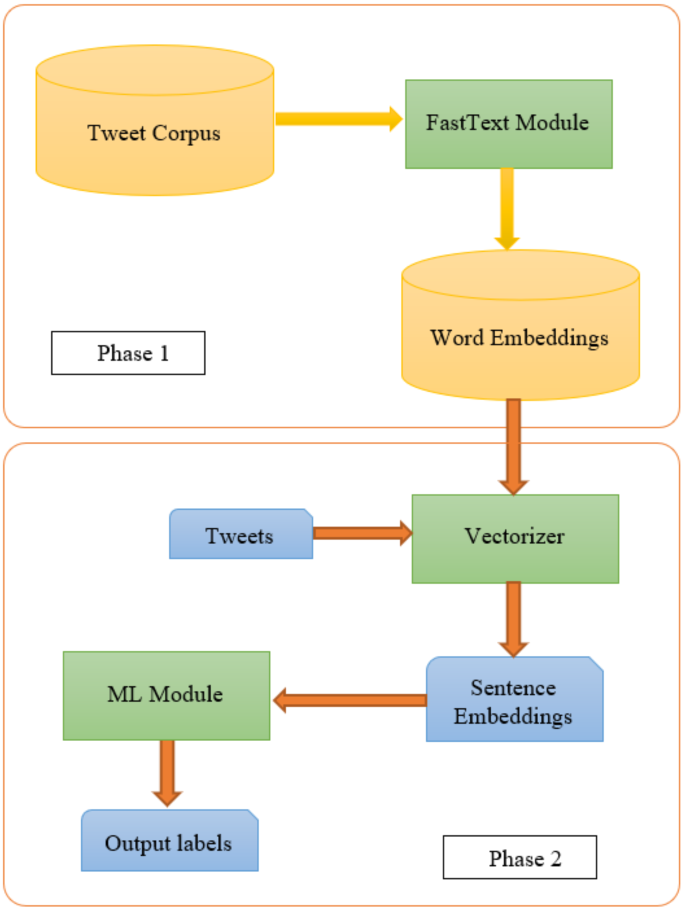

Figure 3. FastText-based system architecture

3.3. Experiment Setup

To get the models of both methods and both tasks that have the best performance, we conducted experiment with scenario, i.e.:

1. The use of mention handling: retain all the mentions or transform into “@USER”
2. The use of Emoticon conversion: transform into word definition text or leave it removed by punctuation removal
3. The use of train-set: 2021 only, or the merge of 2020 and 2021 dataset.
4. The use of stopword removal (for FastText in Hindi only)
5. Variation of machine learning methods in FastText-based system.

[4] https://github.com/stopwords-iso/stopwords-hi

[5] https://colab.research.google.com

4. Results

4.1. Ranked Results

From our submission to the HASOC platform, in general the BERT-based best model has better performance than the FastText-based method. For English task 1A and 1B and Hindi task 1A, we submitted the BERT-based classification results, while for Hindi task 1B, we did not have enough time to complete the BERT training process. Therefore, we only submitted the best model of FastText with Logistic Regression classifier. As for Marathi language, we didn't have time to do experiment at all. The system performance along with our team's ranking can be seen in Table 2 below.

Table 2. Result on Testing 2021 dataset, compared to Rank #1

Language	**Task**	**System**	**Result**			**Compare to Rank #1**	
			Macro F1	Macro Precision	Rank	Macro F1	Macro Precision
English	1A	BERT+NN	0.8024	0.8010	8	0.8305	0.8414
	1B	BERT+NN	0.6417	0.6487	5	0.6657	0.6688
Hindi	1A	BERT+NN	0.7555	0.7784	12	0.7825	0.7862
	1B	FastText+LR	0.4257	0.4864	18	0.5603	0.5873

From the results obtained, we suspect that the lower performances in the Hindi dataset are caused by the drastic imbalance between the portion of NOT (3161) and HOF (1433) label. Moreover, using the entire dataset in training for Hindi task 1B will cause the classification result get worse. It is because the inequality between the NONE label is very large against the HATE, PRFN, OFFN labels, which is around 5.5:1, 14.8:1, and 4.8:1 respectively. We predict that a balancing scheme of training data should be carried out before performing the training process, especially for HINDI Task 1B.

For English, there is also a large discrepancy between the amount of NOT (1342) and HOF(2501) labels. This condition is inversely proportional to Hindi, where the portion of the NOT label is about twice as larger than HOF label. The balancing process should also benefit the training process in regards to improve the classification results, specifically in English task 1B, where the imbalance between the portions of the targeting labels are quite significant, which is around 1:2 between the small amount labels (HATE, OFFN) and the large amount labels (PRFN, NONE).

In terms of the language models, the pre-trained BERT have better text representation (in word vectors) compared to FastText. This is because the training process uses a very large corpus and larger word embeddings size (768). While FastText in this study only uses the HASOC dataset for training, with dimension is set to 128.

