Hate Speech Detection: Twitter Sentiment Analysis

Thanawat Thanaponpaiboon
220038055
Data Science (Postgraduate)
Guy.Thanaponpaiboon@city.ac.uk

1 Problem statement and Motivation

Nowadays with the blossoming of technology, people tend to use social platforms to update their life and socialize with friends or family more and more, great example of this platform would be Facebook or Twitter. The increasing number of users over time the problem of hate messages from users is likely to increase as well.

Hate speech on social platforms is often used to spread hate, intimidate, or threaten people. Making people that receive the message feel worthless, sad, isolated, and anxious. In some cases, there is some evidence, Maurya et al. (2022) show that victims attempt to commit suicide after being bullied and receiving hate speech from social media for a period. Hate speech and cyberbullying online should be stopped immediately in any case. Therefore, in this study, we shall apply natural language processing along with machine learning to create hate speech detection algorithms. In order to separate the hate speech text from normal text using the Twitter dataset. In the hope that this algorithm will be used further to filter hate speech messages before it will be used to harm or abuse someone on the social platform again.

2 Research hypothesis

In this section, we shall explain the research hypothesis which is an important part to guide our research in the right direction and help us avoid mistakes that may occur leading to a failed experiment. It is worth mentioning that in this study we focus on using Natural Language Processing approaches to analyze the text information to extract hate speech sentences from the corpus. According to the Cambridge dictionary, hate speech is defined as words or sentences that aim to harass and insult causing emotional pain to the victim or target that receives the message. Creating the detection of hate speech is required to cope with the problem of offensive content on social platforms and encourage another study to carry on with hate speech topics for NLP research. Taking advantages of applying machine learning to detect the pattern in the sentence and classify whether that sentence is containing hate speech or not.

The main obstacle with hate speech is, most of the time people is unable to agree on what can be classified as hate speech due to diverse perspective. The purpose of this research is to study the context and build a high-precision algorithm for hate speech detection. Below are research questions consisting of two main parts of this study the first part is to take advantage of cutting-edge technology to create hate speech detection in order to allow humans to take benefit using this model and the second part is the comparison between the two models that we created and the pre-trained model that already exists to compare the performance of between two algorithms for extract hate speech tweet from Twitter dataset.

- Is it possible to create effective hate speech detection which is more reliable than humans and requires less time to extract hate sentences from the corpus?

- Does a pre-train model that already exists and created by the expert more efficient in detecting hate speech than a model that we constructed from scratch?

To answer our research question NLP technique along with machine learning needs to be applied to study the underline context and meaning that lies in the sentence for extracting the hate speech sentence. We found that hate speech can morph into

many different shapes depending on the context. Applying linear and non-linear relation models to find the correlation of the word in the context is necessary in order to produce effective hate speech detection to gain fruitful insight to answer the research question in this study.

3 Related work and background

Language is not only used to convey information to the receiver but also to express the emotion of the speaker. However, there are some case that language is used as a tool to insult or attack the target. The growth of sentiment analysis experiments in natural language processing drives people to construct creative studies and research. Due to hate speech being a subset of sentiment analysis, it allows us to gain useful inspiration from other related sentiment analysis research to guide our study in the right direction. Hence, some of these examples are related and beneficial to our research such as Wang and Zhou (2018) and Ge et al. (2019) using an ensemble learning approach to study sentiment analysis and construct an accurate model to categories labels from the corpus.

In addition, many researchers leverage these algorithms and adapt them to social platforms such as Twitter to analyze user behavior or extract important information from social platforms to observe people's thoughts and opinions. For instance, Giachanou and Crestani (2016) and Kouloumpi s et al. (2021) proposed a supervised approach to the sentiment analysis problem to detect the sentiment of Twitter user messages combined with hashtags to increase the performance of their model. Another piece of evidence that supports this statement, shows that a hashtag is a tag that comes after the tweet in the sentence significantly increases the performance of the classification model to detect human emotion. (Mohammad, 2012)

Research that focuses about hates speech topic on social platforms begin to expand because people are concerned about the effect of sentence that contain hate speech word leading to hatred and dissension among people. MacAvaney et al. (2019) start an experiment about the effect of hate speech problems and tried to apply a support vector machine which is a supervised machine learning algorithm to classify whether the text in the sentence contains elements of hate speech or not. Following by research from Ginting et al. (2019) and Koushik (2019) proposed algorithms that use NLP along with Machine learning approaches which are capable to reveal the pattern of hate content in tweet from Twitter. Using natural language processing methods such as a bag of words and TF-IDF approach to preprocessing the text before training machine learning classifiers such as logistic regression.

