PRHLT-UPV at SemEval-2020 Task 12: BERT for Multilingual Offensive Language Detection

Gretel Liz De la Peña Sarracén and Paolo Rosso
Universitat Politècnica de València, València, Spain
gredela@posgrado.upv.es
prosso@dsic.upv.es

Abstract

The present paper describes the system submitted by the PRHLT-UPV team for the task 12 of SemEval-2020: OffensEval 2020. The official title of the task is Multilingual Offensive Language Identification in Social Media, and aims to identify offensive language in texts. The languages included in the task are English, Arabic, Danish, Greek and Turkish. We propose a model based on the BERT architecture for the analysis of texts in English. The approach leverages knowledge within a pre-trained model and performs fine-tuning for the particular task. In the analysis of the other languages the Multilingual BERT is used, which has been pre-trained for a large number of languages. In the experiments, the proposed method for English texts is compared with other approaches to analyze the relevance of the architecture used. Furthermore, simple models for the other languages are evaluated to compare them with the proposed one. The experimental results show that the model based on BERT outperforms other approaches. The main contribution of this work lies in this study, despite not obtaining the first positions in most cases of the competition ranking.

1 Introduction

BERT, the Bidirectional Encoder Representations for Transformers (Devlin et al., 2019), is a model producing context representations that leverage on language model pre-training. It is based on transformers (Vaswani et al., 2017), which are models that process words in relation to all other words in a sentence, rather than word by word in order. That is, as opposed to directional models, which read the text input sequentially (forward and/or backward), the transformer reads the entire sequence of words at once. This characteristic allows the model to learn the context of a word based on all of its surroundings. Hence, BERT models can consider the full context of a word by looking at the words that come before and after it. This is very useful to understand the intent behind sentences. Therefore, unlike other kinds of models, with BERT models it is possible to effectively capture the general meaning of a sentence by detecting relevant words and their relationship to others.

BERT is a deep, bidirectional, unsupervised language representation, pre-trained using only plain text corpus. The pre-training has been performed by using two strategies on a large corpus of unlabelled text which includes the entire Wikipedia and a book corpus. One training strategy is *masked language model*, where the model attempts to predict the original value of some masked words in a sequence, based on the context provided by the non-masked words in the sequence. The other training strategy is *next sentence prediction*, where the model learns to predict if given a pair of sentences, the second one is the subsequent sentence in the original document. Pre-trained BERT can be fine-tuned to many natural language processing (NLP) tasks by adding an additional output layer on a supervised dataset for a target task. Therefore, it is eliminated the need for engineering a specific architecture for a task. This approach has advanced the state-of-the-art performances in many natural language processing tasks ranging from sequence classification to question answering.

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 1605–1614
Barcelona, Spain (Online), December 12, 2020.

Multilingual BERT (MBERT)[1] (Wu and Dredze, 2019) is a language model pre-trained on Wikipedia text from 104 languages in the same way as BERT for English. Therefore, not only is a contextual model, but the training does not requires supervision. That is, no alignment among the languages is done, but rather in the model the tokens from different languages share an embedding space and a single encoder. There are no cross-lingual objectives specifically designed nor any cross-lingual data, like parallel corpora. However, MBERT produces a representation that seems to generalize well from a cross-lingual perspective for a variety of downstream tasks. Some studies indicate that this model has surprising cross-lingual abilities (Wang et al., 2019).

In this paper, we describe a proposed system that is based on BERT for the Multilingual Offensive Language Identification in Social Media (OffensEval 2020[2]) task (Zampieri et al., 2020). OffensEval 2020 is a shared task organized at SemEval 2020 as extension of the previous OffensEval task 2019 (Zampieri et al., 2019b). The general objective of OffensEval is the identification of offensive language in online social media. This is a relevant topic nowadays, since many users take advantage of the perceived anonymity of this kind of communication to incite offensive behaviors. Basically, the principal aim is to determine whether a text is offensive or not. Moreover, other characteristics taken into account, dividing the task in the next three subtasks:

- **A:** Offensive language identification.
- **B:** Automatic categorization of offense types.
- **C:** Offense target identification.

