Deliverable 3: SemEval-2020 Task 12

Sneha Kumaran, Xiaoqing Liu, Phani Preetam Kommineni, Shristi Kota
AIT 526 - NLP - Prof. Marcos Zampieri

Abstract

In this document, we will present the overall findings of SemEval-2020 Task 12 on Multilingual Offensive Language Identification in Social Media (OffensEval-2020). In this project, three subtasks will directly correlate to the 2019 OLID dataset and 2020 SOLID dataset. We used both test and training sets to get numerous results to learn the capabilities of our models. Based on the three given subtasks, the project will be enhanced and deeply comprehended to explore the results and highlight the significance of detecting the potential identifications of offensive languages in social media.

1 Introduction

Social media is a platform where people can express their opinions and have the freedom of speech. At the same time, many users tend to use various offensive languages on these platforms. Facebook, Twitter, Instagram, and Reddit have millions of users that use profanity. Offensive languages have been causing lots of concerns and challenges for many online user interactions. Many abusive languages have resulted in promoting this behavior within the users' communication style. The way people start to engage trends out this behavior within others. The behavioral trends can give out lots of insights about how users are implying the use of offensive, violent, hate speech in many online platforms. Figuring out how big of an issue by collecting and detecting profanity language in social media, online communities, and various other online interactive applications can pull out the major insight into where it is necessary to put measurements on spreading hate speeches.

In this project, as a team, we have selected a system that explores each task of the OffensEval2020 - Task 12 (Zampieri et al., 2020). The system is developed to detect the offensive/abusive language in each tweet. It will be using an OLID/SOLID (Zampieri et al., 2019; Rosenthal et al., 2020) dataset from 2019 and 2020 with a collection of 14,200 annotated English tweets of comments from users. There are three levels of annotated models that we aim to detect:

- Sub-Task A: Offensive Language Detection
- Sub-Task B: Categorization of Offensive Language
- Sub-Task C: Offensive Language Target Identification

Based on these given subtasks, it will be trained and tested in a system to detect the number of offensive languages and categorize the overall profanity. The system that is used for this project varies upon each subtask. Each of the subtasks has specific algorithms used like SVM linear and other techniques that would help detect the needed identification. Our team created a model using the support vector machine algorithm to serve as a powerful tool to classify and identify offensive language. SVM can use NLP-based tasks and utilize to tackle high-dimensional features and obtain text classification. Textual data can be converted to numerical vectors using packages like TF-IDF. It will find features by seeing the patterns and trends and has a strong adaptability plus optimization in handling linguistic patterns.

This paper will explore all the applications we as a team have done to produce models based on each task. Each task has a specialty of using various algorithms efficiently to obtain a proper detection and identification of the necessary offensive language.

2 Taxonomy of Each SubTasks

SubTask A: Offensive Language Detection

- (NOT) - Not Offensive - this particular post does not have or contain any abusive/offensive/hate speech

- (OFF) - Offensive - has profanity in the post

SubTask B: Categorization of Offensive Language

- (TIN) - Targeted insult and threats - this particular post contains an insult/threat to the user/group/others

- (UNT) - Untargeted - the post has non-targeted profanity

SubTask C: Offense target identification

- (IND) - Individual - the offensive post contains a target to a specific individual like a famous person, named individual, or unnamed person

- (GRP) - Group - the offensive post contains a target to a group of people such as ethnicity, sexuality, political viewpoint, religious perspective, or something else

- (OTH) - Other - the offensive post contains a target to other categories other than the ones stated above such as an organization, event, or situation

3 Related Works

In this section, we examine and discuss several papers on the state of the art in offensive language detection, categorization, and target identification. The goal is to explore the present landscape, advancements, challenges, and future directions within this field. Through a thorough analysis of existing literature and methodologies, our aim is to gain a nuanced understanding of the complexities associated with identifying and addressing offensive language. Additionally, we strive to identify feasible methods that can be applied to effectively address the objectives of our project.

Sharma et al. (Sharma et al., 2021) created an innovative framework merging deep learning with Single Valued Neutrosophic Sets (SVNS) for sentiment analysis. They tested this approach on the Offensive Language Identification Dataset (OLID), incorporating advanced neural network models such as Bi-directional Long Short-Term Memory (BiLSTM), Bidirectional Encoder Representations from Transformers (BERT), A Lite BERT (ALBERT), A Robustly Optimised BERT Approach (RoBERTa), and MPNet. The SVNS model demonstrated performance comparable to top neural network models, offering an effective method for sentiment quantification.

Li and Qian (Li and Qian, 2016) present a Recurrent Neural Network (RNN) language model incorporating Long Short-Term Memory (LSTM) for the multi-classification of emotional attributes in text. By training distinct emotion models, it becomes feasible to identify the emotional category to which a specific sentence belongs. Notably, this method exhibits enhanced accuracy and recall rates compared to conventional RNN models.

Zampieri et al. (Zampieri et al., 2023) introduced an innovative multi-task learning (MTL) framework designed to forecast offensiveness not only at the post level but also at the token level in the English language. Additionally, this architecture extends its coverage to address subjective tasks like humor, engaging content, and identifying gender bias in multilingual contexts. This advancement not only enhances the performance of offensive language detection systems but also boosts their interpretability—a vital aspect in assisting human moderators.