Deep learning approach is another popular method to detect non-linear relationships which is much more sophisticated than regular machine learning algorithms. A good example of a comparison between machine learning and deep learning came from Al-Hassan and Al-Dossari (2022). Their research wanted to identify the hate speech from the Arabic tweets into 5 category classes and compared the accuracy of the model between baseline which is the SVM model and other deep learning models such as LSTM and CNN. With the complicated process of constructing deep learning model from scratch, the pre-trained model was introduced to facilitate and shorten the time of modelling. The related study by Suman Dowlagar and Radhika Mamidi (2020) observed the comparison between the pre-trained deep learning model BERT model and the multilingual-BERT model in order to increase the performance of hate speech and offensive content detection models.

Finally, this previous research from expertise shows us the guideline that we are able to accomplish the task of constructing effective hate speech detection which is adequate to answer our research question in this study. Merging with the knowledge that we gained throughout this course will conduct this study to meaningful results and prevent us from getting blind or misleading analysis at the end of this research.

4 Accomplishments

In this section, we will provide a short list of the proposed project tasks and state whether that task is completed or not. Including a brief explanation and a narrative during each task.

Task 1:

- Import Dataset from Hugging Face – Completed
- Preprocess dataset (lower-case, Removing stop words, punctuation etc.) – Completed
- Tokenised the dataset (Using BertTokenizer) – Completed
- Split the dataset into 80:20 for training model and validation – Completed

Task 2:

- Create a simple baseline based on hate-keyword from scratch – Completed Note: **Failed to capture hate-speech sentence assume that using only keywords cannot extract the whole meaning from the sentences**
- Build and train Logistic Regression algorithm baseline model on collected dataset and investigate performance including further analysis and optimization model - Completed

Task 3:

- Build a deep learning model such as CNN or LSTM to examine its performance – Failed due to low accuracy and computational time for training model increase exponentially until GPU on Google colab explodes. In some case model not learning after applying backpropagation, we assume that the architecture of the model that we create or the size of the dataset is too small leading the model to be unable to reveal the hate speech pattern in sentences.

Task 4:

- Build a comparison deep learning model by importing BERT transformer + neuron layer – Completed

Task 5:

- Make comparison model perform better than baseline model by adjusting hyperparameter– Completed

Task 6:

- Perform in-depth analysis to compare two best-fit models and make a conclusion about this study– Completed

5 Approach and Methodology

In this section, we will discuss the approach and methodology that lead us to find the solution to answer our research question. As we mentioned in the previous section that our baseline keyword model fails to detect hate speech from the tweet. Using only one specific word to identify the hate speech sentence may not be suitable for analyzing data gathered from Twitter. Hence, we propose two more sophisticated algorithms which are logistic regression and transformer to detect hate speech from the sentence which we expect that both algorithms will perform better than our keyword baseline model.

1. Our study starts with importing datasets from the website, and pre-processing data using various techniques of natural language processing such as CountVectorise, TF-IDF and BERTtokenise to transform data into vector format before training in machine learning model.

2. We shall import sk-learn library and Pytorch which are the two most useful tools for building machine learning and deep learning algorithms. NLTK library also plays an important part in preprocessing text datasets.

3. Logistic regression is a simple linear approach which acts as a baseline model for this study to compare two algorithms after preprocessing data. In order to apply the machine learning approach, the dataset needs to be transformed into a vector representation. We will consider applying a grid search algorithm to increase the performance of the logit model.

4. For the comparison model we selected BERT (Bidirectional Encoder Representations from Transformers) developed by Google researcher. It is a pre-trained model that could easily apply to our data and provide the benefit of training with less data and compute time than constructing the new model. It helps us to solve the problem of creating a deep learning model which takes too much time to prepare and reduces the complexity of the model.

5. Select the best two performance models indicated by optimizing different hyperparameters across validation data. Compare and contrast two selected approaches and discuss both advantages and disadvantages of using these approaches

including discuss about error result and the behavior of the model that predict incorrect label.

Hate speech text analysis can suffer from extremely unbalanced datasets due to the proportion of the hate speech text is less compared to non-hate speech text. Hence, creating an effective model detection is a challenge. Another issue is deep learning models require large computational operations in terms of memory compared to machine learning. Therefore, Google Colab provide GPU is a great way to execute Deep Learning projects. However, with a large dataset and we execute on a free account that google provides sometimes we might encounter GPU crash problems making this study more challenging. To solve this problem, we divide our data set into small batches before feeding the deep model but still requires an amount of time.

