UTNLP at SemEval-2021 Task 5: A Comparative Analysis of Toxic Span Detection using Attention-based, Named Entity Recognition, and Ensemble Models

Alireza Salemi, Nazanin Sabri, Emad Kebriaei, Behnam Bahrak, Azadeh Shakery
School of Electrical and Computer Engineering, College of Engineering
University of Tehran, Tehran, Iran
{alireza.salemi,nazanin.sabri,emad.kebriaei,bahrak,shakery}@ut.ac.ir

Abstract

Detecting which parts of a sentence contribute to that sentence's toxicity—rather than providing a sentence-level verdict of hatefulness—would increase the interpretability of models and allow human moderators to better understand the outputs of the system. This paper presents our team's, *UTNLP*, methodology and results in the SemEval-2021 shared task 5 on toxic spans detection. We test multiple models and contextual embeddings and report the best setting out of all. The experiments start with keyword-based models and are followed by attention-based, named entity-based, transformers-based, and ensemble models. Our best approach, an ensemble model, achieves an F1 of 0.684 in the competition's evaluation phase.

1 Introduction

When social media platforms were first introduced, they allowed users to post content on any topic they wished, without restricting the type of content they were allowed to put out. This absence of restrictions, along with the anonymity of users through these platforms (Pinsonneault and Heppel, 1997; Mondal et al., 2017), resulted in the spread of offensive language and hate speech online. While one might think there are only a small number of users who produce these types of hateful content, it has been shown that if social media platforms are left unmoderated, over time, the language of the community as a whole will change such that it highly correlates with the speech of hateful users rather than non-hateful ones (Mathew et al., 2020). Given the huge number of social media postings every day, manual moderation of these platforms is not a possibility. As a result, many researchers began to study automatic hate speech detection.

Most studies on hate speech detection only provide labels at the sentence level, showing whether the construct as a whole is toxic or not. But these types of models, offer little explanation as to why the class was predicted, making it hard for human moderators to interpret the results (Pavlopoulos et al.).

In an attempt to solve the aforementioned issue, we took part in SemEval-2021 shared task 5 (Pavlopoulos et al., 2021), where we aim to detect which spans of a sentence cause it to become toxic. Our contributions are as follows: We begin our experimentation by evaluating a random baseline. Next, we test keyword-based methods, trying to find if toxic spans often include words that are known as hateful or negative in available word lists. We then test attention-based models, building on the hypothesis that what the attention model learns to focus on when detecting toxic speech, are the toxic spans. Afterwards, we look at the issue as a named entity recognition problem, by considering *toxic* as a named entity category. Finally, we fine tune *T5-base* and explore the possibility of looking at the task as a text-to-text problem. We compare different neural network architectures and embeddings, and report the model with the best performance. Additionally, we experiment with some hand-crafted features and evaluate their effectiveness in detecting toxic spans. Our best result, an ensemble of named-entity-based models, achieves an F1 of 0.684.

2 Related Work

In this section we provide a brief overview of studies on hate and toxic speech detection, followed by work on span detection in different sub-fields.

2.1 Hate Speech

Hate speech is defined as "any communication that disparages a person or a group on the basis of some characteristic such as race, color, ethnicity, gender, sexual orientation, nationality, religion, or other

Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pages 995–1002
Bangkok, Thailand (online), August 5–6, 2021. ©2021 Association for Computational Linguistics

characteristics" (Nockleby, 2000). However, no clear distinction between toxic and hateful speech has been provided in the scientific literature (D'sa et al., 2019). There are quite a few surveys on the topic of hate speech detection. (Schmidt and Wiegand, 2017), describes available methods, features, and models for such a task. Another survey conducted in 2018 (Fortuna and Nunes, 2018), offers another view of the current state of the field, as well as suggesting ways in which hate speech detection could advance further. Other surveys on the topic published in 2020 include: (Naseem et al., 2020) which examines the impact of pre-processing on the performance of hate speech models. Corpora and resources for the task are studied in (Poletto et al., 2020). Additionally, throughout the years, many shared tasks have been organized to help propel studies in the field (Vu et al., 2020; Bosco et al., 2018; Basile et al., 2019). In addition to the classification of hate speech, significant effort has been put into the analysis of the target of hate (Silva et al., 2016; ElSherief et al., 2018).
Although numerous models have been tested, (Gröndahl et al., 2018) argues that when it comes to hate speech detection, the model is less important than the labeling criteria and the type of data. In confirmation of the importance of labeling, (Arango et al., 2019) also finds that models trained on annotations done by experts outperform systems trained on amateur annotations.

