CSECU-DSG at SemEval-2021 Task 5: Leveraging Ensemble of Sequence Tagging Models for Toxic Spans Detection

Tashin Hossain, Jannatun Naim, Fareen Tasneem,
Radiathun Tasnia, and Abu Nowshed Chy
Department of Computer Science and Engineering
University of Chittagong, Chattogram-4331, Bangladesh
{tashin.hossain.cu,jannatun.naim.cu,fareen.tasneem,
radia.tasnia.cu}@gmail.com and nowshed@cu.ac.bd

Abstract

The upsurge of prolific blogging and microblogging platforms enabled the abusers to spread negativity and threats greater than ever. Detecting the toxic portions substantially aids to moderate or exclude the abusive parts for maintaining sound online platforms. This paper describes our participation in the SemEval 2021 toxic span detection task. The task requires detecting spans that convey toxic remarks from the given text. We explore an ensemble of sequence labeling models including the BiLSTM-CRF, spaCy NER model with custom toxic tags, and fine-tuned BERT model to identify the toxic spans. Finally, a majority voting based fusion method is used to determine the unified toxic spans. Experimental results depict the competitive performance of our model among the participants.

1 Introduction

Social media being a key factor in the world dynamics and toxicity in user-generated contents is a real threat. Threats and hatred instigated in posts and blogs implants fear in users' minds and prevents them from sharing their creative thoughts, valuable opinions to critical information. Sometimes it leads to severe mental trauma and fatalities. Hence, it is a formidable task to precisely detect toxicity in comments and posts to be able to moderate those portions and provide the users a safe online platform to express themselves.

Toxic span detection is a process where the specific toxic segment of a text is detected instead of detecting the whole text as toxic. The goal of this task is to eradicate the vagueness that is present in simple toxic text classification models and help the moderator to precisely moderate the toxic portions instead of the whole post. To elucidate the task, two examples are presented in Table 1.

Text#1: How fucking stupid are you? **Span:** [4, 5, 6, 7, 8, 9, 10, 12, 13, 14, 15, 16, 17]
Text#2: What a sociopathic and parasitic leader we have. **Span:** [7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 23, 24, 25, 26, 27, 28, 29, 30, 31]

Table 1: Example of sample texts with toxic spans.

Here, the "fucking stupid" portion of Text#1 is toxic and is attacking the personality of the second person, so the indices of this portion are included in the toxic span. In Text#2, the "sociopathic and parasitic" fragment is used as a toxic adjective to describe the leader in that context. Consequently, the indices of this fragment are incorporated in the span. We need to detect such spans accurately to remove toxicity from user content and preserve the safe and sound flow of online information.

Toxic content detection on online platforms is a state-of-the-art notion. Numerous works have been done on the binary and multi-label classification of toxic texts. For instance, Georgakopoulos et al. (Georgakopoulos et al., 2018) investigated the impact of CNN in toxic comment classification against the traditional bag-of-words approaches. A multiple word embedding-based approach was adopted by Carta et al. (Carta et al., 2019) for multi-class multi-label toxic comment classification. Besides, the effectiveness of feature extraction in hate speech detection was explored by Schmidt et al. (Schmidt and Wiegand, 2017). Multitude of datasets on toxic comments such as dataset based on Wikipedia discussion comments (Wulczyn et al., 2017), comments on online forums (Borkan et al., 2019a), and offensive language identification dataset (OLID) (Zampieri et al., 2019) were also introduced.

However, very few works detect the precise toxic span from text contents. Katsiolis et al. (Katsiolis, 2020) experimented on both unsupervised and su-

The first four authors have equal contributions.

Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pages 990–994
Bangkok, Thailand (online), August 5–6, 2021.