6 Dataset

The dataset we will use in this study was obtained from the hugging face website. It composes of 31962 sentences of text from Twitter which combine regular messages and hate speech text such as racism and sexism. Below are few examples from the dataset that we shall leverage and creating hate speech detection.

	text_tweet	label_tweet
@user when a father is dysfunctional and is so...		0
@user @user thanks for #lyft credit i can't us...		0
	bihday your majesty	0
#model i love u take with u all the time in ...		0
factsguide: society now	#motivation	0

Figure 1: Example of label Tweet in Dataset

The label 0 represent the tweet that is not found the hate speech and 1 which represent the tweet that contains racist/sexist word or comment. As we mentioned in the previous section, the obstacle from this dataset is the high probability of creating overfitted model due to extremely unbalanced datasets (figure 2) only 7% of this dataset is a hate speech text which is possibly insufficient when we are using this data for training and creating hate speech detection.

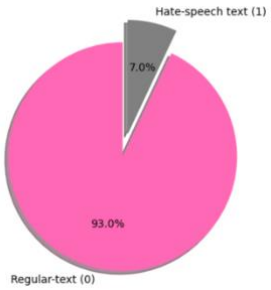

Figure 2: The proportion of hate speech text in Dataset

After import dataset from huggingface website, we store our dataset into dataframe format to make it easier for visualize and preprocessing the dataset which we will discuss in upcoming section.

6.1 Dataset preprocessing

After we obtain this dataset, Data preprocess data using various NLP text preprocessing is required to handle with text dataset in order to prepare the data into the proper format for text mining and training model. For example, extracting meaningful data by removing stop words and punctuation from the text is necessary to extract the meaningful pattern in the particular sentence. Words and sentences that represent characters and text need to be transformed into a number such as a vector representation for the training model. The methods below are used in this study based on popularity in natural language processing to convert text data into a vector representation.

- **Count Vectorization:** the basic approach for text preprocessing using the number of occurrences of each word that appears in the document called bag of words technique to convert text dataset into numerical features that can be used in machine learning algorithms.
- **TF-IDF (Term Frequency – Inverse Document Frequency):** Unlike Count Vectorization TF-IDF also considers the importance of the word into an account. TF-IDF consist of two terms which are Term Frequency (TF) refers to number of occurrences similar to CountVectorization and Inverse Document Frequency (IDF) which measures the significant of the word that appears in the corpus.
- **BERTtokeniser:** A tokenizer library provided by Hugging-face that is used to prepare our dataset into the correct format to make a pipeline for training BERT transformer model. Using WordPiece subword segmentation to convert raw string text into integer sequences.

In both Count Vectorization and TF-IDF, we shall apply n-gram analysis to observe the accuracy of the model across different n-gram sizes during the preprocessing to find the best text-preprocessing method for creating the detection model. The whole dataset was spitted into two groups the first group consists of 80% of the data roughly which is used for training the model and another 20% will be used for validating and testing the performance of our final model. Random state and stratify were set to ensure reproducibility and separate the target class which are hate and non-hate speech to the same amount in both the training set and validation set to avoid the problem of generating overfitted model.

7 Baselines

7.1 Initial Experiment: Hate speech Keyword-based model.

For an initial experiment classifying a label in the dataset by using the keyword based on a hate speech list is a good starting point to compare with more sophisticated algorithms. Although our keyword model can produce a sufficient accuracy of around 92% in both train and validation datasets if we look carefully, we will see that most of the time model will predict non-hate speech (majority class) labels and fail to capture hate speech tweets (minority class). The limitation of this algorithm is dataset may come in the form of a sentence or word that is unseen apart from the given condition list leading to failure to detect hate speech from text most of the time.

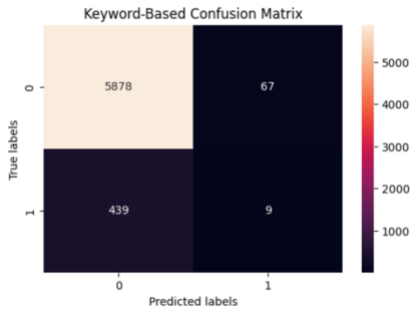

Figure 3: The confusion matrix of Keyword-based model

This is the main reason we must apply a complex model such as machine learning or deep learning to classify hate speech from this dataset. Due to the capability to capture the pattern that is embedded in the corpus.