4.2. Other Runs

As each team is given 5 runs for each sub task, we also submitted other results based on Naïve Bayes for English task, which is multinomial Naïve bayes with word count vectorizer. In order to seek for the best model, we did some experiments with variation on the word cases (cased or uncased), stopword and punctuation (use or remove), choosing to use or leave as it is. We also considered the length of tokens, which are words (space separated tokens) with minimum 2 characters. We didn’t explore Naïve Bayes on Hindi because we do not have any knowledge about words in Hindi, is it similar with English (space separated) or not.

Table 3 shows unranked runs in our team submission compared to the closest ranks of other teams. These includes Bert-based and FastText-based with certain “feature selection” scenarios. Previously, best model on validation data (test 2020) has been explored on each method with its ‘feature variations’.

For English Task 1A, we did further ‘feature selection’ from the BERT+NN v1 method (Run1, ranked 8), by transforming emoji into text description, and replacing mentioned users in tweet by ‘@USER’. Other settings in BERT+NN v1 are remained untouched, e.g. capital letters are changed into lowercases, all hyperlinks and punctuations are removed, and only use train 2021 dataset for training.

We notice that replacing emoticon into words can reduce the detection accuracy for BERT-based method, as for this work, the F1 score is only 0.7876 (Run4). FastText+KNN and Multinomial NB were submitted as Run2 and Run3 respectively, with F1 scores are 0.7395 and 0.6634.

Table 3. Unranked Runs

Language	**Task**	**System**	**Macro F1**	**Compare to closest ranks**			
				Upper Rank	Macro F1	Lower Rank	Macro F1
English	1A	BERT+NN v2	0.7876	22	0.7894	23	0.7823
		Multinomial NB	0.7395	42	0.7413	43	0.7389
		FastText+KNN	0.6634	53	0.6813	54	0.5999
	1B	Multinomial NB v1	0.5378	30	0.5638	31	0.4969
		Multinomial NB v2	0.5236	30	0.5638	31	0.4969
Hindi	1A	FastText+LR	0.6914	31	0.7181	32	0.6848
		FastText+RF	0.6668	33	0.6762	34	0.6628
		FastText+KNN	0.6435	-	-	-	-
	1B	FastText+LR v2	0.4237	18	0.4257	19	0.4077

As FastText+KNN has lower F1 score than Multinomial NB in Task 1A, for English Task 1B we left FastText unexplored. While BERT+NN method for English Task 1B was still in training process, we develop Multinomial NB with one-versus-all scheme to solve multiclass classification. The result using merged train 2020 and 2021 dataset (v2) is lower than using train 2021 dataset only (v1), but not significant.

For Hindi, we only explore word embedding based method as the input features for machine learning block. Our runs in Hindi Task 1A show that BERT-based embedding has higher result than FastText-based with some ML combinations (i.e., logistic regression, random forest and K-Nearest Neighbor) significantly. For FastText, we only train the small size of tweet data to produce word embeddings. We curious if using pre-trained FastText in Hindi could yield competitive results with BERT-based.

In our experiments, combination of FastText with LR, RF and KNN yield F1 score of 0.6914, 0.6668, and 0.6455 respectively, lower than BERT+NN which is 0.7555. While for Task 1B, we only submit FastText based method in two runs. Both runs use the same methods, only differ in using stopword removal in Run2 (v2), which is lower but not significant.

5. Conclusion

This paper explains the description of the systems participating in hate speech and offensive content identification (HASOC) 2021. In general, the results obtained using BERT-based transfer learning have a good robustness when implemented in different languages, English and Hindi. With the same architecture, and almost the same text preprocessing as feature selection, the BERT-based method for binary classification (task 1A) produces good F1 scores, i.e., 0.8024 and 0.7555 for English and Hindi respectively. As for the multi-label classification, the F1 score obtained for English task 1B is also quite good, i.e., 0.6417, with gap about 0.2 from rank #1. The developed method based on BERT, are ranked 8 of 56 and 5 of 37 for English task 1A and 1B respectively, and got rank 12 of 34 for Hindi task 1A.

References

[1] D. Luxton, J. D. June, & J. M. Fairall, (2012). Social media and suicide: a public health perspective. *American journal of public health, 102 Suppl 2*, S195–S200. doi:10.2105/AJPH.2011.300608

[2] P. Nakov, V. Nayak, K. Dent, A. Bhatawdekar, S.M. Sarwar, M. Hardalov, Y. Dinkov, D. Zlatkova, G. Bouchard, I. Augenstein (2021). Detecting Abusive Language on Online Platforms: A Critical Analysis, *arXiv:2103.00153,* 2021

[3] R. Kumar, AK. Ojha, S. Malmasi, M. Zampieri (2018). Benchmarking Aggression Identification in Social Media. In: *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, ACL 2018. p. 1–11.

