Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying, pages 49–54
Language Resources and Evaluation Conference (LREC 2020), Marseille, 11–16 May 2020

IRIT at TRAC 2020

Faneva Ramiandrisoa[1,2], **Josiane Mothe**[1,3]
[1]IRIT, UMR 5505 CNRS Université de Toulouse, France
[2] Université d'Antananarivo
[3] ESPE, UT2J
{faneva.ramiandrisoa, josiane.mothe}@irit.fr

Abstract

This paper describes the participation of the IRIT team in the TRAC *(Trolling, Aggression and Cyberbullying)* 2020 shared task (Bhattacharya et al., 2020) on Aggression Identification and more precisely to the shared task in English language. The shared task was further divided into two sub-tasks: (a) aggression identification and (b) misogynistic aggression identification. We proposed to use the transformer based language model BERT (*Bidirectional Encoder Representation from Transformer*) for the two sub-tasks. Our team was qualified as twelfth out of sixteen participants on sub-task (a) and eleventh out of fifteen participants on sub-task (b).

Keywords: Information systems, Information retrieval, Social media, Cyber-agression, TRAC Trolling, Aggression and cyberbullying, BERT

1. Introduction

Social media has become one of the key ways people communicate and share opinions (Pelicon et al., 2019). These platforms, such as Twitter or WhatsApp, allow people to fully or partially hide their identity and this leads to the proliferation of abusive language and an increase of aggressive and potential harmful content on social media (Zhu et al., 2019). Automatically monitoring user-generated content in order to help moderate social media content is thus an important topic and has attracted significant attention in recent years as evidenced in recent publications (Mishra et al., 2019; Struß et al., 2019; Zampieri et al., 2019). Several studies focus on the automatic detection of abusive language such as hate speech (Warner and Hirschberg, 2012), cyberbullying (Dadvar et al., 2013), aggression (Kumar et al., 2018). Different evaluation forums have also been proposed in order to foster the development of systems to help abusive language detection. Among them, we can mention TRAC (Kumar et al., 2018), GermEval (Struß et al., 2019), and SemEval-2019 Task 6 (Zampieri et al., 2019).

In this work, we report the work we carried out on aggression identification and our participation in the second edition of TRAC *(Trolling, Aggression and Cyberbullying)*. The objective of TRAC shared task is to automatically detect aggression in text. During the first edition, the objective was to develop a classifier that could make a 3-way classification between ”Overtly Aggressive”, ”Covertly Aggressive” and ”Non-aggressive” text data. Deep learning approaches were widely used during the shared task and achieved the best performance (Kumar et al., 2018).

For the second edition of TRAC (Bhattacharya et al., 2020), the organizers proposed two sub-tasks: (a) aggression identification and (b) misogynistic aggression identification. The objective of sub-task (a) is the same as in the first edition of TRAC which is to classify the text according to 3 classes. The objective of sub-task (b) is to develop a binary classifier for classifying the text as ”gendered” or ”non-gendered”.

For our participation in this second edition of TRAC, we proposed variants of a model that use transfer learning based on the BERT model (more details in Section 4.) to tackle the problem of the two sub-tasks.

The rest of this paper is organized as follows: Section 2. presents related work in the area of abusive language detection; Section 3. describes the TRAC data set as well as the pre-processing we developed; Section 4. describes the methodology we propose to answer the TRAC challenge as well as the submitted runs; Section 5. presents the results and discusses them; finally, Section 6. concludes this paper and presents some future work.

2. Related Work

Automatically detecting abusive language from textual analysis has gained momentum (Maitra and Sarkhel, 2018). Schmidt and Wiegand (2017) present a survey on hate speech detection using Natural Language Processing (NLP). The authors report that supervised learning approaches, such as support vector machines (SVM) and recurrent neural networks, are predominantly used to solve the the problem. They also report that features such as simple surface features (eg. bag of words, n-grams, etc.), word generalization (eg. word embedding, etc.), knowledge-based features (eg. ontology, etc.), are widely used for hate speech detection. On the other hand, Mishra et al. (2019) report an overview of abuse detection methods as well as a detailed overview of data sets that are annotated for abusive language detection. The authors noticed that many researchers have relied on text-based features for abuse detection while the recent state of the art approaches rely on deep learning approaches such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN).