7.2 Baseline model: Logistic Regression

Logistic Regression is a supervised machine learning algorithm. It is the probabilistic model used for classified binary classification problems which makes it most suitable to be our baseline model for detecting binary labels due to its simplicity to understand and does not require the amount of time for training or preparing a model. It is a type of predictive modelling technique using a linear algorithm along with log odd to find the best-fit line between a dependent variable and one or more independent variables.

- **Advantages**: Simple to understand the concept behind the algorithm. It provides sufficient results for simple datasets with less effort to prepare compared to more complex models. However, the dataset should have linearly separable features to take full advantage of using the logit model.

- **Disadvantages**: Linear Decision Boundaries cannot predict or create a model with a complex relationship or non-linear relation which is why we must examine Deep Learning along with this approach to observe which algorithm performs better in hate speech detection. In addition, it requires time to maintain and preprocess data before feeding into the model and the risk of overfitting when the number of data is too small.

Hyperparameters		Count Vectorizer (Bag of words)			TF-IDF		
penalty	c	unigram	bigram	trigram	unigram	bigram	trigram
L2	0.1	0.9456	0.9354	0.9346	0.9310	0.9300	0.9300
	1	0.9584	0.9452	0.9416	0.9446	0.9362	0.9358
	10	0.9616	0.9504	0.9478	0.9592	0.9470	0.9454

Figure 4: Accuracy of baseline model across hyperparameters

From the result table above logistic regression model along with text preprocessing methods such as TF-IDF and Count Vectorize produce a better accuracy of around 93-95 from training set data compare to keyword-based model. Not only more reasonable accuracy but this time our model can detect hate speech sentences 263 from 448 which more than 58% of hate speech in the testing dataset are detected by this model.

Obliviously, applying a machine learning model can capture the meaning or pattern of the sentence better than searching for the only keyword which explains why the machine learning approaches become more popular in text mining and natural language processing task. We shall use this result to compare with another pre-train transformer BERT which provides state-of-the-art results in a wide variety of NLP problems and becomes one of the most widespread algorithms to work with text sequence models.

8 Results, error analysis

8.1 Experiment result & Choice of hyperparameters

Our comparison model mainly constructs from bi-directional encoder representations using transformers (BERT) from the hugging face and fine-tuning for our sentiment analysis task. Using PyTorch library to add a small decision layer (Linear decision) for the classification tasks at the output of the hidden state of BERT token. Therefore, BERT model in this comparison task act as a text preprocessing such as TF-IDF and Countvectorize to transform our dataset to a meaningful vector before training the model. BERT transformer can be used for a wide variety of language tasks and provide a remarkable result. BERTtokenizer was applied in this pipeline to convert the text into vector representation before training the model. Loss function, learning rate and the number of iterations is three hyperparameters that we will adjust in order to find the best-fit hyperparameters to compare to our baseline model.

Loss function	BERT(Distilled) + Feed Forward Layer		
	Learning Rate	Epoch	Accuracy
Binary Cross Entropy Loss	1e-5	1	0.9640
		2	0.9740
		3	0.9750
	1e-4	1	0.9740
		2	0.9741
		3	0.9610
	1e-3	1	0.9310
		2	0.9310
		3	0.9310
Cross Entropy Loss	1e-5	1	0.8830
		2	0.9180
		3	0.9550
	1e-4	1	0.8210
		2	0.8510
		3	0.9420
	1e-3	1	0.6800
		2	0.6800
		3	0.7100

Figure 5: Baseline model accuracy after adjusting hyperparameters.

The result from the table shows that using binary cross entropy as a loss function in this pipeline is more appropriate because our prediction class is binary (hate, non-hate) and the study suggests that we should increase the number of epochs for the learning model to increase accuracy. However, with the limitation of the memory that google colab provide and the computational time for creating a model we have to limit the number to only 3 iterations per model. Still provide adequate result compared to our baseline model. The final hyperparameter using Binary cross entropy for loss function, 1e-5 learning rate with 3 epochs produces 97% accuracy approximately. On the other hand, Baseline Logistic regression achieves 96% accuracy when predicting validation or unseen dataset which make BERT outperform Logistic regression in detecting hate speech by 1%.

For the logistic regression result using whether TF-IDF or bag of words method produce a close accuracy because our corpus is not big enough to contain a large amount of unique word that allow TF-IDF to extract meaningful information. As a consequence, the result we have seen is not a huge difference.

