Cisco at SemEval-2021 Task 5: What's Toxic?: Leveraging Transformers for Multiple Toxic Span Extraction from Online Comments

Sreyan Ghosh
Cisco Systems, Bangalore, India
MIDAS Lab, IIIT-Delhi, India
sreyghos@cisco.com

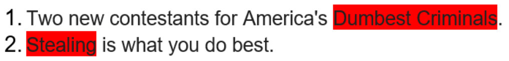
Sonal Kumar
Cisco Systems, Bangalore, India
sonalkum@cisco.com

Abstract

Social network platforms are generally used to share positive, constructive, and insightful content. However, in recent times, people often get exposed to objectionable content like threat, identity attacks, hate speech, insults, obscene texts, offensive remarks or bullying. Existing work on toxic speech detection focuses on binary classification or on differentiating toxic speech among a small set of categories. This paper describes the system proposed by team Cisco for SemEval-2021 Task 5: Toxic Spans Detection, the first shared task focusing on detecting the spans in the text that attribute to its toxicity, in English language. We approach this problem primarily in two ways: a sequence tagging approach and a dependency parsing approach. In our sequence tagging approach we tag each token in a sentence under a particular tagging scheme. Our best performing architecture in this approach also proved to be our best performing architecture overall with an F_1 score of **0.6922**, thereby placing us 7^{th} on the final evaluation phase leaderboard. We also explore a dependency parsing approach where we extract spans from the input sentence under the supervision of target span boundaries and rank our spans using a biaffine model. Finally, we also provide a detailed analysis of our results and model performance in our paper.

1 Introduction

It only takes one toxic comment to sour an online discussion. The threat of abuse and harassment online leads many people to stop expressing themselves and give up on seeking different opinions. Toxic content is ubiquitous in social media platforms like Twitter, Facebook, Reddit, the increase of which is a major cultural threat and has already lead to a crime against minorities (Williams et al., 2020). Toxic text in online social media varies depending on targeted groups (e.g. women, LGBT, gay, African, immigrants) or the context (e.g. pro-trump discussion or the metoo movement). Toxic Text online has often been broadly classified by researchers into different categories like hate, offense, hostility, aggression, identity attacks, and cyberbullying. Though the use of various terms for equivalent tasks makes them incomparable at times (Fortuna et al., 2020), toxic speech or spans in this particular task, SemEval-2021 Task 5 (Pavlopoulos et al., 2021), has been considered as a super-set of all the above sub-types.



1. Two new contestants for America's Dumbest Criminals.
2. Stealing is what you do best.

Figure 1: Toxic spans in sentences

While a lot of models have claimed to achieve state-of-the-art results on various datasets, it has been observed that most models fail to generalize (Arango et al., 2019; Gröndahl et al., 2018). The models tend to classify comments as toxic that have a reference to certain commonly-attacked entities (e.g. gay, black, Muslim, immigrants) without the comment having any intention to be toxic (Dixon et al., 2018; Borkan et al., 2019). A large vocabulary of certain trigger terms leads to a biased prediction by the models (Sap et al., 2019; Davidson et al., 2017). Thus, it has become increasingly important in recent times to determine parts of the text that attribute to the toxic nature of the sentence, for both automated and semi-automated content moderation on social media platforms, primarily for the purpose of helping human moderators deal with lengthy comments and also provide them attributions for better explainability on the toxic nature of the post. This in turn would aid in better handling of unintended bias in toxic text classification. SemEval-2021 Task 5: Toxic Spans Detection focuses on exactly this problem of detecting toxic

Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pages 249–257
Bangkok, Thailand (online), August 5–6, 2021. ©2021 Association for Computational Linguistics

spans from sentences already classified as toxic on a post-level.

In this paper, we approach the problem of multiple non-contiguous toxic span extraction from texts both as a *sequence tagging task* and as a standard *span extraction task* resembling the generic approach and architecture adopted for single-span Reading Comprehension (RC) task. For our sequence tagging approach, we predict for each token, whether it is a part of the span. For our second approach, we predict and compute a couple of scores for each token, corresponding to whether that token is the start or end of the span. In addition to this, we deploy a biaffine model to score start and end indices, thus adopting the methodology for multiple non-contiguous span extraction.

