BRUMS at SemEval-2020 Task 12: Transformer based Multilingual Offensive Language Identification in Social Media

Tharindu Ranasinghe[1], **Hansi Hettiarachchi**[2]
[1]Research Group in Computational Linguistics, University of Wolverhampton, UK
[2]School of Computing and Digital Technology, Birmingham City University, UK
t.d.ranasinghehettiarachchige@wlv.ac.uk
hansi.hettiarachchi@mail.bcu.ac.uk

Abstract

In this paper, we describe the team *BRUMS* entry to OffensEval 2: Multilingual Offensive Language Identification in Social Media in SemEval-2020. The OffensEval organizers provided participants with annotated datasets containing posts from social media in Arabic, Danish, English, Greek and Turkish. We present a multilingual deep learning model to identify offensive language in social media. Overall, the approach achieves acceptable evaluation scores, while maintaining flexibility between languages.

1 Introduction

Social media has become a normal medium of communication for people these days as it provides the convenience of sending messages fast from a variety of devices. Unfortunately, social networks also provide the means for distributing abusive and aggressive content. Given the amount of information generated every day on social media, it is not possible for humans to identify and remove such messages manually, instead it is necessary to employ automatic methods. As offensive language becomes pervasive in social media, scholars and companies have been working on developing systems capable of identifying offensive posts, which can be set aside for human moderation or permanently deleted (Risch and Krestel, 2018).

Along with these studies, a few shared tasks have been organized on detecting offense in social media, such as HatEval (Basile et al., 2019), HASOC (Mandl et al., 2019), TRAC (Kumar et al., 2018) and OffensEval (Zampieri et al., 2019b) co-located with SemEval 2019. In semeval 2020, OffensEval returns for the second time with multilingual offensive language identification in social media. OffensEval 2 focuses on several languages Arabic, Danish, English, Greek and Turkish motivating participants to submit multilingual offensive language identification systems.

This paper revisits the problem of offensive language identification describing our submission to the SemEval-2020 Task 1 (Zampieri et al., 2020). The remainder of this paper is structured as follows: Section 2 describes the related work done in the field of aggression detection, Section 3 has a description of the dataset. Section 4 describes the system that was submitted, split into a description of how the data was processed and the architectures of the classifiers that were used. Section 5 presents an analysis of the results of our evaluation of the different architectures, as well as of the final submission. Finally, Section 6 offers some final remarks and a conclusion.

2 Relevant Work

The majority of text classification approaches in early research work were supported by traditional or feature-based supervised learning methods. Following this tradition, Malmasi and Zampieri (2017) proposed a linear Support Vector Machine (SVM) classifier for hate speech detection in social media. The best accuracy of this approach was obtained by using character n-grams as model features. Another research suggested stacked ensemble system using logistic regression and random forest-based classifiers

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 1906–1915
Barcelona, Spain (Online), December 12, 2020.

(Montani, 2018). For initial layer of classifiers, they used character n-grams, token n-grams, important tokens and word2vec embeddings. The Outputs of initial layer classifiers were used as the inputs of next classifier layer which produces the final result. This approach obtained the best results in GermEval 2018 (Wiegand et al., 2018) by outperforming the deep neural network (DNN) models such as convolutional neural networks (CNN) (von Grünigen et al., 2018) and combination of bi-directional Long-Short term memory and CNN (BiLSTM-CNN) (Wiedemann et al., 2018).

However, most of the recent research conducted in this area prove that DNN-based methods can outperform traditional supervised learning methods (Zampieri et al., 2019a; Modha et al., 2018). But, unlike the traditional methods, DNNs require sufficient amount of training data to adjust their weights properly without overfitting. Focusing on this issue, Modha et al. (2018) found that LSTM with higher dropout (≈ 0.5) can handle overfitting problem. In order to increase the model performance using a high number of training instances in a simple manner, the available data sets related to aggression and cyberbullying detection can be merged (Fortuna et al., 2018). But, when focused on a particular area, it cannot guarantee the availability of multiple data sets. Another research suggested data augmentation and pseudo labelling approaches to increase the amount of training data (Aroyehun and Gelbukh, 2018). As the data augmentation strategy, they proposed to translate each instance into an intermediate language and back to the original language. For pseudo labelling approach, they trained some initial models using available training data and picked the best model to label new data. A LSTM model trained using the augmented data set received the first place in Trolling, Aggression and Cyberbullying (TRAC-2018) shared task designed for Facebook English data set. In the same shared task, for the platform unnamed English data set, third place could be obtained by a CNN-LSTM model trained using the combination of augmented and pseudo labelled data sets.

