OFFENSIVE LANGUAGE IDENTIFICATION IN SOCIAL MEDIA

Haofei Qin, Haozhan Sun, Yi Gao, and Yifei Ke
Duke University
{hq31, hs314, yg185, yk223}@duke.edu

ABSTRACT

Offensive language identification is an important task in natural language processing. In this project, we reproduce and improve the results of a participant team of SemEval-2020 Task 12 challenge. We explore the three Subtasks (offensive language identification, offense type categorization, offense target identification) using models including ALBERT, RoBERTa, GPT-2 and DistilRoBERTa. We successfully reproduce the results using only 1% sampled data, compared to 70% from the original authors and improve the performance on Subtask A and B with checkpoint ensemble and voting ensemble techniques. We conclude that the competition performance can be reproduced with much fewer data, and that the performance can be further improved along the way of model ensemble with properly chosen models and configured training process. The project is available on GitHub[1].

1 INTRODUCTION

1.1 PROBLEM STATEMENT

Offensive language is common in modern social media like Twitter, Facebook and Reddit. This makes abusive language detection tasks one of the most attractive challenges for the NLP community to solve and refine (Waseem et al., 2017; Wiegand et al., 2018). In this project, we research an offensive language detection challenge - Task 12 in SemEval-2020 on Multilingual Offensive Language Identification in Social Media (OffensEval-2020), held during the International Workshop on Semantic Evaluation 2020 (collocated with COLING-2020). The task included three Subtasks corresponding to the hierarchical taxonomy:

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

We work on all three Subtasks. We first try to reproduce the authors' original results as benchmarks, then try to improve their results with new model trials and new ensembling techniques. Subtask A tries to answer the following question: *Is the text in the dataset offensive (OFF) or not offensive (NOT)?* **NOT**: text that is neither offensive, nor profane; **OFF**: text containing inappropriate language, insults, or threats.

In this project, we try to first reproduce the result of a team called "KAFK" (Das et al., 2020) described in their paper *KAFK at SemEval-2020 Task 12: Checkpoint Ensemble of Transformers for Hate Speech Classification*, which mainly uses a *checkpoint ensemble* technique on their GPT-2 language model for Task A, RoBERTa for Task B and DistilRoBERTa for Task C. Then we try a few further steps along the way of their main classifier ensemble framework. This includes the attempts to improve their model by exploring models that are potentially more fit to the dataset, including GPT-2, ALBERT, RoBERTa and DistilRoBERTa for Task A, ALBERT, RoBERTa, DistilRoBERTa

[1]Project GitHub Repository: https://github.com/KennethSunn/Intro-to-NLP-Final-Project-Fall2021

for Task B and ALBERT, RoBERTa, DistilRoBERTa for Task C. We also adopted checkpoint ensembling and 'majority vote ensembling' to the classification result.

In experiment, due to compute power and time limit, we first use only 1% of randomly sampled data as the training set to reproduce a benchmark, then use the aforementioned models to try improving the KAFK team's results on Task A, B and C against the reproduced benchmark. We found that ALBERT on Task A provided very similar performance to the original GPT-2, while significantly boosting the training speed by using a lighter model. Our ensembling method helped a little in Task A. ALBERT on Task B is the best performing model and increased the performance by a significant 7% compared with the RoBERTa benchmark. Our ensembling method did not help in Task B. The benchmark DistilRoBERTa on Task C still provided the best performance compared with other models we tried. We applied our ensembling method on Task C but it did not improve the performance. The conclusions are a little different from the original authors' findings as our settings maintained or exceeded the original paper's results (benchmark).

1.2 Motivation

First, the offensive language identification task is meaningful to be researched, due to its wide occurrence in social media and its value to social media companies like Facebook and Twitter. Second, as demonstrated by the challenge organizer, offensive language in social media has a multitude of terms and definitions, as well as hierarchical taxonomy, which brings complexity to the research (Zampieri et al., 2020). This makes us specifically interested in the classifier ensemble direction that the original paper by Das et al. adopted. This method essentially ensembles the trained model at each checkpoint through the training process. However, in the original paper, the authors did not provide enough justification for their optimization process, for example, how the weight during ensemble is decided; why do they ensemble GPT-2 rather than others etc. These motivate us to explore possible improvements based on a replicate of their original work.