2 Literature

Previous work on automated toxic text detection, and its various sub-types, focuses on developing classifiers that can flag toxic content with a high degree of accuracy on datasets curated from various social media platforms in English(Carta et al., 2019; Saeed et al., 2018; Vaidya et al., 2020), other foreign languages (Zhang et al., 2018; Mishra et al., 2018; Qian et al., 2019; Davidson et al., 2017; Kamal et al., 2021; Leite et al., 2020) including code-switched text (Mathur et al., 2018a,b; Kapoor et al., 2019) and multilingual text (Zampieri et al., 2019). This topic has also evidenced a number of workshops (Kumar et al., 2018) and competitions (Zampieri et al., 2019, 2020; Basile et al., 2019; Mandl et al., 2019).

Recent work shows transformer based architectures like BERT (Devlin et al., 2019) have been performing well on the task of offensive language classification (Liu et al., 2019a; Safaya et al., 2020; Dai et al., 2020). Transformer based architectures have also produced state-of-the-art performance on sequence tagging tasks like *Named Entity Recognition (NER)* (Yamada et al., 2020; Devlin et al., 2019; Yang et al., 2019) *span extraction* (Eberts and Ulges, 2019; Joshi et al., 2020) and *QA tasks* (Devlin et al., 2019; Yang et al., 2019; Lan et al., 2020). Multiple span extraction from texts has been explored both as a *sequence tagging task* (Patil et al., 2020; Segal et al., 2019) and as span extraction as in RC tasks(Hu et al., 2019; Yu et al., 2020).

Very recently HateXplain (Mathew et al., 2020) proposed a benchmark dataset for explainable hate speech detection using the concept of rationales. Attempts have also been made to handle identity bias in toxic text classification (Vaidya et al., 2020) and also to make robust toxic text classifiers which help adversaries not bypass toxic filters (Kurita et al., 2019).

3 Methodology

For our sequence tagging approach, we explore two tagging schemes. First, the well known *BIO* tagging scheme, where *B* indicates the first token of an output span, *I* indicates the subsequent tokens and *O* denotes the tokens that are not part of the output span. Additionally, we also try a simpler *IO* tagging scheme, where words which are part of a span are tagged as *I* or *O* otherwise. Formally, given an input sentence **x** = (x_1,...,x_n), of length n,and a tagging scheme with $|S|$ tags ($|S|$ = 3 for BIO and $|S|$ = 2 for IO), for each of n tokens the probability for the tag of the i-th token is

$$\mathbf{p}_i = softmax(f(\mathbf{h}_i)) \tag{1}$$

where $\mathbf{p} \in R^{m\times|S|}$, and f is parameterized function with $|S|$ outputs.

Our other approach is based on the standard single-span extraction architecture widely used for RC Tasks. With this approach, we extract toxic spans from sentences under the supervision of target span boundaries, but with an added biaffine model for scoring the multiple toxic spans instead of simply taking top k spans based on the start and end probabilities, thus giving our model a global view of the input. The main advantage of this approach is that the extractive search space can be reduced linearly with the sentence length, which is far less than the sequence tagging method. Given an input sentence **x** = (x_1,...,x_n), of length n, we predict a target list **T** = (t_1,...,t_m) where the number of targets is m and each target t_i is annotated with its start position s_i, its end position e_i and the class that span belongs to (only one in our case, *toxic*).

However, to adapt to the problem of extracting multiple spans from the sentence, instead of taking the top k spans based on the start and end probabilities, we apply a biaffine model (Dozat and Manning, 2016) to score all the spans with the constraint $s_i \leq e_i$. Post this we rank all the spans in descending order and choose every span as long it does not clash with higher-ranked spans.

4 Dataset

The dataset provided to us by the organizers of the workshop consisted of a random subset of 10,000 posts from the publicly available Civil Comments Dataset, from a set of 30,000 posts originally annotated as toxic (or severely toxic) on post-level annotations, manually annotated by 3 crowd-raters per post for toxic spans. The final character offsets were obtained by retaining the offsets with a probability of more than 50%, computed as a fraction of raters who annotated the character offsets as toxic. Basic statistics about the dataset can be found in Table 1.

