Toxic Question Classification in Question & Answer Forum Using Deep Learning

MSc Research Project
Data Analytics

Mathiazhagan Sampath
Student ID: x18139973

School of Computing
National College of Ireland

Supervisor: Dr. Muhammad Iqbal

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Mathiazhagan Sampath
Student ID:	x18139973
Programme:	Data Analytics
Year:	2020
Module:	MSc Research Project
Supervisor:	Dr. Muhammad Iqbal
Submission Due Date:	12/12/2019
Project Title:	Toxic Question Classification in Question & Answer Forum Using Deep Learning
Word Count:	5801
Page Count:	21

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	12th December 2019

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Toxic Question Classification in Question & Answer Forum Using Deep Learning

Mathiazhagan Sampath
x18139973

Abstract

In this internet era, Question and Answer forum are majorly used for knowledge sharing. With large amount of questions getting posted in Q&A forum, identifying the question is toxic and removing the question from forum is the major challenge. To maintain the trust and faith among forum users, there is a need to check if the content is toxic or not. This study proposes to build a Deep Learning model to classify the question based on the toxic content in the question. Quora Q&A dataset is used for this study with 1.30 million records. With imbalance in the dataset, F1 score is used as a metrics to evaluate the model. Two different Deep neural network models are built with Attention layer, learning rate hyper-parameter is selected by Cyclic Learning Rate.

The performance of the model is found to be increasing by adding number of nodes in hidden layer. GPU need to be used for building the model with Bidirectional CuDNNLSTM and CuDNNGRU. Threshold value ranging from 0.01 to 1 is passed to find the maximum F1 score of the model. Model with high number of nodes in the hidden layer is successful in classifying the toxic content and highest F1 score of 0.9001 is achieved at Threshold of 0.40 with Attention layer, CuDNNLSTM and CuDNNGRU on comparing to model with less number of nodes and hidden layers.

Keywords: Natural Language Processing, CuDNNLSTM, CuDNNGRU, Attention, Cyclic Learning Rate, Q&A, Deep Learning, Recurrent Neural Network

1 Introduction

1.1 Background

In day to day life people encounter numerous questions and people expect their question to be answered in a rapid phase. Like a business man will have questions related to legal license, a techie would like to know which career path he should take to build his career. In this internet era, all kind of questions can be posted in Social Question and Answering (SQA) forum and related field expert will answer the question. There are many Social Question and Answering forum like Yahoo Answers, stack overflow and Quora are very useful and Knowledge sharing website Khusro et al. (2017)

1.2 Importance

With so many questions getting posted, SQA need to maintain the quality of the question getting posted. Most of the question posted in SQA will have intended to have a genuine answer. Few questions will be designed in toxic wordings with

1. Non-neutral tone
2. Question is grounded in reality
3. Sexual content that does not try to seek the honest or genuine answer
4. Statement against a community

If question comes under above criteria it can be referred as toxic question[1]. As Quora is the fast-growing Social Question and Answering forum in today's internet world Patil and Lee (2015), Quora is chosen to predict the toxic content in the question. Implementing Deep learning techniques is the optimal solution for classifying the toxic question.

1.3 Research Question and Objective

1.3.1 Research Question

"How Neural Network and Natural Language Processing can be used to classify the Question as toxic or not in Question & Answer forum?"

With Deep learning and Natural Language Processing, Question posted in Q&A forum can be flagged toxic if it contains toxic content.

1.3.2 Research Objective

Objective of this research are as follows

- Classifying the questions by human is impossible for the rate of questions getting posted in SQA and it is a complicated problem.
- To classify the question as toxic or not, Deep learning and Natural Language Processing will be used.
- Cyclic Learning Rate to find the learning rate to build the model
- Attention layer to weight the important words with higher value and RNN to build the model.
- Embedding matrix will be created by joining pretrained word embedding glove and FastText.
- Two different Deep Neural Network model will be built using embedding matrix.
- Evaluation metrics F1 score is used to evaluate the model.
- Model result will be passed through Threshold range of 0.01 to 1 to get maximum F1 score.

[1] `https://www.quora.com/q/quoraengineering/`

1.4 Limitation

Identifying the intent and the meaning of the Question posted is the first step to predict the toxic content. There are few traditional text classification techniques but the classification of Deep learning model like LSTM, Neural Network outperforms the traditional model Chakravarty et al. (2019). In this paper multiple attempts were made to solve toxic question classification issue.

Initially data is downloaded from Quora. After cleaning the data, dataset is of 1.30m rows of Questions with 3 columns. Toxic question with value 1 and non-toxic with value 0. Embedding layer will be created using Glove and FastText. Neural Network CuDNNLSTM and CuDNNGRU will be implemented on word embedding layer to classify the toxic question.

2 Related Work

Many Deep learning techniques have been implemented to improve the text classification and it outperform traditional text classification methods Chakravarty et al. (2019). Transfer learning and Neural networks are used to classify the texts. This section discusses about the various works done by researchers in the field of classifying the texts.

2.1 Machine Learning in Q&A Classification

Automatic classification of questions using Machine learning are implemented by developing 4 different techniques along with that bag-of-words and bag-of-ngrams. Support Vector Machine (SVM), Naive Bayes, Nearest Neighbor and Decision tree are the four different Machine Learning models are built Zhang and Lee (2003). Results shows that SVM outperform other models. Research is done on text classification using based on weight of the text. Term Frequency and Inverse document frequency (TF-IDF) is the traditional method of used to classify the data, A new weighing schema is proposed known as Term frequency and Inverse Gravity Moment (TF-IDM) is introduced by Chen et al. (2016) to classify the text based on the weight. SVM and K nearest neighbor (K-NN) has been implemented on TF- IDM and results shows TF-IDM outperformed traditional TF-IDF in classifying the documents. The test is conducted on the public benchmark dataset. Indra et al. (2016) bag of words technique is implemented to find the topic from the extracted tweets. First tweets are extracted, and pre-processing is done to clean the tweets dataset and Logistic Regression is used to build the model. The model is trained and later the trained model is tested using the test dataset and found that the build model is able to classify the topics and confusion matrix is used to find the accuracy and found that model is of 92% accuracy. To classify the Turkish text, 3000 news has been extracted from extracted from two different source and belong to 5 different domains. Support Vector Machine, Multinomial Naive Bayes, K-Nearest Neighbor, Bernoulli Naive Bayes and Decision Trees algorithms are the different models developed to classify the extracted Turkish news dataset Gürcan (2018). 5-fold cross validation is applied on the dataset and the test results shows that Multinomial Naive Bayes model perform well on comparing to other developed models and KNN algorithm perform very low in the analysis.

