SkoltechNLP at SemEval-2021 Task 5: Leveraging Sentence-level Pre-training for Toxic Span Detection

David Dale[‡], **Igor Markov**[‡,†], **Varvara Logacheva**[‡], **Olga Kozlova**[†], **Nikita Semenov**[†], **and Alexander Panchenko**[‡]

[‡]Skolkovo Institute of Science and Technology, Moscow, Russia
[†]Mobile TeleSystems (MTS), Moscow, Russia

{d.dale,igor.markov,v.logacheva,a.panchenko}@skoltech.ru
{oskozlo9,nikita.semenov}@mts.ru

Abstract

This work describes the participation of the Skoltech NLP group team (Sk) in the Toxic Spans Detection task at SemEval-2021. The goal of the task is to identify the most toxic fragments of a given sentence, which is a binary sequence tagging problem. We show that fine-tuning a RoBERTa model for this problem is a strong baseline. This baseline can be further improved by pre-training the RoBERTa model on a large dataset labeled for toxicity at the sentence level. While our solution scored among the top 20% participating models, it is only 2 points below the best result. This suggests the viability of our approach.

1 Introduction

Toxicity and offensive content is a major concern for many platforms on the Internet. Therefore, the task of toxicity detection has attracted much attention in the NLP community (Wulczyn et al., 2017; Hosseini et al., 2017; Dixon et al., 2018). Until recently, the majority of research on toxicity focused on classifying entire user messages as toxic or safe. However, the surge of work on text detoxification, i.e., editing of text to keep its content and remove toxicity (Nogueira dos Santos et al., 2018; Tran et al., 2020), suggests that localizing toxicity within a sentence is also useful. If we know which words of a sentence are toxic, it is easier to "fix" this sentence by removing or replacing them with non-toxic synonyms. Mathew et al. (2020) make human labelers annotate the spans as rationales for classifying a comment as hateful, offensive, or normal. They show that using such spans when training a toxicity classifier improves its accuracy and explainability and reduces unintended bias towards toxicity targets.

This year the SemEval hosts the first competition on toxic spans detection, namely, SemEval-2021 Task 5[1] (Pavlopoulos et al., 2021). It provides training, development, and test data for English. As far as we know, it is the first attempt to explicitly formulate toxicity detection as sequence labeling instead of classification of sentences.

Multiple NLP tasks recently benefited from transfer learning — transfer of probability distributions learned on some task to another model solving a different task. The most common example of transfer learning is the use of embeddings and language models pre-trained on unlabeled data (e.g. ELMo (Peters et al., 2018), BERT (Devlin et al., 2019) and its variations, T5 (Raffel et al., 2020), etc.) on other tasks (e.g. He et al. (2020); Wang et al. (2020) *inter alia* use pre-trained BERT models to perform tasks from the GLUE benchmark (Wang et al., 2018)).

Word-level toxicity classification can be formulated as a sequence labeling task, which also actively uses the pre-trained models mentioned above. BERT comprises the versatile information on words and their context, which allows to successfully use it for sequence labeling tasks of different levels: part-of-speech tagging and syntactic parsing (Koto et al., 2020), named entity recognition (Hakala and Pyysalo, 2019), semantic role labeling (He et al., 2019), detection of Machine Translation errors (Moura et al., 2020).

This diversity of applications suggests that word-level toxicity detection can also benefit from pre-trained models. Besides that, toxicity itself has been successfully tackled with BERT-based models. Research on sentence-level toxicity extensively used BERT and other pre-trained models. Both language-specific and multilingual BERT models were used to fine-tune toxicity classifiers (Leite et al., 2020; Ozler et al., 2020). This shows that BERT has information on toxicity.

[1]https://competitions.codalab.org/competitions/25623

Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pages 927–934
Bangkok, Thailand (online), August 5–6, 2021.

Thus, we follow this line of work. Namely, we fine-tune a RoBERTa model (Liu et al., 2019) to perform a sequence labeling task. Besides that, we train a model for sentence classification on the Jigsaw dataset of toxic comments and use the information from this model to detect toxicity at the subsentential level. This helps us overcome the insufficient data size.