2 Related Work

2.1 SemEval-2020 Workshop Task 12

This project is derived from the Task 12 in International Workshop on Semantic Evaluation 2020: Multilingual Offensive Language Identification in Social Media (OffensEval 2020). This task is a successor of SemEval-2019 Task 6: OffensEval (Zampieri et al., 2019b). OffensEval-2019 used the Offensive Language Identification Dataset (OLID), containing over 14,000 English tweets annotated using a hierarchical three-level annotation schema that takes both the target and the type of offensive content into account. The competition provided a large-scale semi-supervised training dataset containing over 9 million English tweets (Rosenthal et al., 2020b). They also introduced multilingual datasets including Arabic, Danish, Greek and Turkish (note that our project only evaluates English). The size of the test set is also enlarged. This makes us possible to use the supervised test set with 3,897 tweets.

2.2 Other Offensive Content Identification Tasks

Aggression Identification: The TRAC shared task on Aggression Identification (Kumar et al., 2018) used a dataset containing 15,000 annotated Facebook posts and comments in English and Hindi for training and evaluation. For testing, a set from Facebook and another from Twitter were used to discriminate between three classes: non-aggressive, covertly aggressive and overtly aggressive. One of the best-performing systems in this competition used deep learning approaches based on convolutional neural networks (CNN), recurrent neural networks (RNN) and Long Short-term Memory (LSTM) (Aroyehun & Gelbukh, 2018).

Bullying Detection: There are quite several studies on cyber bullying detection. Xu et al. (2012) used sentiment analysis and topic models to identify relevant topics, and Dadvar et al. (2013) used user-related features such as the frequency of profanity in previous messages.

Toxic Comments: The Toxic Comment Classification Challenge was an open competition at Kaggle, providing participants with comments from Wikipedia organized in 6 classes: toxic, severe

toxic, obscene, threat, insult and identity hate. There are several studies working on this (Georgakopoulos et al., 2018), including additional training material from the aforementioned TRAC shared task (Fortuna et al., 2018).

Offensive Language Detection: The GermEval (Wiegand et al., 2018) shared task focused on offensive language identification in German tweets. A dataset containing over 8,500 annotated tweets was provided for a course-grained binary classification task where systems were trained to discriminate between offensive and non-offensive tweets. Another Subtask is about categorizing the offensive class into profanity, insult and abuse. This is very similar to the task we are studying, but contains differences in label hierarchy, label definition and language itself. Our project uses SOLID dataset which uses the hierarchical annotation model proposed in Zampieri et al. (2019a). This is beneficial to uniting these studies by highlighting their commonalities. For example, an insult targeted at an individual is commonly known as cyberbullying and insults targeted at a group are known as hate speech.

3 Approach

In this section, we introduce the models used in our experiment setup, including GPT-2, RoBERTa, ALBERT and DistilRoBERTa. Then we introduce the checkpoint ensemble method and our voting ensemble method. Lastly, we show the task-specific methods we use to generate the experiment results.

3.1 Models

3.1.1 GPT-2

GPT-2 (Radford et al., 2019) is a transformer-based language model created by OpenAI in February 2019. Using attention mechanism, it outperforms previous benchmarks for LSTM-based models. Trained on a dataset of 40 GB of web pages, GPT-2 contains 1.5 billion parameters. Because of our limited resources and GPT-2's extremely large size, our team decided to use DistilGPT-2, a compressed version of GPT-2 model. It allows us to achieve a faster training and inference time without seriously hurting performance. The authors of Distil show that it retains up to 97% performance of the original models (Sanh et al., 2019).

