Intelligent Automation & Soft Computing
DOI: 10.32604/iasc.2023.023277
Article

Tech Science Press

Detection of Toxic Content on Social Networking Platforms Using Fine Tuned ULMFiT Model

Hafsa Naveed[1], Abid Sohail[2], Jasni Mohamad Zain[3,*], Noman Saleem[4], Rao Faizan Ali[5] and Shahid Anwar[6]

[1]Department of Software Engineering, Faculty of Science, University of Lahore, Pakistan
[2]Department of Computer Science, COMSATS University Islamabad, Lahore Campus, Pakistan
[3]Institute for Big Data Analytics and Artificial Intelligence (IBDAAI), Kompleks Al-Khawarizmi, Universiti Teknologi MARA, 40450, Shah Alam, Selangor, Malaysia
[4]TechnoGenics SMC PVT LTD, Lahore, Pakistan
[5]Department of Computer and Information Science, Universiti Teknologi PETRONAS, Bandar Seri Iskandar, Tronoh, Perak, Malaysia
[6]Department of Information Engineering Technology, National Skills University Islamabad, Sector H-8/1, Faiz Ahmed Faiz Road, Islamabad, Pakistan
*Corresponding Author: Jasni Mohammad Zain. Email: jasni67@uitm.edu.my
Received: 01 September 2021; Accepted: 19 January 2022

Abstract: Question and answer websites such as Quora, Stack Overflow, Yahoo Answers and Answer Bag are used by professionals. Multiple users post questions on these websites to get the answers from domain specific professionals. These websites are multilingual meaning they are available in many different languages. Current problem for these types of websites is to handle meaningless and irrelevant content. In this paper we have worked on the Quora insincere questions (questions which are based on false assumptions or questions which are trying to make a statement rather than seeking for helpful answers) dataset in order to identify user insincere questions, so that Quora can eliminate those questions from their platform and ultimately improve the communication among users over the platform. Previously, a research was carried out with recurrent neural network and pretrained glove word embeddings, that achieved the F1 score of 0.69. The proposed study has used a pre-trained ULMFiT model. This model has outperformed the previous model with an F1 score of 0.91, which is much higher than the previous studies.

Keywords: Machine learning; text mining; quora mining; artificial intelligence; natural language processing

1 Introduction

Social media and other blogging sites such as Quora, Twitter, Yahoo Answers, Facebook, and Answerbag have gained so much popularity that they became a necessity for modern users [1–3]. However, Quora evolved into one of the most popular platforms for questioning and answering, with 190 million users and

millions of questions being answered in a timeframe of a month [1,2,4]. With a massive number of users and their queries, the user's safety and security should not be compromised, which can be avoided by filtering inappropriate and unrelated questions. With these types of questions, users can lose interest for the platform [5,6]. Therefore, sincere (meaningful) questions can facilitate users who are searching for their queries. NLP (Natural language processing) can process and understand the communication on the internet and able to extract people opinions by utilizing the pervasive language that people exchange with each other to communicate. Technically speaking NLP's main target is to peruse, interpret, comprehend, and understand human linguistic features significantly to help extract the desired functionality. When text is provided to the computer, the computer process the text and remove unnecessary information related to each sentence and gather the essential information from them [3–7]. The most important part of information and technological advancement is to find out the opinions of other people. With the rise of importance in extracting the individual's opinion, Sentimental Analysis or Opinion mining, serving now and again, is one of the fundamental zones of computational examinations that manage to feel focused on common language handling that people used to exchange words to communicate with each other. This paper highlights its focus on recognizing insincere questions, we have finetuned ULMFit model on the Quora Insincere Questions and our model classifies questions as sincere or insincere. Literature review is presented in Section 2. The unified framework for Quora is presented in Section 3. along with our proposed approach. Section 4 presents the details of our evaluation, including evaluation measures, and experimental settings. The paper concludes in Section 5 and future work is presented in Section 6.