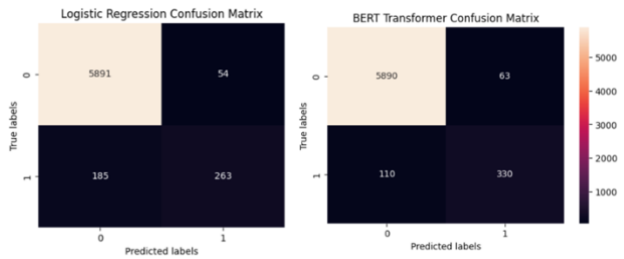

Figure 6: A comparison between two final models after optimization

In order to evaluate the performance of both algorithms, confusion matrix has been applied because it allows us to observe the precise number of correct and incorrect predictions for each model on the test dataset. Our comparison model BERT transformer is more efficient to detect the hate speech from the text than logistic baseline model which is detect 330 correct hate labels from the whole 440 hate-speech tags. Only 173 (110+63) were misclassified by BERT transformer, 63 were misclassified to hate speech but the actual label is non-hate speech. Compared to the result from Logistic Regression baseline which is predicting the wrong class 239 (54+185) 185 were classified as non-hate speech

and the other 54 were classified to hate speech but, it is non-hate speech.

While both algorithms produce similar results, our BERT comparison model slightly improves its capability to detect hate speech in text. Despite this, the BERT algorithm cannot produce results that are far superior to Logistic Regression because of the limitations of GPU memory provided by Colab we have to limit the number of learning iterations. Using an appropriate environment for training BERT transformers, we expect it to produce higher accuracy.

8.2 Error analysis & validation result

Although, we apply machine learning to detect hate speech from tweets slightly improving the model to detect the hate sentence compared to the keyword model. However, there are still some errors and specific conditions that our model cannot detect which impede the creation of an effective model and we will discuss the reason in this section.

sentence	actual label	preict label
immature trying make fool #xenophobe #immature #moron	1	0
porn vids web free sex	1	0
president #woodrowwilson held private screening d.w. griffith's "bih nation" #history #13th	1	0
sums voted #brexit; #littleenglander syndrome sovereignty democracy	1	0
.@user gf used uber without forcing language preferences reading data making #ux choices	1	0
painful truth-another gunman w hate ideology (racism, trumpism, islam, christian) easy access assault weapons	0	1
rightly so! gop hates trump obama	0	1

Figure 7: Error result from machine learning model

The result from figure 7 shows the example result that our model failed to detect or misclassify in some conditions. For instance, in the last two sentences from the table, our model classifies as a hate speech tweet, but the actual label is a non-hate speech tweet. That is because it contains the specific strong hate word such as hate in the last sentence and racism, weapons, and something related to religion respectively in the second one. As we mention that hate speech is mostly related to race, religion, ethnicity, national origin, and gender that intends to insult people. Our model tends to capture this and identify this context into hate labels but example tweets on the table do not intend to attack any individual just only mentioning or telling a sentence.

In some cases, the first few rows on the table model failed to capture hate speech tweets that are because no victim was implicated or insulted. It is obviously evident from example 2 which contain dirty topic suggestion, but the model does not consider it to be an insulting or offensive statement to any person. Therefore, making the model interpreted as a non-hate speech label.

9 Lessons learned and Conclusions

In this study, we use Logistic Regression and pre-trained bi-directional encoder representations using transformers (BERT) for hate speech analysis to detect hate speech tweets from Twitter dataset. We compare these two algorithms by observing accuracy along with confusion matrix when predicting the unseen dataset. Our results show that using the pre-trained BERT for this classification task significantly increases accuracy metrics and be able to capture more hate speech sentence from the text compared to Logistic regression machine learning baseline approaches. Due to deep learning models such as LSTM, transformers can be more advanced in terms of the architecture of the model making it more efficient to capture non-linear relations from the text. Plus, the benefit of using pre-train model provides sophisticated algorithms with less afford for preparing and coding making this study faster and more efficient than ever. However, in exchange deep learning model requires GPUs and computational time for building an effective model. Google colab provide a good platform and GPUs for build and training deep learning model. However, the problem of memory crash when training deep learning is still seen at some point because BERT is a large algorithm. To solve this problem, we decide to split data into a small batch before feeding it into the model but still take quite a lot of time to build deep learning model.