In addition to approaching data set increasing methods, transfer learning techniques also used by previous research to overcome the data limitation issue. Wiedemann et al. (2018) showed that transfer learning can be applied using some labelled data set with relevant labels or unlabelled data set with unsupervised user clusters or topics. Further, as mentioned in SemEval-2019 OffensEval leader-board, the top 5 solutions for offensive language identification are BERT-based and they proved the effectiveness in transfer learning as well as Transformers (Zampieri et al., 2019b). The top solution of this shared task fined tuned a BERT model using the training data and found that it outperforms a LSTM model with higher dropout (Liu et al., 2019a).

In summary, majority of recent research in this area were supported by DNN models and among them, CNN, LSTM and BERT-based models were found to be most effective and commonly used. When focused on the input data representation, there was a tendency to use fastText embeddings to handle the high availability of spelling mistakes and new words in social media (Aroyehun and Gelbukh, 2018; Modha et al., 2018; Wiedemann et al., 2018; Zampieri et al., 2019a). Following this tendency, in this research, we further experimented the effectiveness of DNN models including some recently published transformers for offensive language identification.

3 Dataset

OffensEval 2020, the SemEval 2020 Offensive Language Detection Shared Task, does not provide a new manually labeled training set for the English language (Zampieri et al., 2020). The competitors were recommended to use the Offensive Language Identification Dataset (OLID) which was released for the the SemEval 2019 Offensive Language Detection Shared Task (Zampieri et al., 2019b). The OLID is a hierarchical dataset to identify the type and the target of offensive texts in social media. The dataset is collected on Twitter and publicly available. There are 14,100 tweets in total, in which 13,240 are in the training set, and 860 are in the test set. For each tweet,there are three levels of labels:(A) Offensive/Not-Offensive, (B) Targeted-Insult/Untargeted, (C) Indi-vidual/Group/Other. For English subtask organisers provided an additional dataset similar to OLID (Zampieri et al., 2019b), it still has three levels, but this time only confidence scores, generated by different models, are provided rather than human annotated labels. In addition, the data in level A is separated from levels B and C. In level A, there are 9,089,140 tweets,in levels B and C, there are different 188,973 tweets.

Language	training	dev	test
Arabic	7000	1000	2000
Danish	2961	-	330
Greek	8744	-	1545
Turkish	31757	-	3529

Table 1: Number of rows in Training set(training), Development set(dev) and Test set(test) in each language. ”-” denotes that no data was available.

For non-English languages the organisers released Multilingual Offensive Language Identification Dataset (MOLID). It contains four languages: Arabic (Mubarak et al., 2020), Danish (Sigurbergsson and Derczynski, 2020), Greek (Pitenis et al., 2020), and Turkish (Çöltekin, 2020). Table 1 contains information about number of rows that the training, dev and test datasets had in each language (Zampieri et al., 2020).

4 Methodology

In this section we present the methodology employed for the shared task. All the languages followed a common methodology which had two steps: Data Preprocessing and Machine Learning.

4.1 Data Preprocessing

As mentioned previously, the data preprocessing for this task was kept fairly minimal to make it portable for all the languages. More specifically, we perform only four specialised tasks for this data, followed by tokenisation. The tasks include removing usernames, removing *URL*s, converting all the emojis to text and converting all tokens to lower case.

