KS@LTH at SemEval-2020 Task 12: Fine-tuning multi- and monolingual transformer models for offensive language detection

Kasper Socha
Lund University / Lund, Sweden
fte10kso@student.lu.se

Abstract

This paper describes the KS@LTH system for SemEval-2020 Task 12 OffensEval2: Multilingual Offensive Language Identification in Social Media. We compare mono- and multilingual models based on fine-tuning pre-trained transformer models for offensive language identification in Arabic, Greek, English and Turkish. For Danish, we explore the possibility of fine-tuning a model pre-trained on a similar language, Swedish, and additionally also cross-lingual training together with English. Overall we find that monolingual models achieve higher macro-averaged F1 score. With cross-lingual training of Danish together with English, we achieve better results than by training on the small Danish dataset alone. For Arabic, Danish, English, Greek, and Turkish, we obtained macro-averaged F1 scores of 0.890, 0.775, 0.916, 0.848, and 0.810 ranking 6th, 5th, 6th, 3rd and 4th for each language, respectively.

1 Introduction

Offensive language is a prevalent phenomenon in many online communities and social media platforms. Due to the vast amount of content, it is often infeasible to manually moderate all user submitted content. Computational methods for identifying this type of content is one possible way to help mitigate the problem. Different aspects of the problem such as aggression (Kumar et al., 2018), cyber bulling (Sprugnoli et al., 2018) and hate speech (Malmasi and Zampieri, 2017) have been studied in recent work. OffensEval 2019 used a new three-level hierarchical annotation schema to capture multiple aspects of offensive language in one framework (Zampieri et al., 2019a).

While much of the previous work is focused on English, offensive language detection is a multilingual problem. Apart from country specific communities, large social media platforms such as Facebook and Twitter have many users interacting in their native tongue. Recently, offensive language detection addressed different languages such as German (Wiegand et al., 2018), Arabic (Mulki et al., 2019), Italian (Sanguinetti et al., 2018), and Spanish (Fersini et al., 2018). In OffensEval 2020, the first level task of offensive language detection has been expanded to cover five languages, Arabic, Danish, English, Greek, and Turkish.

Transfer learning is nothing new in NLP but over time, the pre-training has become more complex, incorporating more context. In recent years, language models based on the transformer architecture pre-trained on large amounts of unlabeled text and then fine-tuned on downstream tasks have been used to achieve state-of-the-art (SOTA) results on many natural language benchmarks (Devlin et al., 2018; Liu et al., 2019; Yang et al., 2019). In OffensEval 2019, seven of the top ten models used BERT in some way (Zampieri et al., 2019b). One of the advantages of transfer learning is that it can potentially reduce the amount of labeled data that is needed. The model can learn general features of language from a large unannotated corpus during pre-training. Task specific features can then be learned from a smaller annotated corpus. On some datasets, using a pre-trained language model has shown to match the results of models trained from scratch on ten times more data. Adding language model fine-tuning on unlabeled domain specific text can potentially reduce the need for labeled data even more (Howard and Ruder, 2018).

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 2045–2053
Barcelona, Spain (Online), December 12, 2020.

One obstacle to using large transformer models is that the pre-training step is expensive. The Megatron-LM has 8.3 billion parameters and was trained over 9 days on 512 GPUs (Shoeybi et al., 2019). In comparison, the fine-tuning step is relatively inexpensive. This makes model sharing an important part of applying large transformer models to many tasks. The *HuggingFace Transformers* library provides a platform for sharing models developed by researchers and the community, and a unified API for using them (Wolf et al., 2019).

One additional challenge with multilingual offensive language detection is low resource languages. Such languages might lack both unlabeled data for pre-training and labeled data for fine-tuning. One possible solution in such cases is to use multilingual models. Such models can achieve lower perplexity than monolingual models for language modeling of low resource languages (Conneau and Lample, 2019). In some contexts, multilingual models can even outperform monolingual models on downstream tasks (Conneau et al., 2019). In the case of lacking labeled data, they have also shown to perform well on zero-shot cross-lingual classification tasks. This type of transfer works best between typologically similar languages. However, transfer is possible to some extent even between languages with different scripts (Pires et al., 2019).

