11830 Report: Civility in Communication

Jiyang Tang
jiyangta@andrew.cmu.edu

1 Introduction

Toxic speech detection using machine learning has been a hot research topic in recent years. In this report, we investigate the methods of performing this task, using data sampled from the SemEval2019 challenge (Zampieri et al., 2019) and TwitterAAE dataset (Blodgett et al., 2016).

2 Method

We build three types of toxic speech classifiers and test their classification performance and the biases on out-of-domain non-toxic data.

2.1 PerspectiveAPI-based Classifier

We first use PerspectiveAPI to build a rule-based classifier. Perspective score represents the toxicity level of a piece of text. The classifier recognize a sentence as offensive if its Perspective score is larger than 0.8.

2.2 Linear Classifier using Word Count Vectors

The second baseline model is a linear classifier using word count feature vectors. The model can recognize offensive words in text, but may fail in other situations.

2.3 RoBERTa Sentence Classifier

The third classifier is built on top of the RoBERTa (Liu et al., 2019). There are several advantages of using RoBERTa. Its text tokenizer uses binary pair encoding (BPE) (Sennrich et al., 2015), which means the model utilize unicode characters such as emojis to perform classification. As a language model, RoBERTa recognizes contextual text information which should in theory improve the classification performance on hard cases. The model is also pre-trained on a large amount of data, so we only need finetune it for a small number of iterations.

3 Experiments

For all three classifiers, we report their F1 score, precision, recall, and accuracy on the development set of SemEval2019 data. And we present their false-positive rate (FPR) on TwitterAAE data average across all demographic groups.

3.1 Linear Classifier

For the linear classifier, we clean the text before extracting word count features. We use `spaCy` (Honnibal et al., 2020) `en_core_web_sm` model to tokenize raw strings into lists of lower-case words and remove all punctuations. Then we use `Ekphrasis` (Baziotis et al., 2017) library to normalize the text. `Ekphrasis` specializes in processing social media text which contains typos, urls, emojis and so on. We use it to normalize such components, unpack hashtags, fix elongated words, and convert emoticons to text. Finally, we use `sklearn` (Pedregosa et al., 2011) `CountVectorizer` to convert text into word count vectors.

3.2 RoBERTa Sentence Classifier

For the RoBERTa classifier, we feed the text directly to the tokenizer and rely on the model to handle special text components mentioned in the previous section.

4 Results and Discussion

5 Conclusion

In this report, we have created three toxic speech classifiers and analyzed their performance and biases.

Model	Accuracy	F1	Precision	Recall	FPR (Demo)
Perspective					
Linear					
RoBERTa					

Table 1: Results of three classifiers.

References

Christos Baziotis, Nikos Pelekis, and Christos Doulkeridis. 2017. Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 747–754, Vancouver, Canada. Association for Computational Linguistics.

Su Lin Blodgett, Lisa Green, and Brendan O'Connor. 2016. Demographic dialectal variation in social media: A case study of African-American English. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1119–1130, Austin, Texas. Association for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

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.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *CoRR*, abs/1508.07909.

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.