[4] T. Mandl, S. Modha, G.K. Shahi, A.K. Jaiswal, D. Nandini, D. Patel, P. Majumder, J. Schäfer (2020). Overview of the HASOC track at FIRE 2020: Hate Speech and Offensive Content Identification in Indo-European Languages, *arXiv:2108.05927v1*, 2021

[5] S. Modha, T. Mandl, G.K. Shahi, H. Madhu, S. Satapara, T. Ranasinghe, M. Zampieri (2021). 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

[6] V. Basile, C. Bosco, E. Fersini, D. Nozza, V. Patti, R. Pardo, F. Manuel, P. Rosso, and M. Sanguinetti, (2019). SemEval-2019 Task 5: Multilingual Detection of Hate Speech Against Immigrants and Women in Twitter, in Basile et al (2019), In: *The 13th International Workshop on Semantic Evaluation*, *Association for Computational Linguistics 2019*, doi:10.18653/v1/S19-2007

[7] P. Fortuna, S. Nunes (2018). A Survey on Automatic Detection of Hate Speech in Text. *ACM Computing Surveys 2018*; 51(4):85:1–85:30. doi:10.1145/3232676

[8] T. Davidson, D. Warmsley, M.W. Macy, I. Weber (2017). Automated Hate Speech Detection and the Problem of Offensive Language. In: *The International AAAI Conference on Web and Social Media ICWSM 2017*

[9] S. Zimmerman, U. Kruschwitz, C. Fox (2018). Improving Hate Speech Detection with Deep Learning Ensembles. In: *Language Resources and Evaluation Conference* (*LREC) 2018*.

[10] S. MacAvaney, H-R. Yao, E. Yang, K. Russell, N. Goharian, O. Frieder (2019). Hate speech detection: Challenges and solutions. *PLoS ONE 14 (8)*: e0221152. doi:10.1371/journal.pone.0221152

[11] T. Mikolov, K. Chen, G. Corrado, J. Dean (2013). Efficient Estimation of Word Representations in *Vector Space, arXiv:1301.3781*

[12] J. Pennington, R. Socher, C. Manning (2014). GloVe: Global Vectors for Word Representation In: The 2014 Conference on *Empirical Methods in Natural Language Processing (EMNLP) ACL* 2014. doi:10.3115/v1/D14-1162

[13] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov (2016). Enriching Word Vectors with Subword Information, *arXiv:1607.04606*

[14] M. Mozafari, R. Farahbakhsh, N. Crespi (2020). A BERT-Based Transfer Learning Approach for Hate Speech Detection in Online Social Media. *Studies in Computational Intelligence, 881 SCI, 928–940*. doi:10.1007/978-3-030-36687-2_77

[15] J. Devlin, M.W. Chang, K. Lee, K. Toutanova, (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In: *The North American Chapter of the Association XIV for Computational Linguistics*: *Human language Technologies* (NAACL-HLT) *2019*

[16] J. Howard, S. Ruder (2018). Universal Language Model Fine-tuning for Text Classification, *arXiv:1801.06146*

[17] M.E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer (2018). Deep contextualized word representations. *arXiv:1802.05365*

[18] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever (2018). Improving language understanding by generative pre-training, *Technical report, OpenAI*

[19] R. Liu, Y. Shi, C. Ji, M. Jia (2019). A Survey of Sentiment Analysis Based on Transfer Learning, *Advanced Optical Imaging for Extreme Environments Vol 7 Special Section*, IEEE Access 2019. doi:10.1109/ACCESS.2019.2925059

[20] E. Bataa, J. Wu (2019). An Investigation of Transfer Learning-Based Sentiment Analysis in Japanese, In: *The 57th Annual Meeting of the Association for Computational Linguistics*, ACL 2019, Italy, July 28 - August 2, 2019

[21] B. Chan, S. Schweter, T. Moller (2020). German's Next Language Model, In: *The 28th International Conference on Computational Linguistics*, Barcelona, Spain (Online), December 8-13, 2020

[22] I.A. Farha, W. Magdy (2021). Benchmarking Transformer-based Language Models for Arabic Sentiment and Sarcasm Detection, In*: The Sixth Arabic Natural Language Processing Workshop*, Kyiv, Ukraine (Virtual), April 19, 2021.

[23] A. Joulin, E. Grave, P. Bojanowski, T. Mikolov (2016). Bag of Tricks for Efficient Text Classification, *arXiv:1607.01759*

[24] 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 FIRE2021: 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/.

[25] H.M. Zahera (2019). Fine-tuned BERT Model for Multi-Label Tweets Classification, In: *The 28th Text REtrieval Conference (TREC) 2019*

[26] J. Devlin, M-W. Chang, K. Lee, K. Toutanova (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, arXiv:1810.04805*