2.2 Question & Answer Forum Issues

In Question & Answering (QA) forum finding the question with semantic meaning and avoiding the duplicate question getting posted is the very big challenge in today's internet world Jabbar et al. (2018). Datasets from Quora, Stack Exchange and Ask Ubuntu are used to build the model to find duplicate of the question. Efficient Gradient Tree boosting model (GTB) XGBoost and Siamese Neural Network (SNN) is built to predict the duplicate QA. Transferability of NN is used to increase the performance of under performance of the target domain and it is implemented in SNN to as a Transfer Learning technique. Pre-trained model is used to build word embedding layer and NN is built on top of it to classify the QA based on the Dialog Act (DA) Chakravarty et al. (2019) and DA is to predict the intent of the speaker in a conversation. Proprietary and Tobacco dataset is used. Three different classification model are created with CNN, LSTM, BERT has been created. DA Question specific and Answer specific are the two type of Ontology used. The F1 Score of BERT is 0.84 and outperformed other two classifier model. With rapid growth of Web 2.0, number of questions getting posted increasing rapidly and an automated task is needy in place to classify the Questions to community guidelines and standard Xiao et al. (2017), StackOverflow forum is chosen to find the quality of the question. Naïve Bayesian technique is used to build the model and then accuracy of the model to find the similar question is increased by 2.8%. Most importantly the Recall of the negative scored question is about 4.2%.

An essential domain to today's world is Law and Classifying the questions related to Law in Chinese language is implemented using word2vec and FastText embedding layer created, along with that 4 machine learning algorithm has been implemented and comparison of the model is done to find the best classifying model. Logistic Regression, Naive Bayesian, Stochastic Gradient Boosting, SVM are used to build the classifier model. There was no uniform classification taxonomy of the Chinese language, Khusro et al. (2017) had built a 3 coarse grained and 20 fine grained coarse category. With FastText embedding technique and setting epoch to 12 the model is able to classify the Chinese Question in Law domain. The FastText model outperformed other models developed on Chinese QA. Another important study on classifying the questions based on the semantic meaning using Hierarchical Matching Network (HNM) in Dahou et al. (2016). 10,000 different QA pair is collected from Alibaba.com on Sports, Beauty, shoe and Electronics domain. Positive, Negative, conflict and Neutral are the 4 different semantic categories. Bidirectional Matching Mechanism and Self Matching Attention Mechanism are developed for the semantic classification. Question and Answer sentence are arranged in [Question- Sentence, Answer – Sentence] pair. HNM employs both classification and the research conclude that HNM outperform other baseline approach for sentimental classification of QA. In ALRashdi and O'Keefe (2019) the rise of SQA in explained in today world and shows the importance to maintain the standard of the forum in social web.

2.3 Transfer Learning in Q&A Classification

Pre-trained model is used to create the word embedding layer, this is known as Transfer learning technique. There are many pretrained model available, based on the requirement the suitable model will be chosen. For Image related, ResNet, VGC will be used and for text FastText, Glove and BERT to be used.

2.3.1 Glove Embedding in Classification of Questions

In ALRashdi and O'Keefe (2019) CRISP NLP dataset is used to predict the crisis in an area using Deep Learning techniques. Glove Word Embedding technique is created and then tweets are passed through the neural network. Word Embedding layer along with convolution neural network is evaluated against Bidirectional LSTM (Bi-LSTM). Both models are evaluated and the F1 score is used as evaluation metrics because of the class imbalance. F1 score of Glove model is 62.04% and outperformed the BI-LSTM model.

2.3.2 FastText Embedding in Classification of Questions

Deception detection is nothing but making a false statement in a way that other people believe. Hosomi et al. (2018) FastText is implemented to classify the deception in a spoken statement. FastText is one of the best classifiers to classify based on the tagging of the sentence, Semantic meaning and weight of the words. Useful utterance is extracted from the dataset, dataset consist of 32 hours of audio file which was recorded during the interview. Statistical model is also used to build the model. The model is evaluated and found that FastText word embedding model classify the Deception very accurately. Social networking sites can be used to predict if there is any outage of medical disease in an area. FastText is implemented on the twitter dataset which labelled as Flu and non-flu. After pre-processing and Feature extraction is completed on the dataset, FastText embedding layer is modelled and with different set of features. FastText based classifier is developed to predict the outage of disease in an area using Social network.

2.3.3 Bidirectional Encoder Representations from Transformers (BERT)

Google had recently launched Pre-trained Bidirectional Encoder Representations from Transformers (BERT) Devlin et al. (2019). BERT can be fine-tuned by adding an additional layer to create the state of art model. BERT has produced an outstanding result in many languages on the underlying task Sun et al. (2019). By fine tuning the BERT, high level accurate classifier model can be built. To classify the traditional Chinese medical cases pre-trained BERT has been implemented with final layer as Text-CNN and it is compared with other text classification models Song et al. (2019). BERT based Text-CNN outperformed other text classifier model.

2.4 Classification of Questions using Neural Network

Neural network is created to classify the text along with Word embedding layer created using Transfer learning technique. Different neural network is created for accurate classification of the dataset. Guggilla et al. (2016) CNN and LSTM are the models created to classify the online user comments and the CNN perform well on comparing to LSTM. In Dahou et al. 2016; Essa et al. 2018 Neural network model is built to classify the data based on the label of the data.

2.4.1 Attention Layer and Neural Network in Classification Problems

Classification of data based on sentimental label is most discussed problem Natural language processing and Neural network-based classification is very effective Yang et al. (2016). To get better semantic understanding of the text/image, Attention layer is used to get the weight of different set and implemented along with RNN, GRU, LSTM Zhang et al. (2016) and the results of the model are with high accuracy on comparing to model without attention layer.

A special kind of Recurrent neural network (RNN), Long Short-Term Memory (LSTM) is capable of learning Long term dependencies. Another RNN model Gated Recurrent Unit (GRU) which helps in vanishing gradient vanishing problem. Malware detection is done using LSTM and GRU in Athiwaratkun and Stokes (2017). LSTM and CNN are used to classify the word using the features in text on a small dataset and it outperformed the other models based on character level Gumilang and Purwarianti (2018). Character level Recurrent neural network (CRNN) and Character level Convolution Neural Network (CCNN) are available for sentence categorization, These models are compared with benchmarking corpus and proposed model achieved higher F1 score on Google news data set Fu et al. (2017).

2.5 Selection of Learning Rate through Cyclic Learning Rate

To train a neural network we need to tune the hyperparameters, learning rate is one of the important parameters to be tuned before training the neural network. Instead of manually trying with different learning rate for finding the best one to train, Cyclic learning rate is the best way to fix the learning rate by setting up the few parameters. By setting Minimum bound(base_lr) and Maximum bound(max_lr) and step size is fixed with triangular form in Smith 2015; Niyaz et al. 2018. Image classification with transfer learning in done with help on Cyclic Learning Rate(CLR) to find the learning rate along with Deep neural network.

3 Methodology

This main purpose of this research focus on classifying the toxic content in Question and Answer forum. To build a model to categorize the question, Cross-industry process for data mining(CRISP-DM) methodology is used and k-fold cross validation is implemented. Figure 1 is the pictorial representation of the Cross-industry process for data mining methodology to build a classifier model.