This is in line with previous work, which has shown that sentence-level labels can be used in combination with token labels (Rei and Søgaard, 2019) or completely substitute them (Rei and Søgaard, 2018; Schmaltz, 2019).

In our experiments, we test the hypothesis that the sentence-level toxicity labeling can be used for a sequence labeler that recognizes toxic spans in text. We suggest three ways of incorporating this data: as a corpus for pre-training, pseudo-labeling, and for joint training of sentence-level and token-level toxicity detection models. Our experiments show that the latter method yields the best result. Moreover, we show that using sentence-level labels can dramatically improve toxic span prediction when the dataset with token-level labels is small.

The contributions of this work are the following:

- We successfully use the dataset labeled for toxicity at the sentence level for token-level toxicity labeling,
- We propose a model for joint sentence- and token-level toxicity detection,
- We analyze the performance of our models, showing their limitations and reveal the ambiguities in the data.

2 The task

The training data of the task comprises 7,940 English comments with character-level annotations of toxic spans. The labeling was performed manually by crowd workers.

The spans labeled as toxic often contain rude words: "*Because he's a moron and a bigot. It's not any more complicated than that.*" (toxic spans are underlined). Other toxic spans consist of words that become toxic in context: "*Section 160 should also be amended to include sexual acts with animals not involving penetration*". Borders of some toxic spans fall in the middle of a word; we treat such cases as markup errors.

As a development set, we use the trial dataset of 690 texts provided by the task organizers. We evaluate our final models on the hidden test set of the task consisting of 2,000 texts.

3 Pre-training for toxic span detection

Here we give the motivation behind our models and describe their architecture and training setup.

3.1 Motivation

Our intuition is that the toxicity is often lexically-based, i.e., there are certain words that are considered offensive and make the whole sentence toxic. In this case, we expect that as we add extra data to our toxic span dataset, after some point, the vocabulary of toxic words in it will saturate and stop increasing. However, Figure 1 shows that the size of the toxic vocabulary linearly depends on the dataset size, which suggests that its size is insufficient for the task. In this case, the model will often need to label unseen words. To mitigate the lack of data, we leverage the additional dataset with toxicity information, namely, the Jigsaw toxic comments dataset[2] which features 140,000 user utterances labeled as toxic or safe.

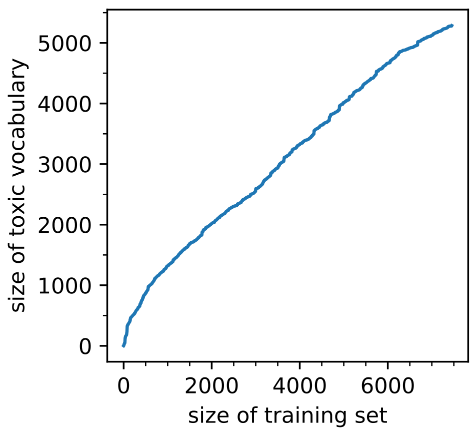

Figure 1: Size of the toxic vocabulary as a function of the corpus size.

3.2 Transfer learning for spans detection

Our base model is RoBERTa. We fine-tune `roberta-base` model on the toxic spans training set for sequence labeling task (further denoted as **RoBERTa tagger**). This model already gives promising results. We further improve it by providing it with the additional training signal from the

[2] https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/data

Jigsaw toxic comments dataset. We propose three ways of incorporating this data.

The first way is **pseudo-labeling** (Lee et al., 2013). We apply the **RoBERTa tagger** to predict the toxic spans in the Jigsaw dataset. We use these predictions to further train the model.

Another option is to use the Jigsaw data to fine-tune RoBERTa with it. We suggest two scenarios. The first is to fine-tune the model on the Jigsaw dataset for the sentence classification task, and then on the toxic spans dataset — this model is referred to as **RoBERTa classifier + tagger**. In this case, the model has different output layers for the two tasks, and other layers are shared.