This time, 5 languages are addressed in the task: English, Arabic, Danish, Greek and Turkish. The 3 previously mentioned subtasks are taken into account for English, while for the rest of the languages, only the subtask A is proposed.

BERT is used as a text feature extractor, using the text representation obtained for the classification of the English texts. In order to perform the classification, a simple feed forward neural network is applied on the top of BERT to detect whether the original text is offensive or not. The rest of the languages are processed in the same way, but using the MBERT model instead.

The paper is organized as follows. Section 2 presents general ideas of related works. Then, Sections 3 and 4 describes the dataset used in the task and the proposed system, respectively. Experiments and results are then discussed in Section 4. Finally, we present our conclusions with a summary of our findings in Section 5.

2 Related Work

Related tasks to abusive language analysis, and particularly the offensive language detection, have attracted significant attention during last years to prevent this kind of online behaviour. This is evidenced by different works (Waseem et al., 2017; Malmasi and Zampieri, 2018; Vidgen and Derczynski, 2020; Tekiroglu et al., 2020), and the organization of different workshops and shared tasks (Kumar et al., 2018; I Orts, 2019; Mandl et al., 2019; Zampieri et al., 2019b; Bosco et al., 2018; Basile et al., 2019).

In general, models in OffensEval 2019 used different approaches, from traditional machine learning such as Support Vector Machines and Logistic Regression, to deep learning such as Convolutional Neural Networks and Recurrent Neural Networks, some of them including attention mechanisms. Moreover, some system employed BERT and reached top-performance in the competition (Zampieri et al., 2019b). In the present work, we propose a system based on BERT and MBERT, analyzing its parameters.

3 Dataset

A multilingual dataset with five languages is provided for the task. Therefore, a corpus of annotated texts have been released for each of the following languages: English (Rosenthal et al., 2020), Arabic

[1] Multilingual Bert Readme Document

[2] https://sites.google.com/site/offensevalsharedtask/home

(Mubarak et al., 2020), Danish (Sigurbergsson and Derczynski, 2020), Greek (Pitenis et al., 2020) and Turkish (Çöltekin, 2020).

The tagset matches the Offensive Language Identification Dataset (OLID) (Zampieri et al., 2019a) that was used in OffensEval 2019. Then, the task is divided into three subtasks according to the tag hierarchy. So that, the 3 subtasks are developed for English, while for the rest of the 4 languages, only the first one is developed.

The subtask A aims to discriminate between offensive and non-offensive text. Therefore, every text is assigned one of the two following labels: Offensive (OFF) and Not Offensive (NOT). On the one hand, offensive texts include insults, threats, and text containing swear words or any form of untargeted profanity, which is unacceptable language. On the other hand, texts that do not contain offense or profanity are considered as not offensive.

The objective of sub-task B is to predict the type of offense, just for those texts labeled as offensive in the subtask A. For this subtask, two labels corresponding to the following categories have been defined: Targeted Insult (TIN) and Untargeted (UNT). The first label corresponds to texts containing an insult or threat and the second one corresponds to texts containing untargeted profanity and swearing.

In sub-task C, the goal is to detect the target of offense, only for those texts labeled as Targeted Insult in the subtask B. In this case, the following three labels have been defined: Individual (IND), Group (GRP) and Other (OTH). Respectively, the first and second labels are for individuals and for groups of people considered as a unity due to a common characteristic. The third label is for the offensive texts that do not belong to any of the previous two labels, such as an organization or an event.

Then, there are 5 possible label combinations for English according the annotation, while for the other languages there are only 2 possible labels.

3.1 Dataset Details

In general, the corpus for each language is made up of tweets in which the user's mentions were replaced by @USER and the URLs have been removed. Other common characteristics in tweets, such as emotions and hashtags were not modified.

