Offensive Language Identification in Social Media: Building a Model for Detecting and Categorising Offensive Tweets

Mohammad Zaeem **Keyan Zhang**

1 Introduction

In today's digital age, social media platforms have provided opportunities for all kinds of harmful content. These negative interactions have an important impact on the well-being of users and the overall environment of online communities. Therefore, identifying and addressing offensive content is a crucial task for platform administrators. This project would like to make a contribution on social media platforms to maintain user safety.

2 Related Work

Many related studies have been carried out in the field of offensive tweet target recognition. For example, the paper published by Wiegand et al. [1] details an evaluation campaign aimed at evaluating offensive language recognition methods.The competition attracted submissions of various deep learning, rule-based, and ensemble based solutions. This work does provide some implications for offensive tweet target recognition, in this project we leverage an pre-trained BERT model and fine-tune it to achieve higher recognition accuracy.

3 Methodology

3.1 Data Collection

The data set we used in this project is the Offensive Language Identification Dataset (OLID) dataset. It was built specifically for this task, which was annotated using a hierarchical three-level annotation model introduced by Zampieri [2].

3.2 Data Pre-Processing

In this project, data preprocessing is a crucial step, which can effectively improve the accuracy and performance of the model. The pre-processing work in this project includes the following aspects:

First, all the non-English words are removed. This step can ensure that the model focuses on processing English text, thus improving the performance of the model on the English tweet target recognition task.

Second, all the words are converted to lowercase. The purpose of this step is to reduce the complexity of the text and unify the form of words. By converting all words to lowercase, we eliminate unnecessary differences caused by case, allowing the model to better focus on the substance of the text.

Next, stop words and punctuation marks are removed. Stop words, such as "of", "and", "in" and so on, do not carry actual meaning and appear more frequently in the text. Removing these words and punctuation would reduce noise in the text and allows the model to focus on more meaningful words.

Finally, words that start with "@" and "#" frequently appear in tweets. These words usually indicate user mentions and hashtags, which are not substantially helpful for the task of tweet target identification. For words containing "-", we keep the part after "-" and remove the part before.

To balance the training data, oversampling and undersampling methods were used for different sub-tasks. Samples are randomly selected from the majority class to reduce the number of samples when undersampling is applied. In oversampling, we randomly copy samples in the minority classes to increase the number of samples to match that of the majority class.

Through the above pre-processing work, the complexity of the text has been effectively reduced and the training data has been balance, as well as the noise in the content.

3.3 Defining Classification Tasks

3.3.1 Sub-task A: Offensive language Identification

In this sub-task, the goal is to distinguish between offensive and non-offensive posts. Offensive posts include insults, threats, and posts containing any form of non-targeted profanity in English. Each instance is assigned one of two labels:

Not Offensive(NOT): Tweets that do not contain any form of offensive content or profanity.

Offensive(OFF): Tweets that contain contains any non-acceptable language or targeted offense.

3.3.2 Sub-task B: Hate Speech Identification

In sub-task B, the goal is to predict the type of offensive language used. Only tweets that have been marked as offensive (OFF) in sub-task A will be included in sub-task B. The two categories of sub-task B are as follows:

Targeted Insult(TIN): Tweets that contain offensive content to an individual, group, or others.

Untargeted (UNT): Tweets containing non-targeted profanity and swearing.

3.3.3 Sub-task C: Offensive Target Identification

Sub-task C focuses on identifying the target of the offense. Only tweets that have been marked as TIN will be included in this third level of annotation. The three labels of sub-task C are as follows:

Individual(IND): The target of the offensive tweet is an individual. The target can be a name, or an unnamed participant in a conversation.

Group(GRP): These offensive tweets target a group of people who are considered as a group based on the same race, gender or sexual orientation, political affiliation, religion, or other common characteristics. Many insults and threats directed at a group of people are often understood as hate speech.

Other(OTH): The targets of these offensive posts do not fall into either of the first two categories, for example, an organisation, a situation, an event, or an issue.

3.4 Model Selection

In order to achieve accurate identification and classification of offensive tweets, we needed to choose an appropriate machine learning model that can handle the complexity and diversity of natural language data. After considering various options, we selected the Bidirectional Encoder Representations from Transformers (BERT) model for this task.

We chose BERT for this project because it has shown excellent performance in various NLP tasks and has become a standard for many NLP applications. BERT has also been used successfully for offensive language identification tasks in previous research, including the OffensEval 2019 shared task [1].