Finally, we propose a novel architecture for joint token- and sentence-level classification, where the score $\mathbf{y}$ for a sentence $\mathbf{x} = \{x_1, x_2, ..., x_n\}$ is computed as the average of word-level scores:

$$\hat{\mathbf{y}} = \sigma\left(\alpha + \beta\frac{1}{n}\sum_{i=1}^{n}\hat{y}_i^{\gamma}\right)$$

where α, β and γ are trainable parameters, and σ is the logistic function. This model does not need to be trained on the data with token-level labeling but can get token-level toxicity information from sentence labels. We fine-tune this model both on Jigsaw and toxic spans datasets. The model is referred to as **tagging classifier**.

3.3 Working with spans

To reformulate toxic spans detection as a token classification problem, we label a token as toxic if at least one of its characters is toxic. When projecting the predicted token-level labels back to the character level, we try two strategies:

1. Consider a token to be toxic if its toxicity score is higher than the threshold, do not force the labels of tokens within a word to agree with each other.
2. Consider a word to be toxic if the aggregated toxicity score of all its tokens is higher than the threshold. We try four different aggregation functions: min, max, mean, and the simplified naive Bayes formula:

$$\tilde{x} = \frac{\prod_i x_i}{\prod_i x_i + \prod_i (1 - x_i)}.$$

In both methods, we label a space character as toxic only if the characters both to the right and to the left of it are toxic.

4 Baselines

In this section, we present a set of common baseline approaches used for sequence tagging, such as CRF and LSTM with pretrained word embeddings. We implement them in order to analyze the performance of our methods in the context of other techniques.

Word-based LogReg This is a vocabulary-based method: we label words as toxic if they appear in our toxic vocabulary. The vocabulary is created as follows. We create a set of toxic and safe phrases, where toxic phrases are toxic spans from our data and safe phrases are sentences from our data with removed toxic spans. We then train a binary logistic regression classifier of toxic and safe phrases using words as features. The by-product of this classifier is the list of weights for all words from the data. We consider words with weights greater than a threshold as toxic.

Attention-based LogReg Another approach to represent words is to take their attention weights from a RoBERTa-based sentence-level toxicity classifier (we train it on the Jigsaw dataset). We assemble attention weights from all RoBERTa heads and layers in a single vector of dimension 144. These vectors are used as features in a logistic regression classifier. This approach is motivated by the fact that a RoBERTa model trained to recognize toxicity puts more emphasis on certain words associated with sentence-level toxicity. Surprisingly, this model underperforms the logistic regression classifier, which uses words as features.

Conditional Random Fields We suggest that the toxicity level of a word can be context-dependent, so we also experiment with sequence labeling models. We try Conditional Random Fields (CRF) (Lafferty et al., 2001) model. It uses the following features: the word itself, the word's part of speech, whether the word is a digit and consists of uppercase letters. Each word is represented with these features of the current, previous, and next words. The model performs closely to the attention-based classifier.

Sequence labeling with LSTM Finally, we experiment with the LSTM architecture (Hochreiter and Schmidhuber, 1997). We implement a Bi-LSTM network and also train an LSTM tagger from the AllenNLP library.[3] We do not use any pre-

[3]`https://github.com/allenai/allennlp`

trained embeddings in the Bi-LSTM model and use two versions of the AllenNLP LSTM: without pre-trained embeddings and with GloVe embeddings (Pennington et al., 2014).

5 Evaluation

5.1 Experimental Setting

For each transfer learning model, we use two-stage fine-tuning. We first train only the output layers of the models with the learning rate of 10^{-3}, and then the whole models with the learning rate of 10^{-5}. In both cases, we use linear learning rate warm-up for 3000 steps. We use the AdamW optimizer (Loshchilov and Hutter, 2019) and the batch size of 8, and early stopping to determine the number of training steps. We use the `transformers`[4] library for training.

For all the models which use the additional data from Jigsaw, we apply this scheme twice. The **pseudo-label** model is first fine-tuned on the original toxic spans dataset, and then on the self-labeled Jigsaw dataset, whereas the **RoBERTa classifier + tagger** and **tagging classifier** are first fine-tuned on the Jigsaw dataset (as classifiers), and then on the toxic spans dataset (as taggers).