2.2 Span Detection

Named entity recognition (NER), code-switching detection, quotation detection, and key-phrase extraction are among many tasks that involve span identification.
(Chen et al., 2020) employs SpanBERT (Joshi et al., 2020) accompanied by a sequence tagging model to detect erroneous spans, proceeding to use detected spans to perform error correction. The combination of conditional random fields (CRF) (Lafferty et al., 2001) and attention mechanisms (Vaswani et al., 2017) are explored in (Xu et al., 2020) to explore aspect sentiment classification. The study finds that the use of multiple CRFs (to some limit) does improve performance. In (Papay et al., 2020) the authors look into systems to predict the performance of span identification tasks. To do so, BIO labels are used, and it is found that BERT (Devlin et al., 2018) helps when there is little data, while CRF is of great help in hard cases. In addition, the frequency of spans is found to help while length hurts the performance of the model. Furthermore, LSTMs (Hochreiter and Schmidhuber, 1997) are reported to require large amounts of data to learn. (Tang et al., 2019) explores using fine-tuned BERT with attention models to extract keywords, showing how such models could enable the text classification model to be human interpretable.

3 Data

In this section, we will provide a brief description of the datasets utilized in this study. We will begin with our main dataset in which span-level toxicity has been labeled (3.1), next we look at other datasets that were used to better train our models, namely the hate word list that was used (3.2.1) and the sentence-level hate speech data (3.2.2).

3.1 Main Task Dataset: Toxic Spans

The main dataset used in this study is that of the SemEval 2021, Toxic Span detection task (Pavlopoulos et al.; Borkan et al., 2019a). In this dataset, which was built upon *Civil Comments* (Borkan et al., 2019b), toxic word sequences (for sentences in the English language) have been labeled. In other words, labels are indexes of characters that are toxic in each sentence. There are a total of 8,629 sentences in the dataset, 8,101 of which include at least one annotated toxic span, and the rest have none. Sentences are on average 35 words long. The word-length of the toxic spans varies from one to 176 words. Toxic spans are, on average, 2.29 words long.
There are some issues with regard to the quality of the annotations in the dataset. Table 1 shows some examples of annotated comments in the dataset. While the first sentence is satisfactorily annotated, the second and third examples display issues with the labels in the dataset. More concretely, in the second example we can see that the indexes result in poorly separated and broken up words. Additionally, the anotated words are not toxic. In the third example we see that some of the words which do have a toxic connotation are not included in the annotation. While these examples are not extremely common in the dataset, these types of issues make automatic detection of such spans much more difficult.

Text	Toxic Spans
what load you trump chumps just do not have any idea how to deal with reality you have terrible judgment and pick exceptionally idiotic arrogant leaders trump admitted he fired comey to stop the russia investigation man is he stupid.	['idiotic', 'man is he stupid']
except for one thing they are liars they only care about being thugs	['r one th']
what harm have you ever heard of someone getting attacked by bear while taking dump in the woods please does just owning gun make someone paranoid and pu55y at the same time	['harm']

Table 1: Examples of comments and the annotated toxic spans in the dataset

3.2 Datasets Used for Training

To better train our models, we made use of several auxiliary datasets.

3.2.1 Word-list Dataset

One of the methods tested in this study is based on word-matching. In other words, we check whether each word in the sentence is among hateful words and if so predict its label to be toxic. While this method is rather simple and we acknowledge that not all hate words are toxic and they could simply be used as a joke, we consider this method as a good first step to help us better understand the task at hand. As a result we need to use a list of hate words. For that purpose, we used a list of 1,616 unique hate words found on Kaggle (nicapotato).

3.2.2 Hate Speech Dataset

To be able to train our attention-based models (4.1) we needed to have sentence-level annotated data. Thus we used the *Civil Comments* dataset (Jigsaw-Conversation-AI). The fact that this dataset and our main dataset have the same domain is the reason why this specific dataset was selected. In this dataset, each sentence is labeled with a number between 0 and 1, representing how hateful the text is. We consider sentences with scores above 0.5 to be hateful, and consider the rest as non-hateful. We then create a balanced sample of 289,298 sentences to train our model. The average length of sentences in this dataset is 48.12 words which is slightly longer than the sentences in the main dataset (3.1).