Offensive language detection in NLP can be approached using various methods, ranging from traditional machine learning techniques to more advanced deep learning methods. The choice of method often depends on the available data, computational resources, and the specific requirements of the application. Combining multiple methods is often more robust in real-world applications

4 Data and Methods

4.1 Dataset

In this project, multiple datasets are utilized. The primary dataset, *olid-training-v1.0.tsv*, serves as the foundation for constructing models for each task. Complementing this, specific test sets are available for each task, facilitating the evaluation of model performance. Below are the details of these datasets:

Dataset#1 olid-training-v1.0.tsv This dataset, presented as Table 1, encompasses 13,240 English tweets that have been annotated using a hierarchical three-level annotation schema considering both the target and the type of offensive content in its annotations. These annotations serve as the classes for our subtasks A, B, and C, providing guidance for constructing models for prediction.

id	tweet	subtask_a	subtask_b	subtask_c
86426	@USER She should ask a few...	OFF	UNT	NULL
90194	@USER @USER Go home you...	OFF	TIN	IND
16820	Amazon is investigating Chinese...	NOT	NULL	NULL

Table 1: Olid-Training-v1.0 Dataset

Datasets#2 Test sets from OLID The test set for subtask A is composed of two files: *testset-levela.tsv* and *labels-levela.csv*. This combination includes 860 tweets labeled as either 'OFF' or 'NOT' for level A. Regarding subtask B, the test set is formed by the pairing of *testset-levelb.tsv* and *labels-levelb.csv*, containing 240 tweets labeled as either 'TIN' or 'UNT' for level B. As for subtask C, the test set comprises two files, namely *testset-levelc.tsv* and *labels-levelc.csv*, including 213 tweets labeled as either 'OTH', 'GRP' or 'IND' for level C (refer to Table 2 and 3 for details).

id	tweet
15923	#WhoIsQ #WheresTheServer #DumpNike...
60133	#NoPasaran: Unity demo to oppose...
83681	. . . What the fuck did he do this time?

Table 2: Test-LevelC.tsv table

id	label
15923	OTH
60133	GRP
83681	IND
65507	OTH
34263	IND

Table 3: Labels-LevelC.csv Table

Datasets#3 Test sets from SOLID We also applied our model to the test sets from The Semi-Supervised Offensive Language Identification Dataset (SOLID).

The test set denoted as *test_a_tweets.tsv* combined with *test_a_labels.csv* comprises 3,887 tweets labeled for level A. Additionally, an extended test set for level A is provided through *test_a_tweets_all.tsv*, along with separate label files, namely *test_a_labels_easy.csv* and *test_a_labels_hard.csv*. This extended set encompasses 5,995 tweets.

The test set named *test_b_tweets.tsv*, accompanied by *test_b_labels.csv*, consists of 1,422 tweets categorized for level B. Furthermore, an expanded test set for level B, denoted as *test_b_tweets_all.tsv*, along with separate label files *test_b_labels_easy.csv* and *test_b_labels_hard.csv*, includes a total of 3,003 tweets.

The test set labeled *test_c_tweets.tsv*, in conjunction with *test_c_labels.csv*, comprises 850 tweets annotated for level C. In addition, an expanded test set for level C, identified as *test_c_tweets_all.tsv*, along with separate label files *test_c_labels_easy.csv* and *test_c_labels_hard.csv*, encompasses a total of 1,547 tweets.

4.2 Data Preprocessing

In our Python implementation, we incorporated the NLTK library for efficient data preprocessing. The initial step involved the removal of punctuations, hashtags, emojis, and other non-essential elements from the text. Subsequently, we tokenized the tweets, organizing them into lists. Further refinement included the elimination of stopwords and numerical values from these tokenized lists. The final phase of our preprocessing pipeline encompassed lemmatization to ensure a more standardized and linguistically meaningful representation of the data.

Subsequently, we conducted an exploration of the dataset based on classes and observed an imbalance in the labels, as illustrated in Figure 1. To mitigate bias, we chose to split the data into an 80% training set and a 20% testing set, and subsequently applied oversampling to the training set. This approach resulted in a balanced representation, as depicted in Figure 2.

It is important to highlight that these described preprocessing steps are not universally applied in all tasks. For each subtask, different methods are employed to customize the input dataset, ensuring its compatibility with the associated models.

4.3 Bag-of-Words model

Commencing our tasks, we constructed a bag-of-words model using a random forest. This model portrays a document as a non-sequential collection of words, overlooking grammar and word sequence

