Identifying Offensive Language in English Tweets using Machine Learning and Deep Learning Methods

Arif Shahriar
ashahri1@ualberta.ca

Alvina Awwal
awwal@ualberta.ca

Abstract

Social media has always been a platform where a vast majority of people display a certain kind of behaviour which they would not otherwise. This is usually in the negative sense which results in comments, posts, tweets, etc being openly abusive, racist or degrading against other users. Twitter is one such platform where this is a common scenario and an increasing amount of work is being done now to handle these cases with the help of NLP. We worked on two such sub-tasks from the SemEval 2020 Competition's Task 12, one of which was identifying Twitter posts as offensive or otherwise and the second was to classify if those offensive tweets were targeted or untargeted. Our project results of a few ML and DL methods show that the BERT model and the TF-IDF with LR model having balanced weights gave the optimal result of macro F1-scores of 0.91 and 0.65 for tasks A and B respectively.

1 Introduction

Social media and its possible anonymity and unrestricted access has always been a comforting blanket for a vast majority of people. Although this cloak has both advantages and disadvantages, on online platforms, it is usually impacted negatively in the form of open abusive, racial or hateful comments against other users. This gives rise to an important and increasingly necessary research area in the field of Natural Language Processing (NLP) where more and more work is now being done in order to help identify, classify and come up with solutions to reduce these occurrences online so that the severity of impact of these on lives and societies can be reduced.

Our paper is oriented around the SemEval (SE) 2020 Task 12 which was "Multilingual Offensive Language Identification in Social Media". We chose this particular task because since the onset of COVID and the decrease in outdoor activities, social media has become the general population's sanity. Twitter has always been one of the most popular methods of communication among the ordinary citizens, celebrities and even important people of the government. Statistics show that in 2020, 500 million tweets were sent per day with 152 million daily active users on the platform (Whitney, 2020). Our focus on this project is using English tweets to identify offensive language on Twitter data.

2 Related Work

Quite a lot of research has been previously done on similar tasks but we concentrated primarily on a few papers from this SE 2020 competition since those are recent and gave admirable results.

One such paper that we closely followed was by Parikh et al. (2020) and they implemented a total of 5 architectures, excluding ensemble for both tasks A and B. For task A, along with SVM and logistic regression (LR), we even executed a Multinomial Naive Bayes (NB) classifier which was dissimilar to this paper. Additionally, we also used a stratify parameter to keep the proportion of the classes similar in the train and dev distribution which they did not explicitly mention doing. Their second best accuracy for this task during the competition was achieved using the BERT model which they ran for 3 epochs without stop words, however, we slightly exceeded their performance with just 1 epoch and stop words which is computationally less expensive. For task B, we essentially have three major differences from this paper. Firstly, we also implemented SVM and average Ensemble on this task, along with other classifiers, which they did not. Secondly, we used less max_feature value than them (40K top features) as our dataset has fewer tweet instances than theirs. Finally, whereas Parikh

et al. (2020) and colleagues reported a macro F1-score of 0.59 for task B, we were able to produce a better result for the LR classifier. Hyperparameter tuning could have contributed to achieving better results as they did not explicitly mention about tuning these parameters.

Another paper that we followed was by Anwar et al. (2020). They used an ensemble of seven machine and deep learning (DL) models combined and achieved good results on the 5 languages that they based their project on, with the highest macro F1 score in the English language. Similar to this paper, we have used 10,000 most frequent words from corpus as features. They also provided both TF-IDF word level and character level features to the LR model, which we did not, and we ended up successfully achieving better performance than them in task A with TF-IDF word-level features only.

One SE paper by Ahn et al. (2020) gave a thorough description of their pre-processing steps which helped us immensely while implementing our initial SVM architecture. Our train dataset was processed and normalized similarly to theirs but we carried out a few extra steps than they did such as removing usernames using Python's regular expressions (RE) module as well as breaking down the texts into meaningful tokens. Another paper by Herath et al. (2020) achieved their best results through an ensemble model for task B which combined 5 models but we have surpassed their accuracy on the test set through the use of just 2 models for the same task. Their ensemble was created using a voting scheme whereas we used a weighted average ensemble.

Among all the teams that participated on this task 12, the state-of-the-art outcome on task A was 0.9204 with the use of an ensemble of ALBERT models of different sizes as well a different team's RoBERTa-large that was fine tuned on the SOLID dataset which was given to them (Rosenthal et al., 2021). For task B, an ensemble method got the best F1 score of 0.7462 (Zampieri et al., 2020).

3 Methodology

3.1 Task A

The main objective of our first task is to classify posts of the dataset into two categories- offensive (OFF) and not offensive (NOT) tweets. We have successfully scraped 14,687 tweets from the Twitter database using the Twitter IDs given in the SOLID dataset for our training set (2021). The test set, which was provided contains 3,887 posts, and has been used for evaluation purposes.

