Are we spreading Reach or Hate? Towards Generating Contextually Relevant Sequence of Reactions for LinkedIn

Abstract

LinkedIn, a platform for posting job-related content, is being used heavily by job-seekers and recruiters for job/candidate search. Due to a large number of active users, LinkedIn gets flooded with various posts, and one can react with any of the available LinkedIn reactions to a particular post with 'like' being the default option followed by other reactions. However, most of the time, such reactions often tend to be against the context of the post resulting in the spread of negative sentiment. To counter this, we propose a system that identifies the context/sentiment of the post and generates a relevant sequence of reactions based on the context thereby enabling the users to see the most relevant reactions first. We show that our system is able to achieve an F1 score of 0.88 while classifying the sentiment of the post.

1 Introduction

LinkedIn, which is a platform for job search is being used actively by people from different domains to post work-related content. As of today, LinkedIn has more than 850 million members in more than 200 countries (LinkedIn, 2022). Due to such large number of active users, LinkedIn gets flooded with different work-related content. Some of these posts are positive, some are negative and some are neutral. There are different emojis/reactions which have already been provided by LinkedIn to react to each of these posts with 'like' being the deafult option. Most of the time, people react with any of the available emojis/reactions just to increase the reach of the post. This might be harmful in a way when the post has a negative context, and the user reacts to the post with an emoji such as 'celebrates', 'likes', 'funny', etc. For instance, a post mentioning that a candidate has been laid off and is looking for a new role might have a negative psychological impact on the candidate if the post is flooded with emojis such as 'funny', 'celebrates', 'likes', etc. A study conducted by researchers from University of Ottawa shows how reacting with negative emojis produce a negative perception of the sender regardless of their true intent (Boutet et al., 2021). Similarly, Tetiana (Avdieieva, 2021) also describes how emojis can be a cause of hate, bullying and harassment, and how it can provoke anger and discourse amongst the public (Matamoros-Fernández, 2018).

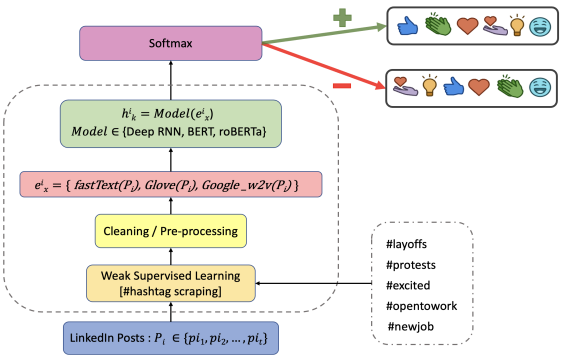

Figure 1: Model overview

A statistical analysis performed on our generated dataset shows that on an average more than 50% of the people react with an opposite reaction to a negative sentiment post. For the posts with a positive sentiment, on an average close to 5% of the people react with an opposite reaction. Figure 2 shows one such example where more than 50% people reacted with irrelevant reactions ('like', 'celebrate', 'love', 'funny' reactions) to a negative sentiment post. It is unclear if they react with such reactions intentionally or unintentionally. It is worth investigating whether most people like the posts because 'like' is the default option or they react intentionally. Either of this results in a spread of opposite opinion. One option to

counter this can be to show only the relevant reactions based on the context of the post. However, this might not be a feasible option since there are already very few emojis available. Additionally, there are only 1 or 2 emojis which support negative sentiment. We, thus propose an alternative option where we generate a sequence of emojis based on the post sentiment such that the most relevant reactions are shown first followed by the least. This will not only motivate the user to react with the right reaction but will also help us investigate our hypothesis in the future as to whether people 'like' the post intentionally or unintentionally. In addition, this will also not limit the users from reacting with only specific reactions.