3.1.2 RoBERTa

RoBERTa (Liu et al., 2019b) is a transformers model pretrained on a large corpus of English data in a self-supervised fashion. It builds on BERT and modifies key hyperparameters, removing the next-sentence pretraining objective and training with much larger mini-batches and learning rates. This allows RoBERTa to improve on the masked language modeling objective compared with BERT and leads to better downstream task performance. We also use the distilled version of RoBERTa for comparison.

3.1.3 ALBERT

ALBERT (Lan et al., 2019), stands for A Lite BERT, is also a transformer-based pretrained model. ALBERT incorporates two parameter reduction techniques: one is a factorized embedding parameterization, and the other one is cross-layer parameter sharing (Lan et al., 2019). Furthermore, the parameter reduction techniques also improve regularization that stabilizes the training. Therefore, compared to BERT, ALBERT reduces the size of BERT to a huge extent but achieves significantly better performance. Interestingly, ALBERT was pretrained on the raw texts only, with no humans labeling them. Labels for texts are generated by an automatic process.

3.2 Ensemble method

3.2.1 Checkpoint Ensemble

The author used a simple *Checkpoint Ensemble* method for creating the transformer-based ensemble classifiers used in this study. *Checkpoint ensemble* introduces the benefits of ensemble methods

within a single network's training process by leveraging the validation scores. Figure 1 shows the difference between traditional ensemble and checkpoint ensemble.

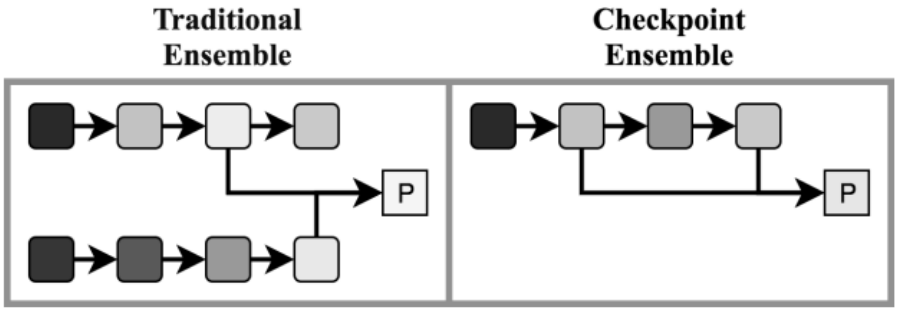

Figure 1: Traditional Ensemble and Checkpoint Ensemble (Chen et al., 2017). The shading represents validation score – lighter shades represent a better score.

In our *checkpoint ensemble* approach, we created the list of models to use in the ensemble based on the validation Macro F-1 score. We picked a model if using its predictions improves the Macro F1, otherwise not. As for prediction, we added the predictions of the chosen models and uses $argmax$ along each row to get the final prediction.

3.2.2 VOTING

Besides *Checkpoint ensemble*, we also used *voting* since we trained several different models on a small proportion of the whole training dataset. The idea behind *voting* is to combine conceptually different classifiers and use a majority vote to predict the class labels. This method can be useful for a set of equally well performing model in order to balance out their individual weaknesses.

3.3 TASK-SPECIFIC METHOD

As for Subtask A, we used checkpoint ensemble of the distilled GPT-2 as our base model for reproduction. Then we tried all three models discussed before, and applied voting as the final prediction.

As for Subtask B, we first chose RoBERTa to be our base model. However, the performance was not as good as we expected due to our compressed training data. Then, we tried DistilRoBERTa and ALBERT model. The ALBERT model generates the result with highest accuracy. We also tried the voting mechanism, but the voting accuracy is lower than the ALBERT accuracy.

As for Subtask C, the classifier was built on the pre-trained distilled version of RoBERTa Sequence Classifier, coupled with a slightly modified Checkpoint Ensemble. Moreover, we also add a drop out to the attention masks in order to reduce the effect of grammar error in Tweets. We also replace distilled RoBERTa classifier by RoBERTa, ALBERT classifier, and voting of these three model. However, the baseline model still performs the best within them.