5.2 Results

The scores of our models and the competing systems are shown in Table 1. Our best submitted system (**tagging classifier**) had the F_1-score of 0.681 on the test set, while the best team over the whole task got 0.708. This brings our team to the top 20% of the leaderboard. The pseudo-labeling approach was only marginally worse, scoring 0.674. Simply fine-tuning RoBERTa only on the tagging problem scored 0.668. On the other hand, none of our baselines could approach this result. Our best baseline is the word-based LogReg classifier. Apparently, other models fail to learn even the toxic vocabulary because their word representations are not informative enough.

While our best-performing model is only 18th best out of 92 participating systems, the results of the top systems are fairly close to ours (the difference is less than 2.5%). The variation of deep learning models often falls in this margin (Reimers and Gurevych, 2017). For our models, the sample standard deviation of the F_1-score is about 0.9%, so the difference between their performance is likely to be statistically insignificant.

[4] `https://huggingface.co/transformers`

Model	F_1 score
Top-5 participants	
HITSZ-HLT	0.708
S-NLP	0.707
hitmi&t	0.698
L	0.698
YNU-HPCC	0.696
Our models	
tagging classifier	0.683
pseudo-labeling	0.682
RoBERTa tagger	0.678
RoBERTa classifier + tagger	0.670
Our baselines	
Word-based LogReg	0.556
LSTM basic embeddings	0.538
Bi-LSTM basic embeddings	0.530
Attention-based Logreg	0.524
CRF	0.523
LSTM Glove embeddings	0.497

Table 1: Performance of our models (baselines and RoBERTa-based models) and their comparison with the 5 best-performing participants. Models within each section are sorted from best to worst.

An important hyper-parameter of the models is the probability threshold. It is usually fine-tuned on the development set. However, the development set provided for the task is too small. The threshold fine-tuned on it performs even worse than the standard threshold of 0.5. Thus, during the evaluation period of the competition, we tried submitting models with different threshold values. While this is not a completely fair practice because we indirectly used the test for tuning a model parameter, we suspect that many teams were overfitting to the test set in a similar way. We suggest that in order to make the evaluation fair, the results of the models on the final test set should not be available before the end of the competition, even in the indirect way (i.e., in the form of teams ranking without scores as it was done in this competition).

The best results which we report here were achieved with the threshold of 0.6 (see Table 1). We compare these results with those of the same models with the default threshold of 0.5 in Table 3. It shows that these scores are lower by up to 1%.

Another hyper-parameter of our models is the method of converting token-level labels to

Aggregation Type	F_1 score
No aggregation	0.685
Aggregation of word-level scores	
max token score	0.673
min token score	0.641
average token scores	0.670
naive Bayes	0.653

Table 2: Scores of the **tagging classifier** model with different token aggregation methods (computed on the development set).

Model	threshold	
	0.5	**0.6**
tagging classifier	0.681	**0.683**
pseudo-labeling	0.674	0.682
RoBERTa tagger	0.668	0.678
RoBERTa classifier + tagger	0.664	0.670

Table 3: F_1-scores of models with different probability thresholds.

character-level labels. We compare different methods on the development set (see Table 2). Surprisingly, the prediction of labels for each token individually with no aggregation works better than assigning labels to the whole words. This might happen because the attempts to decode words consistently lead to the propagation of wrongly predicted labels. Following this observation, we use the no-aggregation strategy for all models.

5.3 Efficiency of pre-training

To understand the effect of the use of additional sentence-labeled data, we compare the performance of **RoBERTa tagger** (a model which uses only the toxic span dataset) and **tagging classifier** (a model which uses sentence-labeled Jigsaw data in addition to the toxic span dataset) models trained on subsets of the data of different sizes. We would like to see if the usefulness of additional sentence-labeled data reduces as we get more data with token-level labeling.