	Sentences	Spans
Train	7939	10298
Dev	690	903
Test	2000	1850

Table 1: Number of sentences and spans

Additionally, we provide a quick look into the length-wise distribution of spans across the train, development, and test set in Table 2. As we observe, the majority of the spans are just a single word in length and mostly comprise of the most commonly used cuss words in the *English* language. In our Results Analysis section, we show how this metric stands important for training and evaluating our systems and for the future development of toxic span extraction datasets.

	Train	Dev	Test
1	7897	687	1650
2-4	1617	153	174
>=5	784	63	26

Table 2: Length-wise segregation of the number of non-contiguous spans

5 Evaluation Metric

To evaluate the performance of our systems we employ F_1 as used by Da San Martino et al. (2019). Let system A return a set $S^t{}_A$ of character offsets, for parts of the post found to be toxic. Let $S^t{}_G$ be the character offsets of the ground truth annotations of post t. We calculate F_1 score of $S^t{}_A$ w.r.t $S^t{}_G$ as follows where $|.|$ denotes set cardinality.

$$P^t\left(A,G\right)=\frac{\left|S_A^t\cap S_G^t\right|}{\left|S_A^t\right|} \tag{2}$$

$$R^t\left(A,G\right)=\frac{\left|S_A^t\cap S_G^t\right|}{\left|S_G^t\right|} \tag{3}$$

$$F_1^t\left(A,G\right)=\frac{2\cdot P^t\left(A,G\right)\cdot R^t\left(A,G\right)}{P^t\left(A,G\right)+R^t\left(A,G\right)} \tag{4}$$

If predicted span i.e $S^t{}_A$ is empty for a post t then we set $F^t{}_1(A,G) = 1$ if the gold truth i.e $S^t{}_G$ is also empty, else if $S^t{}_G$ is empty and $S^t{}_A$ is not empty then we set $F^t{}_1(A,G) = 0$.

6 System Description

6.1 Sequence Tagging Approach

For our sequence tagging approach we employ the commonly used BiLSTM-CRF architecture (Huang et al., 2015) used predominately in many sequence tagging problems, but with added contextual word embeddings for each word using transformer and character-based word embeddings. We experiment with a total of 5 transformer architectures, namely BERT (Devlin et al., 2019), XLNet (Yang et al., 2019), RoBERTa (Liu et al., 2019b), ALBERT (Lan et al., 2020) and SpanBERT (Joshi et al., 2020). For all of the above mentioned transformer architectures, the *large* variant of the transformer was used except ALBERT for which we use its *xlarge-v2* variant. First, the tokenized word input is passed through the transformer architecture and the output of the last 4 encoder layers is concatenated to obtain the final contextualized word embedding E_T for each word in the sentence. Additionally, we also pass each character in a word through a character-level BiLSTM network, to obtain character-based word embeddings for the word E_C as used by Lample et al. (2016). Finally, both these word embeddings, E_T and E_C, for each word are concatenated and passed through a BiLSTM layer followed by a CRF layer to obtain the best probable tag for each word in the sentence.

6.2 Dependency Parsing Approach

For our dependency parsing approach, we employ a similar approach as proposed by Yu et al. (2020), using a biaffine classifier to score our spans post-extraction. This methodology fits best to our purpose of *multiple* toxic span extraction from sentences compared to span extraction systems in general RC tasks which are capable of extracting just a single span from a sentence (Yang and Ishfaq). For each word first we extract it's BERT, FasText