We refer LinkedIn (LinkedInHelp, 2022) to understand the meaning of each emoji and its context. Based on this, we classify the reactions into two groups positive and negative, where negative denotes the sequence of emojis to be displayed if the post has a negative context and positive if the post has a positive context. We decide the sequence of emotions to be displayed based on the description of each emoji provided by LinkedIn (LinkedInHelp, 2022). Figure 1 depicts the different sequences we generate based on the context. For e.g., if the context is negative, then we display the emoticons in the following order : ['supports', 'insightful', 'likes', 'love' 'celebrates', 'funny'] as opposed to the default order present. We focus more on achieving a good precision score since identifying negative sentiment posts accurately is a crucial part of this research. Our research, therefore, aims to answer the following **research question**:

RQ) How can we generate contextually relevant sequence of reactions to restrict the negative hate spread?

2 Related work

To the best of our knowledge, there has not been any work related to hate speech on LinkedIn. However, there are several works related to hate spread across other social media forums. One such example is hate spread on Twitter which can be seen in the work published by (Zhang and Luo, 2019). Since reacting with an inappropriate emoticon to a LinkedIn post is equivalent to spreading hate on social media platforms, we can draw inspiration from the existing research on hate speech detection systems. Recent years have seen an increase in the number of research on automated hate speech detection systems. One example is the work by (Xu et al., 2012) who apply sentiment analysis to detect bullying in tweets and use Latent Dirichlet Allocation (LDA) topic models (Blei et al., 2003) to identify relevant topics in the texts. Other examples of such studies include the work by (Kwok and Wang, 2013), (Djuric et al., 2015), (Burnap and Williams, 2015), and by (Nobata et al., 2016) which use binary classification to differentiate hate and non-hate speech.

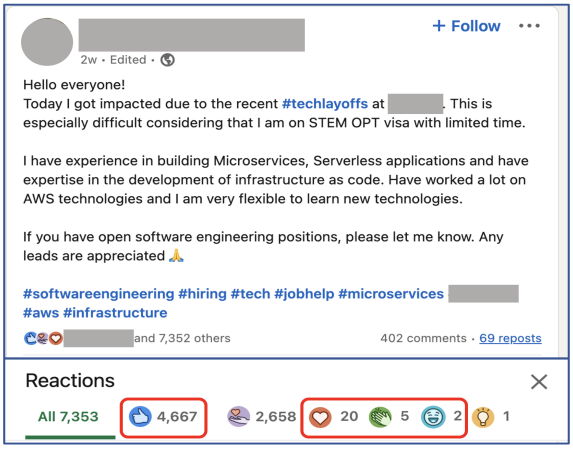

Figure 2: LinkedIn #layoff Post

Another approach is to employ Deep learning based methods to learn abstract feature representations from input data for the classification of hate speech. Since the deep learning methods are designed to automatically extract useful features from a simple input text it is seen that they perform better than the linear classification methods on this task. Typically deep learning methods use word embedding in order to capture similarities between words (Badjatiya et al., 2017). Recent work on this topic makes use of different neural network architectures (such as Gated RNNs, CNNs) (Zhang and Luo, 2019; Zhang et al., 2018; Badjatiya et al., 2017; Kshirsagar et al., 2018; Mishra et al., 2018; Mitrović et al., 2019) and also fine-tuned pre-trained language models, e.g., BERT, RoBERTa, a.o., (Liu et al., 2019; Swamy et al., 2019) for automated hate speech detection.

3 Methodology

3.1 Problem Formulation

In order to understand the context of the post, it is crucial to annotate the post as positive or negative based on its sentiment. A post which is neu-

tral is considered as non-negative and will hence be tagged as positive since the main aim of this research is to avoid hate spread by identfying negative sentiment posts accurately. We, thus formulate our problem as a classification task to predict the sentiments of the LinkedIn posts P_i posted by the users u_i where $P_i \in \{p^i_1, p^i_2, ..., p^i_T\}$, and $u_i \in \{u1, u2, ..., uN\}$, and use the predictions to generate relevant sequence of reactions. In order to classify the sentiments we train 6 different models namely LSTM, Bi-Directional LSTM, Deep Bidirectional LSTM-GRU model with a complex architecture, Bidirectional GRU, BERT, and RoBERTa on our generated dataset. We also use a pretrained RoBERTa-base model trained on 58M tweets and finetuned for emotion recognition with the TweetEval benchmark (Barbieri et al., 2020) (Mohammad et al., 2018) to classify the posts. We then use the generated annotations to create a relevant sequence of reactions for the LinkedIn data. In the next 3 sections, we discuss the architectures of the top 3 performing models across different embeddings.