First, we completely remove all usernames from the tweets, without inserting a placeholder. In the SemEval-2020 dataset, mention of a certain user is represented by *@USER*. We removed all strings containing *@USER* using a regular expression. The reasoning behind this step is mainly to remove noisy text. Then we remove all *URL* mentions in the tweets. In the SemEval-2020 dataset, mention of a certain URL is represented by *URL*. We removed all strings containing *URL* using a regular expression. The reasoning behind this step too is mainly to remove noisy text. Emojis play an important role in showing aggression in social media (Hettiarachchi and Ranasinghe, 2019). Since we can't guarantee that the pretrained embedding models we use will have embeddings for emjois, we converted all the emojis to text using a third party python library named *Emoji* [1]. We only conducted this step for English, since the *Emoji* library does not support other languages.

For Arabic language, we used a separate preprocessing step. Twitter users writing in Arabic can either write in Modern Standard Arabic (MSA) or in their particular dialects. If the tweet is written in MSA, it may or may not include diacritics whereas dialectical Arabic does not include any. Offensive tweets can be formulated in any of these different versions of the Arabic transcript. Thus, in order to avoid false classifications due to a non-defining feature, we used regular expressions to delete diacritics.

Final prepossessing step is only applied to the architectures that used character embeddings. The fastText pretrained character embedding models that we used only contain lower-cased letters. Therefore, we convert the text to lower case letters. However, for the Transformer architecture, we did this preprocessing step only if the model that we used is uncased.

4.2 Machine Learning

In order to determine the most suitable neural network architecture for the task, we experimented with three different neural architectures: Convolutional Neural Network (CNN) (Kim, 2014), Recurrent Neural Network (RNN) (Cui et al., 2018) and Transformer (Devlin et al., 2018).

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

4.2.1 Convolutional Neural Network (CNN)

CNN is a class of deep, feed-forward artificial neural networks that uses a variation of multilayer perceptrons (Kim, 2014). CNNs are generally used in computer vision, however they have recently been applied to various NLP tasks and the results were promising (Kim, 2014; Hughes et al., 2017). In a CNN result of each convolution will fire when a special pattern is detected. By varying the size of the kernels and concatenating their outputs, we can detect patterns of multiples sizes (2, 3, or 5 adjacent words) (Kim, 2014). Patterns could be expressions like "fuck it", "stupid cunt" and CNNs can identify them in the sentence regardless of their position. Since offense is mostly a word pattern, we assumed that CNNs would be a good architecture to detect offensive sentences.

For this architecture we used fasttext character embeddings (Bojanowski et al., 2016). After the embedding layer we used four convolution layers with 1,2,3 and kernal sizes followed by a max pooling layer, (Scherer et al., 2010). Finally the outputs of the max pooling layers were concatenated and passed through a linear layer. For all the languages in the task, we used fasttext character embeddings released by facebook (Grave et al., 2018) [2].

4.2.2 Recurrent Neural Network (RNN)

RNNs are designed to make use of sequential data, when the current step has some kind of relation with the previous steps. (Cui et al., 2018). This makes them ideal for applications with a time component (audio, time-series data) and natural language processing (Pitenis et al., 2020). RNNs perform very well for applications where sequential information is clearly important, because the meaning could be misinterpreted if sequential information is not used. Since sequential information is important to identify offensive sentences, we used the RNN architecture for this task. For this architecture too we used fasttext character embeddings (Bojanowski et al., 2016) trained on English Wikipedia text. After the embedding layer we used a RNN layer. We experimented with LSTM (Hochreiter and Schmidhuber, 1997), bi-directional LSTM (Graves et al., 2005), GRU (Chung et al., 2014) and bi-directional GRU (Chung et al., 2014). The output of the RNN layer was followed a linear layer finally. For this architecture also we used fasttext character embeddings released by facebook.

4.2.3 Transformers

With the introduction of BERT (Devlin et al., 2018) transformer architectures have shown a massive success in wide range of NLP tasks. Transformer architectures have been trained on general tasks like language modelling and then fine-tuned for classification tasks. (Sun et al., 2019; Ranasinghe et al., 2019).