Several European projects and workshops are tackling this challenge (Laurent, 2020; Hoang et al., 2020) and a number of evaluation forums that deal with offensive content, hate speech and aggression have been organized recently. These initiatives confirm the increasing interest in this field (Pelicon et al., 2019). To solve this challenge, participants heavily use deep learning techniques and achieve the best effectiveness. This is the case in GermEval (Struß et al., 2019), SemEval-2019 Task 6 (Zampieri et al., 2019) and the first edition of TRAC (Kumar et al., 2018).

The first edition of TRAC (Kumar et al., 2018), denoted as TRAC 2018 in the remainder of this paper, focused on aggression identification considering both English and Hindi languages. The objective was to classify texts into three classes: **Non-Aggressive (NAG)**, **Covertly Aggressive (CAG)**, and **Overtly Aggressive (OAG)**. Facebook posts and comments were provided for training and validation, while, for testing, two different sets, one from Facebook and one from Twitter, were provided. The best performance during the shared task in English language was achieved with deep learning approaches both on Facebook and Twitter test sets (Kumar et al., 2018). During this shared task, apart from deep learning approaches, participants considered classical machine learning methods (eg. Random Forests) based on features as in (Ramiandrisoa and Mothe, 2018; Arroyo-Fernández et al., 2018; Risch and Krestel, 2018). In Hindi language, Logistic regression over lexical features gave the best performance on Facebook set and second best performance on Twitter sets (Samghabadi et al., 2018).

In the next section, we will describe the second edition of TRAC, denoted as TRAC 2020 in the remainder of this paper, in which we participated as well as the methodology we adopted.

3. Data and preprocessing

In this section, we detail the data set used during the second edition of TRAC as well as how we preprocessed it for text cleaning and added external data to increase the training data set.

3.1. Data set

The second edition of the TRAC shared task (Bhattacharya et al., 2020) (TRAC 2020) was divided into two sub-tasks, namely aggression identification (sub-task (a)) and misogynistic aggression identification (sub-task (b)). The organizers provided a new data set, different from the ones made available during TRAC 2018. The training and validation sets are composed of 5,000 aggression-annotated data from social media each in Bangla (in both Roman and Bangla script), Hindi (in both Roman and Devanagari script) and English. The test set is composed of 1,200 data from social media each in Bangla, Hindi and English. During this edition, we used the English parts only.

For sub-task (a), each text data is labeled as Non-Aggressive (NAG), Covertly Aggressive (CAG), or Overtly Aggressive (OAG). The label NAG is used for text that is generally not intended to be aggressive, CAG is used for text that contains hidden or indirect aggression and finally OAG is used for text that contains open and direct aggression.

For sub-task (b), each text data is labeled as gendered (GEN) or non-gendered (NGEN). The text instances used in both sub-tasks are the same, just labels are different.

Table 1 details the English data set used in this work.

Number of	Train	Validation	Test
texts	4,263	1,066	1,200
OAG	435	113	286
CAG	453	117	224
NAG	3,375	836	690
GEN	309	73	175
NGEN	3,954	993	1,025

Table 1: Distribution of training, validation and test data on English TRAC 2020 data collection.

3.2. Preprocessing

In this section, we describe the preprocessing steps we applied to the data instances in order to clean them. We also describe the two methods we used we used to enlarge the data set in order to get a balanced data set because as we can see in table 1, classes are imbalanced. In various applications balanced data sets have been shown to perform better than imbalanced ones (Chawla et al., 2002; Khan et al., 2017), and various methods have been developed to overcome data imbalance (Prati et al., 2015).

Data Preprocessing : we converted all texts into lowercase and all ”URL” are substituted by ”http”. We also substituted emoticon into their text equivalents by using the online emoji project on github[1]. We treated the substituted phrase as regular English phrase. Finally, we removed non UTF-8 words.