Figure 2 plots the F_1-scores of the two models trained on datasets of sizes between 10 and 7,940 sentences. It shows that when the training set size is between 10 and 1,000, pre-training with sentence-level annotations gives a considerable boost in performance. However, the effect of this pre-training becomes insignificant after the size of the data with word-level labeling reaches around 3,000. Thus, this pre-training strategy is efficient only in cases when the size of the data with word-level labeling is very small.

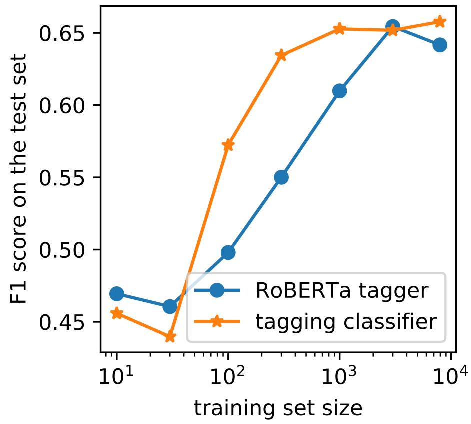

Figure 2: Learning curves for two transfer learning models, with (**tagging classifier**) and without (**RoBERTa tagger**) additional sentence-level data.

5.4 Error analysis

We analyze the errors of our best submitted system (**tagging classifier**) by comparing its predictions with the ground truth labels released after the end of the competition.

The vocabulary of false negative spans (527 unique tokens) is more diverse than that of false positives (275 unique tokens), while the number of false positives and false negatives in the test set is comparable (860 vs 813 tokens). It may indicate that the model is cautious and prefers to highlight only the hypotheses which have high confidence, while human annotators are more creative in their analysis. We give some examples of correct and incorrect labelings by our model in Table 4.

The most frequent false positive words characterize incompetence or lack of mental capacities: *stupid*, *idiot*, *ignorant*, *moron*, *dumb*, etc. Other frequent false positives are derogatory (*pathetic*, *ridiculous*, *ass*, *garbage*, *loser*, etc.), denounce particular misdeeds (*liar*, *troll*, *racist*, *hypocrite*, etc.), or express general negativity (*damn*, *fuck*, etc.). It is not obvious why human annotators label them as toxic in some cases, and as non-toxic in other cases. We suspect that inter-annotator agreement on such words is not very high.

The most frequent false negative words are function words: *and*, *the*, *are*, *a*, *you* etc. It happens because annotators sometimes label the whole text

Correct labeling
See a shrink you **pathetic troll**.
They're not patriots. They're **vandals**, **thieves**, and **bullies**.
Trudeau and Morneau are fiscally and economically **inept** and **incompetent**.
Incorrect labeling
That's right. They are not normal. And I am starting from the premise that they are ABNORMAL. Proceed wth the typical racist, **bigot, sexist rubbish**. Thanks!
ADN is endorsing, without officially endorsing. Bunch of **cowards**!!!
Rabidly **anti-Canadian** troll.

Table 4: Examples of ground truth (underlined) and predicted (**in bold**) toxic spans

Top 20 false positive words		Top 20 false negative words	
stupid	ass	and	of
idiot	liar	the	have
ignorant	garbage	are	loser
moron	loser	a	crap
dumb	fools	ignorant	all
idiots	troll	racist	chemical
fool	crap	you	that
pathetic	damn	in	not
stupidity	fuck	is	bunch
ridiculous	clown	to	dumb

Table 5: The most common false positive and false negative words

or a large chunk of it as toxic. The more meaningful false negatives belong to the same classes as the false positives (*ignorant*, *racist*, *loser*, etc.). The most common false positive and false negative words are listed in Table 5.

In general, the performance on this task might be limited to 0.7 F_1-score (the quality of the best-performing model) by the ambiguity of the annotations. In future work, it

6 Conclusions

We present a number of models for the detection of toxic spans within toxic sentences. All models are RoBERTa language models fine-tuned on the data with character-level labeling of toxic spans. In addition to that, we perform fine-tuning on an additional dataset with sentence-level toxicity labeling. This yields an improvement. However, our analysis shows that the effect of such pre-training is marginal when the main dataset size exceeds 1,000 samples. Therefore, substantial improvement is observed for small dataset sizes. Nevertheless, the models we propose can be useful in extremely low-resource scenarios.