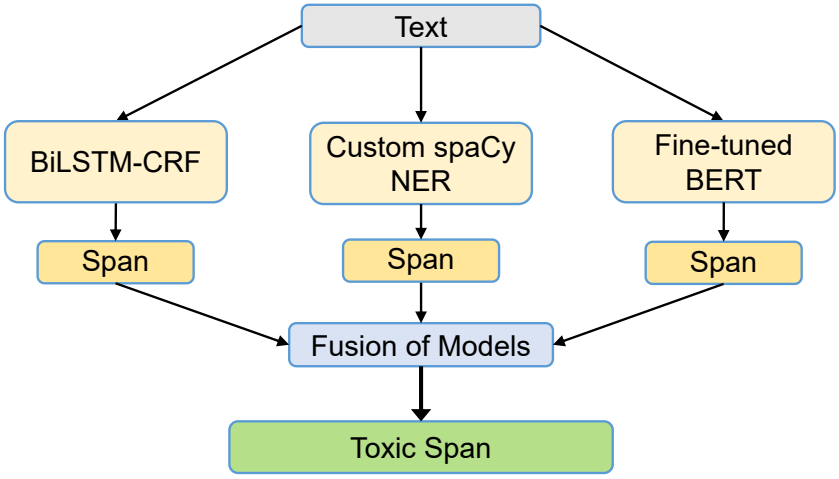

Figure 1: Overview of our proposed framework.

pervised methods to address this challenge. The unsupervised methods include the input erasure method and the LIME algorithm whereas the supervised method implements sequence labeling through a BERT model. The unintended bias created in publicly used toxicity detection models due to many reasons such as the influence of regional culture was investigated by Borkan et al. (Borkan et al., 2019a). John et al. (Pavlopoulos et al., 2017) surveyed the impact of user embeddings, user type embeddings, user biases, or user type biases on the RNN-based moderation method.

In this paper, we portray our insights acquired from experimenting on this task. We propose an approach focusing on an ensemble of sequence labeling models including the BiLSTM-CRF, spaCy NER model with custom toxic tags, and fine-tuned BERT model. We procure the spans from these models through a majority voting scheme to determine the final toxic spans.

The organization of this paper is as follows: we elucidate our proposed framework in Section 2. Section 3 encompasses the experimental details and comparative performance analysis. Finally, we conclude this paper with some future notions in Section 5.

2 Proposed Framework

We cast the toxic span detection as a sequence tagging task and employ an ensemble of sequence tagging models. Our proposed system comprises three individual models. The framework of our system is depicted in Figure 1. The first model is a BiLSTM-CRF model with the BIO tagging scheme. The second model is a custom spaCy named entity recognition (NER) model. The third model is a fine-tuned BERT model for token classification. We leverage these three models as sequence tagging models. These models generate tags in token level for a text. Subsequently, we extract span based on the toxic tags. Finally, we apply a majority voting based fusion scheme on these spans and determine the final toxic spans.

2.1 BiLSTM-CRF

The BiLSTM-CRF model is well-known for sequence-tagging tasks such as named entity recognition (NER). We utilize the model implemented by (Reimers and Gurevych, 2017). For training purposes, the dataset needs to be in CoNLL-2003[2] format where two columns for tokens and BIO tags are required. Since it requires the text to be in a tokenized form, we tokenize the text using NLTK TweetTokenizer (Bird et al., 2009). After tokenization, we label the tokens with custom tags such as B-TOX(begin), I-TOX(inside), and O(outside) utilizing the toxic span from the training dataset. These tokens are then sent to the embedding layer. The embedding layer has three variants of embeddings: word embedding, casing feature or capitalization feature, and character embedding. We employ pre-trained GloVe (6B) (Pennington et al., 2014) word embedding and a CNN based character embeddings (Ma and Hovy, 2016). The embedding vectors are concatenated and the output is fed to the BiLSTM encoder which tags tokens with the BIO tagging scheme. The BiLSTM encoder is followed by a CRF classifier where the tags are optimized enforcing the intermediate logic of tags.

[2]https://www.clips.uantwerpen.be/conll2003/ner/

2.2 Custom spaCy NER

We exploit the spaCy (Honnibal and Montani, 2017) to build an NER type sequence labeling model with the custom tag "TOXIC". We convert the dataset to spaCy entity format and load a spaCy blank English model. We append new word vectors utilizing a pre-trained word2vec (Mikolov et al., 2013) model. Consequently, we add NER pipeline to the model and also a "TOXIC" label. We disable all the pipelines except NER and loop through the training dataset several times.