3.2 Deep Bidirectional-LSTM-GRU Model

To encode each of the LinkedIn posts into numerical vectors, we use 3 different types of pre-trained embeddings namely the Google's pretrained word vectors (Jacob Devlin, 2013), the fastText Embeddings (Joulin et al., 2016) and the Glove Embeddings (Pennington et al., 2014) released by Stanford. In order to ensure the consistency while measuring the performance across the models, we fixed the dimension size of the embeddings to 300 for all the word vectors. We then feed these encoded vectors to a Deep Bi-Directional LSTM-GRU Model to predict the sentiment.

The model consists of 2 Bi-Directional Gated Recurrent unit layers with 256 and 128 neurons each. The fastText encoded embeddings e^i_k=fastText(p^i_k) are passed sequentially to the Gated Recurrent Unit (GRU) layer given by h^i_k=GRU(e^i_k). We, thus obtain a sequence of hidden states, h=[h^1_k,h^2_k,....,h^t_k].The output of these 2 layers is fed to a Bi-Directional LSTM layer with 64 neurons resulting in a sequence of hidden states and then to another Bi-Directional Gated Recurrent unit with 128 neurons. The size of the neurons was fixed after performing deep hyper-parameter tuning with additional parameters. Additionally, the model also consists of 2 dense layers with a ReLU activation function and a dropout of 0.2 to avoid over-fitting. In the last layer, we use a Softmax activation function to predict the prediction vector $\hat{y}$ given by

$$\hat{y} = Softmax(Dense(x))$$

The sentiment with the highest probability is considered as the class of the post. The model architecture is shown in figure 3. We, then train the Deep Bi-Directional LSTM-GRU Model on a corpus of around 16K posts across different annotations as explained in the dataset section of this research paper. The model was tested on a corpus of 3600 posts and a test dataset consisting of 2300 posts. The model achieved highest performance with fastText Embeddings on the test data as evident from Table 1.

3.3 BERT & RoBERTa Model

In our experiment, we have used BERT (*base*) (Jacob Devlin, 2018)., which comprises - 12 transformer layers, 768 hidden units, and 110 M parameters. We fine-tuned the general-purpose BERT (*base*) model to customize its performance for our classification task. Since we want BERT to generate a language representation model, it only needs the encoder part. The input to the encoder for BERT is a sequence of tokens, which are first converted into vectors and then processed in the neural network. Before training, BERT needs the input to be arranged in a specific way using embeddings such as token, segment, and positional. We start with adding special tokens - [CLS] and [SEP]. The [CLS] token will be inserted at the beginning of each post sequence, the [SEP] token will be at the end. The average length of each post in our train data was 430. However, during our experiments, we found the sequence input of size 256 gave better results as compared to others. Sequences shorter than 256 were padded with zeros. Preprocessed training data consisting of 16K posts was tokenized and the tokenized representation is passed to the model.

We designed our training model with a BERT layer followed by two dense layers *(size = 64, 32)* each with a dropout of 0.2 and a dense layer with a softmax activation for classification. The model was trained with the following training parameters - *batch_size = 16, optimizer = Adam (learning rate = $1e^{-5}$)*. The model was trained for *10 epochs* with an early stopping monitored