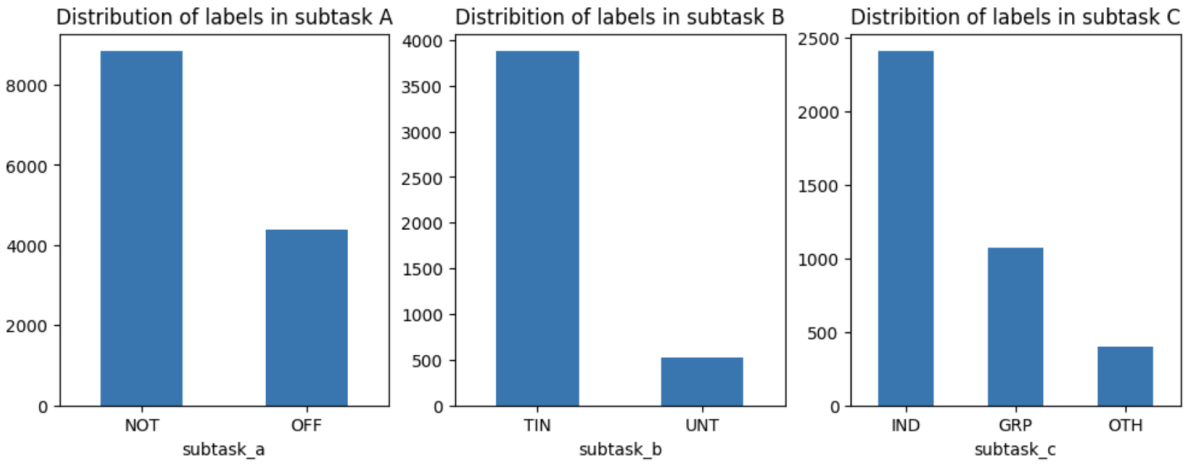

Figure 1: Distribution of Labels before Over Sampling

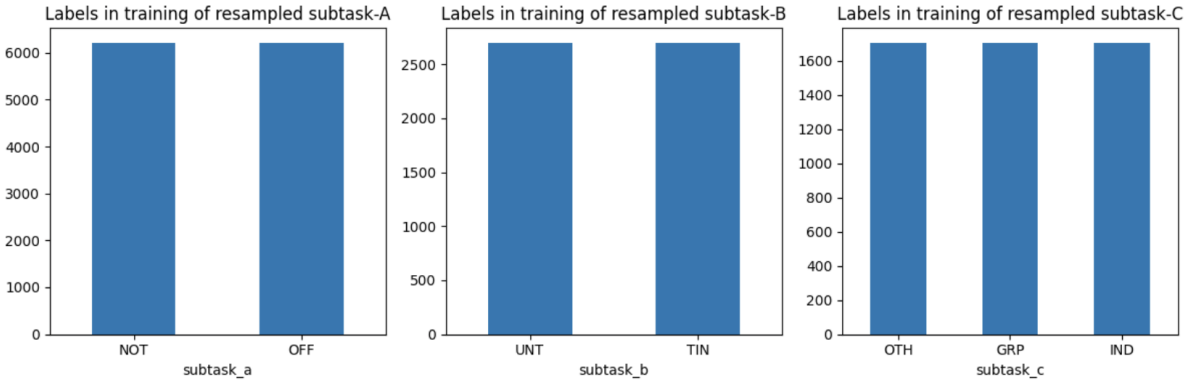

Figure 2: Distribution of Labels in Training Sets after Over Sampling

while preserving word frequency information. The primary objective behind building this model is to establish a straightforward and easily interpretable baseline for comparison with more advanced models.

Table 4, 5, and 6 illustrate the three evaluation metrics used to assess the performance of this model on the OLID training data for subtasks A, B, and C. We used them to assess the performance of other models across our three subtasks.

	Precision	Recall	F1-Score
NOT	0.78	0.91	0.84
OFF	0.73	0.49	0.59
Accuracy			0.77
Micro avg	0.75	0.7	0.71
Weighted avg	0.76	0.77	0.75

Table 4: Confusion Metrics of BoW Model for Subtask A

4.4 Sub-Task A

Subtask A is to detect offensive language in tweets. For this purpose, we employ Long Short-Term Memory (LSTM) networks with the training data after oversampling to construct the model. Figure 3 displays some basic information about the model.

The model architecture involves an input layer that processes sequences of 42 integers, passing them through an embedding layer to convert them

	Precision	Recall	F1-Score
TIN	0.90	0.98	0.94
UNT	0.29	0.08	0.12
Accuracy			0.88
Micro avg	0.59	0.53	0.53
Weighted avg	0.83	0.88	0.85

Table 5: Confusion Metrics of BoW Model for Subtask B

	Precision	Recall	F1-Score
GRP	0.56	0.40	0.47
IND	0.69	0.86	0.77
OTH	0.11	0.05	0.07
Accuracy			0.64
Micro avg	0.46	0.44	0.43
Weighted avg	0.59	0.64	0.61

Table 6: Confusion Metrics of BoW Model for Subtask C

```
Layer (type)                 Output Shape              Param #
=================================================================
input_59 (InputLayer)        [(None, 42)]              0

embedding_58 (Embedding)     (None, 42, 15)            220245

lstm_58 (LSTM)               (None, 42, 20)            2880

global_max_pooling1d_58 (G   (None, 20)                0
lobalMaxPooling1D)

dense_116 (Dense)            (None, 16)                336

dense_117 (Dense)            (None, 1)                 17

=================================================================
Total params: 223478 (872.96 KB)
Trainable params: 223478 (872.96 KB)
Non-trainable params: 0 (0.00 Byte)
```

Figure 3: Information about LSTM Model

into dense vectors of size 15. Subsequently, a LSTM layer with 20 units captures sequential dependencies in the data. Global max pooling condenses information from the sequences, resulting in a tensor of size 20. The model includes a dense layer with 16 units and a final dense output layer with a single unit for binary classification (0 or 1). The entire model comprises 223,478 trainable parameters, making it well-suited for tasks with sequential data. It demonstrates a thoughtful combination of layers for effective classification of offensive language in tweets.