4 Methodology

In this study we have tested and compared various models to perform toxic span detection. In this section we will go over the structure and hyperparameters of these models. The codes of all models are publicly available on GitHub[1].

4.1 Attention-based Methods

We begin with the intuition that if a model with an attention layer is trained to detect hate speech at the sentence-level, the words the attention layer would learn to place importance on, would be the hateful words and spans. Consequently, we create a model made up of the following three layers:
(1) BERT-Base (Uncased) Layer which encodes our input texts.
(2) Attention Layer which is meant to be used for the aforementioned purposes
(3) Dense Layers which connect the attention outputs to two output nodes, detecting if the text is hateful or not.
To train this model we have two training stages:

- Sentence-level classification of hate speech
- Span-level detection of toxic spans

First we perform pre-processing by removing all punctuations (except those in the middle of words such as a$$hole), and lower-casing all words.
Next, the aforementioned model is designed. The BERT-Base (Uncased) layer has an input size of 400 tokens (clipping the input at 400 tokens and dropping the rest). The outputs of this layer are embedding vectors with a hidden size of 768 corresponding to the 400 input tokens. The second layer is an attention layer (attention matrix size = 4096) with a Relu activation function. Our last layers are two fully connected layers (4096 nodes) with dropout of 0.1. There are two neurons in the final layer, the objective of which is to detect whether the sentence is an instance of hate speech or not. The model is trained for 10 epochs with the Adam

[1] https://github.com/alirezasalemi7/SemEval2021-Toxic-Spans-Detection

optimizer and a learning rate of 0.001. We freeze the weights of the BERT layer during this training process as we find through experimentation that fine-tuning BERT in this stage results in lower performance of our model in the toxic span detection task.

Once the model has been trained, we input our sentence and if our sentence-level detector predicts the sentence to be non-hateful we move on and produce a blank output as our toxic span. If, however, the model detects the sentence to be hateful, we extract the attention values and calculate the attention score of each word. If a word is made up of multiple subwords, we average the values of all subwords. After the attention scores have been calculated we use rule-based and machine learning models to label spans as toxic. These models are explained in Table 2. We begin by rule based models, selecting a percentage of spans with attention scores above a certain threshold (shown in Figure 2). Additionally, we test different machine learning models with various sets of features. Our results are shown in Section 5.3.

4.2 Named Entity-based Methods

Our second intuition is to look at this problem as one similar to NER. As such, our toxic span label can be looked at as another NER label. We considered toxic, non-toxit and padding as labels and applied CRF to this NER task. The padding label was added to reduce the model bias toward the non-toxic class.

Our model is depicted in Figure 1. We train the model for 2 epochs with a learning rate of 3×10^{-5}. In contrast to the previous method, the embedding layer is fine-tuned during our training process. Our tests on these models are shown in Section 5.4.

4.3 Ensemble Models

Finally, we test two methods of combining the outputs of various models in order to achieve a better performance on the task. As previously mentioned, the expected outputs of the task are numerical indexes of the parts of the string which are believed to be toxic. Consequently, the first method of mixing could be voting, where if the majority of the models vote for one index, the index is included in the final selection. The second method is based on calculating the intersection of outputted indexes of all three models. In other words, only adding an index if it is detected by all three models. The results are shown in Section 5.6.

5 Results

In this section we will report the results of the models introduced in the previous section (4) on the toxic span detection task. Per the competition evaluation instructions, for all models the F1 score is reported.

5.1 Random Baseline

To help us better understand the complexity of the task at hand, we start with a random baseline. In this method, we first split each sentence into words (using NLTK's functions) and then randomly label each word as toxic or not. We observe that this baseline F1-score for the task is 0.17.

5.2 Keyword-based

The second simple method we test is a word-matching one. Our intuition is that toxic spans will likely include hateful or negative words. Thus we begin with a list of hate words and label any word found on the list as toxic and label the rest as nontoxic. This method results in an F1-score of 0.332 which is almost twice that of the random baseline, showing that while not all hate words are toxic and not all toxic spans are hate words, there is still a considerable amount of overlap. We further test if most words in toxic spans will have a negative sentiment value. Thus we repeat the same method, this time labeling anything with a negative sentiment as toxic. To detect the sentiment score of each word we use *TextBlob* (Loria, 2018). We see that this method achieves an F1 of 0.378, outperforming the aforementioned technique. Finally we mix the two methods (labeling both hate words and words with negative sentiment as toxic), and achieve an F1-score of 0.418.

