LISAC FSDM USMBA at SemEval-2021 Task 5: Tackling Toxic Spans Detection Challenge with Supervised SpanBERT-based Model and Unsupervised LIME-based Model

Abdessamad Benlahbib[1]*, **Ahmed Alami**[2], **Hamza Alami**[2]*,
[1] LISAC Laboratory, Faculty of Sciences Dhar EL Mehraz (F.S.D.M),
Sidi Mohamed Ben Abdellah University (U.S.M.B.A)
[2] Laboratory of Engineering Sciences, National School of Applied Sciences,
Ibn Tofail University, Kenitra, Morocco
abdessamad.benlahbib@usmba.ac.ma, alami.alami1996@gmail.com,
hamza.alami1@usmba.ac.ma

Abstract

Toxic spans detection is an emerging challenge that aims to find toxic spans within a toxic text. In this paper, we describe our solutions to tackle toxic spans detection. The first solution, which follows a supervised approach, is based on SpanBERT model. This latter is intended to better embed and predict spans of text. The second solution, which adopts an unsupervised approach, combines linear support vector machine with the Local Interpretable Model-Agnostic Explanations (LIME). This last is used to interpret predictions of learning-based models. Our supervised model outperformed the unsupervised model and achieved the f-score of 67,84% (ranked 22/85) in Task 5 at SemEval-2021: Toxic Spans Detection.

1 Introduction

By dint of the massive production of user-generated content in social media, moderation becomes crucial to promote healthy online discussions by removing toxic posts and contents. However, it is nearly impossible for a human to keep tracking user-generated content. Thus, the need for the right tools and technologies to help in such a task becomes a necessity.

The Toxic Spans Detection task aims to detect the spans that make a text toxic. While several toxicity detection datasets (Wulczyn et al., 2017; Borkan et al., 2019) and models (Pavlopoulos et al., 2017a, 2019; Schmidt and Wiegand, 2017; Pavlopoulos et al., 2017b; Zampieri et al., 2019; Alami et al., 2020) have been released. However, these works estimate the likelihood of a document being toxic with weak interpretability. In fact, highlighting toxic spans can assist human moderators who often deal with lengthy comments, and who prefer attribution instead of just a system-generated unexplained toxicity score per post.

*contributed equally

In this paper, we propose two solutions to tackle toxic spans detection (Pavlopoulos et al., 2021). The first solution, which follows a supervised approach, is based on SpanBERT (Joshi et al., 2020) model that is pre-trained on span boundary objective and considers masks contiguous spans. Therefore, SpanBERT gives better spans representations and predictions. The second solution, which adopts an unsupervised approach, combines linear support vector machine (Fan et al., 2008) with the Local Interpretable Model-Agnostic Explanations (LIME) (Ribeiro et al., 2016). LIME is an explanation technique that seeks to faithfully interpret the predictions of any classifier.

This paper is organized as follows: Section 2 describes the proposed methods; Section 3 presents the experimental results; Finally, Section 4 concludes and outlines future directions.

2 Methods

In this section, we describe the proposed solutions including SpanBERT-based method which is based on supervised approach, and SVM and LIME-based method that is based on unsupervised approach.

2.1 SpanBERT-based method

We use SpanBERT (Joshi et al., 2020) a pre-trained model built to improve spans of text representation and prediction. It differs from BERT (Devlin et al., 2019) as it (1) masks contiguous random spans, instead of random tokens; and (2) is trained on span-boundary objective, i.e., the model is optimized to predict the masked span given tokens at its boundary. We considered the toxic span text detection as a sequence labeling task. Thus, we performed a transformation to the dataset and fine-tuned SpanBERT to this specific task.

Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pages 865–869
Bangkok, Thailand (online), August 5–6, 2021.

2.1.1 Data preparation

The raw dataset consists of a set of toxic texts where each element is annotated with an array that contains characters' indices. These indices are considered as the toxic span of text. In order to train SpanBERT on this dataset, we applied the pre-trained SpanBERT tokenizer to tokenize sentences, and we built the target arrays by annotating words that contain toxic characters' indices. For instance, given a sentence that contains n tokens, then the target array contains n elements, where the elements that contain a toxic character are labeled as positive "1" otherwise they are labeled as negative "0". Figure 1 illustrates the pipeline of dataset preparation.