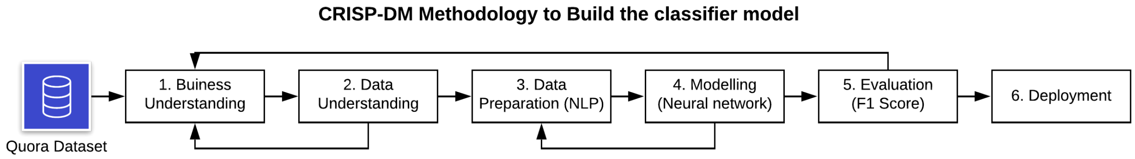

CRISP-DM Methodology to Build the classifier model
Quora Dataset
1. Buiness Understanding
2. Data Understanding
3. Data Preparation (NLP)
4. Modelling (Neural network)
5. Evaluation (F1 Score)
6. Deployment

Figure 1: CRISP-DM methodology

3.1 Business Understanding

In today's internet world, People would like to know the answers for their question in a genuine way Chakravarty et al. (2019). In Question and Answer forum people post question and forum need to maintain the quality of the questions it gets posted. But most of the questions in forum are toxic and this problem to be taken care by platform to make users feel that knowledge shared from forum is safe and not biased. The questions posted in the platform can be a statement against a community and semantic meaning of the question can give false statement, these come under insincere/toxic question category. To handle this issue manual human intervention is not possible, A Deep learning model can be build to flag the question as an toxic question and maintain the quality of the forum and keep the toxic question away from forum.

3.2 Data Understanding

Quora is Question and Answering forum for knowledge sharing and it had released the data of toxic question dataset. The data is downloaded from Quora website which consist of questions, question id and binary value to label it as toxic question or not. The data set is of 1.30 million questions. For transfer learning, glove and FastText embedding layer are created downloaded from the repository of Stanford and Facebook.

3.3 Data Preparation

Train and test dataset questions consist of many special character, null values, misspell of words, stop words removal, punctuation are taken care. Different visualizations are created to see the frequent unigram, bigram and trigram of the dataset.

3.4 Feature Engineering

Meta feature extraction is done, and distribution is visualized. The feature extracted are as follows.

1. Number of words in a question
2. Number of stop words
3. Number of punctuation
4. Number of upper case words
5. Number of unique words in a question
6. Number of characters in a question

3.5 Sequence Creation

Natural language processing (NLP) need to be implemented to make machine to understand the text, A question is a sequence of text and text will be converted into numerical form. keras.preprocessing.text.Tokenizer convert text into tokens. Each question used for training the model should be of same length, using pad sequence the length of the question is fixed.

3.6 Word Embedding

Word embedding plays a key role in Natural Language process (NLP) along with Neural Network. Glove and FastText are the pretrained embedding layers used to classify the questions.

3.6.1 Glove Embedding

Glove embedding creates co-occurrence of matrix (word * context) ALRashdi and O'Keefe (2019), As a result huge matrix will be created, and factorization is done to represent the matrix in lower dimension matrix.

3.6.2 FastText Embedding

FastText embedding layer is quite different on comparing to Glove, it generates embedding for rare words on ngram passed to the layer where other embedding technique cannot achieve this Hosomi et al. (2018).

3.6.3 Embedding Layer Enrichment

Glove and FastText embedding layer are used to build deep learning models, these are pretrained models and few words present in our question will be available in pretrained layers. Words are represented as key and the value of the key as word_vector and length are fixed to 300. Glove and FastText embedding layer are created separately and both are combined together to form the final embedding matrix and the tokenized words will be passed through it to form required embedding.

3.7 Modelling

In this research, Few Deep learning models are built using Bidirectional Long short-term memory (LSTM) and Gated Recurrent Unit (GRU) are built using Attention layer and to find the best learning rate, Cyclic Learning Rate (CLR) is used.

3.7.1 Design of Bidirectional Recurrent Neural Network (RNN)

Convolution Neural Network can read two words together but RNN remembers the sequence and the hidden meaning text and connects it to the current state. Bidirectional RNN remembers the information in both the directions. RNN is better choice for text classification on comparing to the Text CNN. Bidirectional LSTM/GRU is sub class of RNN and it can remember the words for a long period in both directions for classification of text.

3.7.2 Attention Layer for Neural Network

The sequence of data will be taken care by LSTM/ GRU but important words are not given higher weights. On the other hand TFIDF, extracts the feature from the words by keyword extraction. To incorporate both, an Attention layer is created with score. Attention layer is created by extracting the words that give the meaning of the sentence and convert the semantically meaningful words to sentence vector Vaswani et al. (2017) Figure 2 represent the Attention layer for the proposed model with Input as vectors of question text and feed to Embedding layer and RNN with Attention.

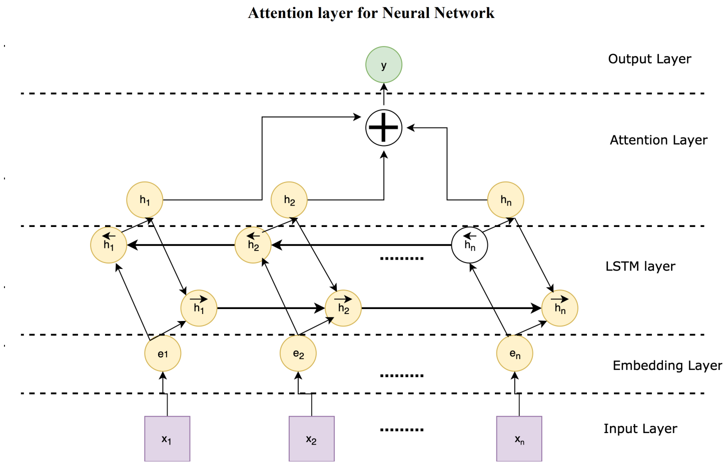


Figure 2: Attention Layer

3.7.3 Cyclic Learning Rate(CLR)

Cyclic learning rate is used to find the best learning rate to build the neural network. Learning Rate is a key hyperparameter Smith 2015; Niyaz et al. 2018. Within few iteration an optimal learning rate (LR) can be determined and approach is similar to Stochastic Gradient Descent with Warm Restarts (SGDR) Loshchilov and Hutter (2017).

3.8 Evaluation of Classifier Model

k-fold cross validation is performed for few epochs and there are few imbalances in the dataset for text classification, With imbalance in the data, F1 score metric need to be used over Accuracy. In recent version of keras precision and recall are removed from metrics, F1 score method is defined to calculate the precision, recall and F1 score for each iteration.

$$precision = \frac{TP}{TP + FN}$$

$$recall = \frac{TP}{TP + FN}$$

$$F1 = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

4 Implementation

Data collected from Quora website and Explanatory data analysis is done in Python using Kaggle kernel with GPU enabled. Glove and FastText Embedding Layer are downloaded from source and concatenated to form the Embedding Matrix. Questions are converted into vectors and feed to embedding matrix and passed to Neural network with attention layer and cyclic learning rate with k-fold cross validation and models are evaluated using F1 score. Figure 3 represent the architecture design to build a classifier model. Implementation phase is divided into Data Collection, Data pre-procesing, Building classifier model and Evaluation.

