YoungSheldon at SemEval-2021 Task 5: Fine-tuning Pre-trained Language Models for Toxic Spans Detection using Token classification Objective

Mayukh Sharma, Ilanthenral Kandasamy, W.B. Vasantha
School of Computer Science and Engineering
Vellore Institute of Technology
Vellore, Tamil Nadu, India
04mayukh@gmail.com, ilanthenral.k@vit.ac.in,
vasantha.wb@vit.ac.in

Abstract

In this paper, we describe our system used for SemEval 2021 Task 5: Toxic Spans Detection. Our proposed system approaches the problem as a token classification task. We trained our model to find toxic words and concatenate their spans to predict the toxic spans within a sentence. We fine-tuned Pre-trained Language Models (PLMs) for identifying the toxic words. For fine-tuning, we stacked the classification layer on top of the PLM features of each word to classify if it is toxic or not. PLMs are pre-trained using different objectives and their performance may differ on downstream tasks. We, therefore, compare the performance of BERT, ELECTRA, RoBERTa, XLM-RoBERTa, T5, XLNet, and MPNet for identifying toxic spans within a sentence. Our best performing system used RoBERTa. It performed well, achieving an F1 score of 0.6841 and secured a rank of 16 on the official leaderboard.

1 Introduction

Internet and social networking sites have brought people together by providing a simple yet effective method of communication. Over the years people used it to exchange positive ideas but recently, there has been a rise in toxic content and hate speech over the internet (Zampieri et al., 2019, 2020). Most datasets (Fortuna et al., 2020) dealing with the problem of toxic, offensive, or hateful content aim to classify the entire text belonging to a particular class. They do not identify the parts of the text that make it toxic. Manual filtering of toxic data is tough and can cause mental and emotional stress to annotators (Zampieri et al., 2019). An automatic system with the ability to identify toxic text and highlighting toxic spans can be useful for the moderators. It will help save time and prevent stress caused by reading long texts. SemEval 2021 Task 5: Toxic Spans Detection(Pavlopoulos et al., 2021) draws attention to the problem of identifying toxic spans present in a sentence.

Our proposed system makes use of a word-level classifier for detecting the offensive words present in a sentence. The offsets of the toxic words can then be concatenated to find the toxic spans. We made use of pre-trained language models (PLMs) for building our classifier. We experimented with BERT(Devlin et al., 2019), ELECTRA(Clark et al., 2020), RoBERTa(Liu et al., 2019b), XLNet(Yang et al., 2020), MPNet(Song et al., 2020), T5(Raffel et al., 2020), and XLM-RoBERTa(Conneau et al., 2020) to compare their performance on the task of toxic spans detection. Owing to the increase in the number of pre-trained language models choosing the correct model is an important decision as these models contain millions of parameters and are expensive to train. So, we present a comprehensive analysis of the performance of different models, which can serve as a baseline for future work.

Our best performing system was fine-tuned using RoBERTa and attained an F1 score of **0.6841**. It was ranked **16** on the official leader board. We used different PLMs for fine-tuning and found exceedingly small variations in their performance. Further analyzing our model's performance on the test set we observed that it is essential for the model to not only detect toxic spans but also decide if it needs to predict toxic spans for that sample or not. Our code is available online[1] for method replicability.

2 Background

Identification of toxic/offensive content is an important task in natural language processing. It is essential for the moderation of harmful content over social media sites that might hurt the sentiments of individuals, groups, or communities at large. Much

[1] https://github.com/04mayukh/YoungSheldon-at-SemEval-2021-Task-5-Toxic-Spans-Detection

Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pages 953–959
Bangkok, Thailand (online), August 5–6, 2021. ©2021 Association for Computational Linguistics

work has been done on the identification of offensive content. OffensEval 19, 20 (Zampieri et al., 2019, 2020) provide a comprehensive analysis of methods useful for the identification of offensive content. SemEval 2020 Task 8: Memotion analysis (Sharma et al., 2020) presented with a dataset of internet memes with one sub-task to detect and quantify offensive content. Work done in (Brassard-Gourdeau and Khoury, 2019) explores different aspects of sentiment detection and their correlation to toxicity. (Pavlopoulos et al., 2020) covers the effect of context on toxicity. (D'Sa et al., 2020) uses BERT and FastText for toxicity detection. (Kurita et al., 2019) covers several attacks to by-pass toxic content filters and methods to make the filters robust to such attacks. Recent state-of-the-art systems (Wiedemann et al., 2020; Wang et al., 2020; Liu et al., 2019a; Nikolov and Radivchev, 2019) performed well in identifying offensive content. Work done in (Gröndahl et al., 2018) shows that although recent systems perform well on given datasets, very slight changes made by adversaries may fool the models. Adding words like "love" to offensive tweets may make it less offensive.