Our model performs closely to the winning systems. We suggest that the differences between the 20 top models might be attributed to the variation of deep learning models and overfitting the test set. In addition to that, the error analysis shows that some errors in our model might be due to inconsistencies in the test data.

We release the code required to reproduce our experiments online.[5]

Acknowledgements

This work was conducted in the framework of the joint Skoltech-MTS laboratory.

References

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*

[5] https://github.com/skoltech-nlp/toxic-span-detection

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.

Kai Hakala and Sampo Pyysalo. 2019. Biomedical named entity recognition with multilingual BERT. In *Proceedings of The 5th Workshop on BioNLP Open Shared Tasks*, pages 56–61, Hong Kong, China. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced BERT with disentangled attention. *CoRR*, abs/2006.03654.

Shexia He, Zuchao Li, and Hai Zhao. 2019. Syntax-aware multilingual semantic role labeling. 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 5350–5359, Hong Kong, China. Association for Computational Linguistics.

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

Hossein Hosseini, Sreeram Kannan, Baosen Zhang, and Radha Poovendran. 2017. Deceiving google's perspective API built for detecting toxic comments. *CoRR*, abs/1702.08138.

Fajri Koto, Afshin Rahimi, Jey Han Lau, and Timothy Baldwin. 2020. IndoLEM and IndoBERT: A benchmark dataset and pre-trained language model for Indonesian NLP. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 757–770, Barcelona, Spain (Online). International Committee on Computational Linguistics.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Proceedings of the 18th International Conference on Machine Learning 2001*.

Dong-Hyun Lee et al. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3.

João Augusto Leite, Diego Silva, Kalina Bontcheva, and Carolina Scarton. 2020. Toxic language detection in social media for Brazilian Portuguese: New dataset and multilingual analysis. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 914–924, Suzhou, China. Association for Computational Linguistics.

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

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

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

João Moura, miguel vera, Daan van Stigt, Fabio Kepler, and André F. T. Martins. 2020. Ist-unbabel participation in the wmt20 quality estimation shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1029–1036, Online. Association for Computational Linguistics.

Kadir Bulut Ozler, Kate Kenski, Steve Rains, Yotam Shmargad, Kevin Coe, and Steven Bethard. 2020. Fine-tuning for multi-domain and multi-label uncivil language detection. In *Proceedings of the Fourth Workshop on Online Abuse and Harms*, pages 28–33, Online. Association for Computational Linguistics.

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*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Marek Rei and Anders Søgaard. 2018. Zero-shot sequence labeling: Transferring knowledge from sentences to tokens. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 293–302, New Orleans, Louisiana. Association for Computational Linguistics.

Marek Rei and Anders Søgaard. 2019. Jointly learning to label sentences and tokens. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6916–6923.

Nils Reimers and Iryna Gurevych. 2017. Reporting score distributions makes a difference: Performance study of LSTM-networks for sequence tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 338–348, Copenhagen, Denmark. Association for Computational Linguistics.

Cicero Nogueira dos Santos, Igor Melnyk, and Inkit Padhi. 2018. Fighting offensive language on social media with unsupervised text style transfer. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 189–194, Melbourne, Australia. Association for Computational Linguistics.

Allen Schmaltz. 2019. Detecting local insights from global labels: Supervised & zero-shot sequence labeling via a convolutional decomposition. *arXiv preprint arXiv:1906.01154*.

Minh Tran, Yipeng Zhang, and Mohammad Soleymani. 2020. Towards a friendly online community: An unsupervised style transfer framework for profanity redaction. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2107–2114, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Wei Wang, Bin Bi, Ming Yan, Chen Wu, Jiangnan Xia, Zuyi Bao, Liwei Peng, and Luo Si. 2020. Structbert: Incorporating language structures into pre-training for deep language understanding. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Ellery Wulczyn, Nithum Thain, and Lucas Dixon. 2017. Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th international conference on world wide web*, pages 1391–1399.