Enlarging the data sets : we added more data in order to increase the number of items in low populated classes. We enlarged the data set for sub-task (a) only because, in that case, we have the data set of TRAC 2018 (the first edition) at our disposal which is annotated with the same class labels as used for sub-task (a).

We proposed two methods to complement the data set for the sub-task (a):

(i) for the first method, we used all the data set provided during the first edition, i.e. we used the training, validation and the two test sets. For this, we took all the text data labeled as CAG or OAG from the TRAC 2018 sets and added them to the training data of TRAC 2020. The resulting data set is called first enlarged data set and is composed of 14,039 texts where there is 6,305 CAG, 4,359 OAG and 3,375 NAG.

(ii) for the second method , we used only the training set of TRAC 2018. More precisely we took some of the text data labeled as CAG or OAG, respectively 2,922 and 2,940, and added them to the training data of TRAC 2020 in order to have the same number of instances per classes to train the model. The resulting data set is called second enlarged data set and is composed of 10,125 of text data where the number of items in each class is 3,375.

In the next section, we describe the models associated to the runs we submitted to TRAC 2020 shared task.

4. Methodology

We submitted five runs during the TRAC 2020 shared task, three for sub-task (a) and two for sub-task (b). These five

[1] `https://github.com/carpedm20/emoji`, accessed on February, 04^{th} 2020

runs are based on a system that uses BERT model (Devlin et al., 2019). More precisely, we used the BERT model combined in parallel with a low-dimensional multi-head attention layer (Projected Attention Layers or PALs) which was proposed by (Stickland and Murray, 2019) and denoted as BERT_Pals in the remainder of this paper. BERT_Pals was designed for multi-task learning but it can be used for a single task learning. We used BERT_Pals because it gave better result than just BERT on the validation set during the model training.

To understand the BERT_Pals model, let us first explain the original BERT model architecture. The original BERT model is simply a stack of BERT layers. In the literature, two types of BERT architecture are widely used: BERT-large (composed of 24 BERT layers) and BERT-base (composed of 12 BERT layers).

BERT takes in a sequence of tokens[2] and outputs a vector representation of that sequence. Each token in the sequence has its own hidden vector and these hidden vectors are transformed with the first BERT layer to get the first hidden states. The first hidden states are transformed through successive BERT layers and get at the end the final hidden states[3].

A BERT layer follows a transformer architecture based on a multi-head attention layer (Vaswani et al., 2017). The multi-head layer consists of n different dot-product attention mechanisms

The BERT_Pals model modify the original BERT by adding a task-specific function in parallel with each BERT layer. Figure 1 provides an illustration of the architecture of the BERT_Pals model with only two layers for simplicity.

For a more detailed explanation of the BERT_Pals model, we refer readers to (Stickland and Murray, 2019). The code of (Stickland and Murray, 2019) is also open-source and is available in github[4].

In their work, Stickland and Murray (2019) used the same configuration of BERT-base architecture as in (Devlin et al., 2019). However, in our work, we changed it to the configuration of BERT-large architecture because Devlin et al. (2019) stated that BERT-large achieved better performances than BERT-base. For the other configurations, which are specific to BERT_Pals, we used the same as in (Stickland and Murray, 2019)' work, except the task sampling that is useless in the case of a single task. Indeed, in our work, we train the model on one task only so we do not need to use the tasks sampling method which is essential for multi-task learning.

4.1. Runs submitted to TRAC 2020

Sub-task (a): For this sub-task, we submitted three runs obtained from BERT_Pals models that were trained with a mini-batch size of 32, a maximum sequence length of 40 tokens, Adam optimizer with learning rate of 2e-5, number of epochs of 3 and learning rate warm-up over the first 10 % of the steps. The difference between these three models is the training data on which they were trained. The first model (model_A_1) was trained on the training data of TRAC 2020 only, while the second model (model_A_2) was trained on the first enlarged data and finally the last model (model_A_3) was trained on the second enlarged data.

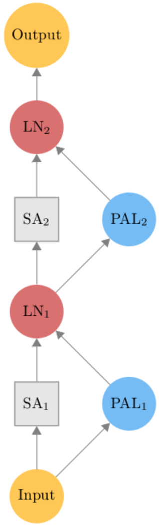

Figure 1: Schematic diagram (Stickland and Murray, 2019) of adding Projected Attention Layers or PALs in parallel with self-attention (SA) layers in a BERT model, with only two layers for simplicity. LN refers to layer-norm.