2 Literature Review

This section has the summaries of the papers we have reviewed for our research along with the reference numbers in square brackets. The first paper we reviewed, determines the attitude of questions and answers to improve the QA in online news forums and discussions on multiple platforms. It has implemented sentence-level attitude classification by focusing on two types of attitudes, one is sentiments, and the other one is arguments, with the help of SVM and RB (rule-based) classifiers [8,9]. Next we have a content-based ranking method to analyze user comments with a lexicon-based approach to set the ranking of Facebook fan pages [10]. It is an automatic questions classification method with questions solely based on facts with the help of SVM. A method for extraction of sentiments on movie user reviews in the presence of the opinion mining domain and the Naïve Bayes classifier [11]. It's been observed that neutral sentiments for tweets are relatively high, showing current work limitations [12]. In another design of sentimental analysis, a large number of tweets are presented. The prototype method is used in this development. A customer perspective via tweets determines sentiments such as positive sentiment towards the tweet or negative sentiment. It is represented in Pie Chart format or in a web page form [13–16].

Numerous researches have been done on the sentimental analysis of social media and microblogging sites like Twitter, Facebook, and Tumblr, etc. [17]. A unique approach to automatically classify the Twitter sentiment, i.e., positive or neutral, concerning user query or tweet with a machine learning approach is used with distant supervision A system for Twitter sentimental analysis on the presidential election of U.S 2012 with the help of Naïve Bayes classifier predicting the user positive, negative, and neutral tweets for the electoral process and getting the clear picture of online political sentiments [18,19]. The new specified version of analysis called entity level, in which a text, product review, news, etc., are analyzed at the document level to predict which tweet's sentiment belongs to which document [20]. The sentimental analysis comes under the umbrella of natural language processing, a field of data science. Many machine learning and deep learning algorithms are used to see whether the content is positive, negative or neutral. Machine learning algorithms such as SVM, Naïve Bayes, Logistic Regression, Random Forest, K-nearest neighbors and Maximum Entropy method are used in the sentimental analysis of Twitter and Quora insincere questions problem [9,21,22]. These machine learning algorithms with

multiple feature extraction techniques such as TF-IDF, BOW, Unigram, Bigram and word2vector achieved the maximum possible efficiencies but deep learning algorithms i.e., RNN, LSTM, BiLSTM, CNN, produced better benchmark results. To eliminate insincere questions from Quora, many machine learning algorithms, neural networks are applied, and RNN with Glove+ paragram achieved a higher F1 score of 69% [23,24], which is quite low with such a large imbalanced dataset posted on Kaggle. There is a need for a better algorithmic technique to improve the results to filter the insincere questions [13,25].Another study about Quora has shown the results on a huge dataset giving astonishing predictions on tropically improved word embedding's [26]. Comparison of multiple state of the art deep learning models are presented to filter noxious statements on social media platforms [27]. BERT classifier is used to remove toxic comments from discussions on social media platforms such as twitter and the classifier is also trained on a comment dataset from kaggle [28]. The study is limited to filter noxious comment rather than all type of anti-social behavior on social media.

3 Proposed Framework for Classification of Quora Incincere Question

Given below is the proposed framework to filter sincere and insincere questions. In Fig. 1, the training and testing phase of the classifier is illustrated.