5.3 Attention-based

As mentioned in Section 4, the intuition behind the attention-based model is that the model which learns to detect hate speech, would learn to pay more attention to the hateful spans in the text. Consequently, we test this idea in Table 2. We can see that the rule-based attention selection method outperforms other span selection techniques. To select the best set of rules for the model, we test both the percentage of top-words (with respect to attention) which we consider for selection, and the threshold we place on the minimum value of attention which is considered. As shown in Figure 2, we can see that the top 75% of attention scores with a thresh-

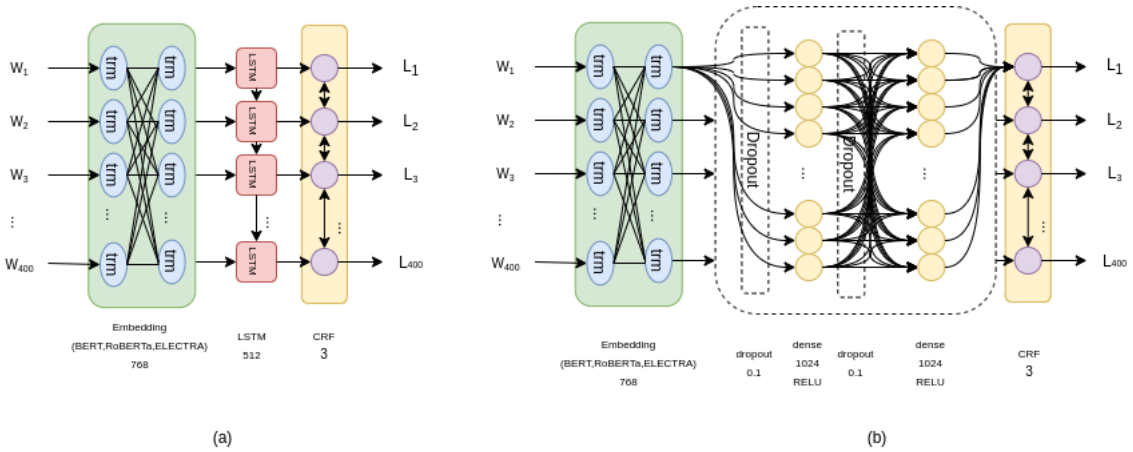

Figure 1: The architecture of the named entity-based models. (a) displays a version of the model in which the two dense layers in (b) have been replaced by an LSTM layer. The results of both versions are shown in Table 3.

old of 10^{-4} is the best set of hyper-parameters for the task.

Upon analysis of the results of the attention-based model, we find that the model performs well on the detection of single word spans (detecting 78% of single-word spans in the evaluation dataset) but does not detect multi-word spans well (only detecting 16% of such spans completely). This is because the distribution of attention scores are observed to be such that there is a large focus on one word and other words receive little attention values.

We further set up another experiment where we assumed that the true sentence-level labels were given. The model then predicted the toxic spans given these true labels achieving an F1 of 0.808. This shows that if the sentence-level classifier performed better, our model would have been able to get higher performance. Thus, more focus should be placed on obtaining higher accuracy in the sentence-level classification task.

5.4 Named Entity-based

Table **??**, displays the results of our named entity based models. We can see that LSTM layers do not improve performance, and among various embeddings, RoBERTa outperforms the others in our 5-fold cross validation testings. However, BERT achieves better results in the competition's evaluation phase.

5.5 Google's T5

Another model we test is Google's T5 (Raffel et al., 2019). To test the T5 model, we use hugging-face's *T5-base* model[2] and frame our problem as one where the context is the Tweet text and the answer is the text of the toxic spans to be detected. Our model achieves an F1 of 0.635 in the evaluation phase of the competition.

5.6 Ensemble Models

As described in section 4.3, we tested intersecting and using a voting scheme for the model outputs. More precisely, we perform these methods on the outputs of the following named entity based models:

(1) BERT + CRF

(2) Electra + CRF

(3) RoBERTa + CRF

We find that the competition evaluation F1 reaches 0.681 when we use voting of indexes, and 0.684 when the indexes are intersected. As can be seen both methods outperform all individual models.