Initially, the training dataset had been divided into train (70%) and dev (30%) sets using sklearn train-test split function. The raw social media data needed to be normalized and thus, data preprocessing was required to facilitate feature extraction, remove inconsistency from the data, and make it more understandable for the NLP models. Firstly, since some of the offensive tweets had already been identified and removed by the Twitter authority, the non-retrievable rows were removed from the dataset. This resulted in a final dataset comprising 9,768 posts. Secondly, all the text data were converted to lowercase, and URL tags, emoji, usernames were removed using Python's RE module. In addition to that, each tweet text was broken into meaningful smaller tokens like words, phrases, symbols with a tokenization module from nltk library. This library also helped to remove punctuations and stop words from the data. Finally, POS tags of each word were identified to perform word lemmatization, which extracted the common lemma of different forms of verbs in the provided text data.

We incorporated a statistical method called TF-IDF with 10,000 top feature values for word vectorization. Unlike other SE papers, which had around 9 million data, our significantly smaller dataset could have resulted in overfitting. To avoid that problem and ensure generalization of the model, SVM algorithm with 5 fold cross-validation has been applied which resulted in 0.85 F1-score. The optimal model which performed better on the dev set was trained with the following parameter values (C=1.0, kernel='linear', degree=3) and this was later evaluated on the test set.

For this task, initially multinomial NB was implemented because it is often used in text classification tasks where features resemble word counts within categories to be classified. This provided us the baseline for comparisons with other classifiers. We then implemented another binary classification model LR on TF-IDF representations obtained from tweets using 10,000 max feature values. After experimenting with different parameter values, we observed better results for inverse regularization strength of C=1.0, 'balanced' class_weight, and 100 max_iter value. Class weight might have been an important factor for this dataset as it has

an uneven class distribution of offensive and not-offensive tweets.

Besides traditional machine learning (ML) approaches, a pre-trained bi-directional model (BERT) was fine-tuned for task A (Devlin et al., 2018). Multiple papers of the SE 2020 competition reported that BERT had outperformed other ML and DL models by achieving the best macro F1 score (Thenmozhi et al., 2020). After different preprocessing steps, generated word embeddings from the BERT layer were further processed by a neural network consisting of three dense layers. The dropout regularization technique was used to overcome the overfitting problem. After investigating different values, the dropout rate was set to 0.2 which randomly eliminated 20% of hidden units in the first two dense layers. The final dense layer implemented a softmax function that computes categorization probability to classify each tweet sample. Finally, we trained the model using Adam optimizer with 'categorical_crossentropy' loss, which works better when there are only two class labels. Due to time constraints and limited computation power, the model was trained for 1 epoch with a batch size of 32.

3.2 Task B

The second task is a binary classification problem which further categorizes offensive tweets identified in task A as targeted insult (TIN) or untargeted insult (UNT). Before model development, a threshold value was selected for AVG_CONF > 0.40 to label the dataset into two classes. We conducted experiments with different threshold values and observed best performance on this threshold. Tweet samples with average confidence greater than 0.40 were mapped to the UNT label, which resembled a positive class for Task B.

For task B, we were able to scrape 17,000 tweets using their tweet IDs from SE dataset. However, we had to remove some tweets from our dataset as they could not be retrieved using Twitter API. Our final dataset consisted of 9,908 tweets which were then divided into train (70%) and dev set (30%) for model development purposes. As explained earlier in task A, similar steps were performed for preprocessing.

TF-IDF scores were used to perform word vectorization, and a binary LR classifier was trained on the top 10,000 features. The reason behind using fewer features than some other papers was to avoid noise features which might lead to misclassification of tweet samples (2020). In order to adjust regularization strength to avoid overfitting, we tuned parameter C with a range of float values between 0.1 and 10. Best macro F1-score of 0.65022 was observed for optimal value of parameter C=4.3. We conducted an experiment with 'balanced' and uniform class weight values. As expected, the LR classifier was able to predict class labels more accurately for task A and B when the 'class_weight' parameter was balanced. As previously stated, balanced mode adjusted class weights to make them inversely proportional to class frequencies to account for the uneven class distribution.

For this task, we also applied an SVM classifier on generated TF-IDF representations as it yielded better results for task A. Then hyperparameter values were tuned using dev set samples. Like LR, we experimented with a range of C values and determined the best result for C value of 3.3. Here, the C value indicated a regularization parameter that implemented squared L2 penalty to prevent the model from putting too much weight on some specific features. Instead, it facilitated the model to spread out weights to address overfitting.

Another model that we experimented with was a weighted average ensemble. This method is used as the classifiers combined with varied weights give better and more stable results through weighted average ensemble. The weights were selected according to the validation accuracies, which defined the importance of each classifier for prediction, and were then fed to the ensemble model. After conducting experiments with different voting methods, "soft" yielded better results for our task.

4 Results and Discussion

We have used the macro F1 score to evaluate the models' performances since we believe it would give us the most accurate depiction of how our models are performing. Most of the SemEval 2020 papers have also used the macro F1 score to evaluate their performances.