Transformer models take an input of a sequence and outputs the representation of the sequence. The sequence has one or two segments that the first token of the sequence is always [CLS] which contains the special classification embedding and another special token [SEP] is used for separating segments. For text classification tasks, Transformer models take the final hidden state **h** of the first token [CLS] as the representation of the whole sequence (Sun et al., 2019). A simple softmax classifier is added to the top of the transformer model to predict the probability of label c as shown in Equation 1 where W is the task-specific parameter matrix .

$$p(c|\mathbf{h}) = softmax(W\mathbf{h}) \tag{1}$$

We fine-tuned all the parameters from transformer models as well as W jointly by maximising the log-probability of the correct label. For English language, there exists many pretrained transformer models released by Hugging Face[3]. Therefore, we experimented several transformer architectures like BERT (Devlin et al., 2018), XLNet (Yang et al., 2019), XLM (Conneau et al., 2019), RoBERTa (Liu et al., 2019b) and DistilBERT (Sanh et al., 2019). We used the HuggingFace's implementation of the transformer models (Wolf et al., 2019) and the pre-trained models available in the HuggingFace's model repository[4]. However, for non-English languages, Google only provides a single BERT multilinugal model that supports 104 languages[5]. DistilBERT also has a multilingual model that supports the same 104

[2] https://fasttext.cc/docs/en/crawl-vectors.html

[3] https://huggingface.co

[4] https://huggingface.co/models

[5] https://github.com/google-research/bert/blob/master/multilingual.md

languages that multilingual BERT supports. We used these two models for all the non-English languages. To experiment more, we also experimented several community built transformer models.

For Arabic, other than the multilingual BERT model and multilingual DistilBERT we used AraBERT (Antoun et al., 2020) [6]. AraBERT is an Arabic pretrained lanaguage model based on Google's BERT architechture. The model was trained on 70M sentences or 23GB of Arabic text with 3B words. The training corpora are a collection of publicly available large scale raw Arabic text (Arabic Wikidumps, The 1.5B words Arabic Corpus (El-Khair, 2016), The OSIAN Corpus (Zeroual et al., 2019), Assafir news articles, and 4 other manually crawled news websites (Al-Akhbar, Annahar, AL-Ahram, AL-Wafd). There are two version off the model AraBERTv0.1 and AraBERTv1, with the difference being that AraBERTv1 uses pre-segmented text where prefixes and suffixes were splitted using the Farasa Segmenter (Abdelali et al., 2016). We used AraBERTv1 since it performed better from the two models in most of the downstream NLP tasks (Antoun et al., 2020).

For Danish language, we used Danish-BERT [7], a Danish pretrained language model based on BERT architecture. The Danish corpus used to train the Danish-BERT was compiled by combining multiple sources: All Danish language text from Common Crawl, The Danish Wikipedia, Custom scraped data from the two biggest Danish debate forums (dindebat.dk and hestenettet.dk) and Danish OpenSubtitles.

We used GreekBERT[8] for Greek Language. This model too has been trained on combining multiple text sources: Greek Wikidumps, Greek part of European Parliament Proceedings Parallel Corpus and Greek language text from OSCAR (Ortiz Suárez et al., 2019), a cleansed version of Common Crawl.Common Crawl. Unlike the transformer models we used for other languages GreekBERT is uncased. Therefore, we had to convert all tokens to lowercase before feeding them to the model.

For Turkish we used BERTurk[9] which is also based on the BERT architecture. The current version of the model was trained on a filtered and sentence segmented version of the Turkish OSCAR corpus (Ortiz Suárez et al., 2019), a recent Wikipedia dump and various OPUS corpora (Tiedemann, 2012).

5 Results

In this section we present the evaluation results that were obtained during testing for each language. We also provide a brief look at the final submission results of the shared task. In all the languages multilingual DistilBERT does not perform better than multilingual BERT model we used. Therefore, we have omitted it from the results.

5.1 Arabic

For Arabic language, we trained our system on the training set and evaluated the system on development data that the organisers provided. Table 2 shows the results we obtained for the development set. As shown Transformer models outperform traditional word embedding models. Also it should be noted that AraBERT slightly outperformed multilingual BERT. On the test set our system achieved 0.788 macro F1 score and ranked 41st out of 53 teams.

