LING 573 Project Report: D2

Levon Haroutunian*, **Yi-Chien Lin***, **Bridget Tyree***, **David Yi***
Department of Linguistics
University of Washington
{levonh, yichlin, btyree, davidyi6}@uw.edu

Abstract

This project considers the task of offensive language detection. We use OLID, the data set from OffensEval 2019 (Zampieri et al., 2019b), to train an initial model to classify English tweets as offensive or non-offensive. In our next steps, we plan to make our model more complex by incorporating sequential information into our representations and improving the way we handle emojis and punctuation. We plan to adapt our model to classify multilingual tweets.

1 Introduction

This project focuses on detecting offensive speech in social media. The detection of offensive or abusive language has been a rich area of research in NLP, which is overviewed in Waseem et al. (2017). Automated abuse detection can be a helpful tool for content moderation, though it should be noted that such a tool is only one component of a system that can adequately address and prevent toxic behavior (Jurgens et al., 2019).

For our initial model, we use static GloVe word embeddings (Pennington et al., 2014) trained on Twitter data[1] to form a representation for each tweet, and feed those representations into a logistic regression classifier. This model achieves a Macro F1 score of around 0.598.

The rest of this paper is organized as follows. Section 2 details the task description, including both the primary task of detecting offensive language in English tweets and the secondary task of detecting offensive language in other languages. Sections 3 and 4 provide a system overview and describe our approach for our initial model. Section 5 provides the results from our initial classifier, which are analyzed in Section 6. Our analysis includes planned directions for improvements to future iterations of this project. Finally, Section 7 provides a conclusion.

2 Task Description

The tasks in this project are based on OffensEval 2019[2] (Zampieri et al., 2019b) and OffensEval 2020[3] (Zampieri et al., 2020), both of which were SemEval shared tasks. Description regarding the primary task and adaptation task are presented respectively in the following subsections.

2.1 Primary Task

For our primary task, we tackle the subtask A of OffensEval 2019 – identifying offensive language in English (i.e. determining whether a given statement is offensive). More specifically, the affect type of this task is interpersonal stance and the target is sentiment. The identification of offensive language is obtained by implementing binary text classification on tweets.

We use the same training and test sets as the ones used in the subtask A of OffensEval 2019: the Offensive Language Identification Dataset (OLID)[4] (Zampieri et al., 2019a). OLID is annotated tweets using crowdsourcing based on a three-layer annotation scheme. This annotation scheme enables developers to use the same dataset for different subtasks.

Our training data for the primary task contains 14,100 English tweets, in which 4,640 (33%) are annotated as offensive (OFF) and 9,460 (67%) are labeled as not offensive (NOT). The provided test set contains 240 OFF tweets and 620 NOT tweets.

*Equal contribution, sorted in alphabetical order.

[1] https://nlp.stanford.edu/projects/glove/

[2] https://sites.google.com/site/offensevalsharedtask/offenseval2019

[3] https://sites.google.com/site/offensevalsharedtask/results-and-paper-submission

[4] https://sites.google.com/site/offensevalsharedtask/olid

For evaluation, compute the Macro F1 score, which is the official evaluation metric of OffensEval 2019 (Zampieri et al., 2019b).

2.2 Adaptation Task

For our adaptation task, we plan to extend our system to classify tweets in the four other languages included in OffensEval 2020: Arabic, Danish, Greek, and Turkish. These datasets are annotated and preprocessed using the same guidelines as OLID (Zampieri et al., 2020). These datasets have a range of sizes: there are 3,020 Danish tweets; 10,000 Arabic tweets; 10,287 Greek tweets; and 35,284 Turkish tweets (Zampieri et al., 2020). The evaluation method is same as the one used for our primary task – calculating the Macro F1 scores of the classification results.

3 System Overview

As a baseline system, we implement a linear classification model that utilizes static pre-trained word embeddings. First, we tokenize the raw tweets from the OLID dataset into fixed-length token sequences. Then, for each tweet, we create a feature vector by concatenating the vector representation of each token in the sequence while preserving the order. Finally, we feed the concatenated feature vector into a linear classifier to predict whether a tweet is offensive (OFF) or not offensive (NOT).

4 Approach

This section presents detailed descriptions of the approaches used in our system. First we present the way that we preprocess and tokenize tweets from their raw text form. Then, we show how the tweets are featurized using static pretrained word embeddings. Lastly, we provide a description of the classifier used to classify the featurized tweets.

4.1 Featurizing the Tweets

To convert a raw tweet into a fixed-length token sequence, we use the text tokenizer from Keras[5] which removes all punctuation by default and splits text into a space-separated sequence of tokens. Each token is additionally assigned a unique index which serves as its integer representation. Because tweets can be of variable length and a linear classification model like Logistic Regression requires a fixed-length input vector, we pad the end of the sequence with 0 (where 0 is an empty index that does not correspond to any token) if it is less than 50 tokens in length and we truncate the sequence if it has more than 50 tokens.

For word vector representations, we leverage the 200-dimensional pre-trained Twitter embeddings from GloVe (Pennington et al., 2014) which were trained on 2 billion tweets. We form a feature vector for each tweet by horizontally stacking the embedding vector representation for each token in the tweet to form a 10000-dimensional vector. For tokens that are not found in the pretrained GloVe embeddings, we simply assign the zero vector. Instead of summing or averaging the word vectors, we choose to concatenate the token vectors with the goal of preserving sequential properties from the original tweet.