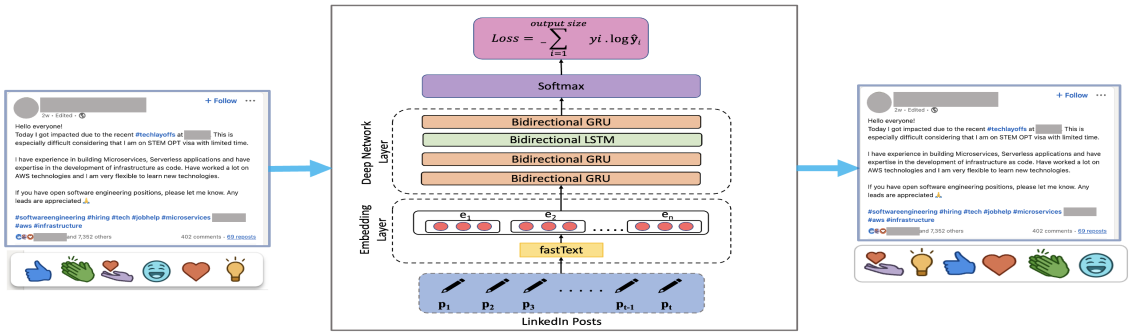

Figure 3: Deep Bidirectional-LSTM-GRU Model architecture and pipeline

on accuracy. The training took around 9 minutes per epoch on a K80 GPU. In the inference phase, the model was evaluated on test dataset (3K posts). The model performed with an F1-score of 88%. Similar to BERT, we fine-tuned the general-purpose RoBERTa (*base*) model (Yinhan Liu, 2019) to customize its performance for our classification task.

For tokenization, we use a pre-trained tokenizer from Hugging-Face with the check-point name roberta-base. The preprocessed train data is tokenized using a byte version of Byte-Pair Encoding (BPE). We chose a sequence input length of size 256, so the posts with more than 256 words would be truncated, and posts shorter than 256 would be padded. Tokenization results in two NumPy arrays - the numeric tokens, and the attention mask (shows which tokens are real and which ones are padded). The resultant tokenized representation is passed to the model. We designed our training model with a RoBERTa layer followed by a single dense layer *(size = 256)* each with a dropout of 0.1 and a dense layer with a softmax activation for classification. The model was trained with the following training parameters - *batch_size = 8, optimizer = Adam (learning rate = $1e^{-5}$)*. In the inference phase, the model was evaluated on test dataset (3K posts) and the model achieved an F1-score of 88%.

4 Experimental Setup

4.1 Dataset & Annotation

The dataset with the LinkedIn posts and its annotations was not readily available. As a result, we formulated a weak supervised learning approach to annotate the LinkedIn posts. Since the number of training posts were more than 15K, it was not feasible to annotate them manually. However, the posts in the unseen data were annotated by the researchers manually to identify whether the model has a decent performance even after training on weakly supervised annotated data. Every researcher manually annotated 3K posts and then the annotations were discussed and conflicts were resolved to decide the final annotation. LinkedIn posts were scrapped using a self-written scrapper script. In order to generate diverse data, every member of the group ran the scrapping script daily for 10 days. Since, the LinkedIn feed varies from person to person, it was estimated that the posts generated will be different.

Around 20K posts were generated by the scrapping script from 30 different hashtags such as **#layoffs, #protests, #unemployment, #excited, #newbeginnings, #newjob** , etc. Figures 4 and 5 depict positive and negative hashtags which we use to scrap LinkedIn data. Since the dataset was extracted based on the hashtags, the posts were annotated based on the hashtag sentiment. For instance, posts scrapped from hashtags such as #protests, #unemployment, were annotated as negative and posts scrapped from hashtags such as #excited, #newbeginnings, #newjob were annotated as positive. In order to avoid any bias while classification, the dataset was balanced before encoding. The data was filtered and duplicate posts were removed to avoid redundancy. The final data had around 8.5K negative posts and 7.5K positive posts after balancing and removing duplicates.