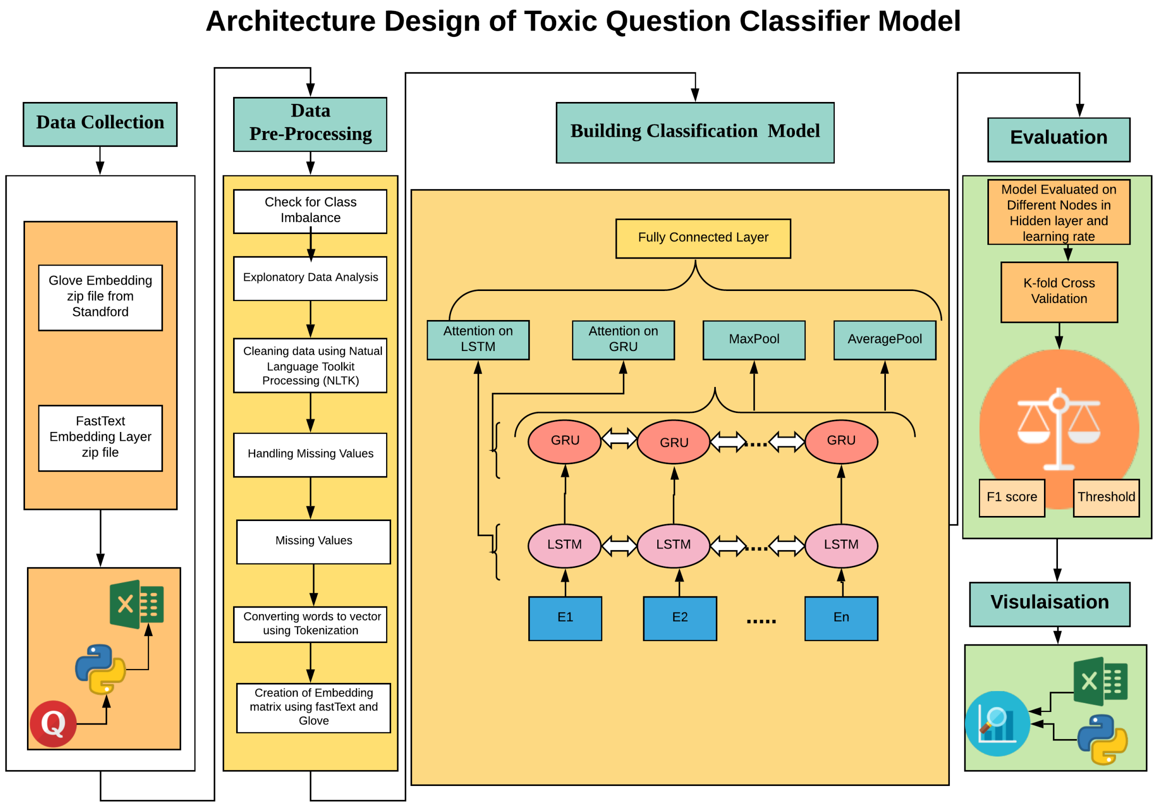

Architecture Design of Toxic Question Classifier Model
Data Collection
Glove Embedding zip file from Standford
FastText Embedding Layer zip file
Q
Data Pre-Processing
Check for Class Imbalance
Explonatory Data Analysis
Cleaning data using Natual Language Toolkit Processing (NLTK)
Handling Missing Values
Missing Values
Converting words to vector using Tokenization
Creation of Embedding matrix using fastText and Glove
Building Classification Model
Fully Connected Layer
Attention on LSTM
Attention on GRU
MaxPool
AveragePool
GRU
GRU
GRU
LSTM
LSTM
LSTM
E1
E2
.....
En
Evaluation
Model Evaluated on Different Nodes in Hidden layer and learning rate
K-fold Cross Validation
F1 score
Threshold
Visulaisation

Figure 3: Implementation of Classifier Model

Data is downloaded from Quora data repository, by using Kaggle kernel and Python Explanatory data analysis and pre-processing of the dataset is as follows. Quora Q&A dataset is of 1,306,122 rows.

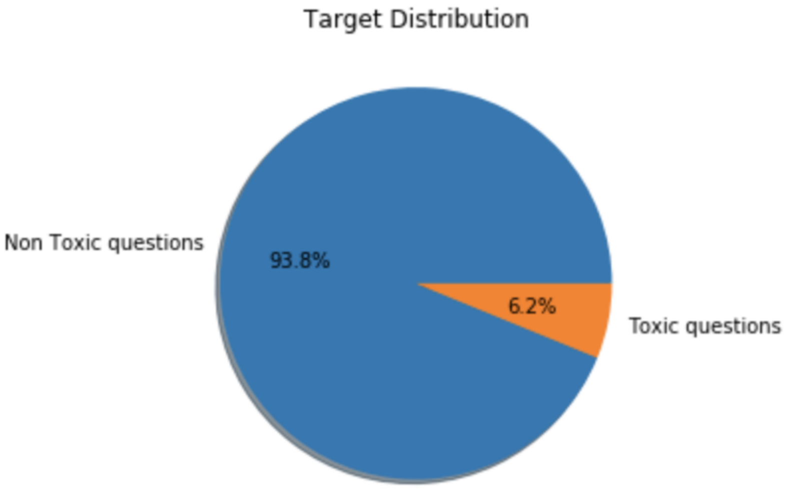


Figure 4: Class Imbalance in data

Data distribution of target value is plotted using plotly in Python and found there is an imbalance in the target data in pie chart. From Figure 4 it is clear that 6.2% of question are toxic/insincere question and remaining are non-toxic/sincere question. Frequently used words in the training data are visualized by using word cloud. Maximum number of words in word cloud is set to 200. Figure 5 represent the word cloud of top 200 words in dataset.

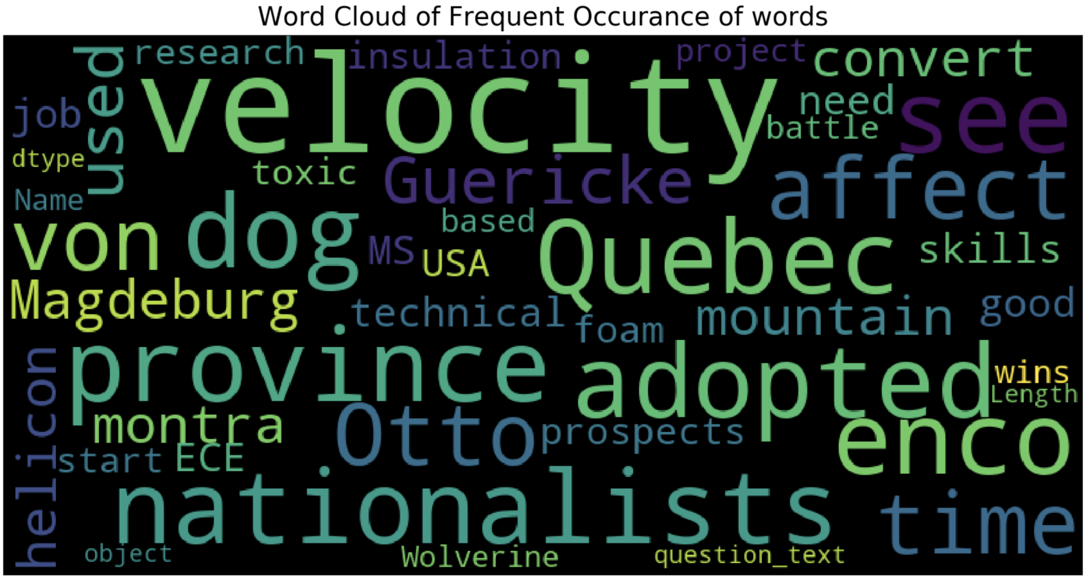


Figure 5: Word Cloud of text in Questions

In many combinations there are many frequently used words, to explore the combination ngram is implemented to do it. Most commonly used single words is explored as onegram. Similar to that bigram and trigram are used to find the frequently used combination of two and three words in both toxic and non-toxic questions.