	Not Offensive			Offensive			Weighted Average			
Model	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**F1 Macro**
CNN	0.80	0.82	0.81	0.71	0.66	0.69	0.82	0.82	0.78	0.70
RNN-BILSTM	0.76	0.78	0.77	0.67	0.62	0.65	0.78	0.78	0.74	0.66
BERT-multilingual-cased	0.82	0.84	0.83	0.74	0.69	0.72	0.86	0.87	0.86	0.77
AraBERT	0.84	0.86	0.85	0.76	0.71	0.74	0.88	0.89	0.88	**0.79**

Table 2: Results for English Task For each model, Precision (P), Recall (R), and F1 are reported on all classes, and weighted averages. Macro-F1 is also listed (best in bold).

[6] https://github.com/aub-mind/arabert
[7] https://github.com/botxo/nordic_bert
[8] https://github.com/nlpaueb/greek-bert
[9] https://github.com/stefan-it/turkish-bert

5.2 Danish

Since the organisers didn't provide a separate development set for Danish language, we separated 20% of the training data and treated it as the development set to evaluate the models. Table 3 shows the results. In Danish language also transformer models outperformed traditional word embedding models. However, Danish-BERT model could not outperform the BERT Multilingual model. We suspect that this is mainly due to the fact that Danish-BERT model we used was uncased. For the test data our best system scored 0.656 Macro F1 score ranking 32nd out of 39 teams.

	Not Offensive			Offensive			Weighted Average			
Model	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**F1 Macro**
CNN	0.91	0.92	0.93	0.71	0.67	0.67	0.85	0.85	0.85	0.72
RNN-BILSTM	0.90	0.91	0.92	0.70	0.66	0.66	0.84	0.84	0.84	0.71
BERT-multilingual-cased	0.95	0.96	0.97	0.75	0.75	0.75	0.93	0.92	0.93	**0.80**
DANISH-BERT	0.92	0.93	0.93	0.71	0.71	0.71	0.89	0.89	0.89	0.76

Table 3: Results for English Task For each model, Precision (P), Recall (R), and F1 are reported on all classes, and weighted averages. Macro-F1 is also listed (best in bold).

5.3 English

For English language also we separated an evaluation set which was 20% from the training set and evaluated our models on that. Since there were many pretrained transformer models for English, we were able to experiment with various transformer architectures. Table 4 shows the results for the evaluation set. *XLNET large cased* transformer model outperformed all the other architectures. Our best system scored 0.90056 Macro F1 score for the test set and ranked 62nd out of 86 teams.

	Not Offensive			Offensive			Weighted Average			
Model	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**F1 Macro**
CNN	0.87	0.88	0.87	0.81	0.77	0.77	0.81	0.83	0.82	0.79
RNN-BILSTM	0.86	0.87	0.86	0.80	0.76	0.76	0.80	0.81	0.81	0.77
BERT-large-cased	0.91	0.92	0.91	0.85	0.81	0.81	0.85	0.85	0.86	0.82
XLNet-large-cased	0.92	0.93	0.92	0.84	0.83	0.83	0.87	0.87	0.88	**0.85**

Table 4: Results for English Task For each model, Precision (P), Recall (R), and F1 are reported on all classes, and weighted averages. Macro-F1 is also listed (best in bold).

5.4 Greek

For Greek we prepared a separate evaluation set as we did with Danish and English since the organisers did not provide a development set. In Greek, GreekBERT performed best outperforming multilingual BERT model slightly. Table 5 shows the results for the evaluation set. For the test set it had 0.814 Macro F1 score and ranked 15th out of 37 teams.

	Not Offensive			Offensive			Weighted Average			
Model	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**F1 Macro**
CNN	0.81	0.82	0.83	0.74	0.75	0.74	0.86	0.85	0.84	0.80
RNN-BILSTM	0.80	0.80	0.81	0.73	0.72	0.73	0.77	0.77	0.75	0.71
BERT-multilingual-cased	0.87	0.87	0.88	0.79	0.79	0.79	0.83	0.83	0.82	0.77
GREEKBERT	0.89	0.89	0.90	0.81	0.81	0.81	0.85	0.85	0.85	**0.81**

Table 5: Results for Greek subtask For each model, Precision (P), Recall (R), and F1 are reported on all classes, and weighted averages. Macro-F1 is also listed (best in bold).