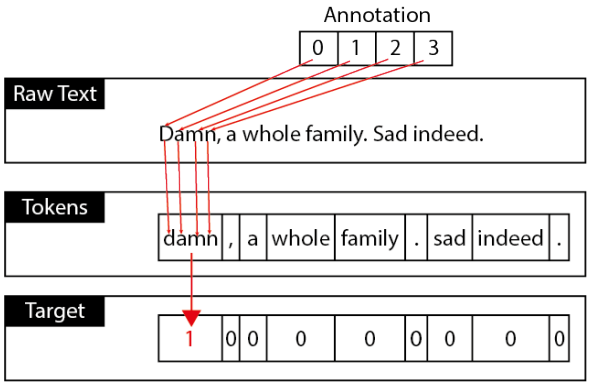

Figure 1: The pipeline of dataset preparation

2.1.2 Toxic spans detection

We considered the toxic span detection as a sequence labeling task. Therefore, we fine-tuned SpanBERT pre-trained model on token classification task. First, we tokenize the sentence and map its tokens into indices according to SpanBERT vocabulary. Next, we fed the model with tokens indices. Then, it computes tokens embeddings by applying SpanBERT pre-trained model. After that, we compute the probability if a given token is toxic by applying a linear layer followed by a softmax on tokens embeddings. Finally, the model is trained to optimize the cross-entropy loss. Figure 2 shows the flowchart of the SpanBERT-based model. It is worth noting that we filter predicted spans by removing toxic character offsets that have a size equal to one.

2.2 SVM and LIME-based method

2.2.1 Data preparation

The data preparation for our unsupervised method can be summarized as follows:

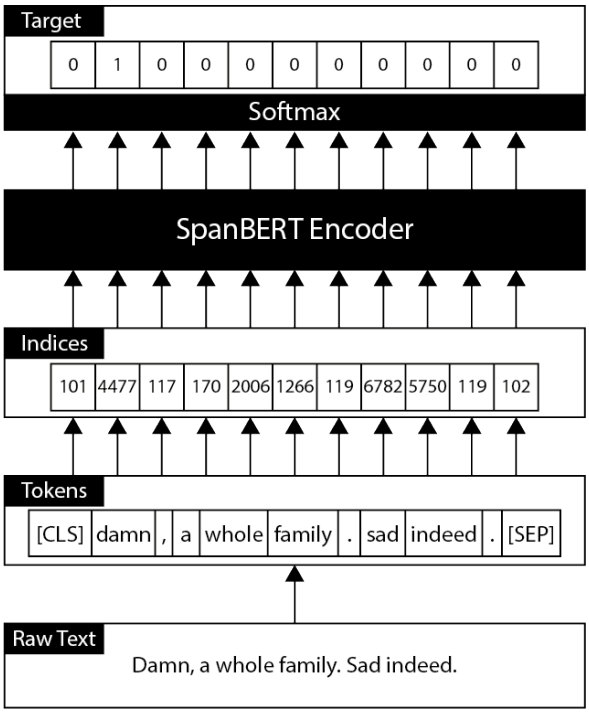

Figure 2: The flowchart of SpanBERT-based model

1. We combine both SemEval 2021 Task 5: Toxic Spans Detection training set which contains 7939 toxic comments, SemEval 2021 Task 5: Toxic Spans Detection test set that contains 2000 toxic comments, and 159571 comments (16225 toxic comments and 143346 non-toxic comments) from Kaggle Jigsaw Toxic comment classification challenge [1] in order to use them for training the linear support vector machine classifier. Later, We label the toxic comments with 1 and non-toxic comments with 0.