Table 1 summarizes the distribution of labels per languages for the subtask A, where a large imbalance is observed between the OFF and NOT classes. Moreover, it should be noted that the corpus in English is very large, with more than 9 million texts, unlike the other languages, among which the largest (Turkish) has only slightly more than 30 thousand texts.

Language	OFF	NOT
English	1446768	7628650
Arabic	1410	5590
Danish	384	2577
Greek	2486	6257
Turkish	6131	25625

Table 1: Labels distribution for subtask A.

subtask A	subtask B	subtask C
OFF: 1446768	TIN: 149550	IND: 152562
		GRP: 24917
		OTH:11494
	UNT: 39424	
NOT: 7628650		

Table 2: Labels distribution for the 3 subtasks in the English corpus.

In addition, Table 2 shows the distribution of labels for the 3 subtasks. As previously discussed, the subtasks B and C are only for English. The number of texts in each class corresponds to the labeling provided for each of the subtasks, where the sum of the number of texts in the UNT and the TIN classes is 188974. This amount does not match the number of texts in the class OFF. Thus, it should be noted that labels were not provided for all texts in the subtask B, so the corpus is much smaller in this case and in the subtask C.

4 System Description

4.1 Preprocessing

The first step in the texts analysis is a preprocessing. In this step the English tweets are cleaned. Firstly, misspelled words are corrected with the support of the *TextBlob*[3] tool. We think it is an important step since many users tend to misspell words and this can lead to a large number of elements outside of vocabularies used for the texts analysis. Another process is the analysis of emojis. We replaced each emoji with a phrase that describes its meaning. For this purpose we used the *emoji*[4] tool.

4.2 Features

The feature analysis of the texts has been included in the system. The aim is to use the information for discrimination between classes. The first group of features is based on some texts *basic properties (B_prop)*: (i) the length of the tweets (L), (ii) the number of misspelled words (MW), as well as iii) the use of punctuation marks (PM). For the misspelled words analysis, the same tool used for the preprocessing of the texts has been used. In this way, this feature is analyzed before the preprocessing where the texts are corrected. In fact, the preprocessing of each text is performed after obtaining a vector in which the components correspond to the values of the features from to the text. The case of the feature corresponding to punctuation marks is the number of times that one of the signs in the set {?! '[...]} is used in the text, which indicate exclamation, question or omission of phrases. The element [...] corresponds to a sequence of more than one dot.

Another group of features analyzed is based on *semantic properties (S_prop)* present in the texts: (i) the use of emoticons (E), as well as (ii) the noun phrases (NP). In the emoticons analysis, the same tool for the preprocessing of emoticons is used. In this case, a vector is constructed with the emoticons present in a text. The representation in this vector space is based on TF-IDF and the dimensionality of the vectors is reduced by using the Principal Component Analysis (PCA) technique. Then, it is added to this vector a last component indicating the number of emoticons in the original text. The resulting vector is the feature corresponding to the emoticons analysis. A similar process is carried out to obtain the feature corresponding to the noun phrases set present in each text. In this case, instead of obtaining the emoticon set, the noun phrases set is extracted with the *TextBlob* tool. The resulting vector is combined with the vector obtained with the emoticons analysis to obtain the second group of properties.

All features are used for English texts analysis. Hence, the feature vector (F_vector) corresponds to the equation 1, where $[.,.]$ represents the concatenation operation.

$$\begin{aligned} B\_prop &= [[L, MW], PM] \\ S\_prop &= [E, NP] \\ F\_vector &= [B\_prop, S\_prop] \end{aligned} \tag{1}$$

For the rest of the languages not all the features are analyzed. Only the length of the tweets and the analysis of emoticons are taken into account.

4.3 Method

The general architecture of the proposed system is showed in Figure 1. This architecture has been used for the 3 subtasks defined for English. The size of the output is the only parameter that varies, since for the subtask C there are three possible labels instead of two as in the other two subtasks.