In conclusion, we achieve the point that we can create adequate hate speech detection by applying machine learning and deep learning algorithms both approaches provide a sufficient result to detect hate speech from the sentence. Moreover, the results show that applying pre-train model can also achieve good results with the benefit of using less data to build a model and optimized faster compared to starting from scratch. Further work should focus on the impact of the extremely unbalanced dataset that we underline in the previous section and how to apply some algorithms

to solve this problem such as under or over-sampling the dataset.

References

Maurya C. Muhammad T. Dhillon P. et al. 2022. The effects of cyberbullying victimization on depression and suicidal ideation among adolescents and young adults. a three year cohort study from India. BMC Psychiatry 22, 599.

Suyu Ge, Tao Qi, Chuhan Wu, and Yongfeng Huang. 2019. THU_NGN at SemEval-2019 task 3: Dialog emotion classification using attentional LSTM-CNN. In Proceedings of the 13th International Workshop on Semantic Evaluation, pages 340–344, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Min Wang and Xiaobing Zhou. 2018. Yuan at SemEval2018 task 1: Tweets emotion intensity prediction using ensemble recurrent neural network. In Proceedings of The 12th International Workshop on Semantic Evaluation, pages 205–209, New Orleans, Louisiana. Association for Computational Linguistics.

Kouloumpis E. Wilson T. and Moore J. 2021. Twitter Sentiment Analysis: The Good the Bad and the OMG!. Proceedings of the International AAAI Conference on Web and Social Media. 5, 1 (Aug. 2021), pages 538-541.

Anastasia Giachanou and Fabio Crestani. 2016. Like It or Not: A Survey of Twitter Sentiment Analysis Methods. ACM Comput. Surv. 49, 2, Article 28 (June 2017), pages 41.

Saif Mohammad. 2012. #emotional tweets. In *SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012), pages 246–255, Montréal, Canada. Association for Computational Linguistics.

MacAvaney S, Yao H-R, Yang E, Russell K, Goharian N, Frieder O 2019. Hate speech detection: Challenges and solutions. PLoS ONE 14(8): e0221152.

P. S. Br Ginting, B. Irawan and C. Setianingsih. 2019. Hate Speech Detection on Twitter Using Multinomial Logistic Regression Classification Method, IEEE International Conference on Internet of Things and Intelligence System (IoTaIS), Bali, Indonesia, 2019, pages 105-111.

G. Koushik, K. Rajeswari and S. K. Muthusamy 2019 Automated Hate Speech Detection on Twitter 5th International Conference On Computing, Communication, Control And Automation (ICCUBEA), Pune, India, pages. 1-4

Al-Hassan A. and Al-Dossari H. 2022. Detection of hate speech in Arabic tweets using deep learning. Multimedia Systems 28, 1963–1974.

Suman Dowlagar and Radhika Mamidi. 2020. HASOCOne@FIRE-HASOC2020: Using BERT and Multilingual BERT models for Hate Speech Detection. International Institute of Information Technology-Hyderabad (IIIT-Hyderabad), Gachibowli, Hyderabad, Telangana, India, 500032

Appendices

Any additional data that you consider useful and essential to support your analyses that you think is interesting to include can be included here.

- **Ensemble learning result**: Combination of weak learner and use majority vote to predict hate speech class. As a result, it did not improve model accuracy as we expected.

	Logistic Regression	Support Vector Machine	Ensemble learning
Uni-gram	0.9851	0.9849	0.9829
Bi gram	0.9790	0.9734	0.9730
Tri-gram	0.9700	0.9714	0.9699

- **LSTM error result**: Found the problem of GPU on colab explode and sometimes model did not learn after implement backpropagation. This result led us to employing BERT pretrained model which is faster and require less data to train model.

```
[ ] model = LSTM()
    criterion = nn.CrossEntropyLoss()
    optimizer = torch.optim.Adam(model.parameters(), lr=0.5)

    model.train()
    epoch = 20
    for epoch in range(epoch):
        optimizer.zero_grad()
        y_pred = model(input)
        test, value = torch.max(y_pred,1)

        loss = criterion(value.type(torch.float), label)
        print('Epoch {}: train loss: {}'.format(epoch, loss.item()))
        loss.requires_grad = True

        loss.backward()
        optimizer.step()

    Epoch 0: train loss: 18211.35546875
    Epoch 1: train loss: 18211.35546875
    Epoch 2: train loss: 18211.35546875
    Epoch 3: train loss: 18211.35546875
    Epoch 4: train loss: 18211.35546875
    Epoch 5: train loss: 18211.35546875
    Epoch 6: train loss: 18211.35546875
    Epoch 7: train loss: 18211.35546875
```