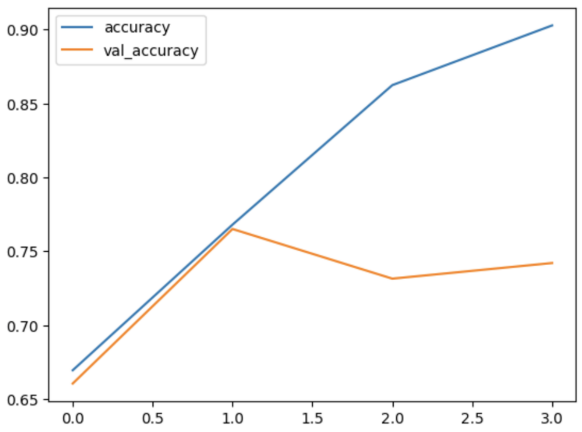

Figure 4: Training accuracy versus Validation accuracy

The trends of training and validation accuracy are illustrated in Figure 4. The validation accuracy is approximately 0.73. A thorough evaluation metrics presented in table 7 indicates that the LSTM model demonstrates enhanced recall and F1-score for the 'OFF' class in comparison to the previous bag-of-words model. Consequently, the LSTM model is recommended for application to the OLID and SOLID test sets for subtask A.

	Precision	Recall	F1-Score
NOT	0.80	0.79	0.79
OFF	0.60	0.61	0.60
Accuracy			0.73
Micro avg	0.70	0.70	0.70
Weighted avg	0.73	0.73	0.73

Table 7: Confusion Metrics for LSTM in Subtask A

4.5 Sub-Task B

Sub-task B involves the classification of offensive language into Targeted Insult and Threat (TIN) and Un-targeted Insult and Threat (UNT) categories. A BERT (Bidirectional Encoder Representations from Transformers) model is employed for this classification task. The training phase utilizes the Olid-training-v1.0.tsv data set, consisting of 13,240 English tweets, each with five features. The model's performance is evaluated using the test-levelb.tsv dataset, comprising annotated tweets from both 2019 and 2020. Gold labels and IDs for annotations at level B are provided in the labels-levelb.csv data set refer Table 8.

id	label
15923	TIN
60133	TIN
12588	UNT
34263	TIN

Table 8: Labels-Levelb.csv Table

Firstly, the "labels-levelb.csv" dataset is loaded to extract gold labels and IDs, which are essential for model evaluation. Subsequently, the "testset-levelb.tsv" dataset and the "olid-training-v1.0.tsv" dataset are loaded to prepare the test set and training set, respectively. The "labels-levelb.csv" data undergoes initial processing to separate 'id' and 'label' information. Pre-processing steps include the removal of hashtags, mentions, emojis, and numerical values, as well as tokenization, removal of non-alphanumeric tokens, elimination of stop words and lemmatization are applied to both testset-levelb data set and OLID 2020 data set. This preprocessing ensures that the text data is appropriately formatted and ready for subsequent model training and evaluation.

In the model implementation, we employ a BERT (Bidirectional Encoder Representations from Transformers) model for sequence classification, initialized from the 'bert-base-cased' pre-trained model. The model is configured for binary classification

with two output labels, representing un-targeted insult and threat (UNT) and targeted insult and threat (TIN). Utilizing the GPU if available, the model is trained on the olid-training-v1.0.tsv dataset, with labels transformed to numeric form. We tokenize and encode both the training and test datasets using the BERT tokenizer, incorporating attention masks and padding for consistent input lengths. PyTorch dataloaders are constructed for efficient batching during training and testing. After training, the model undergoes evaluation on the test set, yielding predictions that are compared to the gold labels from labels-levelb.csv. Performance metrics, including accuracy and a classification report, are computed, demonstrating the model's effectiveness in distinguishing between UNT and TIN tweets. Additionally, a confusion matrix and a precision-recall curve, including the area under the curve (AUC), provide a comprehensive assessment of the model's classification capabilities. These evaluation metrics collectively contribute to a robust analysis of the BERT model's performance in the context of offensive language classification.

4.6 Sub-Task C

The table 3 the labels-levelc.csv dataset. It contains the gold labels and IDs of the instances test set layer c. The table 1 is the olid-training-v1.0.tsv data set used for all the models. It contains contains 14,100 annotated tweets. The table 2 is the test-levelc.tsv data set of 2019 and 2020. The SOLID dataset from 2020 SemEval is similar to the OLID, but it has more variations of the tweets. It has about 9,000,000 annotated tweets. It contains the test set instances of level c.