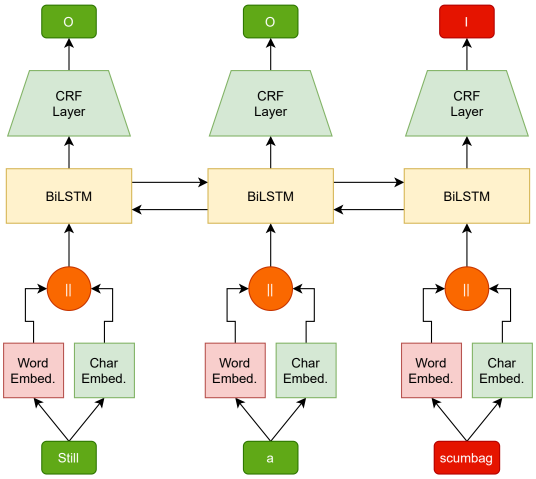


Figure 2: Sequence Tagger Model

and character-based word embeddings. We used $\text{BERT}_{\text{Large}}$ for all our experiments and used the recipe followed by Kantor and Globerson (2019) to extract contextual embeddings for each token. After concatenating both the word embeddings and character embeddings for each word, we feed the output to a BiLSTM layer. We then apply two separate FFNNs to the output word representations x to create different representations (h_s / h_e) for the start/end of the spans. These representations are then passed through a biaffine model for scoring all possible spans (s_i,e_i), where s_i and e_i are start and end indices of the span, under the constraint $s_i \leq e_i$ (the start of the span is before its end) by creating a $l \times l \times c$ scoring tensor r_m, where l is the length of the sentence and c is the number of NER categories + 1(for non-entity). We compute the score for a span i by:

$$h_s(i) = \text{FFNN}_s\left(x_{s_i}\right) \tag{5}$$

$$h_e(i) = \text{FFNN}_e\left(x_{e_i}\right) \tag{6}$$

$$\begin{aligned} r_m(i) =& h_s(i)^{\top} \text{U}_m h_e(i) \\ &+ W_m\left(h_s(i) \oplus h_e(i)\right) + b_m \end{aligned} \tag{7}$$

We finally assign each span a category $y\prime$ based on

$$y'(i) = \arg\max r_m(i) \tag{8}$$

Post this, we rank each span that has a category other than non-entity and consider all the spans for our final prediction as long as it does not clash with higher ranked spans with an additional constraint, whereby, an entity containing or is inside an entity ranked before it will not be selected.

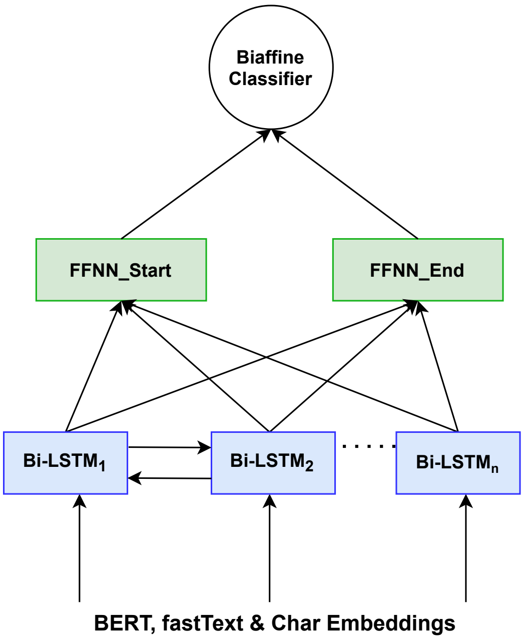


Figure 3: Biaffine Model

7 Experimental Setup

Data was originally provided to us in the form of sentences and the corresponding character offsets for the toxic spans of the sentence. Before converting the character offsets to our required format for our respective approaches, we apply some basic text pre-processing to all our sentences. First, we normalize all the sentences by converting all white-space characters to spaces. Second, we split all punctuation characters from both sides of a word and also break abbreviated words. These pre-processing steps help improve the F_1 score of both our approaches as shown in Table 6. Post these pre-processing steps, we formulate our targets for both our approaches. For our sequence tagging approach, we tag each word in the sentence with its corresponding tag based on the tagging scheme we follow, *BIO* or *IO*. For our span extraction approach, we convert the sequence of character offsets into its corresponding word-level start and end indices for each span. In Fig. 4, we provide a pictorial representation of the above mentioned procedures we follow for data preparation for both our approaches.

We use PyTorch[1] Framework for building our Deep Learning models along with the Transformer

[1] https://pytorch.org/

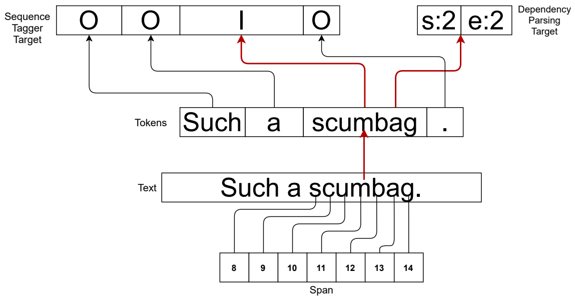


Figure 4: Data Preparation

implementations, pre-trained models and, specific tokenizers in the HuggingFace[2] library.