The system consists of a BERT based model at the text level. This model is used as an embedding generator from the text. Hence, a vector representation (BERT_out) is obtained given a text. Basically, the vector is the output of the special token [CLS] included in the processing in BERT. Afterward, this vector is concatenated with the features vector obtained before, and a normalization layer is applied to the result. Finally, the vector is fed to a softmax layer to predict the output as equation 2 indicates.

$$\phi = \sigma(W^s \cdot \eta([BERT\_out, F\_vector]) + b^s) \tag{2}$$

[3] https://textblob.readthedocs.io/en/dev/

[4] https://github.com/carpedm20/emoji/

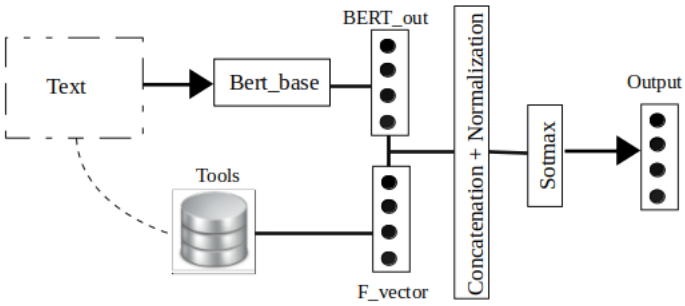

Figure 1: General system architecture

where σ is the softmax function and η represents the normalization layer. $W^s \in \mathbb{R}^{N_c*(B+F)}$ and $b^s \in \mathbb{R}^{N_c}$ are the parameters for the softmax layer. N_c is the size of the model output, corresponding to the number of possible labels analyzed for the model. B and F are the dimensions of the BERT output and features vector, respectively. Finally, cross entropy is used as the loss function, defined as equation 3, where y_i is the true classification of a text i.

$$L = -\Sigma_i \; y_i * log(\phi_i) \tag{3}$$

4.3.1 BERT

BERT is a model based on transformer that applies an attention mechanism to learn contextual relations between words in a text. A transformer model includes an encoder that reads the text and a decoder that produces a prediction. The objective of BERT is to generate a language model, therefore it only uses the encoder mechanism. Hence, the entire sequence of words is read at once in BERT, so that the model is no directional. This characteristic allows the model to learn the context of a word based on all of its surroundings. For the task at hand, this is important, since offensive language is often not expressed only with certain words, but in the entire context of the text.

Basically, BERT is a stack of a number L of encoders, identical in structure but without sharing weights. Each encoder is divided into two sub-layers: a multi-head attention and a feed-forward neural network. Moreover, each sub-layer has a residual connection around it, and is followed by a layer-normalization step.

In general, the input first flows through a multi-head attention, which helps the encoder look at other words in the input sentence as it encodes a specific word. The multi-head attention consists of a given number A of self-attention mechanisms, which are combined to obtain the result. Then, the outputs of this sub-layer are fed into a feed-forward neural network. The same feed-forward network is independently applied to each position.

In conclusion, every encoder layer does some computation on the output of the previous layer, or on the input representation for the first layer, to create a new representation. The input is a sequence of tokens, corresponding to each word or sub-words from the text and including the specials tokens [CLS] and [SEP]. The token [CLS] is added at the beginning of the tokens sequence, and [SEP] at the end of each sentence. The output is a sequence of vectors of a given size H, in which each vector corresponds to an input token with the same index.

The proposed system uses the BERT-base model, where L = 12, H = 768 and A = 12. Therefore, the output of BERT in our case is the output vector of the [CLS] token in the layer 12.

4.3.2 RoBERTa and ALBERT

RoBERTa is a model that modifies some of the hyperparameters in BERT. The train follows the architecture of the BERT-Large, that is L = 24, H = 1024 and A = 16. It removes one task from the pre-training of BERT (next sentence prediction) and introduces dynamic masking so that the masked token changes during the training epochs (Liu et al., 2019).

ALBERT is a model with state-of-the-art results in many tasks. It is much lighter and smarter than BERT. The changes allow both outperform and dramatically reduce the model size. This model improves parameter efficiency by sharing all parameters across all layers. Feed forward network parameters and attention parameters are all shared (Lan et al., 2020).