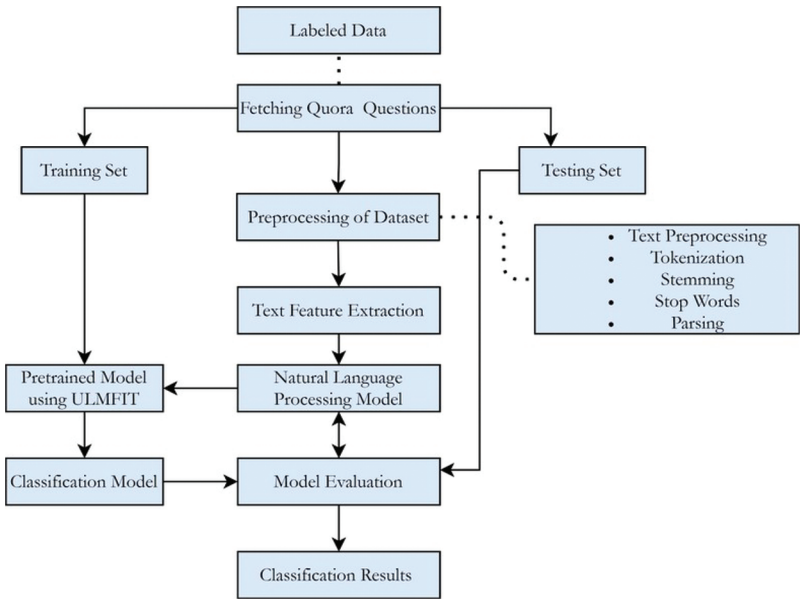


Figure 1: Proposed framework for quora insincere questions classification

After fetching the questions from the Quora dataset consisting of 1.44k questions, framework applies pre-processing on the complete dataset that includes converting the text into lower case and removal of non english text and stopwords, then the freamework splits the dataset into training and testing data. We have fine-tuned ULMFiT, a pre-trained language model trained on wikitext 103, with our training data. Universal Language Model Fine-tuning (ULMFiT) is applied because, its a powerful exchange learning strategy. It is an open source, pre-trained model that can be easily implemented. It can be applied to any new dataset in such a manner that it does not neglect and forget the previous learning iterations and makes it useful for the next sub-tasks. ULMFiT includes 3 significant stages. These stages are LM pre-training, LM fine-training, and Classifier fine-tuning.

The training dataset used in this project is taken from Kaggle, consisting of 1.3 m questions and 300k questions are utilized in the test data set. The training data includes the question asked and whether the questions being asked were identified as an insincere target = 1 or a sincere target = 0. The training dataset is quite large in size and highly unbalanced therefore it is sliced into 80000 sincere questions and 64770 as insincere questions. There is a total of 115816 questions taken as training data. The model evaluates and learn from training dataset. Also, we have taken 14770 questions as validation data, which is a type of data that will provide an unbiased judgment of the model along with tuning the hyper-parameters of the model. The model will occasionally be exposed to the validation data, but it will not recalibrate from the validation data. The high-level hypermeters would be updated from the validation dataset and could be further used for test data. Furthermore, we have 14770 questions taken as test data samples which gives the unbiased results of the final model already fit on both training and test datasets. The frequency of different questions in the dataset is shown in Tab. 1.

Table 1: Dataset distribution of quora questions

Total number of questions	Training	Testing	validation
Before splitting	1.3 m	300k	–
After splitting	115816	14770	14770
Sincere	800000	7966	8015
Insincere	64770	6511	6462

3.1 Data Preprocessing

Text pre-processing is preparing the text for experimentation, transforming it for additional handling. The advantage of pre-trained models is that, they are trained on the huge datasets that we might not create. The fast Artificial Intelligence library focused typically on working with pre-trained language models, and their fine-tuning. Data pre-processing flow is shown in Fig. 2.