2.3 Fine-tuned BERT

We finetune the state-of-the-art bert-large-cased model (Devlin et al., 2019) to identify the toxic spans. We employ the BertForTokenClassification (Wolf et al., 2019) method to perform the token level tagging. This method classifies level for each tokenized word in a sentence. To generate the training data, we convert the sentence into tokens and annotate them with spans. We tag the tokens as "non-toxic" and "toxic" whereas the tokens that are tagged as "toxic" are in between the spans.

2.4 Fusion of Models

An ensemble approach is a simulation that constructs multiple models and then blends them to bring out improved results. To obtain a more accurate solution than a single model, we apply majority voting (Rokach, 2010) on the spans generated from three models as shown in Figure 1. The primary idea is based on the frequency of the span elements. If a span is predicted by at least two models, it is included in the final predicted span. Thus, we obtain our final toxic spans through majority voting.

3 Experiments and Evaluations

3.1 Dataset Description

For detecting toxic spans in posts, we used the Civil Comments Dataset (Borkan et al., 2019b) which consists of 10K toxic comments. The whole dataset is divided into three subsets where the train, trial, and test set comprises 7939, 690, and 2000 comments, respectively. Toxic comments are mainly divided into two portions: 1. Having no toxic spans and 2. Having toxic spans that are identified as spans with specific character positions. Analyzing the ratio of empty and toxic spans in our dataset we found that 90% of data occupies toxic spans where only 10% data have empty spans. F1-Score is used as the primary evaluation metric in this task.

3.2 Experimental Setup

In our CSECU-DSG system submitted to the SemEval-2021 Task 5 (Pavlopoulos et al., 2021), we make use of three sequence and entity tagging models to get better predictions. We present the configuration of our best submitted system in Table 2. Based on the predicted spans from these models, a majority voting has been applied.

System	Settings
BiLSTM-CRF	1. *dropout*: (0.25, 0.25) 2. *LSTM-Size*: [100, 100] 3. *maxCharLength*: 50 4. *Tokenizer*: TweetTokenizer 5. *Word embedding*: GloVe (6B) 6. *Optimizer*: nadam 7. *miniBatchSize*: 32 8. *Epochs*: 25
Custom spaCy NER	1. *spaCy Model*: blank ('en') 2. *Pipeline*: ner 3. *Word embedding*: word2vec 4. *Iteration*: 30 5. *drop*: 0.5
Fine-tuned BERT Model	1. *Tokenizer*: bert-large-cased 2. *Optimizer*: AdamW 3. *Batch Size*: 16 4. *Learning_rate*: 2e-5 5. *weight_decay_rate:* 0.01 6. *Epochs*: 25

Table 2: System settings.

3.3 Results Analysis

Now, we compare the performance of our system against other competitors' systems. Among the 91 valid submissions, the comparative performance with top-performing teams depicted in Table 3.

Team Name	F1-Score
HITSZ-HLT9 (1st)	0.7083028253
hitmi&t (3rd)	0.6984762534
IITK@Detox (9th)	0.6895352367
CSECUDSG (21st)	**0.6795264755**
mnfourka (45th)	0.6581458018
ST_TSResearch (64th)	0.6133591537

Table 3: Comparative performance analysis.

It depicts that our system achieved competitive performance compared to the participants' systems. It only lacks by 3% from the top-performing team HITSZ-HLT.

4 Discussion

To estimate the impact of individual components on the overall system's performance, we examine the performance of individual models on the test set. To do this, we make use of the test set and the findings are presented in Table 4.

Method	F1-Score
CSECU-DSG	0.6795264755
Performance of Individual Model	
−Fine-tuned BERT	0.6381618923
−Custom spaCy NER	0.6474175682
−BiLSTM-CRF	0.6404340000

Table 4: Performance analysis of individual models.

It shows that all three models obtained a similar kind of performance. However, employing the majority voting based scheme on these three models improves the overall result by almost 3% which leads to better detection of toxic spans from the text. Thus, we demonstrate the efficacy of utilizing the ensemble strategy to ameliorate the performance.