Identifying toxic content is an important NLP task. It is useful in moderating online content over the web having millions of users. Most problems deal with labeling the entire content as toxic/non-toxic. None of the previous work has tried to identify spans within a text that makes it toxic. SemEval-2021 Task 5: Toxic Spans Detection aims to bring attention to this problem via the task defined as: Given a dataset D of sentences, the objective of the task is to learn a classification function that can predict the toxic spans T present in the given sentence. The content of the provided dataset D was in English.

Dataset statistics: The dataset for the task consisted of character offsets for toxic spans present for each text sample. The span consisted of single words as well as a collection of words. Table 1 shows the count of samples having different number of toxic words.

From Table 1 we can infer that samples with toxic words within the range of one to three form a major component of the dataset. In the test set, samples with no toxic words were significantly more than the training and development set. Toxic words with the highest frequency of occurrence present in the training set are given in Table 2. We observed that toxic words contained stopwords (the, a, and, of) which are generally not toxic when used independently. These stopwords can exist as part of multiword toxic spans.

3 System Overview

3.1 Pre-trained Language Models

Natural language processing tasks are data intensive. Training deep neural networks for NLP tasks requires large amounts of training data that might not always be available. To overcome this problem researchers proposed pre-training large language models which can be fine-tuned on various downstream tasks. Pre-training involves training general representations of text to understand its syntactic and semantic relations. The main advantage of pre-training is that it can be done on unlabelled text corpus allowing training on a large amount of textual data. The pre-trained language models can then be used across various downstream tasks by fine-tuning them on task-specific datasets.

3.2 Brief overview of used PLMs

BERT: It is a bidirectional language model based on the Transformer architecture(Vaswani et al., 2017). It uses Masked Language Modeling (MLM) and Next Sentence Prediction (NSP) as a pre-training objective.

ELECTRA: It is one of the most recent models and is inspired by generative adversarial networks. It introduces Replaced Token Detection (RTD) pre-training objective.

RoBERTa: It is a modification of BERT proposed by Facebook. It uses dynamic masking as a part of the pre-training objective. NSP was removed and the model was pre-trained on larger data for more time.

XLNet: It is a generalized auto-regressive pre-training method using the best of both Auto Regressive(AR) and Auto Encoding(AE) modeling techniques. It makes use of permutation language modeling (PLM) objective for pre-training.

MPNet: It was proposed by Microsoft. It overcomes the pre-train fine-tune discrepancy in XLNet. It makes use of both PLM and MLM to map the dependencies among predicted tokens as well as use full positional information in a sentence.

T5: It was proposed by Google and aimed to reframe all NLP tasks into a single text-to-text format where both inputs and outputs are always strings. It used a masking objective similar to BERT and used teacher forcing for pre-training.

No. of Toxic words per sentence (N)	N = 0	N>0 and N<=3	N>3 and N<=7	N>7	Total
Train	486	6216	742	495	7939
Development	43	543	72	32	690
Test	394	1541	53	12	2000

Table 1: Number of samples with different frequencies of toxic words.

Toxic Word	Frequency
stupid	1237
idiot	668
the	581
idiots	428
a	383
and	350
of	288
ignorant	277
stupidity	276

Table 2: Most frequent toxic words.

XLM-RoBERTa: It is a multi-lingual model trained by Facebook AI on more than 100 languages. It made use of the Transformer architecture(Vaswani et al., 2017) with multilingual MLM (Devlin et al., 2019; CONNEAU and Lample, 2019) using only monolingual data as a pretraining objective.