Sub-task (b): For this sub-task, we submitted two systems also obtained from BERT_Pals models trained with a mini-batch size of 32, a maximum sequence length of 40 tokens, adam optimizer with learning rate of 2e-5, number of epochs of 3 and learning rate warm-up over the first 10% of steps. The difference between the models was also the data on which they were trained. The first model (model_B_1) was trained on the training data of TRAC 2020 only while the second model (model_B_2) was trained on both the training and validation data of TRAC 2020.

The training was carried out on an Nvidia Geforce GTX 1080TI GPU and took between 3 to 6 minutes in total.

In the next sections, we report the results we obtained during the TRAC 2020 shared task.

5. Results

This section reports the results our team obtained on the English data sets when participating to TRAC 2020. More details on other participants' systems are presented in (Kumar et al., 2020).

Table 2 presents the results we obtained for sub-task (a) and table 3 the ones for sub-task (b).

For sub-task (a), we can see that the model (model_A_3) trained on the balanced data set gives the best performance (weighted F1 of 0.6352). Nonetheless, this model achieved just the twelfth rank over sixteen participants runs during the TRAC 2020 challenge, where the best team achieved a weighted F1 of 0.8029.

For sub-task (b), we can see that the model (model_B_1) trained on the training data of TRAC 2020 only gives the

[2] A special classification embedding ([CLS]) is always inserted as the first token of every sequence.

[3] Only the final hidden state of [CLS] is used as the aggregate sequence representation for classification or regression tasks.

[4] https://github.com/AsaCooperStickland/Bert-n-Pals

System	F1 (weighted)	Accuracy
model_A_1	0.6179	0.6958
model_A_2	0.5894	0.645
model_A_3	**0.6352**	**0.6967**

Table 2: Results of our three models for sub-task (a) on English test set. Bold font highlights the best performance.

best performance according to weighted F1 (0.8202) while the model (model_B_2) trained on both training and validation sets of TRAC 2020 gives the best result when considering accuracy score. Nonetheless the model_B_1 achieved just eleventh rank over fifteen participants runs during the TRAC 2020 challenge where the best team achieved a weighted F1 of 0.8716. We should mention that the performance of our models are closer to the best in this sub-task (b) than in sub-task (a).

System	F1 (weighted)	Accuracy
model_B_1	**0.8202**	0.8433
model_B_2	0.7870	**0.8542**

Table 3: Results of our two models for sub-task (b) on English test set. Bold font highlights the best performance.

5.1. Discussion

When analyzing the results of our models according to confusion matrix on sub-task (a), we can see that they hardly identify CAG. From the confusion matrix presented in figure 2, we can see that our best model confuses NAG and CAG, and the same holds for CAG and OAG. It confirms our hypothesis, when reading some texts from the training set, that it is easier to distinguish texts labelled as NAG from texts labelled as OAG than from texts labelled as CAG. This difficulty to detect CAG is the main weakness of our model, this is why our ranking is so poor during the competition. With BERT_Pals, we are able to detect the six CAG while using normal/original BERT, we do not even predict CAG at all.

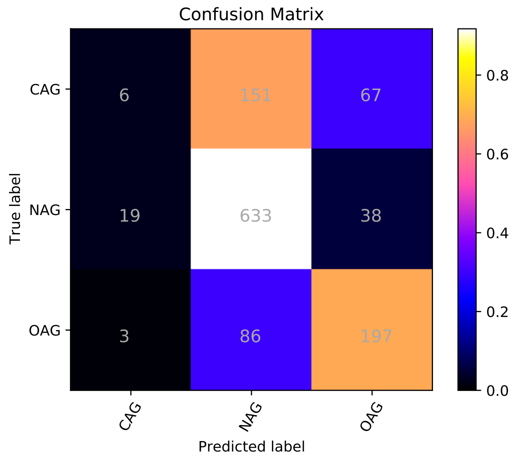

Figure 2: Heatmap of the confusion matrix for our best model (model_A_3) on sub-task (a)

For the sub-task (b), our model performs better than for sub-task (a) but hardly predict GEN cases as we can see in Figure 3 for our best model; the same holds for the other model which does not even predict GEN cases at all. This is likely to be due to the imbalanced nature of the data set as there are about thirteen times more NGEN cases than GEN cases. This finding confirms what (Pelicon et al., 2019) said in their work that transfer learning with BERT does not perform well on imbalanced data sets.