We mention the major hyperparameters of our best-performing systems experimental setting for our dependency parsing approach and span extraction approach in Tables 3 and 4 respectively.

Parameter	Value
BiLSTM size	256
BiLSTM layer	1
BiLSTM dropout	0
Transformer size	1024
Transformer encoder layers	last 4
Char BiLSTM Hidden Size	25
Char BiLSTM layers	1
Optimiser	Adam
Learning rate	[1e-3,1.56e-4]

Table 3: Major hyperparameters of Sequence Tagger model

Parameter	Value
BiLSTM size	200
BiLSTM layer	3
BiLSTM dropout	0.4
FFNN size	150
FFNN dropout	0.2
BERT size	1024
BERT encoder layers	last 4
fastText embedding size	300
Char CNN size	50
Char CNN filter width	[3,4,5]
Embeddings dropout	0.5
Optimiser	Adam
Learning rate	1e-3

Table 4: Major hyperparameters of Dependancy Parsing model

We train all our sequence tagging models with stochastic gradient descent in batched mode with a batch size of 8. In the training phase, we keep all layers in our model, including all the transformer layers trainable. We start training our model at a learning rate of 0.01, with a minimum threshold limit of 0.0001, and half the learning rate after every 4 consecutive epochs of no improvement in the F_1 score of the development set. We train our model to a maximum of 100 epochs or 4 consecutive epochs of no improvement at our minimum learning rate.

We train our our model for dependency parsing approach with Adam optimizer in batched mode with a batch size of 32 and a learning rate of 0.0001 for a maximum of 40,000 steps. With this approach too, we keep all layers trainable in the training phase except the BERT Transformer layers. Pre-trained BERT and fastText embeddings were just used to extract context-dependent and independent embeddings respectively and BERT was *not fine-tuned* in the training phase.

The training was performed on 1 NVIDIA Titan X GPU. Our code is available on Github[3].

8 Results

In Table 5 we present F_1 scores for all our systems trained for both our sequence tagging and span extraction approaches. For our sequence tagging approach, we divide our results according to the transformer architecture and tagging scheme used for that experiment.

Model	Scheme	Test	Dev
XLNet	IO	**0.6922**	0.6945
XLNet	BIO	0.6653	0.6683
spanBERT	IO	0.6777	0.6744
spanBERT	BIO	0.6887	0.6730
RoBERTa	IO	0.6647	**0.6967**
RoBERTa	BIO	0.6849	0.6789
BERT	IO	0.6830	0.6814
BERT	BIO	0.6852	0.6815
ALBERT	IO	0.6621	0.6702
ALBERT	BIO	0.6679	0.6431
Biaffine	-	0.6731	0.6627

Table 5: Test and Dev Results of different models on various tagging scheme

Our best performing architecture proved to be the sequence tagging system with XLnet trans-

[2] http://huggingface.co/

[3] https://github.com/Sreyan88/SemEval-2021-Toxic-Spans-Detection

former trained with *IO* tagging scheme. Additionally, in Table 6 we show how the LSTM and CRF over the transformer architecture , and our preprocessing step mentioned in Section 7 affect the performance of our best performing architecture.

	F1	**Δ**
Our Model	0.6922	-
- LSTM	0.6912	0.0010
- CRF	0.6850	0.0072
- Pre-processing	0.6759	0.1630

Table 6: Impact of LSTM, CRF and pre-processing on learning

9 Results Analysis

9.1 Length vs Performance

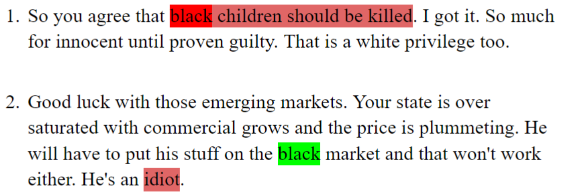
We wanted to understand how the performance of the system varied with varying lengths of spans. Table 7 summarizes the performance of our best performing systems on all approaches experimented by us, on the test dataset spans, divided into 3 sets according to their length in terms of the number of words that help to make the span.