In our proposal for English we have tested the substitution of BERT for each of these variants. Therefore, 3 different systems were studied for the task depending on the model used.

4.4 Multilingual Approach

In the analysis of the Arabic, Danish, Greek and Turkish languages, the architecture presented above was used, including the features vector extracted as was explained in the previous section. The main difference lies in the model used to obtain the vector of text embeddings. In this case the MBERT model is used, which has been trained with 104 languages in the same way as BERT for English.

In the model the tokens from different languages share an embedding space and a single encoder. There are no cross-lingual objectives specifically designed nor any cross-lingual data, like parallel corpora. However, MBERT produces a representation that seems to generalize well a cross languages for a variety of tasks.

5 Experiments and Results

The experiments are carried out with the 10-fold cross validation stratified technique. The measure is macro F1-score, according to the one used for the ranking of the systems in the competition, for each language and subtask.

Due to the large size of the English corpus, we select a sample subset for the analysis of the subtask A. Thus, the experimental results presented in this paper have been obtained with a subset of the original texts in the case of the subtask A for English. We followed two strategies to randomly take 1,000,000 texts. Firstly, the texts were selected keeping, in the subset, the same class proportion in the original corpus. The other strategy was to take the same amount of texts from both classes. The best results were obtained with the second strategy and are those shown in the tables that are discussed later. We use random, a python library, with seed 4 for the selection, taking the necessary number of elements per class. In the second case we select 500,000 texts from each class.

5.1 Our Baselines

We used four models as baselines to evaluate the proposed model. These models are based on traditional machine learning methods and the others are deep leaning models. One one side, we used Support Vector Machines (**SVM**) and Logistic Regression (**LR**). The parameters were selected by optimization with the GridSearchCV[5] tool from the sklearn library. On the other hand, we used a Convolutional Neural Network (**CNN**) with a convolutional layer of 32 filters of 3x3 and a maxpooling layer of 2x2. Moreover, we employed a Bidirectional LSTM network (**BiLSTM**), where the number of units is 64 and the FastText[6] words embeddings were used for text representation.

5.2 Implementation Details

For all the models, we use the same batch size of 50 instances in the training with 20 epochs. The number of BERT layers trained for fine-tuning was 5. For the baselines, the representation based on TF-IDF word ngrams was used for the texts.

5.3 Results

Table 3 shows a summarization of the experimental results obtained for English. In the proposed model (Proposal) all the features are taken into account and BERT is used. The other systems correspond to the baselines and different variants of the proposal that were evaluated. First, we can check the superiority

[5] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

[6] https://fasttext.cc/

of the proposal compared to the baselines. Among the baselines, the model with the best performance is BiLSTM, which obtains close values to the proposal but does not exceed them. Moreover, the table shows a comparison of the results based on the use of the features sets. Comparing the results of the proposal with the version where the features are not used (None), it can be seen that in general there is an improvement with the features. Furthermore, when analyzing the results obtained by using each features set separately, it can be seen that the main contribution lies in the semantic properties (S_prop). Regarding the use of ALBERT and RoBERTa, very similar results are obtained to those obtained with BERT. We expected better results with ALBERT, but it may be because we have used ALBERT-base in the experiments, possibly the results will improve by using a larger model such as ALBERT-large.

Model	**Subtasks**		
	A	**B**	**C**
Proposal	**0.9496**	0.8587	0.7290
Baselines			
SVM	0.8825	0.8073	0.6140
LR	0.8925	0.7901	0.6095
CNN	0.8910	0.8109	0.6007
BiLSTM	0.9204	0.8421	0.6936
Feature analysis			
B_prop	0.9281	0.8387	0.6998
S_prop	0.9482	0.8401	0.7235
None	0.9048	0.8312	0.6934
Varying the model based on BERT			
RoBERTa	0.9460	**0.8596**	**0.7325**
ALBERT	0.9435	0.8589	0.7103

Table 3: Macro F1 for English

Table 4 shows the results for languages other than English. On the one hand, we can see that the features are not very relevant, since the difference in the results is not significant with respect to those obtained with the model where the features are not used. For Arabic, Greek and Turkish, the proposed model achieves better results with respect to the baselines as well as in English. The difference is that in some of these cases the best results among the baselines is not always for BiLSTM. This is the case of the Greek and Turkish, where the best baseline is LR and SVM respectively. An interesting result is that in the case of Danish, the model based on SVM performs better than the proposed system.