This paper describes our system for OffensEval 2020 (Zampieri et al., 2020). We participated in *Sub-task A: Offensive language identification* for all language tracks. Based on the recent success of the transformer architecture, we compared monolingual BERT models for Arabic, English, Greek, and Turkish with the XLM-R multilingual model (Conneau et al., 2019). We found that the monolingual models outperform the multilingual models for all languages on the development data. We used models available through the HuggingFace Transformers library. Since no monolingual models were available for Danish, we initially compared a Swedish BERT model with multilingual XLM-R. We found that the Swedish model worked reasonably well on the development data, while XLM-R only predicted the majority class for most runs. We hypothesized that this is due to the small and imbalanced Danish dataset; similar high variance results have been seen for BERT in Devlin et al. (2018) and Phang et al. (2019). To get around the problem of the small dataset, we tried cross-lingual training of Danish and English using XLM-R which outperformed the Swedish BERT model.

In Section 2 we give a short description of the task and data used. Section 3 presents our approach, describing data preprocessing, models and training approach. Section 4 shows our results on the test data for OffensEval 2020.

2 Task and Data

OffensEval 2020 uses a multilingual dataset of posts from Twitter, tweets, with annotations following the hierarchical annotation schema proposed by Zampieri et al. (2019a). Only the first level of annotation is provided for all languages. This level discriminates between two kinds of tweets:

- **Offensive (OFF):** Tweets containing any form of offensive language. This includes insults, threats, and profanity.
- **Not Offensive (NOT):** Tweets not containing any form of unacceptable language.

The goal of the task is to distinguish between offensive and not offensive tweets. Macro-averaged F1-score is used as evaluation metric.

Table 1 shows a summary of the labeled training datasets for each language. All the datasets are imbalanced to some extent, with the majority of tweets being labeled as not offensive. Danish is the most extreme in this regard, having only 13% of tweets labeled as offensive. We can also see that the size of the datasets varies quite a bit, with Turkish having about ten times as many labeled instances as Danish.

For English the labeled dataset is the same as for OffensEval 2019. In addition to the manually annotated data, about nine million additional tweets labeled using unsupervised methods are provided for English (Rosenthal et al., 2020). These tweets have a confidence score and a standard deviation. For English, we also use additional publicly available data from Kaggle[1] and Davidson et al. (2017)[2].

[1] https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/data
[2] https://github.com/t-davidson/hate-speech-and-offensive-language

Language	OFF	NOT	Total
Turkish (Çöltekin, 2020)	6131 (19%)	25625 (81%)	31756
English (Zampieri et al., 2019a)	4640 (33%)	9460 (67%)	14100
Greek (Pitenis et al., 2020)	2486 (28%)	6257 (72%)	8743
Arabic (Mubarak et al., 2020)	1589 (20%)	6411 (80%)	8000
Danish (Sigurbergsson and Derczynski, 2020)	384 (13%)	2576 (87%)	2960

Table 1: Training dataset size and class distribution for each language.

3 Approach

3.1 Data Preprocessing

A minimal amount of preprocessing was done. We applied only two operations to all languages:

1. Multiple consecutive user mentions were replaced with a single @User to reduce sequence length and noise.
2. All tweets were truncated or padded to a common length. This length was chosen separately for each language to be the smallest length longer than 95% of all tweets in the training set.

Additional processing was done on the external datasets for English. We sampled about 10,000 additional tweets from Davidson et al. (2017). Samples were chosen such that the complete labeled tweet dataset became balanced. Tweets with at least 3 annotators labeling it as either *offensive language* or *hate speech* were labeled as OFF. Tweets with all annotators agreeing on the *neither*-class were labeled as NOT. A balanced dataset of 13,000 Wikipedia comments, from the Kaggle dataset, were also added. To be consistent with the Twitter data, all comments were at most 280 characters. Any comment having at least two of the labels toxic, severe toxic, obscene, threat, insult, or identity hate was labeled as OFF. Comments with no negative labels were labeled as NOT. For both datasets, we replaced URLs with a URL token, and for the tweet dataset, we replaced user mentions with @User.

Additionally we sampled 400,000 tweets from the English silver standard data using confidence scores as weights. These were then filtered down further to the 40,000 tweets with highest confidence using our model as described in section 3.3.

3.2 Models

Vaswani et al. (2017) initially introduced the transformer architecture in the context of machine translation. While previous approaches relied on convolutional and recurrent neural networks, they showed that a relatively simple architecture based on feed-forward neural networks and attention mechanisms could provide better results while being more parallelizable and faster to train. Like previous sequence-to-sequence models the transformer consists of two main components: an encoder component and a decoder component.