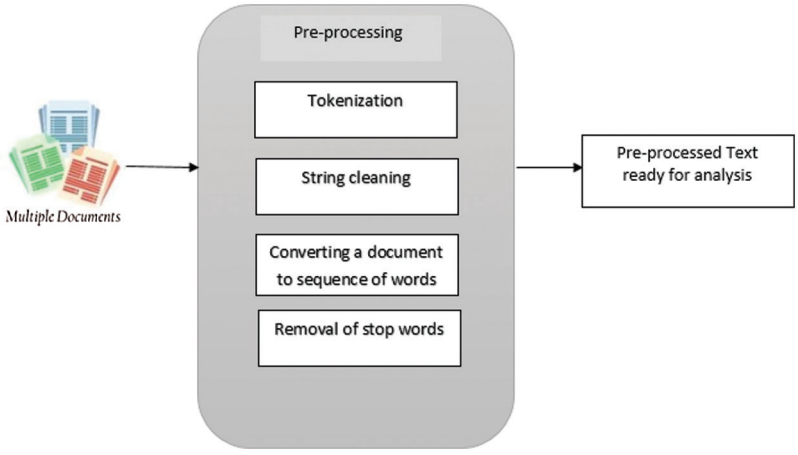


Figure 2: Data pre-processing

With the help of the FASTAI library, Researchers have done pre-processing of data in several steps:

1. Tokenization/string cleaning
2. Converting a document to the sequence of words
3. Removing stop words

3.2 Text Feature Classifier

The FASTAI library streamlines preparing quick, accurate and efficient neural nets utilizing present-day best practices. It takes questions text as input and predicts the sincere/insincere question into respective groups. The content module of the FASTAI library contains essential parameters to characterize a dataset, appropriate for the different NLP related tasks and rapidly create models. In Fig. 3 a complete framework of Fast AI library is shown to predict the results. In Tab. 2 a complete text transformation in FASTAI library is presented to see the word tokens.

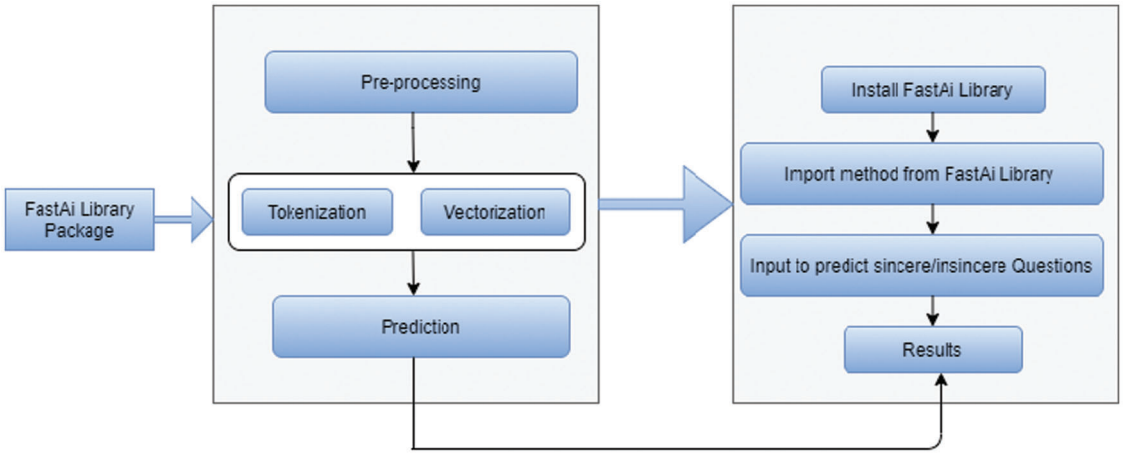


Figure 3: Framework of FastAI library

Table 2: Text transformation in fast AI library

Text	Target	Prediction
xxbos what is math overbrace sum xxunk 8 infty vec frac sum xxunk 7 infty overbrace 1x 0 text read carefully 3x 1 div 1x 5 sqrt 3 2x 3 1x 0 vec vec 3x 3 1x 2 sum dagger 9 infty vec boxed boxed 3x 1 3x 1 times 1x 5 div sin \ (boxed boxed vec 3x 5 sqrt 4 2x 4 vec 2x 3 div sin	insincere	Insincere
xxbos alright, i know people have the habit of asking for help with homework , but i 'm completely lost with this the problem is given s \ (x \ ) 4x 1 and 0 \ (x \ ) x 2 3 , find \ (s p \ ) \ (x \ ) can anyone explain the way to figure this out to me \	Sincere	Sincere
xxbos int f \ (int n \ ) static int r 0 if \ (n 0 \ ) return 1 if \ (n 3 \ ) r n return f \ (n 2 \ ) 2 return f \ (n 1 \ ) r what will be the value of f \ (5 \ ) when executed this code in c language \ ?	Sincere	Sincere
xxbos any object used to cause harm is an assault weapon \ (e.g., xxunk knife i stab you \ ) , so how do you ban all of them \ (cars , hammers , baseball bats \ ) \ ? or is this guy on about xxunk rifles and has no idea what i is talking about \ ? \ (rhetorical question \ )	insincere	Insincere