3.3 Modelling as Token Classification Task

The given dataset provided spans of toxic content in a statement. Each sentence could contain multiple toxic spans. Another important thing to note was that a toxic span could comprise more than one word. We extracted all toxic words using the toxic spans. If a span contains over one word, it was further processed to extract individual words. Once we found all the toxic words, we split the original sentence to label the toxic/non-toxic words. Before splitting the original sentence, we removed extra whitespace and newline characters. We removed any punctuation before or after the word. Punctuations present within the words were not removed. Figure 1 shows an example of the process. The toxic spans have been highlighted in red in the original sentence which, we convert into an array of words labeled as toxic/non-toxic.

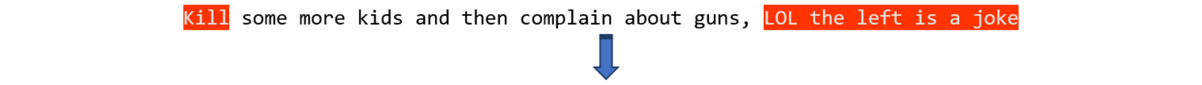
The next step is to prepare the data for fine-tuning on pre-trained language models. PLMs use tokenization to break the original words into sub-words. Different models use different tokenization techniques like Byte-Pair-Encoding(BPE) (Sennrich et al., 2016), WordPiece (Schuster and Nakajima, 2012), and SentencePiece (Kudo and Richardson, 2018). One advantage of using tokenization is that it helps to reduce the vocabulary size. One challenge it poses for token classification tasks is which sub-word to use for classification. Different models also add special tokens like [CLS], [SEP], start, end tokens which are not required for the token classification task. In our approach, we used the first sub-word of the tokenized word for classification. We masked the remaining sub-words and special tokens while computing the loss. The sub-words were masked only during loss computation and not while being passed through the model. This allowed all sub-words to learn dependencies within the sentence. Figure 2 shows the tokenized words and their corresponding labels using the BERT tokenizer.

3.4 Fine-tuning

We used a simple approach for fine-tuning the model for token classification. We used a token classifier on top of features learned by PLMs. Our classifier consisted of three layers on top of PLM features. First was the batch normalization layer, followed by a dropout layer. The final layer was a time-distributed dense layer over features of each tokenized word containing a single neuron and a sigmoid activation to predict if the given token is toxic/non-toxic.

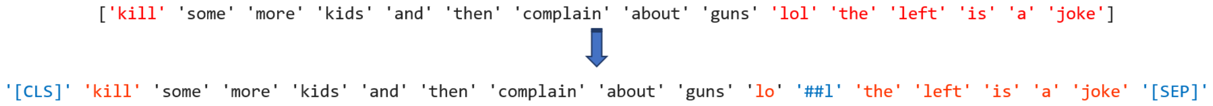
3.5 Masked Loss

As described, we reduced the problem to a token classification task where we predict the label for each word. We used binary cross-entropy loss for the fine-tuning process. In cases where the original word is broken down into multiple sub-words, we used only the first sub-word for calculating the loss. We created masks for each sentence to store the position of words/sub-words. Cross-entropy loss was calculated for required sub-words/words using the masks and then summed up over all tokens in a sentence. The summed value was the loss for a given sentence.



```
Kill some more kids and then complain about guns, LOL the left is a joke
['kill' 'some' 'more' 'kids' 'and' 'then' 'complain' 'about' 'guns' 'lol' 'the' 'left' 'is' 'a' 'joke']
```

Figure 1: Converting toxic spans to toxic words.



```
['kill' 'some' 'more' 'kids' 'and' 'then' 'complain' 'about' 'guns' 'lol' 'the' 'left' 'is' 'a' 'joke']
'[CLS]' 'kill' 'some' 'more' 'kids' 'and' 'then' 'complain' 'about' 'guns' 'lo' '##l' 'the' 'left' 'is' 'a' 'joke' '[SEP]'
```

Figure 2: Tokenization for using PLMs. Sub-words(except first sub-word) and special symbols were masked.

4 Experimental Setup

4.1 Hyperparameters and Training

Our models were developed on Keras[2] (Chollet et al., 2015) using HugginFace's [3] implementation of transformer[4] (Wolf et al., 2020) models. We fine-tuned the models on TPU's on Google Colab. We fixed the sequence length of input to 150 tokens. We padded/truncated the sequences according to their length. Our model was fine-tuned using the AdamW optimizer(Loshchilov and Hutter, 2019) with a linear learning rate decay against masked binary cross-entropy loss. We experimented with learning rates of 1e-4, 3e-5, 4e-5, 5e-5 for each PLM architecture. Fine-tuning was done for 4 epochs. Each PLM architecture with the best performance on the development set was used for making final predictions on the test set.