To qualitatively demonstrate the effectiveness of the ensemble approach compared to the individual models an instance is illustrated in Table 5. It clearly shows that majority voting helps to detect the accurate span.

Text: They are more animal than the goat, disgusting.!!!!!
Gold: [**36, 37, 38, 39, 40, 41, 42, 43, 44, 45**]
BiLSTM-CRF: [30, 31, 32, 33, **36, 37, 38, 39, 40, 41, 42, 43, 44, 45**]
Custom spaCy NER: [**36, 37, 38, 39, 40, 41, 42, 43, 44, 45**]
Fine-tuned BERT: [14, 15, 16, 17, 18, 19, **36, 37, 38, 39, 40, 41, 42, 43, 44, 45**]
Majority voting: [**36, 37, 38, 39, 40, 41, 42, 43, 44, 45**]

Table 5: Comparative performance analysis of models according to the predicted toxic spans.

We further investigate the reason behind the erroneous span detection by our proposed system. In this regard, we articulate some examples in Table 6.

Text	Predicted Span	Gold Span
1. See a shrink you pathetic troll.	[17, 18, 19, 20, 21, 22, 23, 24, 26, 27, 28, 29, 30]	[17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30]
2. ADN is endorsing, without officially endorsing. Bunch of cowards !!!	[58, 59, 60, 61, 62, 63, 64]	[49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64]
3. The mascot was a ridiculous pick twenty years ago, too. Did you ever see the welcome sign going into Keenesburg? "Home to 500 people and a few soreheads."	[17, 18, 19, 20, 21, 22, 23, 24, 25, 26]	17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31]

Table 6: Examples of erroneous span detection.

We observed that our system could not detect the in-between spaces of toxic words. Such as in the first example, the predicted span is "pathetic" (17-24) and "troll" (26-30). Whereas, the gold span is "pathetic troll"(17-30). The probable reason for this can be that our models are trained with tokens of the training dataset. Another observation indicates that our system failed to detect the phrasal spans of some texts. In example #2 and #3, we see that instead of capturing the toxic phrases "Bunch of cowards" and "ridiculous pick", it detects the toxic words only. Since two of our models are trained on token-level and only the BiLSTM-CRF model follows the BIO tags convention, the ensemble of models lacks in perceiving the context of the phrasal toxic texts and sometimes fragments the toxic sequences. Though majority voting improves the overall score, it shrinks some important features of the discrete models.

5 Conclusion and Future Directions

In this paper, we introduced an ensemble of three distinct models to detect the toxic spans. Among these models, BiLSTM-CRF and Custom spaCy NER models are implemented as NER type sequence and entity tagging models whereas fine-tuned BERT model is exploited as a token classification model. We also leveraged a majority voting strategy to overcome the limitations of individual models. Our model tackles the task challenge effectively and achieved a competitive performance compared to the participants' systems.

Our future plan incorporates exploring a better sequence tagging model with an ensemble of various fine-tuned language models i.e. ALBERT, DistilBERT, RoBERTa, and GPT.

References

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

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.

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.

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 (NAACL:HLT)*, pages 4171–4186.

Spiros V Georgakopoulos, Sotiris K Tasoulis, Aristidis G Vrahatis, and Vassilis P Plagianakos. 2018. Convolutional neural networks for toxic comment classification. In *Proceedings of the 10th hellenic conference on artificial intelligence*, pages 1–6.

Matthew Honnibal and Ines Montani. 2017. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 7(1):411–420.

Athanasios Katsiolis. 2020. Toxic span detection in online posts.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

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*.

John Pavlopoulos, Prodromos Malakasiotis, Juli Bakagianni, and Ion Androutsopoulos. 2017. Improved abusive comment moderation with user embeddings. *arXiv preprint arXiv:1708.03699*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Nils Reimers and Iryna Gurevych. 2017. Optimal hyperparameters for deep lstm-networks for sequence labeling tasks. *arXiv preprint arXiv:1707.06799*.

Lior Rokach. 2010. Ensemble-based classifiers. *Artificial Intelligence Review*, 33(1-2):1–39.

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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

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.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Predicting the type and target of offensive posts in social media. *arXiv preprint arXiv:1902.09666*.