Radford et al. (2018) trained a left-to-right language model, GPT, using only the decoder part of the transformer and fine-tuned it on multiple downstream tasks with minimal task specific changes. Devlin et al. (2018) showed the importance of bi-directional pre-training for certain types of tasks by obtaining new SOTA results on 11 NLP benchmarks, including an almost 8 point improvement on GLUE. Their model architecture, named BERT (Bidirectional Encoder Representations from Transformers), is the architecture we used for all monolingual models apart from English.

Since the decoder component of the transformer already does masking of subsequent positions, it is a natural choice for the next word prediction language modeling task used by GPT. To be able to train a bidirectional language model, BERT instead uses the encoder part of the transformer. Apart from increasing the size, it is almost identical to the initial transformer implementation. BERT consists of a stack of encoders, 12 for $\text{BERT}_{\text{BASE}}$ and 24 for $\text{BERT}_{\text{LARGE}}$, compared to 6 in the original transformer.

Each encoder, in turn, consists of two main parts: a self-attention layer followed by a feed-forward neural network. Self-attention is the mechanism which allows the transformer to consider other words

in the sequence when encoding the current word. BERT increases the number of attention heads from 8 in the original Transformer to 12 for $\text{BERT}_{\text{BASE}}$ and 16 for $\text{BERT}_{\text{LARGE}}$. Finally the number of hidden units in the feed-forward neural networks is also increased from 512 to 758 and 1024 for $\text{BERT}_{\text{BASE}}$ and $\text{BERT}_{\text{LARGE}}$, respectively.

We used pre-trained BERT language models without changes to the base architecture. For the fine-tuning step, we followed the approach for single sentence classification suggested by Devlin et al. (2018). A single fully connected classification layer was added to the base model. A special `[CLS]` token was prepended to all inputs. The contextual representation of this token was used as an embedding for the complete sentence, and passed to the classification head. The complete base model was fine-tuned during training.

Liu et al. (2019) showed that BERT is undertrained. Their model, RoBERTa, uses exactly the same architecture as BERT. RoBERTa outperforms BERT simply by training on more data, with larger batches, for a longer time. Some additional simple changes in the pre-training approach, such as removing one of the pre-training objectives and training on longer sequences, improved the results even further. This is the monolingual model we used for English. There were no pre-trained RoBERTa models available for the other languages. The fine-tuning approach is identical to the one used for BERT.

Similarly, in the multilingual context, the XLM-RoBERTa (XLM-R) model we used achieves much of its improvement over previous multilingual models by using several orders of magnitude more data (Conneau et al., 2019). Conneau et al. (2019) also find that vocabulary size has a large impact when many languages are used. Again XLM-R uses the same model architecture as BERT. However, the increase of vocabulary size from 30K to 250K leads to an increase of the total number of parameters from 110M and 335M to 270M and 550M for the $_{\text{BASE}}$ and $_{\text{LARGE}}$ models, respectively. All five languages are present among the 100 languages used during pre-training of XLM-R. The fine-tuning approach is identical to the one used for the previous models.

A summary of the different pre-trained models that we used for each language is provided below:

Arabic

- Monolingual: Arabic $\text{BERT}_{\text{BASE}}$[3]
- Multilingual: XLM-R_{BASE} (Arabic corpus of 28.0GB).

Danish

- Monolingual: Swedish $\text{BERT}_{\text{BASE}}$[4]
- Multilingual: XLM-R_{BASE} (Danish corpus of 45.6GB).

English

- Monolingual: English $\text{RoBERTa}_{\text{LARGE}}$ (Liu et al., 2019).
- Multilingual: XLM-R_{LARGE} (English corpus of 300.8GB).

Greek

- Monolingual: Greek $\text{BERT}_{\text{BASE}}$[5]
- Multilingual: XLM-R_{BASE} (Greek corpus of 46.9GB).

Turkish

- Monolingual: Turkish $\text{BERT}_{\text{BASE}}$[6]
- Multilingual: XLM-R_{BASE} (Turkish corpus of 20.9GB).