4.2 Predicting Toxic Span Offsets

Our model was trained to find the toxic words. In case the word was tokenized into sub-words, we used the first sub-word to determine the toxic nature of the entire word. We stored flag values for each sentence to find the correct label for each word during prediction. Once we found the toxic words, we searched for them in the original un-processed sentences. We concatenated the spans for all predicted toxic words which was the final expected output.

4.3 Evaluation Metric

The performance of the model was evaluated using the F1 score as described in (Da San Martino et al., 2019). Let system A_i return a set $S^t_{A_i}$ of character offsets found toxic for post t. Let G_t be ground truth annotation for t. F1 score of system A_i with respect to ground truth values G for post t is calculated as follows:

Model	F1 Score	
	Dev	Test
BERT-base	0.6654	0.6812
ELECTRA-base	0.6710	0.6804
RoBERTa-base	0.6676	**0.6842**
XLM-RoBERTa-base	0.6519	0.6775
T5-large	0.6658	0.6811
XLNet-base	0.6714	0.6817
MPNet-base	**0.6750**	0.6800

Table 3: Model performance on Test set.

$$F^t_1(A_i, G) = \frac{2.P^t(A_i, G).R^t(A_i, G)}{P^t(A_i, G) + R^t(A_i, G)}$$

$$P^t(A_i, G) = \frac{|S^t_{A_i} \cap S^t_G|}{|S^t_{A_i}|}$$

$$R^t(A_i, G) = \frac{|S^t_{A_i} \cap S^t_G|}{|S^t_G|}$$

where $|.|$ represents the cardinality of the set. If $S^t_G = 0$ i.e no toxic spans are present in t then $F^t_1(A_i, G) = 1$ if $|S^t_{A_i}| = 0$ else $F^t_1(A_i, G) = 0$. Finally, $F^t_1(A_i, G)$ was averaged over all posts t present in dataset D to obtain single score for system A_i.

5 Results and analysis

Table 3 shows the performance of our proposed model on different PLMs. Learning rate of 1e-4 was used for ELECTRA, 4e-5 for MPNet, and 5e-5 for the remaining PLMs to obtain the above-mentioned results. RoBERTa had the best performance on the test set while MPNet had the best

[2] https://keras.io

[3] https://huggingface.co/transformers

[4] https://huggingface.co

Model	No. of toxic words = 0		No. of toxic words >0	
	F1 = 1	F1 = 0	F1 = 1	F1 = 0
RoBERTa	24	370	1061	96
BERT	24	370	1050	99
ELECTRA	22	372	1007	76
MPNet	18	376	1063	101
T5	24	370	1041	92
XLNet	32	362	1059	112
XLM-RoBERTa	19	375	1056	104

Table 4: Performance analysis on test samples containing no toxic words vs containing one or more toxic words.

Words	Frequency
stupid	55
ignorant	32
idiot	27
garbage	16
fool	13
pathetic	13
moron, ass, white, dumb, stupidity, idiots	12
racist	10
trash, crap	9

Table 5: Words predicted as toxic in Test samples containing no toxic spans.

performance on the development set. Our best performing model achieved a best F1 score of 0.6842 on the test set and was ranked 16 on the official leader board.

We further analyzed the performance of our model on the test set. We evaluated the performance of our model on samples containing any number of toxic words vs no toxic words. Table 4 shows the results of the analysis. We found that our models performed significantly well for samples having one or more toxic words present and, our best performing model had a perfect F1 score on 66.06 % of them. Our model was unable to find toxic words in only 5.97% of samples containing one or more than one toxic word.

In the case of samples that had no toxic words in a sample, our model could not perform well. Only 6.09% of samples with no toxic words were classified correctly. The dataset statistics for the test set show that samples with no toxic words constitute 19.7 % of the test set. The training and development set had only 6.12% and 6.23% samples without any toxic words. We also found the top 15 most common words which were predicted as toxic from samples containing no toxic words in the test set. The words are given in Table 5 along with their frequency of occurrence.