BERT's ability to capture contextual information makes it well-suited for offensive language identification, where the meaning of a sentence can be heavily influenced by its context. BERT's pre-training on a large corpus of text also allows it to recognise subtle linguistic patterns and relationships between words, which can help in identifying offensive language in tweets.

3.5 Model Architecture

Our approach to detecting offensive tweets involves using a pre-trained BERT model to generate embeddings for the tweets, followed by a simple linear classifier to predict the probability of a tweet being offensive or not.

The BERT model has been pre-trained on a large corpus of text data and can understand the context of the input text. We use the pre-trained BERT model as a feature extractor and fine-tune the linear classifier on our dataset. The BERT model outputs embeddings for each token in the input text, and we use the embedding of the '[CLS]' token, which is a special token used in BERT for classification tasks, as the input to the linear classifier.

Our linear classifier has a single hidden layer with ReLU activation and a dropout layer to prevent overfitting. The output of the classifier is passed through a sigmoid function to obtain a probability value between 0 and 1, which represents the likelihood of the tweet being offensive.

We use binary cross-entropy loss as our loss function to train the model. We use the AdamW optimizer with a learning rate of 5e-5 and a linear scheduler with warmup steps to optimize the model parameters.

3.6 Training Objectives

The goal of this project is to train a deep learning model to classify tweets based on three sub-tasks:

Offensive Language Identification: The model should be able to identify tweets that contain offensive language, as well as classify the degree of offensiveness.

Hate Speech Identification: The model should be able to identify tweets that contain hate speech, which is defined as any language that is used to demean or insult a particular group of people based on their race, gender, religion, etc.

Target Identification: The model should be able to identify the target of the offensive or hate speech in a tweet, as well as classify the degree of severity.

For each sub-task, we trained a BERT model using the labelled dataset. The models were fine-tuned using a transfer learning approach, where we first initialised the models with pre-trained BERT weights, and then fine-tuned the model on the specific task's labelled dataset.

Each of the models were trained with the architecture and configuration defined above, with specific adjustments made to the output layer to match the task at hand. The training objectives were set accordingly to achieve optimal performance for each sub-task.

3.7 Model Training

We trained three separate BERT models on the dataset for each task. We used a 10-fold cross-validation approach to train our models. This involved randomly dividing the dataset into 10 folds, training our models on 9 of the folds, and using the remaining fold for validation. We repeated this process 10 times, using a different fold for validation each time. This allowed us to train and evaluate our models on different subsets of the dataset and ensured that our models were robust to different variations in the data.

During the training process, we used the AdamW optimizer with a learning rate of 5e-5. We also used a linear learning rate scheduler with a warm-up period to gradually increase the learning rate over the course of the training. We set the number of warm-up steps to 500 and the total number of training steps to 10,000.

For each fold, we trained the model for a maximum of 5 epochs, monitoring the training and validation loss to ensure that the model was not overfitting. We used early stopping to prevent overfitting and saved the best performing model based on the validation loss. Once all 10 folds had been trained and evaluated, we selected the model with the highest average performance across all folds as our final model.

Overall, the 10-fold cross-validation approach allowed us to obtain reliable estimates of our models' performance and ensured that our models were robust to different variations in the data.

4 Evaluation

The experimental setup involved training the BERT model on the dataset for sub-task A using 10-fold cross-validation. The aim of this experiment was to evaluate the performance of the model in identifying offensive tweets in the dataset. We expected the precision and recall to be high for the "OFF" class, as we wanted the model to accurately identify tweets containing offensive content. Conversely, we expected a lower precision and recall for the "NOT" class since it contains tweets that do not contain offensive content.

Sub-Task A

	precision	recall	f1-score	support
OFF	0.92	0.73	0.81	620
NOT	0.54	0.84	0.66	240
accuracy			**0.76**	860
macro avg			**0.74**	860

Table 1: Classification Report for Sub-Task A

The results show that the model achieved high precision and recall for the "OFF" class, which suggests that it can effectively identify tweets containing offensive content. However, the precision and recall for the "NOT" class were lower, indicating that the model struggled to identify tweets that did not contain offensive content. The overall accuracy of 0.76 suggests that the model is moderately successful in distinguishing between offensive and non-offensive tweets.