4.1 Feature Engineering of Questions

Basic feature of questions with number of words, character, unique character, stopwords, average length of the words, upper case words and title case words are extracted.

For better visualization of the extracted features, number of words higher than 60 in each question are set to 60, Questions with punctuation count higher than 10 are set to 10 and if number of character count are in each question higher than 350 are truncated to 350. By visualising the features in boxplot using Matplotlib and seaborn, Insincere question has the highest number of character and words on comparing to insincere questions. Unique words are higher in insincere question. Correlation Matrix of the extracted features is visualised and Figure 6 represent the same.

Data collected from different sources need to be cleaned under different criteria, to do a standard procedure of cleaning the dataset to be done for Natural Language Processing (NLP) task.

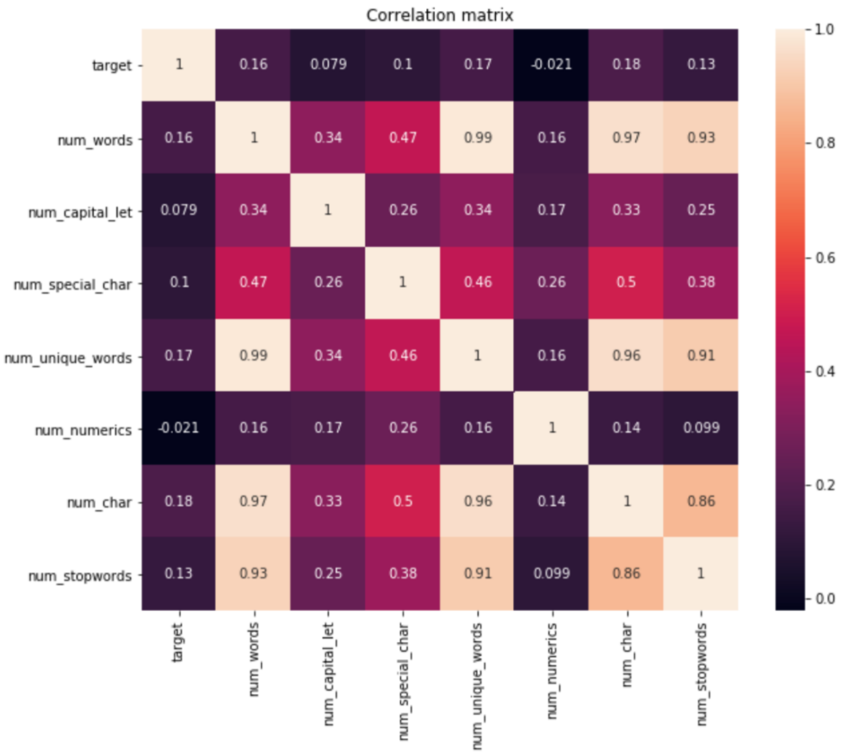


Figure 6: Correlation Matrix of Features

4.2 Handling the Class Imbalance

With 6.2% of toxic question and 93.8% if nontoxic questions, Undersampling is performed on training dataset to handle the class imbalance.

4.3 Data Cleaning

Cleaning of Maths related symbols by removing the math symbol to “MATH EQUATION” and website related information are changed into “URL” using Regex.

4.3.1 Removal of Punctuation in the question

Different special characters are present in question are extracted and stored in an array. If extracted special characters are available in question it will be replaced with space.

4.3.2 Converting text to lowercase

The essential step in preprocessing is to convert each text in dataset to lower case letter.

4.3.3 Missing values

Handling null value is the basic step in cleaning the data, if there are any null value in the train and test dataset, that particular row will be updated to “##”.

4.3.4 Removal of misspelled words

Removal of misspelled words is also an important step of natural language processing. Frequently misspelled words are changed to their semantically correct words and a python function correction_mispell is created to clean the dataset.

4.3.5 Removal of common stop words

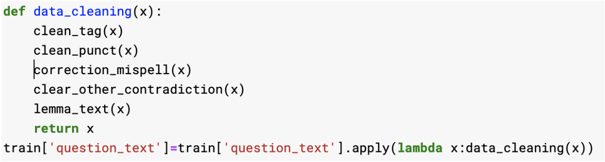
In Natural language Processing useless words are known as stop words. English words like the, is, was, a, an, that are known as stop words. By importing Natural language took kit (NLTK), list of common stop words is predefined in nltk are imported by default and those words are removed from question text.

4.3.6 Removing the Contraction

Dataset consist of so many texts with apostrophe (‘) is known as contraction. As part of natural language processing we need to remove the contraction words to standardize the text. Contraction mapping is done and stored in an array and using regex the contraction are removed to mapped text.

4.3.7 Lemmatization of words

Lemmatization comes with NLTK library. Morphological analysis of words are done. Similar meaning of words are linked to one word by Lemmatization in Natural Language Processing. Figure 7 represent the code snippet of Data cleaning process.



```
def data_cleaning(x):
    clean_tag(x)
    clean_punct(x)
    correction_mispell(x)
    clear_other_contradiction(x)
    lemma_text(x)
    return x
train['question_text']=train['question_text'].apply(lambda x:data_cleaning(x))
```

Figure 7: Python code for Data Cleaning

4.4 Sequence Creation of Questions

4.4.1 Tokenizer

Next step in Natural Language Processing is to convert the text into machine understandable language, to do that we need to convert the text into tokens using tokenizer from keras library. num_words is the parameter defines the number of words need to be tokenized in tokenizer. The next step is to fit the train and test data using tokenizer. ‘num_words’=6100. Table 1 are the configuration used to build the model.

Table 1: Configuration Value

Configuration	embed_size	max_features	maxlen	pad_sequence
Value	300	6100	70	70

4.4.2 Pad Sequence

In deep learning, Model will expect the data to be of same length, to set the length of the text sequence for each question “maxlen” feature will be used on dataset and will be passed as a parameter in “pad_sequence”. Figure 8 represent the tokenization of words in questions.