4.2 Performing Binary Classification

For classification, we adopt the logistic regression classification model. Logistic regression uses the logistic function to model a binary output in terms of probabilities. We incorporate the logistic regression classifier from the scikit-learn library (Pedregosa et al., 2011). More specifically, for the regression classifier, we use L2 regularization. To avoid our model being biased by the class imbalance data, we set the `class_weight` parameter to be `balanced`. Lastly, we select the limited-memory BFGS as the algorithm for the optimization problem.

5 Results

For the evaluation of our system, since the OLID dataset does not contain a distinguished development dataset, we designated 10% of our training data to be our development dataset. Table 1 presents the F1-Macro of baseline model provided in Zampieri et al. (2019b) (**SVM**) and our system (**OUR MODEL**). In addition, Table 1 also includes baseline Macro-F1 on all offensive tweets (**ALL OFF**) and all non-offensive tweets (**ALL NOT**) (Zampieri et al., 2019a). As shown in Table 1, the SVM model performs slightly better than our system. However, our system has a higher Macro-F1 than ALL OFF and ALL NOT – our model performs better than just choosing not offensive every time or just choosing offensive every time.

Table 2 shows a confusion matrix of the results of running our model on the development dataset.

[5] https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/text/Tokenizer

	F1 Score
SVM	0.690
ALL OFF	0.220
ALL NOT	0.420
OUR MODEL	0.598

Table 1: F1-Macro of baseline systems and our system

Our model has a higher accuracy classifying non-offensive tweets than offensive ones. It correctly predicts offensive tweets roughly 43% of the time, and correctly predicts non-offensive tweets roughly 76% of the time.

Confusion Matrix		Predicted Label	
		negative	positive
True Label	negative	667	210
	positive	254	193

Table 2: Confusion Matrix based on the results from our system

6 Discussion

In this section, we analyze the results described above. We first perform some error analysis, which guides our plans for next steps.

6.1 Error Analysis

Based on the results of the initial model, there are a couple of areas to improve on.

First, it appears that the initial model did not leverage word identity in its classification. We hypothesized that a simple classifier may be able to detect profanity, since the presence of any swear word determines that a tweet is offensive by OLID guidelines. Task B of SemEval 2019 (Zampieri et al., 2019b) concerns the classification of an offensive tweet as a targeted insult (TIN) or untargeted offensive language (UNT), i.e. profanity, so we were able to use the data from Task B to test this hypothesis directly. In the development set, the initial model achieved a recall of about 0.41 for untargeted offensive tweets and about 0.43 for targeted offensive tweets. Ensuring that future iterations of our model can detect profanity will likely lead to improvements in performance.

Second, the performance of our model likely suffered from a lack of sequential information. The current model does weakly encode some sequential information, since tweet embeddings are concatenated word embeddings, but this may not be enough to capture syntactic information about the sentence. We hypthesize that a more principled method of incorporating the sequential nature of the text will improve our model's performance, particularly in detecting insults and threats.

6.2 Next Steps

There are several improvements that we are hoping to make to our current baseline system. For word embeddings, we currently assign the zero vector to tokens that are not found in the pre-trained GloVe embeddings. Although the GloVe embeddings we are using were specifically trained on a Twitter corpus, recent abbreviations and acronyms may not be found in the embeddings. Additionally, emojis and punctuation symbols that likely encode a lot of information are all being treated as out of vocabulary (OOV) tokens. In future experiments, we aim to convert common acronyms/abbreviations to their full form using a dictionary mapping and we also hope to leverage emoji-aware embeddings like Emoji2Vec (Eisner et al., 2016).

Additionally, we hypothesize that the performance of our system will significantly improve by replacing our linear classification model with a variable-length, sequence-based model like an RNN and LSTM.

7 Conclusion

This paper presents an initial model for offensive language detection in English. This model, which uses static word embeddings and a logistic regression classifier, attains a 0.598 Macro F1 score. We hope to improve upon this result by incorporating techniques that make use of the presence of profane words and embed sequential information into learned representations.

References

Ben Eisner, Tim Rocktäschel, Isabelle Augenstein, Matko Bošnjak, and Sebastian Riedel. 2016. emoji2vec: Learning emoji representations from their description. In *Proceedings of The Fourth International Workshop on Natural Language Processing for Social Media*, pages 48–54, Austin, TX, USA. Association for Computational Linguistics.

David Jurgens, Libby Hemphill, and Eshwar Chandrasekharan. 2019. A just and comprehensive strategy for using NLP to address online abuse. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3658–

3666, Florence, Italy. Association for Computational Linguistics.

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.

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.

Zeerak Waseem, Thomas Davidson, Dana Warmsley, and Ingmar Weber. 2017. Understanding abuse: A typology of abusive language detection subtasks. In *Proceedings of the First Workshop on Abusive Language Online*, pages 78–84, Vancouver, BC, Canada. Association for Computational Linguistics.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019a. Predicting the Type and Target of Offensive Posts in Social Media. In *Proceedings of NAACL*.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019b. 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.

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*.