We can observe that Table 2 and 5 has common words. We trained our model using token classification objective which tries to capture toxic words. The model cannot identify if the word is part of a toxic/non-toxic sentence. Sometimes these words may be part of a sentence intended to present humor or sarcasm. This may lead the model to incorrectly identify toxic words in samples containing no toxic spans.

6 Conclusion

In this paper, we describe our approach for SemEval 2021 Task 5: Toxic Spans Detection. We propose a word-level classifier for identifying the toxic words in a sentence. We experimented with different PLMs to provide a comprehensive analysis of their performance for identifying toxic spans. We performed well, getting a rank of 16 on the leader board. Our analysis shows that a word-level classifier performs extremely well for sentences that contain at least one toxic word. However, it cannot identify cases with no toxic spans efficiently. In the future, we would like to work on solving this problem by using a classifier to simply predict if the sentence is toxic/non-toxic along with span detection.

References

Eloi Brassard-Gourdeau and Richard Khoury. 2019. Subversive toxicity detection using sentiment information. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 1–10, Florence, Italy. Association for Computational Linguistics.

François Chollet et al. 2015. Keras. https://keras.io.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis CONNEAU and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeño, Rostislav Petrov, and Preslav Nakov. 2019. Fine-grained analysis of propaganda in news article. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5636–5646, Hong Kong, China. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

A. G. D'Sa, I. Illina, and D. Fohr. 2020. Bert and fasttext embeddings for automatic detection of toxic speech. In *2020 International Multi-Conference on: "Organization of Knowledge and Advanced Technologies" (OCTA)*, pages 1–5.

Paula Fortuna, Juan Soler, and Leo Wanner. 2020. Toxic, hateful, offensive or abusive? what are we really classifying? an empirical analysis of hate speech datasets. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6786–6794, Marseille, France. European Language Resources Association.

Tommi Gröndahl, Luca Pajola, Mika Juuti, Mauro Conti, and N. Asokan. 2018. All you need is "love": Evading hate speech detection. In *Proceedings of the 11th ACM Workshop on Artificial Intelligence and Security*, AISec '18, page 2–12, New York, NY, USA. Association for Computing Machinery.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Keita Kurita, Anna Belova, and Antonios Anastasopoulos. 2019. Towards robust toxic content classification.

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, Minneapolis, Minnesota, USA. 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. 2019b. Roberta: A robustly optimized bert pretraining approach.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Alex Nikolov and Victor Radivchev. 2019. Nikolov-radivchev at SemEval-2019 task 6: Offensive tweet classification with BERT and ensembles. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 691–695, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

John Pavlopoulos, Léo Laugier, Jeffrey Sorensen, and Ion Androutsopoulos. 2021. Semeval-2021 task 5: Toxic spans detection (to appear). In *Proceedings of the 15th International Workshop on Semantic Evaluation*.

John Pavlopoulos, Jeffrey Sorensen, Lucas Dixon, Nithum Thain, and Ion Androutsopoulos. 2020. Toxicity detection: Does context really matter? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4296–4305, Online. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

M. Schuster and K. Nakajima. 2012. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational*

Linguistics (Volume 1: Long Papers), pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Chhavi Sharma, Deepesh Bhageria, William Scott, Srinivas PYKL, Amitava Das, Tanmoy Chakraborty, Viswanath Pulabaigari, and Björn Gambäck. 2020. SemEval-2020 task 8: Memotion analysis- the visuo-lingual metaphor! In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 759–773, Barcelona (online). International Committee for Computational Linguistics.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. In *NeurIPS 2020*. ACM.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Shuohuan Wang, Jiaxiang Liu, Xuan Ouyang, and Yu Sun. 2020. Galileo at SemEval-2020 task 12: Multi-lingual learning for offensive language identification using pre-trained language models. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1448–1455, Barcelona (online). International Committee for Computational Linguistics.

Gregor Wiedemann, Seid Muhie Yimam, and Chris Biemann. 2020. UHH-LT at SemEval-2020 task 12: Fine-tuning of pre-trained transformer networks for offensive language detection. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1638–1644, Barcelona (online). International Committee for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2020. Xlnet: Generalized autoregressive pretraining for language understanding.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. 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. 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 the Fourteenth Workshop on Semantic Evaluation*, pages 1425–1447, Barcelona (online). International Committee for Computational Linguistics.