4 EXPERIMENTS

4.1 DATASETS AND EVALUATION

In this section, we introduce the SOLID dataset we use for experiment, and the evaluation metrics we apply for Subtask A, B and C.

4.1.1 SUBTASK A

We use SOLID dataset (Rosenthal et al., 2020a) as our base dataset. SOLID contains over 9,000,000 tweets in the training set. Due to limited time and computing resources, we randomly sampled around 1% of the whole training set as our new training set, and maintained the positive-to-negative ratio. SOLID has only 3897 tweets for testing, so we keep the test set unchanged. The detailed data distribution is shown in Table 1. The positive class is OFF for Subtask A.

	OFF	NOT
Training	14488	76402
Test	1080	2807

Table 1: Training and test dataset for Subtask A

4.1.2 SUBTASK B

For task B, we have 56692 training tweets shown in Table 2. We use roughly 90% of the data to do training, and the rest to do validation. The training data is unbalanced because most of the offensive sentences have their targets. The testing set is provided by the official website. It contains 1422 data points.

	TIN	UNT
Training	40491	10674
Validation	2243	591
Test	850	572

Table 2: Training and test dataset for Subtask B

4.1.3 SUBTASK C

For task C, we have 51165 training tweets shown in Table 3. We use roughly 90% of the data to do training, and the rest to do validation. The testing set is provided by the official website. It contains 850 data points.

	IND	GRP	OTH
Training	41307	6746	3112
Validation	2288	374	172591
Test	580	190	80

Table 3: Training and test dataset for Subtask C

4.2 MODEL AND TRAINING DETAILS

In this section, we show the models we explored in Subtask A, B and C, together with the training setup details including hyperparameter settings and other configurations.

4.2.1 SUBTASK A

In the reproduction task, we first converted each text sample into a sentence-matrix by extracting 786 dimensional word embeddings from pre-trained DistilGPT-2. These were then fed into a dense layer having 2 units which makes the classification, because the number of classes is 2. We fine-tuned the entire model using a cross-entropy loss-function with a small learning rate of 0.0001 for

10 epochs and applied Checkpoint Ensemble as described in 3.2.1. We also used Ranger which is a combination of two optimizers, RAdam(Liu et al., 2019a) wrapped with LookAhead (Zhang et al., 2019). We set the (k, α) parameters of the optimizer to (5, 0.5). The batch-size and maximum-sequence-length were set to 14 and 349 respectively.
For improvement, we tried RoBERTa, DistilRoBERTa and ALBERT classifier and their ensembling, and we did not change most of the hyperparameter settings. We only changed maximum-sequence-length based on different tokenizers, 283 for ALBERT, 278 for RoBERTa and 318 for distilRoBERTa.

4.2.2 SUBTASK B

For Subtask B, the problem is to identify whether the offensive is targeted to a group or not. This is a binary classification problem. The result is either "TIN", meaning targeted, or "UNT", meaning untargeted. We mapped the string "TIN" to integer 0, and string "UNT" to integer 1. Then, for each tweet data, we add the special token '$< s >$', the start symbol, to the beginning the tweet, and '$< /s >$', the end symbol, to the end of the tweet. After that, we used the Tokenizer library developed by Hugging Face to vectorize our training text.

For the neural net, we set the number of epoch to be 5, because the pretrained RoBERTa seems very likely to overfit the data. We have our learning rate being 0.0001, and batch size being 16. We used Rectified Adam, known as RAdan, as our base optimizer. To prevent overfitting, we add the early stopping mechanism of patience equal to 4, meaning that our model will stop running when validation accuracy dropping 4 times. We have tried RoBERTa, DistilRoBERTa and ALBERT classifier. Using ensembling method to aggregate three results together, we generate an ensembling solution to this Subtask.