5.5 Turkish

For Turkish also we prepared a separate evaluation set as we did with Danish and English. For Turkish, the transformer model with BERTURK performed best surpassing BERT multilingual as shown in table 6. Our best system had 0.7858 Macro F1 score and ranked 7th out of 46 teams.

	Not Offensive			Offensive			Weighted Average			
Model	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**F1 Macro**
CNN	0.80	0.81	0.79	0.70	0.59	0.69	0.75	0.76	0.75	0.67
RNN-BILSTM	0.79	0.80	0.78	0.69	0.58	0.68	0.74	0.75	0.74	0.66
BERT-multilingual-cased	0.83	0.84	0.82	0.73	0.62	0.72	0.78	0.79	0.78	0.70
BERTURK	0.92	0.93	0.91	0.82	0.82	0.81	0.87	0.88	0.87	**0.79**

Table 6: Results for Turkish subtask For each model, Precision (P), Recall (R), and F1 are reported on all classes, and weighted averages. Macro-F1 is also listed (best in bold).

6 Conclusion

In this paper, we have presented the team *BRUMS* system for identifying offensive language in social media posts in Arabic, Danish,English, Greek and Turkish. The system uses minimal preprocessing, and relies on word and context embeddings. We experimented with different deep neural network architectures in order to determine the most suitable for this task. Our implementation has been made available on Github.[10]. According to our evaluation, and the results provided by the task organisers, it is clear that fine tuning transformer architectures score highest overall. Also we discovered that pretrained monolingual transformer models perform better than the *bert-multilingual* model. In future, we are hoping to experiment more with transformer architectures in different languages.

References

Ahmed Abdelali, Kareem Darwish, Nadir Durrani, and Hamdy Mubarak. 2016. Farasa: A fast and furious segmenter for Arabic. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 11–16, San Diego, California, June. Association for Computational Linguistics.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding.

Segun Taofeek Aroyehun and Alexander Gelbukh. 2018. Aggression detection in social media: Using deep neural networks, data augmentation, and pseudo labeling. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 90–97.

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.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*.

Çağrı Çöltekin. 2020. A corpus of Turkish offensive language on social media. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 6174–6184, Marseille, France, May. European Language Resources Association.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

[10] https://github.com/tharindudr/offenseval_2020

Jianjing Cui, Jun Long, Erxue Min, Qiang Liu, and Qian Li. 2018. Comparative study of cnn and rnn for deep learning based intrusion detection system. In Xingming Sun, Zhaoqing Pan, and Elisa Bertino, editors, *Cloud Computing and Security*, pages 159–170, Cham. Springer International Publishing.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ibrahim Abu El-Khair. 2016. 1.5 billion words arabic corpus. *ArXiv*, abs/1611.04033.

Paula Fortuna, José Ferreira, Luiz Pires, Guilherme Routar, and Sérgio Nunes. 2018. Merging datasets for aggressive text identification. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 128–139.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. 2005. Bidirectional lstm networks for improved phoneme classification and recognition. In *Proceedings of the 15th International Conference on Artificial Neural Networks: Formal Models and Their Applications - Volume Part II*, ICANN'05, page 799–804, Berlin, Heidelberg. Springer-Verlag.

Hansi Hettiarachchi and Tharindu Ranasinghe. 2019. Emoji powered capsule network to detect type and target of offensive posts in social media. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 474–480, Varna, Bulgaria, September. INCOMA Ltd.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November.

M Hughes, I Li, S Kotoulas, and T Suzumura. 2017. Medical text classification using convolutional neural networks. *Studies in health technology and informatics*, 235:246.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October. Association for Computational Linguistics.

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, Santa Fe, New Mexico, USA, August. Association for Computational Linguistics.