In this model that was created for subtask c has been developed to identify the purpose of this task. The overall goal of the system is to find/predict the necessary categories. In this case, it is other (OTH), individual (IND), and Group (GRP). First, the data loading was done. The system begins by loading in the three different data set: labels data, test set data, and the olid training data set. The imported packages are pandas, sklearn, demoji, sklearn.feature_extraction.text, string, re, sklearn.metrics,and sklearn.svm. The system is preprocessed by removing the mentions, hashtags, emojis and also punctuation's. The raw twitter data is prepared for the model, so it will replace all the mentions of the username into a generic token and anonymize the data. Then the hashtags are removed by providing the least semantic value. Punctuation is stripped out. The emojis are removed as well. The stop word removal is in English. The preprocessed tweets are converted and vectorized into a TF-IDF representation by an array and generate feature vectors. We decided not to tokenize or use lemmatization for this model due structure of the algorithm. We are converting the text into numerical representations and adding it into vectors using the Term Frequency-Inverse Document Frequency. Later, it will take the "id" column from each dataframe and find the strings and merge the dataframe based on the id from both the datasets. This means each of the dataframe have these common ids, that is how it merges. The training data is used and it is split into 80% of the training set and 20% as the validation set. This is fit into the SVM linear model and will eventually evaluate the performance. It will produce the classification report that contains the accuracy, precision, f1score, recall, and support. As a team, we will use the 2019 data sets as well as the SOLID 2020 test data set to change the parameters and enhance the system to obtain more results. There can be a good amount of differentiation and knowing how the model works. We have created word clouds based on what are the popular words associated within the categories of GRP, IND, and OTH. This includes for the SOLID dataset.

5 Results

Below are results pertain to the models evaluated on the test sets that were both created and analyzed. In the subsequent discussion, we will present insights into the model's performance, offering a comprehensive understanding of its effectiveness.

5.1 Sub-Task A Results

In this subsection, the LSTM model is implemented for subtask A on the test set from OLID, as well as the test and extended test sets from SOLID. The outcomes of these applications are discussed in detail within this section.

In Table 9, a summary of the LSTM model's performance is provided for the 'OFF' class in subtask A across four datasets. In the OLID Test Set A, the model exhibits moderate performance, achieving an F1-Score of 59%. Conversely, in the SOLID Test Set A, the model performs robustly, showcasing an impressive F1-Score of 84%. Notably, it

Test Set	Precision	Recall	F1-Score
OLID Test Set a	0.62	0.55	0.59
SOLID Test Set a	0.77	0.93	0.84
SOLID Extended Easy	0.92	0.99	0.95
SOLID Extended Hard	0.83	0.91	0.87

Table 9: Evaluation Metrics on Different Test Sets by LSTM for Subtask A

excels in the SOLID Extended Easy subset, demonstrating a high precision of 92%, recall of 99%, and an outstanding F1-Score of 95%. Regarding the SOLID Extended Hard subset, the model maintains good performance with a precision of 83%, recall of 91%, and an F1-Score of 87%.

The model showcases its effectiveness in identifying offensive language across datasets, with notably strong performance in the SOLID Extended Easy subset. The diverse precision and recall levels underscore the model's adaptability to distinct dataset characteristics and complexities.

5.2 Sub-Task B Results

In Sub-Task B we employed a BERT-based Model, attaining an impressive overall accuracy of 88.75% on the test data. The classification report highlights the model's performance with high precision (1.00) and recall (0.00) for the 'Un-targeted Insult' class ('UNT'). Conversely, for the 'Targeted Insult' class ('TIN'), precision is 0.89, and recall is 1.00. Notably, the F1-score is considerably lower for 'Un-targeted Insult' (0.00) compared to 'Targeted Insult' (0.94). Both a macro-average F1-score of 0.47 and a weighted-average F1-score of 0.83 were computed, indicating potential challenges in handling class imbalance. It is worth mentioning that the evaluation also involved the utilization of a confusion matrix, along with precision and recall curves. Further exploration is suggested to enhance the model's capacity in identifying 'Un-targeted Insult' instances, recognizing the interlinked nature of precision and recall in influencing overall model performance.

We Have also evaluated confusion matrix for OLID data shown in Figure 5. The confusion Matrix offers a comprehensive perspective on the BERT-based model's performance in offense evaluation, specifically distinguishing between 'Untargeted Insult' (UNT) and 'Targeted Insult' (TIN) classes. The matrix indicates that the model effectively identified 178 instances of 'Targeted Insult' (True Positives), underscoring its proficiency in recognizing Targeted content. However, it also misclassified 20 instances of 'Un-targeted Insult' as 'Targeted Insult' (False Positives), revealing occasional mislabeling of Un-targeted Insults as Targeted insults. Additionally, the model accurately identified 7 instances of 'Un-targeted Insult' (True Negatives), showcasing its ability to recognize Un-Targeted insults. Nevertheless, there were 35 instances of 'Targeted Insult' misclassified as 'Untargeted Insult' (False Negatives), suggesting challenges in accurately capturing Targeted insults.

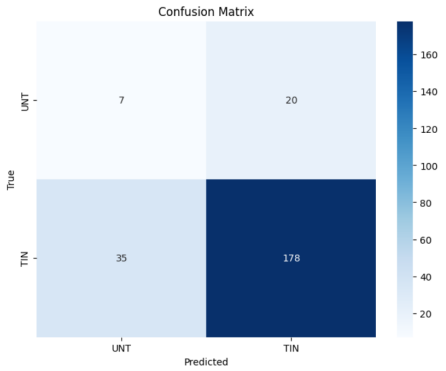

Figure 5: Confusion Matrix for OLID Dataset for Subtask B

In summary, while the model excels in identifying targeted insults, refinement is needed to mitigate false positives and enhance its capacity to recognize un-targeted insults. The confusion matrix offers valuable insights for targeted model improvement, contributing to the ongoing optimization of the offense evaluation system.