4.2.3 SUBTASK C

In subask C, we add a drop out to the attention masks. With probability p, we randomly dropped d% of attention masks of the tokens in the sentence. We set p to 0.50 and d to 30%. We used attention mask dropout Similar to the original dropout technique (Hinton et al., 2012), it was only used during training. The maximum sequence length was set to 245. The pre-trained model 2027 was fine-tuned using a cross-entropy loss function for 10 epochs with early stopping patience set to 4. Generally we just need to run no more than 7 epochs in practice. We used a batch-size of 120 and learning rate of 0.0001 with Ranger optimizer. Like in previous classifiers, we set the (k, μ) parameters of the optimizer to (5, 0.5).

4.3 RESULTS

In this section, we show the results we get on Subtask A, B and C, together with the models we explored. We also show the results after adding our voting ensemble technique.

4.3.1 SUBTASK A

In the original paper of KAFK, they used GPT-2 as their base model and achieved a Macro F1 of 0.90989 on the test set. As a reproduction of Subtask A, we achieved a Macro F1 score of 0.91228, which is a little higher than their score. In addition, we tried several different pretrained models and applied ensemble method. Due to the poor performance of RoBERTa, we did not include it in the ensembling. Fortunately, we reached a higher Macro F1 score. The detailed results are shown in Table 4.

4.3.2 SUBTASK B

In the original paper of KAFK, they used RoBERTa as their base model for Subtask B and achieved Macro F1 of 0.55 on the test set. From the table below 5, reproducing the model from KAFK, we achieved the Macro F1 score of 0.50, which is not as good as expected. We realized that this benchmark could be improved by using more advanced classifiers. Hence, we tried another transformer-based pretrained model. On test set, both ALBERT and Distil-RoBERTa achieve 0.57 of Macro F1,

Model	Train	Dev	Test
GPT-2	0.9693	0.9663	0.91228
RoBERTa	0.4574	0.4567	0.50898
Distil-RoBERTa	0.9632	0.9667	0.91176
ALBERT	0.8396	0.9109	0.90664
Ensemble	NaN	NaN	**0.91296**

Table 4: Subtask A Macro F1 comparision

higher than the benchmark of KAFK paper. This result show that we have successfully improved the work from KAFK paper.

Model	Train	Dev	Test
RoBERTa	0.7785	0.8113	0.50
Distil-RoBERTa	0.7842	0.8461	**0.57**
ALBERT	0.7302	0.8324	**0.57**
Ensemble	NaN	NaN	0.54

Table 5: Subtask B Macro F1 comparision

4.3.3 SUBTASK C

In the original paper of KAFK, they used Distil-RoBERTa as their base model for Subtask C and achieved Macro F1 of 0.62 on the test set. From the table below 6, reproducing the model from KAFK, we achieved the Marco F1 score of 0.65, which is worse than the KAFK's result, since we only use less than 30% data to train the model. We do not improve the F1 value by changing classifiers types. We try ALBERT and RoBERTa and Macro F1 are 0.62 and 0.57, which means both of them perform worse than baseline model. When using ALBERT, the difference between Macro F1 on training set and validation set indicates that the model overfits.

Model	Train	Dev	Test
Distil-RoBERTa	0.7369	0.7693	**0.65**
RoBERTa	0.7622	0.7659	0.62
ALBERT	0.8303	0.7716	0.57
Ensemble	NaN	NaN	0.64

Table 6: Subtask C Macro F1 comparision

5 ANALYSIS

As for Subtask A, all pre-trained transformers we try separately are able to correctly classify almost every offensive sample. Among all the single models, GPT-2 performs the best, which is able to cross the 0.912 mark. Ensembling improves the test set Macro F1 from 0.91228 to 0.91296. Using the full dataset instead of just 1% might have resulted in a better score. This shows the need for techniques like model quantization to deal with massive datasets.
The models in Subtask B and Subtask C perform quite poorly, because they overfit the data as the difference between the dev set and test set f1 scores is quite large. As for Subtask B, the classifier performs the best when changing the baseline model to ALBERT classifier. When built the classifier based on distilled RoBERTa or by ensembling all models, the Macro-F1 improves slightly, but it still performs poorly. Even using ALBERT, the classifier fails to distinguish properly between targeted and untargeted samples. We found that samples that contained the "@USER" token were

mostly mistaken as targeted.
As for Subtask C, we replace the baseline model by RoBERTa and ALBERT classifier and ensemble them together, but none of them performs well than baseline model itself. In addition, we found that the model cannot work when the text consists of emojis. The model might have performed better when the emoji are translated into its meanings.