Ping Liu, Wen Li, and Liang Zou. 2019a. Nuli at semeval-2019 task 6: transfer learning for offensive language detection using bidirectional transformers. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 87–91.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Shervin Malmasi and Marcos Zampieri. 2017. Detecting Hate Speech in Social Media. In *Proceedings of the International Conference Recent Advances in Natural Language Processing (RANLP)*, pages 467–472.

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*, FIRE '19, page 14–17, New York, NY, USA. Association for Computing Machinery.

Sandip Modha, Prasenjit Majumder, and Thomas Mandl. 2018. Filtering aggression from the multilingual social media feed. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 199–207.

Joaquın Padilla Montani. 2018. Tuwienkbs at germeval 2018: German abusive tweet detection. In *14th Conference on Natural Language Processing KONVENS*, volume 2018, page 45.

Hamdy Mubarak, Ammar Rashed, Kareem Darwish, Younes Samih, and Ahmed Abdelali. 2020. Arabic offensive language on twitter: Analysis and experiments.

Pedro Javier Ortiz Suárez, Benoît Sagot, and Laurent Romary. 2019. Asynchronous Pipeline for Processing Huge Corpora on Medium to Low Resource Infrastructures. In *7th Workshop on the Challenges in the Management of Large Corpora (CMLC-7)*, Cardiff, United Kingdom, July.

Zeses Pitenis, Marcos Zampieri, and Tharindu Ranasinghe. 2020. Offensive language identification in greek. In *Proceedings of the Twelfth International Conference on Language Resources and Evaluation (LREC 2020)*, Marseille, France, May. European Language Resources Association (ELRA).

Tharindu Ranasinghe, Marcos Zampieri, and Hansi Hettiarachchi. 2019. Brums at hasoc 2019: Deep learning models for multilingual hate speech and offensive language identification. *In Proceedings of the 11th annual meeting of the Forum for Information Retrieval Evaluation (December 2019)*.

Julian Risch and Ralf Krestel. 2018. Delete or not Delete? Semi-Automatic Comment Moderation for the Newsroom. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 166–176.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Dominik Scherer, Andreas Müller, and Sven Behnke. 2010. Evaluation of pooling operations in convolutional architectures for object recognition. In Konstantinos Diamantaras, Wlodek Duch, and Lazaros S. Iliadis, editors, *Artificial Neural Networks – ICANN 2010*, pages 92–101, Berlin, Heidelberg. Springer Berlin Heidelberg.

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*, pages 3498–3508, Marseille, France, May. European Language Resources Association.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In Maosong Sun, Xuanjing Huang, Heng Ji, Zhiyuan Liu, and Yang Liu, editors, *Chinese Computational Linguistics*, pages 194–206, Cham. Springer International Publishing.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey, May. European Language Resources Association (ELRA).

Dirk von Grünigen, Fernando Benites, Pius von Däniken, Mark Cieliebak, Ralf Grubenmann, and AG SpinningBytes. 2018. spmmmp at germeval 2018 shared task: Classification of offensive content in tweets using convolutional neural networks and gated recurrent units. In *14th Conference on Natural Language Processing KONVENS 2018*, page 130.

Gregor Wiedemann, Eugen Ruppert, Raghav Jindal, and Chris Biemann. 2018. Transfer learning from lda to bilstm-cnn for offensive language detection in twitter. *arXiv preprint arXiv:1811.02906*.

Michael Wiegand, Melanie Siegel, and Josef Ruppenhofer. 2018. Overview of the GermEval 2018 Shared Task on the Identification of Offensive Language. In *Proceedings of GermEval*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.

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: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1415–1420, Minneapolis, Minnesota, June. Association for Computational Linguistics.

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*, pages 75–86, Minneapolis, Minnesota, USA, June. Association for Computational Linguistics.

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*.

Imad Zeroual, Dirk Goldhahn, Thomas Eckart, and Abdelhak Lakhouaja. 2019. OSIAN: Open source international Arabic news corpus - preparation and integration into the CLARIN-infrastructure. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 175–182, Florence, Italy, August. Association for Computational Linguistics.