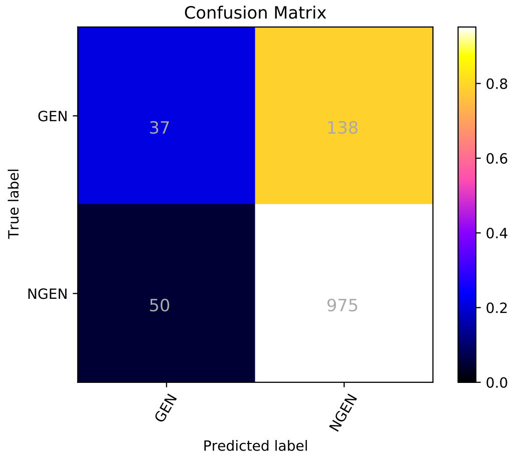

Figure 3: Heatmap of the confusion matrix for our best model (model_B_1) on sub-task (b)

6. Conclusion and Future Work

In this paper, we presented our participation in the second edition of TRAC shared task in English language for both sub-tasks: (a) aggression identification and (b) misogynistic aggression identification. We used BERT model to tackle the problem of the two sub-tasks. On the first sub-task, our best model achieved a weighted F1 of 0.6352 which ranked our team on the twelfth place over sixteen participants runs. On the second sub-task, our best model achieved a weighted F1 of 0.8202 which placed our team to the eleventh rank over fifteen participants runs. However, in this second task, the performance of our models are closer to the best.

We noticed that the class imbalances in the data set had a significant impact on the performance of our models. Adding instances from an external data set to the minority classes on sub-task (a) proved to be the most consistent technique to improve the performance of our models. Nevertheless on this sub-task, our models met another problem which is to differentiate *covertly* aggressive cases and non-aggressive cases.

Our aim for the short term future work is to balance the data set for sub-task (b) in order to see if it improves the results. We also plan to test different techniques to tackle the problem of imbalanced data sets. For long term future work, we aim to make our proposed models more robust to imbalanced data set. We also plan to investigate why it is hard for our models to detect covertly aggressive by analyzing the text in the training data set with keywords extraction technique such as the one we developed in (Mothe et al., 2018). We may also investigate more on keywords by using them instead of long text as input to our models.

Ethical issue. While TRAC challenge has its proper ethical policies, detecting aggressive content from user's posts raises ethical issues that are beyond the scope of the paper.

7. Bibliographical References

Arroyo-Fernández, I., Forest, D., Torres-Moreno, J., Carrasco-Ruiz, M., Legeleux, T., and Joannette, K. (2018). Cyberbullying detection task: the EBSI-LIA-UNAM system (ELU) at coling'18 TRAC-1. In *Proceedings of the 1st Workshop on Trolling, Aggression and Cyberbullying, TRAC@COLING 2018*, pages 140–149.

Bhattacharya, S., Singh, S., Kumar, R., Bansal, A., Bhagat, A., Dawer, Y., Lahiri, B., and Ojha, A. K. (2020). Developing a multilingual annotated corpus of misogyny and aggression.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.

Dadvar, M., Trieschnigg, D., Ordelman, R., and de Jong, F. (2013). Improving cyberbullying detection with user context. In *Advances in Information Retrieval - 35th European Conference on IR Research, ECIR 2013, Moscow, Russia, March 24-27, 2013. Proceedings*, pages 693–696.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. (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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Hoang, T. B. N., Marchand, P., Milard, B., and Mothe, J. (2020). *Workshop on Machine Learning for Trend and Weak Signal Detection in Social Networks and Social Media, Toulouse, France, Feb. 27-28, 2020, Proceedings*. -.