[3] https://huggingface.co/asafaya/bert-base-arabic

[4] https://huggingface.co/af-ai-center/bert-base-swedish-uncased, https://github.com/af-ai-center/SweBERT

[5] https://huggingface.co/nlpaueb/bert-base-greek-uncased-v1

[6] https://huggingface.co/dbmdz/bert-base-turkish-cased

Language	Model	Mean F1	Max F1
Arabic	BERT	0.864	**0.868**
	XLM-R	0.709	0.833
Danish	$\text{BERT}_{\text{Swedish}}$	0.745	0.777
	$\text{XLM-R}_{\text{Danish}}$	0.464	0.464
	$\text{XLM-R}_{\text{Danish+English}}$	0.795	**0.813**
English	RoBERTa	0.913	**0.917**
	XLM-R	0.892	0.901
Greek	BERT	0.820	**0.845**
	XLM-R	0.766	0.798
Turkish	BERT	0.814	**0.820**
	XLM-R	0.805	0.814

Table 2: Mean and maximum F1 macro on the development sets for five random restarts on each language and model.

3.3 Experiments

We carried out the initial experimentation and the hyperparameter selection using the English data from OffensEval 2019. We followed the fine-tuning procedure recommended for BERT by Devlin et al. (2018). We tested the following parameters, where the best performing values are underlined:

- **Batch size**: 16, 32
- **Learning rate (Adam)**: 5e-5, 3e-5, 2e-5
- **Epochs**: 2, 3, 4

The dropout was kept constant at 0.1 for all layers. Overall we found that fine-tuning was relatively insensitive to batch size and learning rate. However, most random restarts seemed to overfit when using more than 2 epochs. The same hyperparameters were then used for all further experiments.

For each language, 20% of the data was set aside as a development set and used for model selection. For each model, we ran five random restarts with different data shuffling and classifier head layer initialization. The model with the best macro-averaged F1-score on the development set was then used for submission. Table 2 summarizes the results we obtained.

For English, the training was done in two steps. Initially, we trained the model using only the labeled data. We then used this model to label 400,000 samples from the silver standard data. We labeled the 20,000 instances with the highest scores as OFF and the 20,000 instances with the lowest scores as NOT. We finally added these 40,000 tweets to the training set used to train the final model.

For Danish, we initially failed to train XLM-R to predict anything other than the majority class. Since XLM-R has shown promising cross-lingual transfer results, we tried training Danish together with English. We did this by shuffling the Danish training data with the English data from OffensEval 2019. We evaluated the models only on the Danish development dataset.

4 Results

Table 3 shows our results on the official test data. The figures are similar to those we obtained on the development dataset. Danish shows the largest drop in performance, going from 0.813 on the development dataset to 0.775 on the test dataset. Nonetheless, since the development set was rather small, it might be difficult to conclude on the generalization performance.

4.1 Impact of external and silver standard data

Previous work has shown that models for offensive language detection often generalize poorly to other datasets (Karan and Šnajder, 2018; Swamy et al., 2019; Arango et al., 2019). This is especially true when

Language	F1	Rank
Arabic	0.890	6
Danish	0.775	5
English	0.916	6
Greek	0.848	3
Turkish	0.810	4

Table 3: Macro averaged F1-score on the test data and competition placement.

Training Data	F1
OffensEval19	0.906
Wikipedia	0.909
Davidson	0.848
OffensEval19 + Davidson	0.917
OffensEval19 + Davidson + Wikipedia	0.917
OffensEval19 + Davidson + Wikipedia + Silver	0.916
OffensEval19 + Silver	0.915

Table 4: Results on the English test set using different subsets of the training data. For the combination *OffensEval19 + Silver*, the silver standard data was processed using the approach described previously, but only using OffensEval19 for the initial training.

evaluating across domains, e.g. between Twitter and Wikipedia, but also within the same domain. Some features are likely platform specific and some datasets focus on specific aspects of offensive language. The data collection process can also lead to some types of content being overrepresented.

We tried to determine the impact of the different English datasets we used by retraining the model on different subsets of the data. The results on the test set are shown in table 4. All the labeled datasets perform reasonably well on their own. Surprisingly the sampled Wikipedia data performs just as well as the OffensEval 2019 data. The sampled data from (Davidson et al., 2017) performs worse. This might be due to it being smaller and oversampled to contain more offensive tweets. This hypothesis is also supported by the fact that when used with the OffensEval 2019 data, the results are comparable with the submitted model. Finally, the silver standard data seems to be most useful when the original labeled dataset is small.