6 Contribution

The contribution of this project is twofold:

1. The results achieved by the KAFK team can be reproduced with much fewer data and a lighter training model. We reproduced the Macro F1 score very close to the original ones on Subtask A, B and C using only 1% randomly sampled data, compared with 70% of the data in the original paper.
2. We show that the performance can be further improved along the way of using model ensemble techniques. We achieved higher performance by properly choosing models and configuring the training process on Subtask A and B. We also proposed our own voting ensemble technique which made the Subtask A performance better.

Author Contributions

Haozhan and Yi are responsible for the reproduction and improvement for Task A. Also, Haozhan is the owner for our team Github repository. Yi is also responsible for the file management and library update for our project. Yifei is responsible for the reproduction and improvement for Task B. Haofei is responsible for the reproduction and improvement for Task C.

References

Segun Taofeek Aroyehun and Alexander Gelbukh. 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)*, pp. 90–97, 2018.

Hugh Chen, Scott M. Lundberg, and Su-In Lee. Checkpoint ensembles: Ensemble methods from a single training process. *CoRR*, abs/1710.03282, 2017. URL http://arxiv.org/abs/1710.03282.

Maral Dadvar, Dolf Trieschnigg, Roeland Ordelman, and Franciska de Jong. Improving cyberbullying detection with user context. In *European Conference on Information Retrieval*, pp. 693–696. Springer, 2013.

Kaushik Amar Das, Arup Baruah, Ferdous Ahmed Barbhuiya, and Kuntal Dey. Kafk at semeval-2020 task 12: Checkpoint ensemble of transformers for hate speech classification. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pp. 2023–2029, 2020.

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

Spiros V Georgakopoulos, Sotiris K Tasoulis, Aristidis G Vrahatis, and Vassilis P Plagianakos. Convolutional neural networks for toxic comment classification. In *Proceedings of the 10th hellenic conference on artificial intelligence*, pp. 1–6, 2018.

Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580, 2012. URL http://arxiv.org/abs/1207.0580.

Ritesh Kumar, Atul Kr Ojha, Shervin Malmasi, and Marcos Zampieri. Benchmarking aggression identification in social media. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pp. 1–11, 2018.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A lite BERT for self-supervised learning of language representations. *CoRR*, abs/1909.11942, 2019. URL http://arxiv.org/abs/1909.11942.

Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. *CoRR*, abs/1908.03265, 2019a. URL http://arxiv.org/abs/1908.03265.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019b. URL http://arxiv.org/abs/1907.11692.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

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

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

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

Zeerak Waseem, Thomas Davidson, Dana Warmsley, and Ingmar Weber. Understanding abuse: A typology of abusive language detection subtasks. *arXiv preprint arXiv:1705.09899*, 2017.

Michael Wiegand, Melanie Siegel, and Josef Ruppenhofer. Overview of the germeval 2018 shared task on the identification of offensive language. 2018.

Jun-Ming Xu, Kwang-Sung Jun, Xiaojin Zhu, and Amy Bellmore. Learning from bullying traces in social media. In *Proceedings of the 2012 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pp. 656–666, 2012.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. Predicting the type and target of offensive posts in social media. *arXiv preprint arXiv:1902.09666*, 2019a.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). *arXiv preprint arXiv:1903.08983*, 2019b.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. Semeval-2020 task 12: Multilingual offensive language identification in social media (offenseval 2020). *arXiv preprint arXiv:2006.07235*, 2020.

Michael R. Zhang, James Lucas, Geoffrey E. Hinton, and Jimmy Ba. Lookahead optimizer: k steps forward, 1 step back. *CoRR*, abs/1907.08610, 2019. URL http://arxiv.org/abs/1907.08610.