Khan, S. H., Hayat, M., Bennamoun, M., Sohel, F. A., and Togneri, R. (2017). Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE transactions on neural networks and learning systems*.

Kumar, R., Ojha, A. K., Malmasi, S., and Zampieri, M. (2018). Benchmarking aggression identification in social media. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying, TRAC@COLING 2018, Santa Fe, New Mexico, USA, August 25, 2018*, pages 1–11.

Kumar, R., Ojha, A. K., Malmasi, S., and Zampieri, M. (2020). Evaluating aggression identification in social media. In Ritesh Kumar, et al., editors, *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying (TRAC-2020)*, Paris, France, may. European Language Resources Association (ELRA).

Laurent, M. (2020). Hatemeter Project: Analysis of hate speech on twitter at the crossroads of computer science, humanities and social sciences. In *Workshop on Machine Learning for Trend and Weak Signal Detection in Social Networks and Social Media*, 2.

Maitra, P. and Sarkhel, R. (2018). A k-competitive autoencoder for aggression detection in social media text. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying, TRAC@COLING 2018, Santa Fe, New Mexico, USA, August 25, 2018*, pages 80–89.

Mishra, P., Yannakoudakis, H., and Shutova, E. (2019). Tackling online abuse: A survey of automated abuse detection methods. *CoRR*, abs/1908.06024.

Mothe, J., Ramiandrisoa, F., and Rasolomanana, M. (2018). Automatic keyphrase extraction using graph-based methods. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing, SAC 2018, Pau, France, April 09-13, 2018*, pages 728–730.

Pelicon, A., Martinc, M., and Novak, P. K. (2019). Embeddia at semeval-2019 task 6: Detecting hate with neural network and transfer learning approaches. In *Proceedings of the 13th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2019*, pages 604–610.

Prati, R. C., Batista, G. E., and Silva, D. F. (2015). Class imbalance revisited: a new experimental setup to assess the performance of treatment methods. *Knowledge and Information Systems*, 45(1):247–270.

Ramiandrisoa, F. and Mothe, J. (2018). IRIT at TRAC 2018. In *Workshop on Trolling, Aggression and Cyberbullying, in International Conference of Computational Linguistics (TRAC@COLING 2018)*, pages 19–27, http://www.aclweb.org. Association for Computational Linguistics (ACL).

Risch, J. and Krestel, R. (2018). Aggression identification using deep learning and data augmentation. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying, TRAC@COLING 2018, Santa Fe, New Mexico, USA, August 25, 2018*, pages 150–158.

Samghabadi, N. S., Mave, D., Kar, S., and Solorio, T. (2018). Ritual-uh at TRAC 2018 shared task: Aggression identification. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying, TRAC@COLING 2018, Santa Fe, New Mexico, USA, August 25, 2018*, pages 12–18.

Schmidt, A. and Wiegand, M. (2017). A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media, SocialNLP@EACL 2017, Valencia, Spain, April 3, 2017*, pages 1–10.

Stickland, A. C. and Murray, I. (2019). BERT and pals: Projected attention layers for efficient adaptation in multi-task learning. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 5986–5995.

Struß, J. M., Siegel, M., Ruppenhofer, J., Wiegand, M., and Klenner, M. (2019). Overview of germeval task 2, 2019 shared task on the identification of offensive language. In *Proceedings of the 15th Conference on Natural Lan-*

guage Processing, KONVENS 2019, Erlangen, Germany, October 9-11, 2019.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Warner, W. and Hirschberg, J. (2012). Detecting hate speech on the world wide web. In *Proceedings of the Second Workshop on Language in Social Media*, pages 19–26, Montréal, Canada, June. Association for Computational Linguistics.

Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., and Kumar, R. (2019). Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). In *Proceedings of the 13th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2019*, pages 75–86.

Zhu, J., Tian, Z., and Kübler, S. (2019). UM-IU@LING at SemEval-2019 task 6: Identifying offensive tweets using BERT and SVMs. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 788–795, Minneapolis, Minnesota, USA, June. Association for Computational Linguistics.