In figure 6, the precision and recall curve was Implemented for OLID data in sub-task B, it demonstrates exceptional performance, with precision and recall both consistently reaching 1.00. This signifies that the model exhibits perfect precision, accurately identifying all instances predicted as positive, and perfect recall, capturing all actual positive instances. The curve's shape, spanning from precision 0.8 to 1.0 and recall 0.8 to 1.0, suggests a high degree of confidence in positive predictions, maintaining precision even as recall increases. The area

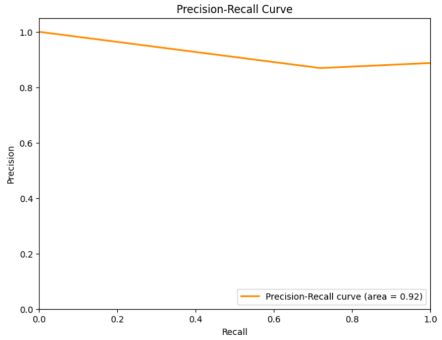

Figure 6: Precision-Recall Curve for OLID data

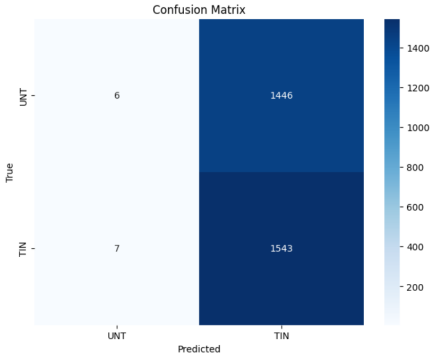

Figure 7: Confusion Matrix for SOLID 2020 Dataset for Subtask B

under the precision and recall curve, quantified at 0.92, further underscores the model's robustness in balancing precision and recall. This high area indicates a strong ability to distinguish between the 'Un-targeted Insult' and 'Targeted Insult' classes, showcasing the model's effectiveness and reliability in offense evaluation.

Using the SOLID 2020 dataset and a BERT model yielded an overall accuracy of 66.07% on the test data. The classification report provides detailed insights into the model's performance across the 'Un-targeted Insult' (UNT) and 'Targeted Insult' (TIN) classes. Notably, the model achieved perfect precision (1.00) for 'Un-targeted Insult,' indicating that when it predicted instances as un-targeted insults, it was accurate. However, the recall for 'Un-targeted Insult' is considerably low (0.00), suggesting the model struggled to identify instances of un-targeted insults, resulting in an F1-score of 0.00 for this class. Conversely, for 'Targeted Insult,' the model achieved a precision of 0.66, indicating that when it predicted instances as targeted insults, it was correct approximately two-thirds of the time. The recall for 'Targeted Insult' is 1.00, indicating that the model identified all actual instances of targeted insults, leading to a higher F1-score of 0.80 for this class. The macro and weighted averages of precision, recall, and F1-score highlight the imbalanced nature of the dataset, with an emphasis on the challenges associated with un-targeted insults.

As shown in figure 7, the confusion matrix presents a detailed breakdown of the BERT model's performance using SOLID 2020 Dataset, specifically distinguishing between 'Untargeted Insult' (UNT) and 'Targeted Insult' (TIN) classes. The matrix indicates that the model correctly identified 1542 instances of 'Targeted Insult' (True Positives), demonstrating its proficiency in recognizing instances of directed offensive language. However, the model misclassified 1446 instances of 'Untargeted Insult' as 'Targeted Insult' (False Positives), suggesting a challenge in accurately distinguishing between targeted and untargeted insults. On the 'Untargeted Insult' side, the model correctly identified only 6 instances (True Negatives), while 7 instances of 'Targeted Insult' were misclassified as 'Untargeted Insult' (False Negatives). This pattern reflects the model's tendency to be more conservative in predicting untargeted insults, leading to a higher number of false positives. The overall performance is reflective of the imbalanced nature of the dataset, with targeted insults being more effectively identified than untargeted insults.

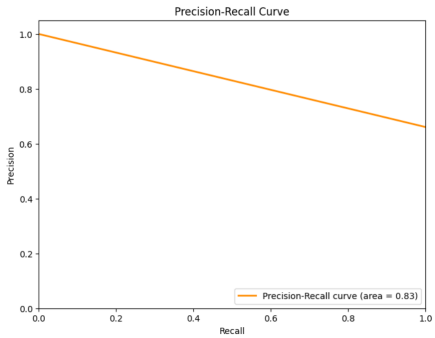

Figure 8: Precision and Recall curve for SOLID 2020 Dataset

In figure 8, the precision and recall curve for the offense evaluation task on the SOLID 2020 dataset depicts a scenario where both precision and recall attain perfect values of 1.00. This signifies that the model achieves flawless precision, correctly identifying all instances predicted as positive (Targeted Insult), and perfect recall, capturing all actual positive instances. The slanting line from precision to recall suggests a deterministic prediction strategy, where precision remains consistently high as recall

increases. The precision and recall curve's area, quantified at 0.83, underscores the model's effectiveness in balancing precision and recall across the dataset. This high area under the curve reflects the model's robustness in distinguishing between 'Untargeted Insult' and 'Targeted Insult' classes, indicating a strong overall performance in offense evaluation on the SOLID 2020 dataset.