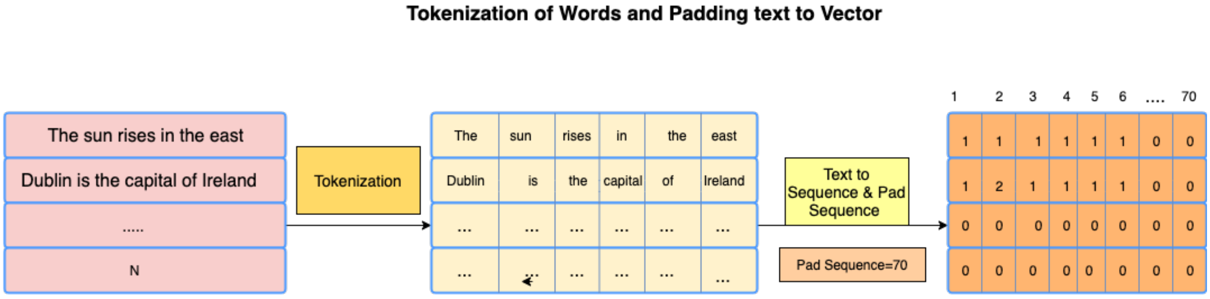

Tokenization of Words and Padding text to Vector
The sun rises in the east
Dublin is the capital of Ireland
.....
N
Tokenization
The | sun | rises | in | the | east
Dublin | is | the | capital | of | Ireland
... | ... | ... | ... | ... | ...
... | ... | ... | ... | ... | ...
Text to Sequence & Pad Sequence
Pad Sequence=70
1 2 3 4 5 6 70
1 1 1 1 1 1 0 0
1 2 1 1 1 1 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0

Figure 8: Tokenization of words

4.5 Pretrained Word Embedding for Classifier Model

An embedding matrix will be created by combing the glove and fastText word embedding. Pretrained glove embedding prepared on large corpus of Wikipedia data and fastText is extension of word2vec. Glove Embedding matrix with “max_feature” is created as embedded_matrix_glove and embedded_matrix_fast is created with “max_feature” on fastText embedding. Both embedded_matrix_fast and embedded_matrix_glove are combined together to form embedding_matrix. Input data will be passed to the embedding_matrix. Figure 9 represent the creation of Embedded Matrix using Glove and fastText

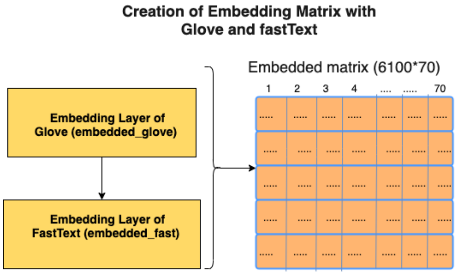

Creation of Embedding Matrix with Glove and fastText
Embedding Layer of Glove (embedded_glove)
Embedding Layer of FastText (embedded_fast)
Embedded matrix (6100*70)
1 2 3 4 70

Figure 9: Embedded matrix of Glove and fastText

4.6 Prediction of Learning Rate by using CLR for Text classification

Cyclic Learning Rate (CLR) class is to find the optimal learning rate. With below arguments passed to the CyclicLR class, learning rate can be predicted for the model is an optimal way.

- base_lr: The minimum boundary of the cycle. base_lr is set to 0.001.

- max_lr: Upper boundary of the cycle. Technically amplitude of the cycle will be set based on the max_lr (max_lr - base_lr). In most of the cases, max_lr will not be reach max_lr depending on the scaling function defined. is set to 0.003.
- step_size: For each iteration how many iterations need to be done is defined in step_size. step_size is set to 300 for each epoch.
- mode: It depends on the scale_fn argument, if scale_fn is none, mode will not be ignored.
- scale_fn: Is set to None and exp_range mode is set to the class by default.
- gamma: The value of gamma is constant in finding the learning rate. Gamma value is set to 0.99994.
- scale_fn: Based on the value set, mode will be ignored or considered. To determine the learning rate using the defined class, mode need to be passed with value to do that scale_fn is set to None.
- scale_mode: It is passed to set the epoch as iteration or cycle, for text classification scale_mode is set to cycle.

Figure 10 represent the Cyclic Learning Graph with exp_range mode used to build the proposed model.

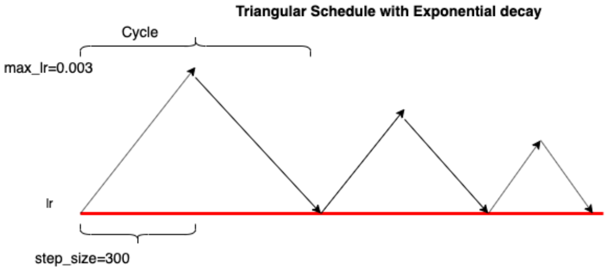


Figure 10: Cyclic Learning Graph for proposed model with exponential decay.

Apart from finding the learning rate using CyclicLR, function has been written to find the F1 score. The inbuilt F1 score calculation from keras is not used to evaluate the model, as the results are not the appropriate for batch wise. Batch wise average of recall and precision is computed. Using the computed value of precision and recall, F1 score is calculated.

Table 2 are the arguments passed to the CyclicLR function to predict the learning rate.

Table 2: Cyclic Learning Rate (CLR) Arguments

Arguments	**base_lr**	**max_lr**	**step**	**mode**	**gamma**	**scale_fn**	**scale_mode**
Value	0.001	0.003	300	exp_range	0.99994	None	Cycle

Learning Rate is a important Hyperparameter to train the model. Three different neural network are build with different combination max_lr(0.002, 0.003, 0.004) and base_lr=0.001. Based on the highest F1 score achieved on both test and train dataset, max_lr of 0.003 is selected. Table 3 provides the detail of the arguments used to predict the learning rate of the Neural Netowrk.

Table 3: Learning Rate of the Model

max_lr	Train	Train_threshold	Test	Test_threshold
0.002	0.899480937	0.44	0.901019942	0.41
0.003	0.900086934	0.4	0.900193056	0.42
0.004	0.902714145	0.44	0.904519877	0.41

From Figure 11 it is clear that at Learning Rate 0.003 model F1 score on both train and test are at same equilibrium. max_lr is choosen as 0.003 to trian the models.

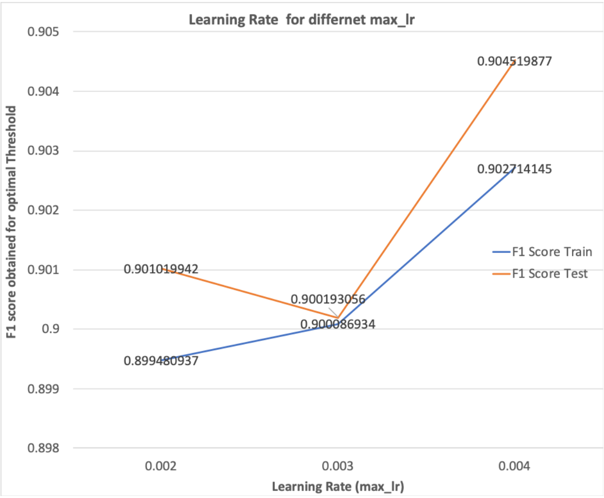


Figure 11: Learning Rate

4.7 Bidirectional Recurrent Neural Network

LSTM and GRU and two Recurrent Neural Network which remembers the word in a question from previous text in a sequence of hidden state Zhang et al. (2016). Both LSTM and GRU are used together to form the neural network. CuDNNLSTM and CuDNNGRU are similar to LSTM and GRU but they run faster on GPU. Many different combinations are tried to build the model, Out of that only 2 models are shortlisted. With different combination of hidden layers and nodes many different models are created and out of that 2 models provided high F1 score on comparing to other models.