2. Word-level uni-grams and bi-grams are extracted, then vectorized using TF-IDF scores.

2.2.2 Toxic spans detection

The toxic spans detection task adopted by our unsupervised method can be summarized as follows:

1. We train the linear support vector machine classifier on 26164 toxic comments and 143346 non-toxic comments (the combination of SemEval 2021 Task 5: Toxic Spans Detection training set, SemEval 2021 Task 5: Toxic Spans Detection test set, and a subset of Kaggle Jigsaw Toxic comment classification challenge dataset).

[1] https://raw.githubusercontent.com/iampukar/toxic-comments-classification/master/train.csv

2. We apply the trained model on the SemEval 2021 Task 5: Toxic Spans Detection test set comments to predict their toxicity, then, we use the LIME technique to explain the predictions (Figure 3).

3. We discard words that contribute less to the toxic category by applying a thresholding technique. Words with a high influence score, greater or equal to the threshold, are considered toxic, therefore, we retrieve their character offsets (toxic spans).

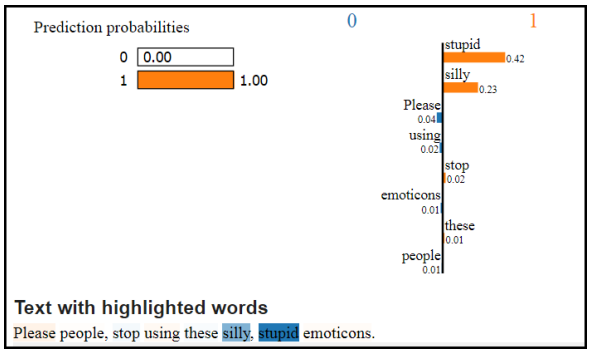

Figure 3: Lime explanations

By training the linear support vector machine classifier on the SemEval 2021 Task 5: Toxic Spans Detection test set, we guarantee that the model accurately predicts the toxicity of its comments with precision, recall, f-score, and accuracy of 1 (the model correctly predict the toxicity of all 2000 reviews in the test set). Besides, we ensure that the LIME explanations are somewhat accurate. In fact, if the model misclassifies the toxicity of the comments, the LIME explanations will be inaccurate since the latter will explain wrong predictions.

From Figure 3, we can see that the words "silly" and "stupid" contribute to the toxic category 42% and 23% respectively in the following toxic comment "Please people, stop using these silly, stupid emoticons". Since we only consider words with high influence scores for the toxic category (greater or equal to 0.13), we keep the two words "silly" and "stupid", and we discard the remaining words. Next, we retrieve their character offsets from the comment as shown in Table 1.

3 Experimental results

We experimented our models on the SemEval 2021 Task 5: Toxic Spans Detection dataset. The training set and test set contain 7939 and 2000 toxic comments labeled with their toxic spans. All our experiments have been conducted in Google Colab environment[2], The following libraries: Hugging Face[3], LIME[4], Scikit-Learn[5], and PyTorch[6] were used to train and to asses the performance of our models.

3.1 Evaluation Metric

In order to measure the performance of our models, we employ the F1 score proposed in (Da San Martino et al., 2019). Considering a post t and a system A_i which predict a set $S^t_{A_i}$ of toxic character offsets. Let denote by G^t the expected character offsets. Then, the $F1$ score of the model A_i with respect to G for t is computed in the following manner:

$$P^t(A_i, G) = \frac{\left|S^t_{A_i} \cap S^t_G\right|}{\left|S^t_{A_i}\right|} \quad (1)$$

$$R^t(A_i, G) = \frac{\left|S^t_{A_i} \cap S^t_G\right|}{|S^t_G|} \quad (2)$$

$$F^t_1(A_i, G) = \frac{2 \cdot P^t(A_i, G) \cdot R^t(A_i, G)}{P^t(A_i, G) + R^t(A_i, G)} \quad (3)$$

where $|\cdot|$ denotes set cardinality.