5.3 Sub-Task C Results

After the model implementation using the Support Vector Machine (SVM) algorithm, we ended up getting various results to show the model's performance. Word clouds were designed to show which words from the annotated tweets are most frequent. With the TF-IDF vectorization implementation and adding it into the SVM linear model by splitting the training and testing sets proceeded to give a prediction of the first 10 values. It predicted ['GRP' 'IND' 'GRP' 'IND' 'IND' 'GRP' 'IND' 'IND' 'GRP' 'IND'], these are the prediction for OLID dataset. The accuracy of this model was 69.974%. A confusion matrix was printed out to provide the relationship of the predicted values and the true values.

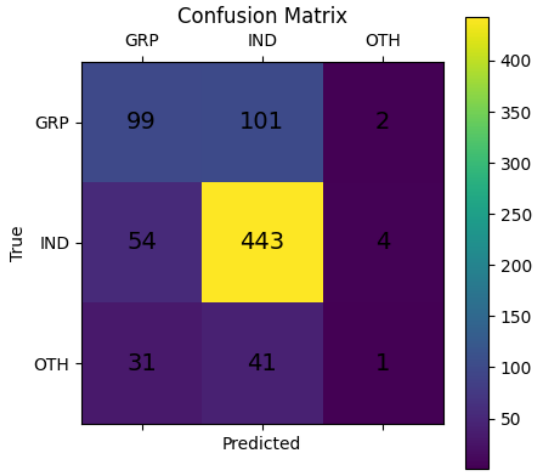

Figure 9: Confusion Matrix for Subtask C using OLID data

As shown in Figure 9, for GRP true and GRP predicted values, we received a value of 99, GRP true and IND predicted were 101, GRP true and OTH predicted are 2 values. In the IND true and GRP predicted, it is 54 values, IND true and IND predicted is 443. IND true and OTH predicted values were 4. Finally, OTH true and GRP predicted is 31. OTH true and IND predicted is 41. OTH true and OTH predicted 1.

We also printed out the sampling predictions from the SVM model using the OLID data. Table 10 displays some of the sampling prediction results, it shows how well the sampling prediction has been portrayed to show the accuracy of the first few samples.

True Label	Predicted Label
IND	GRP
IND	IND
IND	GRP
GRP	IND
IND	IND
IND	GRP
IND	IND

Table 10: Sampling Predictions OLID Data for Subtask C

Based on these sampling predictions, the true values tend to be the same as the predicted, but at the same time, there are some predictions with different true and predicted values.

Here is the classification report of the OLID data model using SVM algorithm. Each identification has been stated by displaying the precision, recall, f1-score, and the support. The precision for GRP, IND, and OTH are 54%, 76%, and 14%. The recall for GRP, IND, and OTH are 49%, 88%, and 1%. The f1-score for GRP, IND, OTH are 51%, 82%, and 3%. The accuracy is good for a NLP model, because it is 69-70%.

Three word clouds were created to portray the frequent words in the tweets that were grouped in each category in the OLID data.

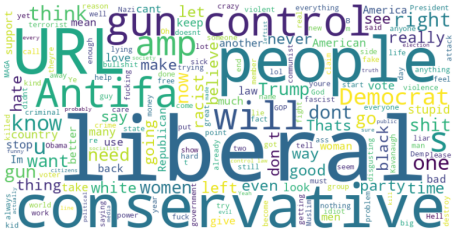

Figure 10: Word Cloud - GRP - OLID data

In figure 10, the most frequent words that relied in the GRP annotated tweets are "liberal", "people", "conservative", and "URL".

In figure 11, the most frequent words that relied in the IND annotated tweets are "URL", "liberal", "will", and "shit".

Figure 11: Word Cloud - IND - OLID data

Figure 12: Word Cloud - OTH - OLID data

In figure 12, the most frequent words that relied in the OTH annotated tweets are "URL", "gun", "will", and "control".

The evaluation for the SOLID test sets are presented in Figure 13. It displays the classification report for the SOLID test sets applied with the SVM linear algorithm.

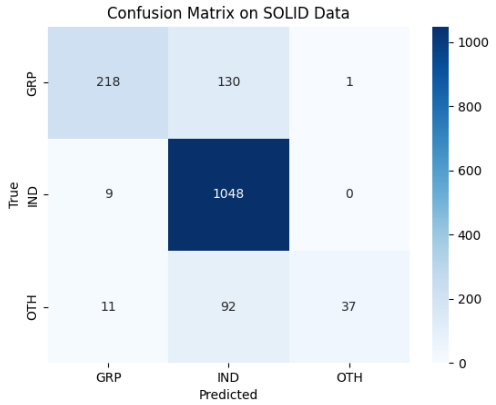

Figure 13: Confusion Matrix for Subtask C using SOLID data

For GRP true and GRP predicted values, we received a value of 218, GRP true and IND predicted were 130, GRP true and OTH predicted are 1 values. In the IND true and GRP predicted, it is 9 values, IND true and IND predicted is 1048. IND true and OTH predicted values were 0. Finally, OTH true and GRP predicted is 11. OTH true and IND predicted is 92. OTH true and OTH predicted is 37.