For task A, we had 9,768 training data samples after pre-processing, of which 7,416 samples were from 'OFF' class and 2,352 samples from 'NOT' class. This uneven distribution was due to fewer offensive tweets being posted on social media than non-offensive ones. We did a few experiments using SVM, NB, and LR using TF-IDF representations of tweets and our achieved results are shown

in Table 1. However, the test set used for evaluation contained a 2:1 ratio of not offensive to offensive tweets. We have been able to improve our accuracy using TF-IDF word level features than Anwar and Baig (2020) after following their approach. Next, in order to experiment with a similar proportion of classes in both train and test data, we temporarily removed 2500 tweets from our train data. As a result, train data demonstrated the same 2:1 ratio of non-offensive to offensive tweets. The models yielded a F1-macro of 0.88295 and 0.80125, respectively for SVM and Naïve Bayes classifiers on the test set. Based on the empirical evidence, we think collecting more data might have increased the classification performances of our implemented models since the mentioned results were slightly lower, especially for SVM, than what was reported in (Parikh et al., 2020). One reason for NB yielding such poor results could have been that the classifier got confused between classes and failed to separate offensive and not-offensive tweets as they had some overlapping features.

However, we have also observed an improved macro F1-score for our proposed BERT model than what had been previously recorded by Parikh et al.. There could be two potential reasons for achieving such results. We tried with and without dropout regularization, and it can be stated from our findings that the dropout technique had contributed to this slight performance gain. Another reason could be that a comparatively smaller amount of tweets were used to train our model than their dataset.

Methods	**F1-macro**
TF-IDF with SVM	0.87839
TF-IDF with Naïve Bayes	0.68534
TF-IDF with LR	0.88198
BERT Classifier fine-tuned	0.90924

Table 1: Results of Task A on Test set

Task B also gave satisfactory results, and our LR model even outperformed the primary paper we were following in this competition which was by Parikh et al. (2020). Experiments were conducted with SVM, NB and LR using TF-IDF features of tweets and observed results from test set (1,422) are presented in Table 2. Hyperparameter tuning of our LR model resulted in improved results. Moreover, traditional ML models usually perform better than DL models used in SE papers in smaller datasets. So, our comparatively smaller dataset could have also contributed to the mentioned enhanced results. Our second best performer was the SVM classifier in this task. Finally, a weighted average ensemble model was also implemented and different combinations experimented to visualize if the results could boost further. Our best ensemble performance was achieved by the combination of LR and NB models. The result was not as high as we had hoped for but it still outperformed Herath et al. (2020) which was a robust ensemble of 5 models. The unsatisfactory performance could be because in a lot of cases, averaging linear classifiers together still gives a linear model which is not likely to be better than the models that we originally had. We think that increasing the number and types of individual classifiers to ensemble with would have greatly increased our score for this model along with increasing our overall number of training samples.

Methods	**F1-macro**
TF-IDF with SVM	0.62162
TF-IDF with Naïve Bayes	0.41947
TF-IDF with Logistic Regression	0.65021
Ensemble Approach	0.60406

Table 2: Results of Task B on Test set

5 Conclusion

From our research we have seen that ML models perform the best with these NLP tasks, especially when there is limited training data, classes have similar proportions between test and train data and the classifiers are tuned properly. For our first task, BERT outperformed all the other individual models after just 1 epoch run. In the future, we hope to run more epochs of BERT to try and achieve even better results. For our second task, LR gave admirable results and even beat some of the previous SE 2020 papers with its F1 score. However, we would like to extend our research further and implement more models for this task, including BERT, which would require more computational resources and time. We also believe that an weighted average ensemble of at least 4-5 different models would yield a more stable and superior result.

6 Repository URL

https://github.com/UOFA-INTRO-NLP-F21/f2021-proj-arif-shahriar-anik

References

Hwijeen Ahn, Jimin Sun, Chan Young Park, and Jungyun Seo. 2020. Nlpdove at semeval-2020 task 12: Improving offensive language detection with cross-lingual transfer. *arXiv preprint arXiv:2008.01354*.

Talha Anwar and Omer Baig. 2020. Tac at semeval-2020 task 12: Ensembling approach for multilingual offensive language identification in social media. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 2177–2182.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Mahen Herath, Thushari Atapattu, Hoang Anh Dung, Christoph Treude, and Katrina Falkner. 2020. Adelaidecyc at semeval-2020 task 12: Ensemble of classifiers for offensive language detection in social media. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1516–1523.

Apurva Parikh, Abhimanyu Singh Bisht, and Prasenjit Majumder. 2020. Irlab_daiict at semeval-2020 task 12: Machine learning and deep learning methods for offensive language identification. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 2006–2011.

Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Marcos Zampieri, and Preslav Nakov. 2021. Solid: A large-scale semi-supervised dataset for offensive language identification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 915–928.

D Thenmozhi, Nandhinee Pr, S Arunima, and Amlan Sengupta. 2020. Ssn_nlp at semeval 2020 task 12: Offense target identification in social media using traditional and deep machine learning approaches. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 2155–2160.

Margot Whitney. 2020. 40 twitter statistics marketers need to know in 2020.

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*.