3.3 Classification of Quora Questions

The Fig. 4 below demonstrates steps piloting to accomplish our targets. At first, the dataset was assembled through Kaggle which comprises of 1.3 million inquiries, because of thoroughly disproportioned dataset, it is splitted into training, validation and test data. After splitting, the data is then cleansed using the pre-trained model techniques. In the ULMFIT model, the data is trained in batches to see if it's being trained precisely. Its accuracy is checked on both validation data and test data to authentically predict that either the question is genuine or deceitful. During the training of batches, some hypermeters are used to avoid over fitting or under fitting of data. Some of the hypermeters that are used in this study is learning rate and Dropout value. Learning rate is a hyperparameter that manages how much change is required in the model regarding the approximated error every time the weights are upgraded. Dropout value is a technique that sets the hidden neuron layer value to 0 after each iteration. If the model is over-fitting, we will increase our Dropout value and decrease it for under-fitting. In this project, the dropout value is set to 0.65 to control over-fitting. This value best fits the model as it gives accurate results on the test data set.

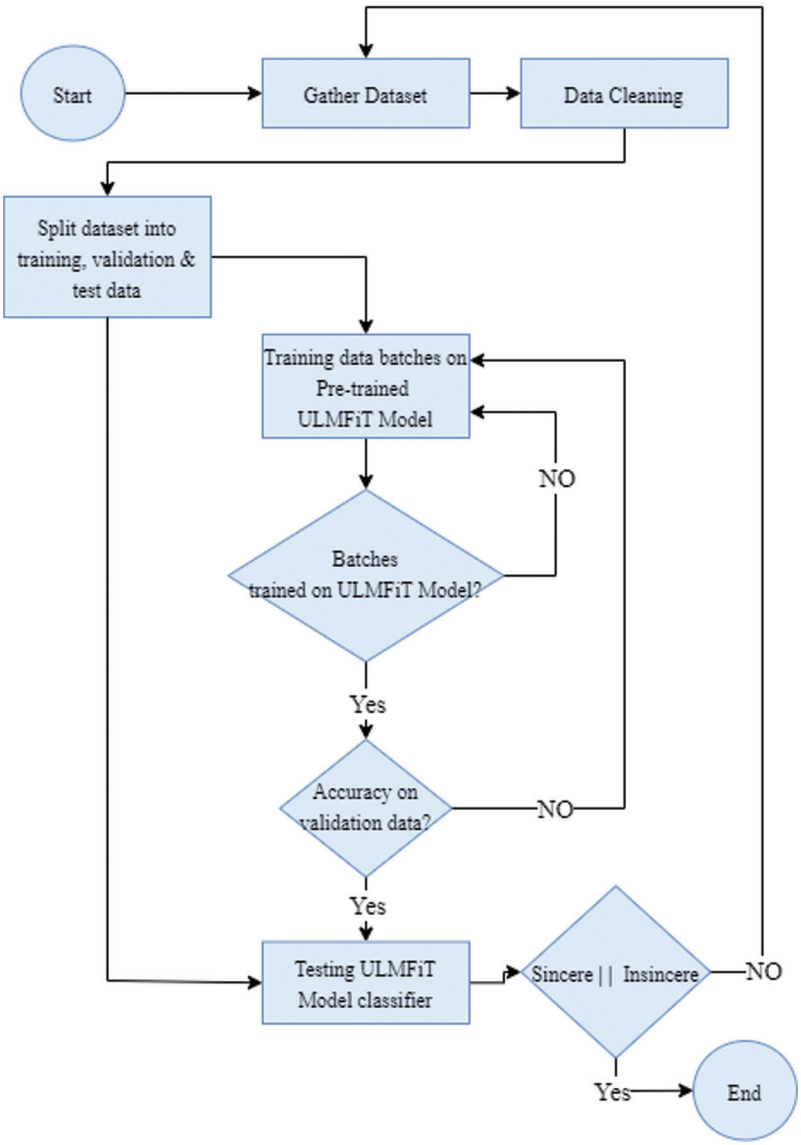


Figure 4: Workflow of quora questions classifier

3.4 Create a Language Model

FASTAI library, with its great deal in NLP, provides Wikitext Dataset, comprising of a pre-processed subset of 103 million tokens separated from Wikipedia. The type of a model that comprehends a lot of information and knowledge about language and what it actually depicts in computational terms. Fig. 5 shows the subsequent stages. In the beginning, a pre-trained language model is chosen on a large dataset, i.e., wikitext. It is then used to fine-tune the language model on unlabeled data with the help of novel techniques. In the last step the language model is fine-tuned on a labeled data classifier as shown in Fig. 5.

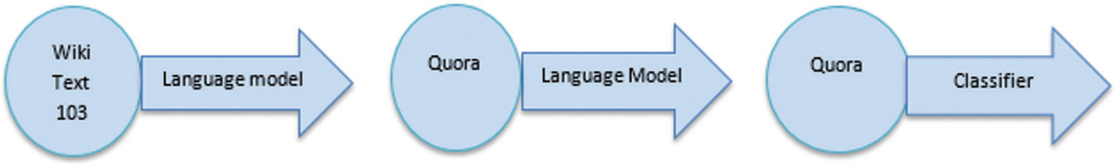


Figure 5: Quora language model

A hyper-parametric value is also used to make a unified Language model fine-tuned with its special characteristic values at the beginning of the training. A pre-trained model ULMFiT is used with its weights and then fine-tuned on the Quora questions. A learner is created by passing it into two stages: The actual data in our language model, i.e., data_lm. A pre-trained model, which is the Wikitext 103 dataset model that is downloaded from a library known as FASTAI.