We also printed out the sampling predictions from the SVM model using the SOLID data. Table 11 below displays few of the sample predictions.

True Label	Predicted Label
IND	IND
IND	IND
IND	IND
IND	IND
IND	IND
IND	IND
IND	IND
IND	IND
GRP	GRP
IND	IND
IND	IND
GRP	GRP

Table 11: Sampling Predictions SOLID Data for Subtask C

Based on these sampling predictions, the true values tend to be the same as the predicted, but at the same time there are some predictions with different true and predicted values.

In the classification report of the SOLID data model using SVM algorithm. Each identification has been stated by displaying the precision, recall, f1-score, and the support. The precision for GRP, IND, and OTH are 92%, 83%, and 97%. The recall for GRP, IND, and OTH are 62%, 99%, and 26%. The f1-score for GRP, IND, OTH are 74%, 90%, and 42%. The accuracy is good for a NLP model, because it is 84.282%.

Three word clouds are created to portray the frequent words in the tweets based on the SOLID data.

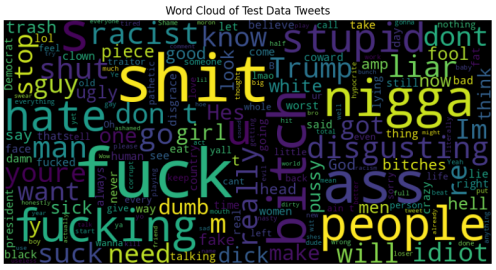

Figure 14: Word Cloud SOLID data

In figure 14, the annotated tweets are have many frequent words which are various bolded profanity words that seem to be the most highlighted words in the annotation.

Future Work

Multilingual Expansion: Extend the offensive language identification to more languages. The current focus is on English, but incorporating multiple languages would make the model more versatile and applicable to a broader range of social media content.

Fine-Tuning Models: Explore further fine-tuning of existing models, such as BERT, to enhance performance on specific nuances of offensive language across different cultures and contexts. This can involve additional training on domain-specific datasets to improve the model's understanding of context-dependent offensive language.

Dynamic Learning: Implement a dynamic learning system that adapts to evolving language trends and new forms of offensive content. Regularly update the model with recent data to ensure it stays relevant and effective in capturing emerging patterns of offensive language.

User Feedback Integration: Develop a mechanism to integrate user feedback into the model's training process. This can help in refining the model based on real-world user interactions and continuously improve its accuracy and efficiency.

Cross-Platform Integration: Extend the model's application to various social media platforms and communication channels. This involves adapting the model to different data formats and structures present in platforms beyond Twitter, making it a more comprehensive tool for content moderation.

Ensemble Approaches: Investigate ensemble methods by combining predictions from multiple models or model architectures. This can enhance the overall performance by leveraging the strengths of different models and mitigating individual weaknesses.

Conclusion

In conclusion, the OffensEval 2020 project presents a significant stride towards addressing the pervasive issue of offensive language in social media. Through the meticulous evaluation of various models, including LSTM and BERT, we have achieved noteworthy results in identifying and categorizing offensive language across different subtasks.

The LSTM model, applied to Subtask A, demonstrated a respectable performance with a precision of 0.92 and a recall of 0.99 on the SOLID 2020 extended easy test set. This indicates the model's effectiveness in accurately pinpointing offensive language, especially in less challenging instances.

The BERT model, applied to Subtask B, exhibited a commendable precision of 0.87 and a recall of 0.88 on OLID data. This suggests its robustness in categorizing offensive language, providing a solid foundation for further improvements.

However, it is essential to note that there is room for refinement, particularly in handling more complex instances of offensive language. The BERT model's performance on SOLID data indicates a lower precision of 0.83 and recall of 0.50, suggesting potential challenges in handling diverse and nuanced offensive content.

To propel this work forward, future endeavors should focus on multilingual expansion, fine-tuning models for specific contexts, and incorporating user feedback. This will contribute to a more comprehensive and culturally sensitive offensive language identification system, fostering safer and more inclusive online environments. The presented results underscore the progress made, while the proposed future work outlines the path for continued advancements in this critical domain.

References

Dan-Yang Li and Jiang Qian. 2016. Text sentiment analysis based on long short-term memory. *2016 First IEEE International Conference on Computer Communication and the Internet (ICCCI)*, pages 471–475.

Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Marcos Zampieri, and Preslav Nakov. 2020. A large-scale semi-supervised dataset for offensive language identification. *arXiv preprint arXiv:2004.14454*.

Mayukh Sharma, Ilanthenral Kandasamy, and Vasantha Kandasamy. 2021. Deep learning for predicting neutralities in offensive language identification dataset. *Expert Systems with Applications*, 185:115458.

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

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). In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, Barcelona (online). International Committee for Computational Linguistics.

Marcos Zampieri, Tharindu Ranasinghe, Diptanu Sarkar, and Alex Ororbia. 2023. Offensive language

identification with multi-task learning. *Journal of Intelligent Information Systems*, pages 1–18.