3.2 Performance Evaluation

On the one hand, we compared various pre-trained models, including BERT-base, BERT-large, DistilBERT (Sanh et al., 2019), and SpanBERT-large, to compute tokens embeddings. All the models are based on transformers (Vaswani et al., 2017) technique. The SpanBERT model achieves the best results due to the fact that is trained with contiguous masked spans and optimizes the span boundary objective. On the other hand, we compared the logistic regression LIME (LR-LIME) to linear support vector machine LIME (LSVM-LIME). The latter produces superior scores. Table 2 reports the obtained results for both supervised and unsupervised techniques. SpanBERT outperforms all the models by scoring about 0.6783 F1 score. During the fine-tuning of SpanBERT model, we set the hyper-parameters as follows: $1.5e-5$ as the learning rate, 3 epochs, 256 as the max sequence length, 4 as batch size, 476 as the warmup steps, and 0.01

[2]https://colab.research.google.com/
[3]https://huggingface.co/
[4]https://lime-ml.readthedocs.io/en/latest/lime.html
[5]https://scikit-learn.org/stable/
[6]https://pytorch.org/

Comment	Toxic spans
Please people, stop using these silly, stupid emoticons.	[32, 33, 34, 35, 36, 38, 39, 40, 41, 42, 43, 44]

Table 1: Example of unsupervised toxic spans detection

Method	F1 score
LR-LIME	0.5887938605
LSVM-LIME	0.592141639
DistilBERT	0.6636129383
BERT-base	0.6714433707
BERT-large	0.6677294902
SpanBERT-large	0.6783641122

Table 2: Toxic spans detection results

as the weight decay. For the unsupervised technique, several experiments have been conducted to reach the suitable threshold. Actually, 0.12 and 0.13 thresholds achieved the best performances for LR-LIME and LSVM-LIME, respectively.

4 Conclusion

In this paper, we described our models for tackling SEMEval 2021 Task 5: Toxic Spans Detection. Two approaches have been employed. A supervised approach based on transformers technique, where toxic sequences are tokenized and embedded using pre-trained models. We optimize the likelihood of a token to be toxic by minimizing the cross-entropy loss. SpanBERT scored the best results by achieving about 0.6783 F1 score. An unsupervised approach based on shallow machine learning and LIME, which is an explanation technique that explains the prediction of any classifier in an interpretable and faithful manner. Since the top-ranked score was about 0.7083 F1 score, future studies and works will focus on improving the performance of toxic spans detection task.

References

Hamza Alami, Said Ouatik El Alaoui, Abdessamad Benlahbib, and Noureddine En-nahnahi. 2020. LISAC FSDM-USMBA team at SemEval-2020 task 12: Overcoming AraBERT's pretrain-finetune discrepancy for Arabic offensive language identification. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 2080–2085, Barcelona (online). International Committee 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*, WWW '19, page 491–500, New York, NY, USA. Association for Computing Machinery.

Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeño, 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 5636–5646, Hong Kong, China. Association for Computational Linguistics.

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 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. Liblinear: A library for large linear classification. *J. Mach. Learn. Res.*, 9:1871–1874.

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 Pavlopoulos, Ion Androutsopoulos, Jeffrey Sorensen, and Léo Laugier. 2021. Semeval-2021 task 5: Toxic spans detection. In *Proceedings of the 15th International Workshop on Semantic Evaluation*. International Committee for Computational Linguistics.

John Pavlopoulos, Prodromos Malakasiotis, and Ion Androutsopoulos. 2017a. Deep learning for user comment moderation. In *Proceedings of the First Workshop on Abusive Language Online*, pages 25–35, Vancouver, BC, Canada. Association for Computational Linguistics.

John Pavlopoulos, Prodromos Malakasiotis, and Ion Androutsopoulos. 2017b. Deeper attention to abusive user content moderation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1125–1135, Copenhagen, Denmark. Association for Computational Linguistics.

John Pavlopoulos, Nithum Thain, Lucas Dixon, and Ion Androutsopoulos. 2019. ConvAI at SemEval-2019 task 6: Offensive language identification and categorization with perspective and BERT. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 571–576, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1135–1144, New York, NY, USA. Association for Computing Machinery.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *NeurIPS EMC2 Workshop*.

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, Valencia, Spain. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

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*, WWW '17, page 1391–1399, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

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). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 75–86, Minneapolis, Minnesota, USA. Association for Computational Linguistics.