Model	**Languages**			
	Arabic	**Danish**	**Greek**	**Turkish**
Proposal	**0.8064**	0.7048	**0.7350**	0.7218
Baselines				
SVM	0.6912	**0.7258**	0.7295	0.7033
LR	0.6770	0.7066	0.7319	0.6633
CNN	0.7343	0.6503	0.6782	0.6675
BiLSTM	0.7556	0.6620	0.7049	0.6932
Feature analysis				
None	0.8049	0.7010	0.7298	**0.7225**

Table 4: Macro F1 for Arabic, Danish, Greek and Turkish

5.4 Error Analysis

This section briefly presents an error analysis in the subtask A for English. We try to show an idea of possible errors in general. The main type of error are false negatives regarding the class of offensive texts, even when a balanced dataset is used. An example of a false negative is the next:

"Hate is heavy. Don't let it consume you. Just let it go."

In this case, the offense is not explicitly, but the writer is implicitly calling as hateful the target user. This type of phenomenon is more difficult to deal with.

5.5 Results on the Test Set

Tables 5 and 7 summarize the results obtained in the test set. The number of participants for English was 85, 43 and 39 for the subtasks A, B and C respectively, where our system reached the positions 27, 18 and 3. In all cases the proposed system is positioned in the first half of the overall ranking of the competition, achieving the third position in the case of the subtask C. This result is not obtained in the rest of the languages, where the results do not rank among the first positions.

Model	**Subtask A**		**Subtask B**		**Subtask C**	
	Position	**Macro F1**	**Position**	**Macro F1**	**Position**	**Macro F1**
Best system	1	0.9222	1	0.7418	1	0.7145
Our proposal	27	0.9097	18	0.5987	3	0.6692
Last system	85	0.0728	43	0.2777	39	0.0574

Table 5: Summary of the results in the test set for English

Model	**Arabic**		**Danish**		**Greek**		**Turkish**	
	Pos	**Macro F1**	**Pos**	**Macro F1**	**Pos**	**Macro F1**	**Pos**	**Macro F1**
Best system	1	0.9017	1	0.812	1	0.852	1	0.8258
Our proposal	42	0.7868	32	0.637	26	0.776	35	0.7127
Last system	53	0.44512	39	0.491	37	0.269	46	0.3109

Table 6: Summary of the results in the test set for Arabic, Danish, Greek and Turkish

Model	**Arabic**		**Danish**		**Greek**		**Turkish**	
	Pos	**Macro F1**	**Pos**	**Macro F1**	**Pos**	**Macro F1**	**Pos**	**Macro F1**
Best system	1	0.9017	1	0.812	1	0.852	1	0.8258
Our proposal	42	0.7868	32	0.637	26	0.776	35	0.7127
Last system	53	0.44512	39	0.491	37	0.269	46	0.3109

Table 7: Summary of the results in the test set for Arabic, Danish, Greek and Turkish

6 Conclusion

In this work, we studied the problem of Multilingual Offensive Language detection taking part in the OffensEval shared task of SemEval 2020. We proposed a system for each subtask and language that is based on BERT. Feature analysis is included in the system and its contribution to the improvement in the performance was validated with the experiments. Furthermore, we evaluated the use of ALBERT and RoBERTa, but no significant improvement was obtained with them. We achieved a good ranking position in English regarding the subtask C. However, the results for the rest of the English subtasks and for the rest of the languages were not satisfactory.

Acknowledgements

The research work was partially funded by the Spanish MICINN under the project MISMIS-FAKEnHATE on Misinformation and Miscommunication in social media: FAKE news and HATE speech (PGC2018-096212-B-C31).