Sub-Task B

	precision	recall	f1-score	support
UNT	0.28	0.70	0.40	27
TIN	0.95	0.77	0.85	213
accuracy			**0.77**	240
macro avg			**0.63**	213

Table 2: Classification Report for Sub-Task B

For Sub-Task B, we trained a BERT model to classify tweets as either Targeted Insult or Non-Targeted Insult. Table 2 presents the classification report for the model evaluation. The model achieved an overall accuracy of 0.77 and a macro-average F1-score of 0.63. The precision score for the TIN class is 0.95, indicating that the model is good at correctly identifying targeted insults, while the precision score for the UNT class is only 0.28, indicating that the model is not very good at identifying non-targeted insults. The recall score for TIN is 0.77, which suggests that the model is not able to correctly identify all targeted insults in the dataset. Conversely, the recall score for UNT is 0.70, indicating that the model is good at correctly identifying non-targeted insults.

We can hypothesize that the imbalanced class distribution between TIN and UNT labels in the dataset might have impacted the model's performance. The dataset has significantly more TIN examples than UNT examples, which could have caused the model to be biased toward the TIN class, leading to lower performance on the UNT class. We can also observe that the model's performance on identifying targeted insults is better than that on non-targeted insults, which could indicate that identifying non-targeted insults from regular language is a more challenging task. Overall, we can say that the model's performance on Sub-Task B is good but has room for improvement, particularly in correctly identifying non-targeted insults.

Sub-Task C

	precision	recall	f1-score	support
OTH	0.28	0.14	0.19	35
GRP	0.59	0.81	0.68	78
IND	0.84	0.52	0.64	100
accuracy			**0.75**	213
macro avg			**0.51**	213

Table 3: Classification Report for Sub-Task C

For Sub-Task C, as shown in Table 3, our model achieved an overall accuracy of 0.75. The precision and recall for each category varied, with the highest precision score of 0.84 achieved for IND, and the highest recall score of 0.81 achieved for GRP. However, the precision and recall scores for OTH were both relatively low, at 0.28 and 0.14, respectively.

These results indicate that our model was relatively successful in identifying tweets belonging to the IND and GRP categories, but struggled to accurately classify tweets in the OTH category. One possible reason for this is that the OTH category is more ambiguous and can encompass a wider variety of tweets that are not clearly identifiable as belonging to the other two categories.

4.1 Comparison to Related Work

Compared with the research in related work, our model performs as follows on each sub-task: On sub-task A, the F1 score is 0.74, which means that our model has high accuracy and recall in distinguishing offensive and non-offensive posts compared with related work. On sub-task B, the F1 score is 0.63, which is slightly worse than related work in predicting attack types but still shows a certain effect. However, on sub-task C, the F1-score is 0.51, which is about the average compared to related work. A comprehensive comparison shows that on sub-tasks A and B, our model shows good performance, but the performance on sub-task C would have more room to work. This may be due to the higher complexity of sub-task C, which involves identifying more complex attack targets. In general, our model achieves good results on the offensive tweet target recognition task, but it still needs to be compared and improved with related work in some aspects to improve the overall performance.

5 Conclusion

In this project, we trained three separate BERT models, each one for a specific sub-task. The models were trained on the OLID dataset that was built specifically for this task.

Our experimental results show that the models achieved reasonable performance on the task. Sub-task A achieved an F1-score of 0.74, Sub-task B achieved an F1-score of 0.63, and Sub-task C achieved an F1-score of 0.51. These results demonstrate that our approach can effectively identify and categorise offensive tweets.

Based on the results of our experiments, we can conclude that our approach can be used as a viable solution for offensive tweet detection on Twitter. However, our work is not without limitations. For instance, our dataset is relatively small, and it did not fully capture the diversity of offensive tweets on social media. Additionally, our models were trained on a specific type of social media, Twitter, and it is unclear how well they would perform on other platforms.

Some reasonable follow-on work based on what we have done could include expanding the dataset to include more examples of hate speech, exploring the use of different pre-trained language models, and evaluating the models on different social media platforms. Additionally, it may be useful to explore the use of additional features such as user information and metadata in the models.

References

[1] Marcos Zampieri et al. "Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval)". In: *arXiv preprint arXiv:1903.08983* (2019).

[2] Marcos Zampieri et al. "Predicting the type and target of offensive posts in social media". In: *arXiv preprint arXiv:1902.09666* (2019).