6 Conclusion

In this study we presented and compared various methods for toxic span detection. We examined the problem from various points of views reporting our results using each model. Our best system, an ensemble model, achieved an F1 of 0.684 in the SemEval-2021 Task 5 evaluation phase. Among the named-entity-based models, BERT+CRF performs best achieving an F1 of 0.67. Our attention-based model achieved an F1 of 0.609 in the competition's

[2]We were not able to test a larger version of the model due to system constraints

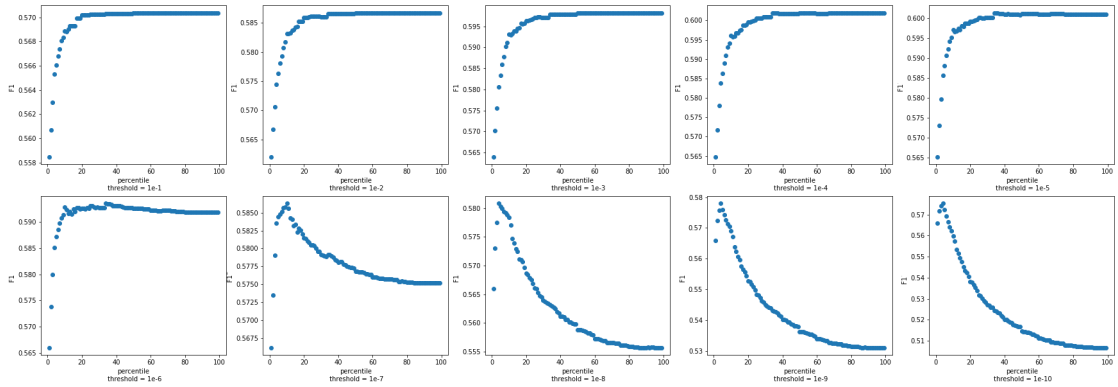

Figure 2: The effects of placing various thresholds on the minimum value of attention scores allowed to be selected and the percentage of top scores that have been selected on the F1-score of the toxic span detection task. Each plot displays one threshold value, and the x-axis in each plot is the percentile of scores we select and the y-axis is the F1 value achieved by this combination of threshold and percentile.

Model	Hate Speech Detection				Toxic Span Detection	
Span Selection Rules	Accuracy	Precision	Recall	F1	F1	F1 (Competition Evaluation)
R1 [a]	0.85	0.85	0.85	0.85	0.601	0.609
R2 [b]					0.601	-
R3 [c]					0.496	-
Decision Tree [d]					0.360	-
Neural Network [e]					0.354	-

[a] **R1**: selecting words with top 75% of attention scores with threshold 10^{-4} and then removing stop-words
[b] **R2**: R1 + removing positive sentiment words among the top 75%
[c] **R3**: R2 + adding all hate words (using the hate word list) in the sentence regardless of attention scores
[d] **Decision Tree**: the input features of the model are: 1-attention score of word, 2-part of speech of the word, 3-sentiment of word 4-whether the word is a hate word or not (0/1)
[e] **Neural Network**: the features inputted to the model are 1-attention score of word, 2-part of speech of the word, 3-sentiment of word, 4-whether the word is a hate word or not (0/1) - categorical features (e.g. POS) are modeled as learnable embeddings.

Table 2: Results of the attention-based models, the model structure is *BERT + Attention + Dense* and we have tested out different span selection rules

Embedding	Layers	F1 (train)	F1 (test)	F1 (Competition Evaluation)
BERT	CRF	0.702	0.648	0.67
RoBERTa	CRF	0.682	0.652	0.66
Electra	CRF	0.687	0.646	0.65
BERT	LSTM + CRF	0.668	0.62	-
RoBERTa	LSTM + CRF	0.669	0.647	-
Electra	LSTM + CRF	0.678	0.641	-

Table 3: Results of the named-entity based models evaluated using 5-fold cross-validation

evaluation phase. Future work could focus on the improvement of the sentence-level detection in our attention scheme, as we showed improvement in that regard would improve this task's performance.

References

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

Valerio Basile, Cristina Bosco, Elisabetta Fersini,

Nozza Debora, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, Manuela Sanguinetti, et al. 2019. Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In *13th International Workshop on Semantic Evaluation*, pages 54–63. Association for Computational Linguistics.

Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2019a. Nuanced metrics for measuring unintended bias with real data for text classification. In *Companion Proceedings of The 2019 World Wide Web Conference*, pages 491–500.

Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2019b. Nuanced metrics for measuring unintended bias with real data for text classification. *CoRR*, abs/1903.04561.

Cristina Bosco, Dell'Orletta Felice, Fabio Poletto, Manuela Sanguinetti, and Tesconi Maurizio. 2018. Overview of the evalita 2018 hate speech detection task. In *EVALITA 2018-Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian*, volume 2263, pages 1–9. CEUR.

Mengyun Chen, Tao Ge, Xingxing Zhang, Furu Wei, and Ming Zhou. 2020. Improving the efficiency of grammatical error correction with erroneous span detection and correction. *arXiv preprint arXiv:2010.03260*.

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

Ashwin Geet D'sa, Irina Illina, and Dominique Fohr. 2019. Towards non-toxic landscapes: Automatic toxic comment detection using dnn. *arXiv preprint arXiv:1911.08395*.

Mai ElSherief, Vivek Kulkarni, Dana Nguyen, William Yang Wang, and Elizabeth Belding. 2018. Hate lingo: A target-based linguistic analysis of hate speech in social media. *arXiv preprint arXiv:1804.04257*.

Paula Fortuna and Sérgio Nunes. 2018. A survey on automatic detection of hate speech in text. *ACM Computing Surveys (CSUR)*, 51(4):1–30.

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*, pages 2–12.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Jigsaw-Conversation-AI. *Detect toxicity across a diverse range of conversations*. https://cutt.ly/XhOYKPR.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Steven Loria. 2018. textblob documentation. *Release 0.15*, 2.

Binny Mathew, Anurag Illendula, Punyajoy Saha, Soumya Sarkar, Pawan Goyal, and Animesh Mukherjee. 2020. Hate begets hate: A temporal study of hate speech. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW2):1–24.

Mainack Mondal, Leandro Araújo Silva, and Fabrício Benevenuto. 2017. A measurement study of hate speech in social media. In *Proceedings of the 28th acm conference on hypertext and social media*, pages 85–94.

Usman Naseem, Imran Razzak, and Peter W Eklund. 2020. A survey of pre-processing techniques to improve short-text quality: a case study on hate speech detection on twitter. *Multimedia Tools and Applications*, pages 1–28.

nicapotato. *Bad Bad Words*. https://www.kaggle.com/nicapotato/bad-bad-words.

JT Nockleby. 2000. 'hate speech in encyclopedia of the american constitution.

Sean Papay, Roman Klinger, and Sebastian Padó. 2020. Dissecting span identification tasks with performance prediction. *arXiv preprint arXiv:2010.02587*.

John Pavlopoulos, Ion Androutsopoulos, Jeffrey Sorensen, and Léo Laugier. *Toxic Spans Detection*. https://sites.google.com/view/toxicspans.

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*.

Alain Pinsonneault and Nelson Heppel. 1997. Anonymity in group support systems research: A new conceptualization, measure, and contingency framework. *Journal of Management Information Systems*, 14(3):89–108.

Fabio Poletto, Valerio Basile, Manuela Sanguinetti, Cristina Bosco, and Viviana Patti. 2020. Resources and benchmark corpora for hate speech detection: a systematic review. *Language Resources and Evaluation*, pages 1–47.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Anna Schmidt and Michael Wiegand. 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*, pages 1–10.

Leandro Silva, Mainack Mondal, Denzil Correa, Fabrício Benevenuto, and Ingmar Weber. 2016. Analyzing the targets of hate in online social media. *arXiv preprint arXiv:1603.07709*.

Matthew Tang, Priyanka Gandhi, Md Ahsanul Kabir, Christopher Zou, Jordyn Blakey, and Xiao Luo. 2019. Progress notes classification and keyword extraction using attention-based deep learning models with bert. *arXiv preprint arXiv:1910.05786*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30:5998–6008.

Xuan-Son Vu, Thanh Vu, Mai-Vu Tran, Thanh Le-Cong, and Huyen Nguyen. 2020. Hsd shared task in vlsp campaign 2019: Hate speech detection for social good. *arXiv preprint arXiv:2007.06493*.

Lu Xu, Lidong Bing, Wei Lu, and Fei Huang. 2020. Aspect sentiment classification with aspect-specific opinion spans. *arXiv preprint arXiv:2010.02696*.