References

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 2019. Semeval-2019 Task 5: Multilingual Detection of Hate Speech against Immigrants and Women in Twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 54–63.

Cristina Bosco, Dell'Orletta Felice, Fabio Poletto, Manuela Sanguinetti, and Tesconi Maurizio. 2018. Overview of the EVALITA 2018 Hate Speech Detection Task. In *EVALITA 2018-Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian*, volume 2263, pages 1–9. CEUR.

Çağrı Çöltekin. 2020. A Corpus of Turkish Offensive Language on Social Media. In *Proceedings of the 12th International Conference on Language Resources and Evaluation*. ELRA.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Òscar Garibo I Orts. 2019. Multilingual Detection of Hate Speech Against Immigrants and Women in Twitter at SemEval-2019 Task 5: Frequency Analysis Interpolation for Hate in Speech Detection. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 460–463.

Ritesh Kumar, Atul Kr Ojha, Shervin Malmasi, and Marcos Zampieri. 2018. Benchmarking Aggression Identification in Social Media. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 1–11.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *International Conference on Learning Representations*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized Bert Pretraining Approach. *arXiv preprint arXiv:1907.11692*.

Shervin Malmasi and Marcos Zampieri. 2018. Challenges in Discriminating Profanity from Hate Speech. *Journal of Experimental & Theoretical Artificial Intelligence*, 30(2):187–202.

Thomas Mandl, Sandip Modha, Prasenjit Majumder, Daksh Patel, Mohana Dave, Chintak Mandlia, and Aditya Patel. 2019. Overview of the Hasoc Track at Fire 2019: Hate Speech and Offensive Content Identification in Indo-European Languages. In *Proceedings of the 11th Forum for Information Retrieval Evaluation*, pages 14–17.

Hamdy Mubarak, Ammar Rashed, Kareem Darwish, Younes Samih, and Ahmed Abdelali. 2020. Arabic Offensive Language on Twitter: Analysis and Experiments. *arXiv preprint arXiv:2004.02192*.

Zeses Pitenis, Marcos Zampieri, and Tharindu Ranasinghe. 2020. Offensive Language Identification in Greek. In *Proceedings of the 12th Language Resources and Evaluation Conference*. ELRA.

Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Marcos Zampieri, and Preslav Nakov. 2020. A Large-Scale Semi-Supervised Dataset for Offensive Language Identification. In *arxiv*.

Gudbjartur Ingi Sigurbergsson and Leon Derczynski. 2020. Offensive Language and Hate Speech Detection for Danish. In *Proceedings of the 12th Language Resources and Evaluation Conference*. ELRA.

Serra Sinem Tekiroglu, Yi-Ling Chung, and Marco Guerini. 2020. Generating Counter Narratives against Online Hate Speech: Data and Strategies. *arXiv preprint arXiv:2004.04216*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Bertie Vidgen and Leon Derczynski. 2020. Directions in Abusive Language Training Data: Garbage In, Garbage Out. *CoRR*, abs/2004.01670.

Zihan Wang, Stephen Mayhew, Dan Roth, et al. 2019. Cross-Lingual Ability of Multilingual BERT: An Empirical Study. *arXiv preprint arXiv:1912.07840*.

Zeerak Waseem, Thomas Davidson, Dana Warmsley, and Ingmar Weber. 2017. Understanding Abuse: A Typology of Abusive Language Detection Subtasks. *arXiv preprint arXiv:1705.09899*.

Shijie Wu and Mark Dredze. 2019. Beto, Bentz, Becas: The Surprising Cross-Lingual Effectiveness of BERT. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 833–844. Association for Computational Linguistics.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019a. Predicting the Type and Target of Offensive Posts in Social Media. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 1415–1420.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019b. SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffensEval). In *Proceedings of The 13th International Workshop on Semantic Evaluation (SemEval)*.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media (OffensEval 2020). In *Proceedings of SemEval*.