5 Results

We train each model over 50 epochs with early stopping, and report the averaged Macro F1, Pre-

Model	Embedding	Precision	Recall	F1
Deep Bidirectional-LSTM	fastText	0.84	0.84	0.84
	Glove	0.85	0.85	0.85
	Google-W2V	0.86	0.85	0.85
Deep Bidirectional-GRU	fastText	0.86	0.86	0.86
	Glove	0.86	0.86	0.86
	Google-W2V	0.85	0.85	0.85
Deep Bidirectional-LSTM-GRU	**fastText**	**0.88**	**0.88**	**0.87**
	Glove	0.86	0.85	0.84
	Google-W2V	0.85	0.85	0.85
BERT	BERT-embeddings	**0.88**	**0.88**	**0.88**
RoBERTa[1]	RoBERTa-embeddings	**0.88**	**0.88**	**0.88**
RoBERTa[2]	RoBERTa-embeddings	**0.77**	**0.74**	**0.73**

[1] Trained on our corpus.
[2] Trained on 58M tweets (Barbieri et al., 2020) and (Mohammad et al., 2018)

Table 1: Performance Evaluation on Unseen Test Data. Macro-F1 score results for precision, recall, and F1-score.

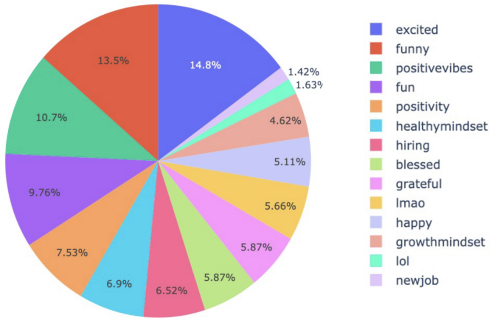

Figure 4: Positive Hashtags and count in percentage

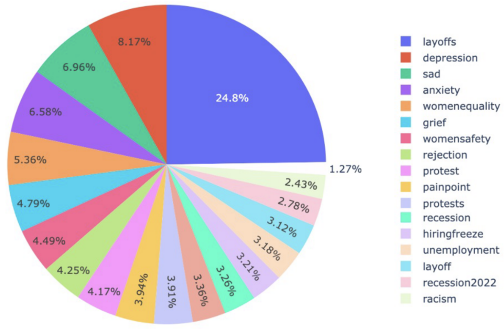

Figure 5: Negative Hashtags and count in percentage

cision and Recall Scores. Since, the false positives are of more concern, we focus more on the precision scores of different models. A post marked positive when actually negative can ultimately lead to reverse results which is why focusing on precision is extremely important. We demonstrate the performances of 7 different models across 3 different embeddings. Table 1 shows the performance metrics of different models on the unseen test data. From the table it is clear that BERT, RoBERTa and Deep-Bidirectional-LSTM-GRU are the top performing models. We show that the Deep-Bidirectional-LSTM-GRU model performs best when used with fastText word embeddings. It is also clear from the table that the pretrained RoBERTa model has a significantly lower performance as compared to other models trained on our corpus.

6 Conclusion

With a motivation to reduce and avoid hate spread across LinkedIn, we present a Deep-learning based solution that detects the context of the post to generate a relevant sequence of reactions thereby motivating the users to react with relevant emoticons. We demonstrated the effectiveness of our system through analysis on real-world LinkedIn data by comparing the performances of various models which were either pretrained or trained on our corpus. Our current research also serves as a baseline model for our future hypothesis which is to test whether people react with irrelevant reactions deliberately. This can be done by testing how different people react when presented

different sequence of reactions, and then calculating the statistics as to how many people react with irrelevant reactions.

7 Limitations

We demonstrated that we used a weak supervised learning approach to annotate the training data. However, this approach generates noisy data due to direct scraping from hashtags. This is because most of the times negative hashtags can have a positive sentiment. For eg, a post such as "In the event of layoffs, I am excited that we have openings..." will be tagged in #layoffs but the sentiment is actually positive. The performance of the models might improve further if such posts are human-annotated and then fed to the model. Also, presently, we just focus on English language texts. In the future, we also plan to build a Neural Machine Translation model which will convert any non-English text to English, and will then be fed to the model for classification.

References

Tetiana Avdieieva. Emojis, not words: Hate, Bullying AND Harassment. 2021. https://cedem.org.ua/en/analytics/emojis-not-words-hate-bullying-and-harassment/ [Online; accessed 01-Dec-2022].

Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th international conference on World Wide Web companion*, pages 759–760, 2017.

Francesco Barbieri, Jose Camacho-Collados, Leonardo Neves, and Luis Espinosa-Anke. Tweeteval: Unified benchmark and comparative evaluation for tweet classification. *arXiv preprint arXiv:2010.12421*, 2020.

David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

Isabelle Boutet, Megan LeBlanc, Justin A. Chamberland, and Charles A. Collin. Emojis influence emotional communication, social attributions, and information processing. *Computers in Human Behavior*, 119:106722, 2021. ISSN 0747-5632. doi: https://doi.org/10.1016/j.chb.2021.106722. URL https://www.sciencedirect.com/science/article/pii/S0747563221000443.

Pete Burnap and Matthew L Williams. Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making. *Policy & internet*, 7(2): 223–242, 2015.

Nemanja Djuric, Jing Zhou, Robin Morris, Mihajlo Grbovic, Vladan Radosavljevic, and Narayan Bhamidipati. Hate speech detection with comment embeddings. In *Proceedings of the 24th international conference on world wide web*, pages 29–30, 2015.

Kenton Lee Kristina Toutanova Jacob Devlin, Ming-Wei Chang. Efficient estimation of word representations in vector space. arXiv, 2013. arXiv preprint arXiv:1301.3781.

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

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. arXiv, 2016. arXiv preprint arXiv:1607.01759.

Rohan Kshirsagar, Tyrus Cukuvac, Kathy McKeown, and Susan McGregor. Predictive embeddings for hate speech detection on Twitter. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 26–32, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5104. URL https://aclanthology.org/W18-5104.

Irene Kwok and Yuzhou Wang. Locate the hate: Detecting tweets against blacks. In *Twenty-seventh AAAI conference on artificial intelligence*, 2013.

LinkedIn. LinkedIn Webpage. 2022. http://about.linkedin.com/ [Online; accessed 01-Dec-2022].

LinkedInHelp. LinkedInHelp. 2022. https://www.linkedin.com/help/linkedin/answer/a528190/use-linkedin-reactions?lang=en [Online; accessed 01-Dec-2022].

Ping Liu, Wen Li, and Liang Zou. 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,

June 2019. Association for Computational Linguistics. doi: 10.18653/v1/S19-2011. URL https://aclanthology.org/S19-2011.

A. Matamoros-Fernández. Inciting anger through Facebook reactions in Belgium: The use of emoji and related vernacular expressions in racist discourse. 2018. https://doi.org/10.5210/fm.v23i9.9405.

Pushkar Mishra, Helen Yannakoudakis, and Ekaterina Shutova. Neural character-based composition models for abuse detection. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 1–10, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5101. URL https://aclanthology.org/W18-5101.

Jelena Mitrović, Bastian Birkeneder, and Michael Granitzer. nlpUP at SemEval-2019 task 6: A deep neural language model for offensive language detection. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 722–726, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/S19-2127. URL https://aclanthology.org/S19-2127.

Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. Semeval-2018 task 1: Affect in tweets. In *Proceedings of the 12th international workshop on semantic evaluation*, pages 1–17, 2018.

Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. Abusive language detection in online user content. In *Proceedings of the 25th international conference on world wide web*, pages 145–153, 2016.

Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1162. URL https://aclanthology.org/D14-1162.

Steve Durairaj Swamy, Anupam Jamatia, and Björn Gambäck. 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 2019. Association for Computational Linguistics. doi: 10.18653/v1/K19-1088. URL https://aclanthology.org/K19-1088.

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*, pages 656–666, 2012.

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

Ziqi Zhang and Lei Luo. Hate speech detection: A solved problem? the challenging case of long tail on twitter. *Semantic Web*, 10(5):925–945, 2019.

Ziqi Zhang, David Robinson, and Jonathan Tepper. Detecting hate speech on twitter using a convolution-gru based deep neural network. In *European semantic web conference*, pages 745–760. Springer, 2018.