Model	Span length	F1
Seq. Tagger (IO)	1	**0.6546**
	2-4	0.1750
	>=5	0.0596
Seq. Tagger (BIO)	1	**0.6588**
	2-4	0.1524
	>=5	0.09198
Dependency Parsing	1	**0.6486**
	2-4	0.0514
	>=5	0.0

Table 7: Span Length vs. Performance

9.2 Learning context

Majority of single word spans in the dataset are the most commonly used cuss words or abusive words in the English language, i.e., words that can be directly classified as toxic and are not context-dependant, e.g. *"stupid"*,*"idiot"* etc., with spans longer than a single word having a lesser ratio of such words. We acknowledge the fact that an AI-based system should be able to do much more, like learning the context behind which a word is used, than just detect common *English* cuss words from a sentence, which can be otherwise done by a simple dictionary search. The deteriorating performance of the model with an increase in span length makes us dig deeper into our test set results to find out if our model is being able to detect *context-based* toxic spans from sentences. We follow a two step procedure to analyze this. First, we calculate our model performance on single-word spans consisting of just the top 25 most commonly occurring context-independent cuss words[4]. Table 8 shows an analysis of these results. Second, we take the word *"black"* and analyze two sentences in our test where the word black was mentioned in a toxic and non-toxic context. Fig. 5 shows how our model indeed tags the latter black as toxic and the former one as non-toxic.



1. So you agree that black children should be killed. I got it. So much for innocent until proven guilty. That is a white privilege too.
2. Good luck with those emerging markets. Your state is over saturated with commercial grows and the price is plummeting. He will have to put his stuff on the black market and that won't work either. He's an idiot.

Figure 5: Toxicity classification of the word "black" in toxic and non-toxic context

Single Word Cuss Spans	Others
0.6894	0.1736

Table 8: F_1 score of context independent cuss words

10 Conclusion

In this paper, we present our approach to SemEval-2021 Task 5: Toxic Spans Detection. Our best submission gave us an F_1 score of **0.6922**, placing us 7^{th} on the Evaluation Phase Leaderboard. Future work includes independently incorporating both post level and sentence level context for determining the toxicity of a word, and also collating a dataset with toxic spans comprising of a healthy mixture of simple cuss words (which can always be attributed as toxic independant of the context) and words for which the toxicity of the word depends on the context in which it appears, thereby making better systems towards *contextual* toxic span detection.

[4] List of cuss words used for analysis can be found in our GitHub repository

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

Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2019. 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.

Salvatore Carta, Andrea Corriga, Riccardo Mulas, Diego Reforgiato Recupero, and Roberto Saia. 2019. A supervised multi-class multi-label word embeddings approach for toxic comment classification. In *KDIR*, pages 105–112.

Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeno, Rostislav Petrov, and Preslav Nakov. 2019. Fine-grained analysis of propaganda in news article. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5640–5650.

Wenliang Dai, Tiezheng Yu, Zihan Liu, and Pascale Fung. 2020. Kungfupanda at semeval-2020 task 12: Bert-based multi-task learning for offensive language detection. *arXiv preprint arXiv:2004.13432*.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 11.

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

Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2018. Measuring and mitigating unintended bias in text classification. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 67–73.

Timothy Dozat and Christopher D. Manning. 2016. Deep biaffine attention for neural dependency parsing. *CoRR*, abs/1611.01734.

Markus Eberts and Adrian Ulges. 2019. Span-based joint entity and relation extraction with transformer pre-training. *arXiv preprint arXiv:1909.07755*.

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

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.

Minghao Hu, Yuxing Peng, Zhen Huang, Dongsheng Li, and Yiwei Lv. 2019. Open-domain targeted sentiment analysis via span-based extraction and classification. *arXiv preprint arXiv:1906.03820*.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

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.

Ojasv Kamal, Adarsh Kumar, and Tejas Vaidhya. 2021. Hostility detection in hindi leveraging pre-trained language models.