1. CuDNNLSTM(256)+CuDNNGRU(128)+Attention
2. CuDNNLSTM(40)+CuDNNGRU(40)+Attention

Above two are the two different combination of neural network used for building further classification model. The architecture of the neural network starts with Input from embedding matrix and followed by SpatialDropout1D(0.1). CuDNNLSTM with 256 nodes and output is passed to CuDNNGRU with 128 nodes. Attention layer is created and placed after CuDNNLSTM and CuDNNGRU. GlobalAveragePooling1D, GlobalMaxPooling1D, Attention of both GRU and LSTM are concatenated together and then

feed to Dense output layer with one node. Sigmoid activation is applied on the output layer. The final layer of neural network is output layer is of Dense with one node. Model is complied with “binary_crossentropy” as loss function, “adam” optimizer and metrics as F1.
Another neural network with Embedding matrix as input with SpatialDropout1D of 0.1. Along with that CuDNNLSTM is placed with 40 nodes followed by 40 node of CuDNNGRU. Attention layer behind CuDNNLSTM and CuDNNGRU is placed and will be concatenated with GlobalAveragePooling1D and GlobalMaxPooling1D of CuDNNLSTM and CuDNNGRU. Output of the concatenated layer is feed into Dense layer with 16 nodes with ”relu” activation. Dropout with 0.1 is passed to Dense Layer and final output layer with 1 node. ”sigmoid” activation is passed to the output layer. Model is complied with “binary_crossentropy” as loss function, “adam” optimizer and metrics as F1. Evaluation of the above two models are discussed elaborately at Evaluation Section.

4.8 Attention Layer for LSTM&GRU

For long sequence text data, LSTM and GRU remembers the previous sequence through hidden states. High weight for the important words in the input text are not given based on the semantic meaning. To achieve this a Attention layer need to be placed next to Bidirectional LSTM/GRU Yang et al. (2016), weight of the important words are set by dot product with context vector and raised to exponentiation.

4.9 Threshold Search for Maximum F1 Score

F1 score is used to measure the success of binary classification when there is an imbalance in one class. Maximum F1 score can be calculated by optimising the threshold for binary classification Lipton et al. (2014). Maximum F1 score for classifying the question is toxic or not is calculated for various threshold ranging from 0.01 to 1. threshold_search function will return the maximum F1 score and threshold value at which maximum F1 score is achieved.

4.10 K-Fold Cross Validation on Train Data

k-fold cross validation is used to avoid overfitting of the model by running model with k different samples. StratifiedKfold is used to split the data into k number of subset for n number of epochs. After training the model with different number of k, 4 fold is chosen to train the model with 4 epochs.Two different models are built with StratifiedKfold and discussed briefly in evaluation section.

5 Evaluation

Two different models are built based on the learning rate obtained from CyclicLR. Number of epochs and number of nodes in hidden layer are need to be determined, to do that different combination of neural network are built and the models are evaluated. Model with GRU+LSTM+Attention layer is explored with following rate to find the optimal Cyclic Learning Rate to train the model. By changing the number of nodes in hidden layer different experiments are conducted to built the neural network. As there was imbalance in the dataset, F1 score metric is used to evaluate the model.

Table 4: Experiment on selection of Layers and Nodes for Neural Network

Layer and Nodes	Epochs	K-fold	F1 Train	F1 Test
(CuDNNLSTM (40) + CuDNNGRU(40) + [LSTMAttention, GRUAttention, MaxPool, AvgPool] + DenseLayer(16) + Dense(1)	4	4	0.8993	0.8995
(CuDNNLSTM (256) + CuDNNGRU(128) + [LSTMAttention, GRUAttention, MaxPool, AvgPool] + Dense(1)	4	4	0.9001	0.9007

To build a neural network, choosing the hidden layers and number of nodes to the hidden layers need to be determined. By performing different combination of layers and node, Two combination provided the best results for 4 fold Cross Validation with 4 epochs. Results are available in Table 4 and model with high F1 score is chosen for building the classification model. The batch size for train data is 512. For prediction of validation and test data, batch size is set to 1024.

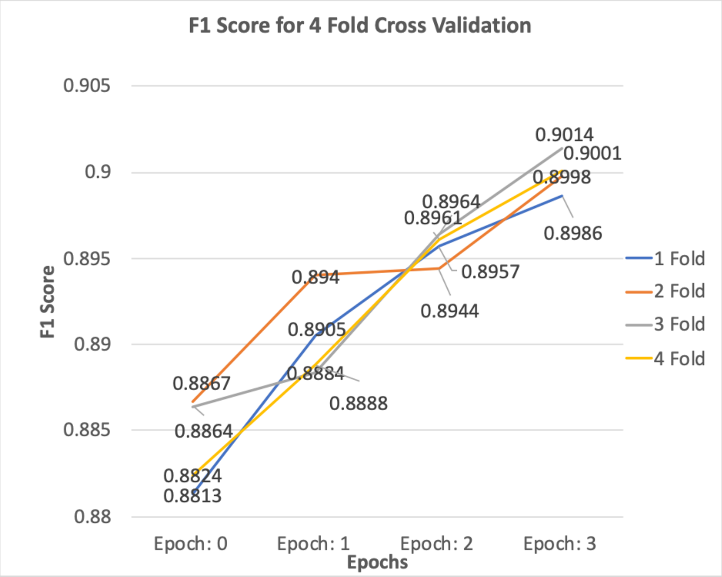


Figure 12: Result of Model on 4 Epochs and 4 fold Validation

5.1 Discussion

Model with 256 nodes in CuDNNLSTM and CuDNNGRU with 128 nodes provide the highest F1 score on both train and test data. With 4 epochs and 4 fold Cross Validation data model is trained and validated on the validation data. Figure 12 represents the F1 score for each fold and epoch.

Based on the Precision and Recall, Best F1 score of the model is calculated for threshold for different range. Figure 13 shows the F1 score for threshold range from 0.1 to 0.5

for both train and test dataset. Without use of TPU model has been built and best F1 score of 0.90008693 is achieved on train dataset at threshold of 0.40. On test dataset, Maximum F1 score of 0.900193 is achieved at 0.42 threshold.