4.2 Error analysis

To get a better understanding of the kind of mistakes the system makes we studied some of the misclassified instances. To get some indications of what words are important for the classification of a given sentence, we applied LIME (Ribeiro et al., 2016). In short, LIME estimates the importance of a word by:

1. Generating many distorted versions of the original tweet.
2. Applying the original classifiers to the distorted tweets.
3. Training a white-box model to predict the output of the original classifier given a version of the tweet.

Table 5 shows five instances from the English OffensEval 2019 dataset, where the classifier assigned a high confidence to the wrong class. Examples 1 and 2 are very short and the profanity dominates the other words. Both examples look like reasonable classifications. However, the same thing seems to happen in Example 3. The word *shit* dominates the otherwise inoffensive sentence. Example 4 has no direct profanity. Looking at bigrams using LIME, *stinking cute* is correctly identified as inoffensive. Example 5 doesn't seem to have any offensive language. It is possible that it could be considered offensive given external knowledge about the people mentioned. Given only the tweet, the classification looks reasonable.

#	Tweet	Prediction	Label
1	Are you fucking^{+} serious? URL	OFF	NOT
2	And dicks^{+}. URL	OFF	NOT
3	#Room25 is actually incredible, Noname is the shit^{+}, always has been, and I'm seein her in like 5 days in Melbourne. Life is good. Have a nice day.	OFF	NOT
4	@User Aw she is so stinking^{-} cute^{+}! How old is she now?	NOT	OFF
5	#ChristineBlaseyFord is your #Kavanaugh accuser. #Liberals try this EVERY time. #ConfirmJudgeKavanaugh URL	NOT	OFF

Table 5: Examples of misclassifications for English. Using LIME, we marked words that have a large impact on the classification. A + indicates agreement with the predicted label and a – indicates disagreement.

5 Conclusions

In the context of offensive language detection for multiple languages, we found that fine-tuning transformer models works well. Monolingual models outperform multilingual models for all languages studied. However, multilingual models can still be a viable alternative when no monolingual models are available. When the amount of labeled data is small, they can also be used for cross-lingual transfer. We showed the positive effect of cross lingual transfer when augmenting Danish with English.

Acknowledgments

I would like to thank Pierre Nugues at Lund University for helpful discussions and feedback on this paper.

References

Aymé Arango, Jorge Pérez, and Barbara Poblete. 2019. Hate speech detection is not as easy as you may think: A closer look at model validation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 45–54.

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.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 7059–7069. Curran Associates, Inc.

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*.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Eleventh international aaai conference on web and social media*.

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

Elisabetta Fersini, Paolo Rosso, and Maria Anzovino. 2018. Overview of the task on automatic misogyny identification at ibereval 2018. In *IberEval@ SEPLN*, pages 214–228.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia, July. Association for Computational Linguistics.

Mladen Karan and Jan Šnajder. 2018. Cross-domain detection of abusive language online. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 132–137, Brussels, Belgium, 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.

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.

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 2017*, pages 467–472, Varna, Bulgaria, September. INCOMA Ltd.

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*.

Hala Mulki, Hatem Haddad, Chedi Bechikh Ali, and Halima Alshabani. 2019. L-HSAB: A Levantine twitter dataset for hate speech and abusive language. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 111–118, Florence, Italy, August. Association for Computational Linguistics.

Jason Phang, Thibault Févry, and Samuel R. Bowman. 2019. Sentence Encoders on STILTs: Supplementary Training on Intermediate Labeled-data Tasks. *arXiv:1811.01088 [cs]*, February. arXiv: 1811.01088.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual bert? *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

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

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should i trust you?". *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*.

Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Marcos Zampieri, and Preslav Nakov. 2020. A large-scale semi-supervised dataset for offensive language identification. *arXiv preprint arXiv:2004.14454*.

Manuela Sanguinetti, Fabio Poletto, Cristina Bosco, Viviana Patti, and Marco Stranisci. 2018. An Italian twitter corpus of hate speech against immigrants. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May. European Language Resources Association (ELRA).

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. Megatron-lm: Training multi-billion parameter language models using model parallelism.

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.

Rachele Sprugnoli, Stefano Menini, Sara Tonelli, Filippo Oncini, and Enrico Piras. 2018. Creating a WhatsApp Dataset to Study Pre-teen Cyberbullying. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 51–59, Brussels, Belgium, October. Association for Computational Linguistics.

Steve Durairaj Swamy, Anupam Jamatia, and Björn Gambäck. 2019. Studying generalisability across abusive language detection datasets. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 940–950, Hong Kong, China, November. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

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, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding.

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*.