Ben Kantor and Amir Globerson. 2019. Coreference resolution with entity equalization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 673–677, Florence, Italy. Association for Computational Linguistics.

Raghav Kapoor, Yaman Kumar, Kshitij Rajput, Rajiv Ratn Shah, Ponnurangam Kumaraguru, and Roger Zimmermann. 2019. Mind your language: Abuse and offense detection for code-switched languages. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9951–9952.

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

Keita Kurita, Anna Belova, and Antonios Anastasopoulos. 2019. Towards robust toxic content classification. *arXiv preprint arXiv:1912.06872*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Joao A Leite, Diego F Silva, Kalina Bontcheva, and Carolina Scarton. 2020. Toxic language detection in social media for brazilian portuguese: New dataset and multilingual analysis. *arXiv preprint arXiv:2010.04543*.

Ping Liu, Wen Li, and Liang Zou. 2019a. Nuli at semeval-2019 task 6: Transfer learning for offensive language detection using bidirectional transformers. In *Proceedings of the 13th international workshop on semantic evaluation*, pages 87–91.

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

Thomas Mandl, Sandip Modha, Prasenjit Majumder, Daksh Patel, Mohana Dave, Chintak Mandlia, and Aditya Patel. 2019. Overview of the hasoc track at fire 2019: Hate speech and offensive content identification in indo-european languages. In *Proceedings of the 11th Forum for Information Retrieval Evaluation*, pages 14–17.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2020. Hatexplain: A benchmark dataset for explainable hate speech detection. *arXiv preprint arXiv:2012.10289*.

Puneet Mathur, Ramit Sawhney, Meghna Ayyar, and Rajiv Shah. 2018a. Did you offend me? classification of offensive tweets in hinglish language. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 138–148.

Puneet Mathur, Rajiv Shah, Ramit Sawhney, and Debanjan Mahata. 2018b. Detecting offensive tweets in hindi-english code-switched language. In *Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media*, pages 18–26.

Pushkar Mishra, Marco Del Tredici, Helen Yannakoudakis, and Ekaterina Shutova. 2018. Author profiling for abuse detection. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1088–1098.

Rajaswa Patil, Somesh Singh, and Swati Agarwal. 2020. Bpgc at semeval-2020 task 11: Propaganda detection in news articles with multi-granularity knowledge sharing and linguistic features based ensemble learning. *arXiv preprint arXiv:2006.00593*.

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*.

Jing Qian, Anna Bethke, Yinyin Liu, Elizabeth Belding, and William Yang Wang. 2019. A benchmark dataset for learning to intervene in online hate speech. *arXiv preprint arXiv:1909.04251*.

Hafiz Hassaan Saeed, Khurram Shahzad, and Faisal Kamiran. 2018. Overlapping toxic sentiment classification using deep neural architectures. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 1361–1366. IEEE.

Ali Safaya, Moutasem Abdullatif, and Deniz Yuret. 2020. Kuisail at semeval-2020 task 12: Bert-cnn for offensive speech identification in social media. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 2054–2059.

Maarten Sap, Dallas Card, Saadia Gabriel, Yejin Choi, and Noah A Smith. 2019. The risk of racial bias in hate speech detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1668–1678.

Elad Segal, Avia Efrat, Mor Shoham, Amir Globerson, and Jonathan Berant. 2019. A simple and effective model for answering multi-span questions. *arXiv preprint arXiv:1909.13375*.

Ameya Vaidya, Feng Mai, and Yue Ning. 2020. Empirical analysis of multi-task learning for reducing identity bias in toxic comment detection. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 14, pages 683–693.

Matthew L Williams, Pete Burnap, Amir Javed, Han Liu, and Sefa Ozalp. 2020. Hate in the machine: anti-black and anti-muslim social media posts as predictors of offline racially and religiously aggravated crime. *The British Journal of Criminology*, 60(1):93–117.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. Luke: deep contextualized entity representations with entity-aware self-attention. *arXiv preprint arXiv:2010.01057*.

Chenjie Yang and Haque Ishfaq. Question answering on squad.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

Juntao Yu, Bernd Bohnet, and Massimo Poesio. 2020. Named entity recognition as dependency parsing. *arXiv preprint arXiv:2005.07150*.

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

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

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