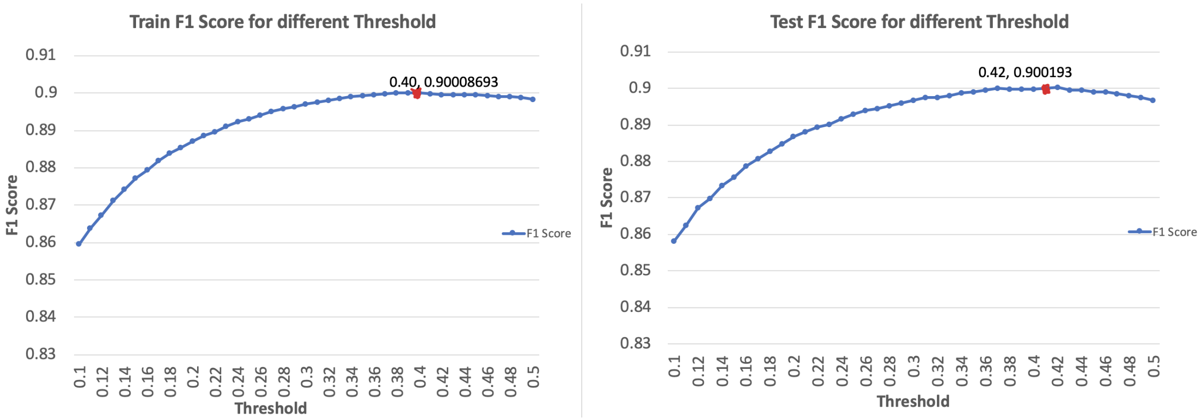


Figure 13: Result of Model on Train and Test data

ALRashdi and O'Keefe (2019) model to classify the tweets with LSTM and CNN achieved F1 score of 62.04%, Our proposed model had achieved higher F1 score with LSTM and GRU. It can be inference from the result that CNN is not good for the text classification as it cannot remember the long sequence of text with on comparing to LSTM and GRU. Attention layer behind the LSTM and GRU are very helpful in handling the words by giving high weight to important words in the question. Essa et al. (2018) results might be higher if Attention layer is implemented to the RNN.

6 Conclusion and Future Work

This research work focus on classifying the question in Online Question and Answer forum based on the toxic content in the question. With RNN and transfer learning techniques model is built using GPU. With imbalance in dataset, Undersampling is done to get the class count to equal and F1 metrics is used to evaluate the model. Using k-fold cross validation on dataset, 0.9001 F1 score is achieved at 0.40 threshold. Other sampling techniques can be performed in future to get the best from the data. Due to limitations not able to explore various parameter of keras library, Exploring the keras library and tuning the Neural Network will be useful in building a high standard classifier model.

PyTorch library can be used along with Keras, Metrics of PyTorch and sklearn on calculating the Precision, Recall, F1 Score need to manually created to find the average of batch epochs in cross validation. TPU computation can be done using Bidirectional Encoder Representations from Transformers (BERT) developed by Google on large number of balanced dataset in COLAB. The computation speed will be very high on comparing to GPU and CPU, Different combination of hidden layer and nodes with Dense layer can be developed at very short time. In future this model can be extended to different domain like Law, Medicine to predict the false/toxic statement and ease the workload of people and government.

References

ALRashdi, R. and O'Keefe, S. (2019). Deep learning and word embeddings for tweet classification for crisis response, *ArXiv* **abs/1903.11024**.

Athiwaratkun, B. and Stokes, J. W. (2017). Malware classification with lstm and gru language models and a character-level cnn, *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* pp. 2482–2486.

Chakravarty, S., Chava, R. V. S. P. and Fox, E. A. (2019). Dialog acts classification for question-answer corpora, *ASAIL@ICAIL*.

Chen, K., Zhang, Z., Long, J. and Zhang, H. (2016). Turning from tf-idf to tf-igm for term weighting in text classification, *Expert Syst. Appl.* **66**: 245–260.

Dahou, A., Xiong, S., Zhou, J., Haddoud, M. H. and Duan, P. (2016). Word embeddings and convolutional neural network for arabic sentiment classification, *COLING*.

Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding, *NAACL-HLT*.

Essa, A. A., Faezipour, M. and Alhassan, Z. (2018). Text classification of flu-related tweets using fasttext with sentiment and keyword features, *2018 IEEE International Conference on Healthcare Informatics (ICHI)* pp. 366–367.

Fu, X., Ch'ng, E., Aickelin, U. and See, S. (2017). CRNN: A joint neural network for redundancy detection, *CoRR* **abs/1706.01069**.
URL: *http://arxiv.org/abs/1706.01069*

Guggilla, C., Miller, T. and Gurevych, I. (2016). Cnn- and lstm-based claim classification in online user comments, *COLING*.

Gumilang, M. and Purwarianti, A. (2018). Experiments on character and word level features for text classification using deep neural network, *2018 Third International Conference on Informatics and Computing (ICIC)*, pp. 1–6.

Gürcan, F. (2018). Multi-class classification of turkish texts with machine learning algotirthms, *2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)* pp. 1–5.

Hosomi, N., Sakti, S., Yoshino, K. and Nakamura, S. (2018). Deception detection and analysis in spoken dialogues based on fasttext, *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)* pp. 139–142.

Indra, S., Wikarsa, L. and Turang, R. T. B. (2016). Using logistic regression method to classify tweets into the selected topics, *2016 International Conference on Advanced Computer Science and Information Systems (ICACSIS)* pp. 385–390.

Jabbar, M. S. M., Kumar, L. N., Samuel, H., Kim, M., Prabhakar, S., Goebel, R. and Zaïane, O. R. (2018). On generality and knowledge transferability in cross-domain duplicate question detection for heterogeneous community question answering, *CoRR* **abs/1811.06596**.
URL: *http://arxiv.org/abs/1811.06596*

Khusro, S., Alam, A. and Khalid, S. (2017). Social question and answer sites: the story so far, *Program* **51**(2): 170–192.
URL: *https://app.dimensions.ai/details/publication/pub.1085548093*

Lipton, Z. C., Elkan, C. and Narayanaswamy, B. (2014). Thresholding classifiers to maximize f1 score.

Loshchilov, I. and Hutter, F. (2017). Sgdr: Stochastic gradient descent with warm restarts, *ICLR*.

Niyaz, U., Sambyal, A. S. and Devanand (2018). Advances in deep learning techniques for medical image analysis, *2018 Fifth International Conference on Parallel, Distributed and Grid Computing (PDGC)* pp. 271–277.

Patil, S. and Lee, K. (2015). Detecting experts on quora: by their activity, quality of answers, linguistic characteristics and temporal behaviors, *Social Network Analysis and Mining* **6**: 1–11.

Smith, L. N. (2015). Cyclical learning rates for training neural networks, *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)* pp. 464–472.

Song, Z., Xie, Y., Huang, W. and Wang, H. (2019). Classification of traditional chinese medicine cases based on character-level bert and deep learning, *2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)* pp. 1383–1387.

Sun, C., Qiu, X., Xu, Y. and Huang, X. (2019). How to fine-tune bert for text classification?, *CCL*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. and Polosukhin, I. (2017). Attention is all you need.

Xiao, G., Chow, E., Chen, H., Mo, J., Guo, J. and Gong, Z. (2017). Chinese questions classification in the law domain, *2017 IEEE 14th International Conference on e-Business Engineering (ICEBE)* pp. 214–219.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A. J. and Hovy, E. H. (2016). Hierarchical attention networks for document classification, *HLT-NAACL*.

Zhang, D. and Lee, W. S. (2003). Question classification using support vector machines, *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR '03, ACM, New York, NY, USA, pp. 26–32.
URL: *http://doi.acm.org/10.1145/860435.860443*

Zhang, Y., Er, M. J., Venkatesan, R., Wang, N. and Pratama, M. (2016). Sentiment classification using comprehensive attention recurrent models, *2016 International Joint Conference on Neural Networks (IJCNN)* pp. 1562–1569.