4 Results and Discussions

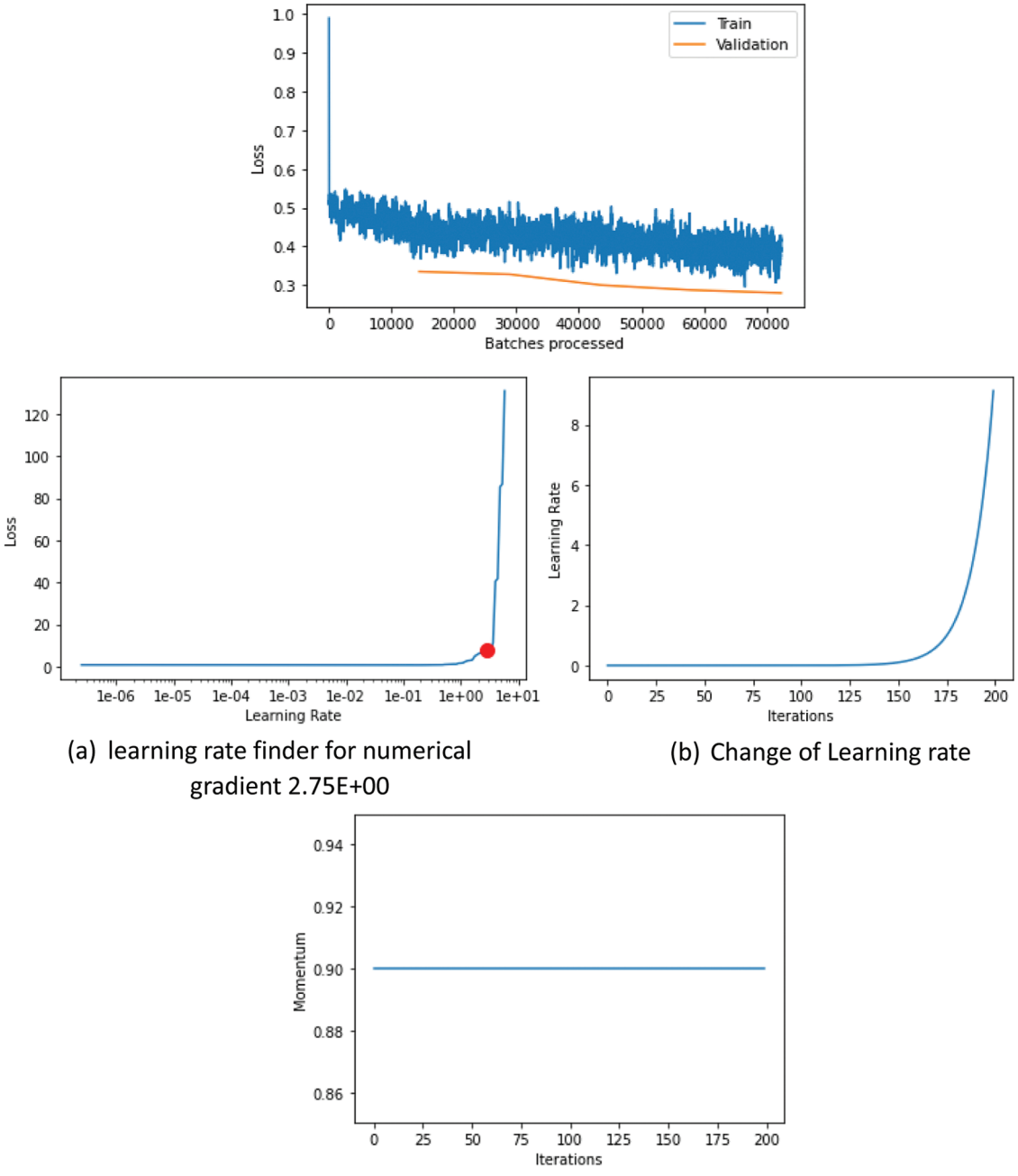
4.1 Fine Tuning with Quora

With its great value, the learning rate hyper-parameter shows significant parametric values to train and instruct any type of model. FASTAI gives a helpful utility learn_lr finder to look among the various available learning rates to track down the perfect one available. The learning rate discoverer (learn_lr) will update the learning rate after every short time interval. In the end, the learning rate will touch its peak. Presently take a view at the plot section of the learning rate against the expected loss and decided the absolute bottom (around 1e+0 for the plot beneath) and return by one immensity and pick that as a learning rate (something around 1e-1). To find an adequate learning rate at a peak LR, we have taken all the LRs in between 0.000001 and 1. With learn.recorder.plot(), the learning rate can be depicted against loss. With this plot function's support, we can determine how huge that LR can expand so that the descent of the loss is still enormous. We have set the peak LR to the highest LR in a space where the loss has its sharpest drop, which is approximately 0.01 in our case. Fig. 6a shows the learning rate finder for a numerical gradient known as the stochastic gradient descent method. It approximates the error gradient for the current prototype using specimens from the training and validation dataset, that upgrades the weights of the prototype using the back propagation method, which in turn identifies errors. Similarly, the number of times weights are upgraded during data training is known as the learning rate.it is used as customizable hypermeter value ranges between 0.0, and 1.0 in training neural networks. Fig. 6b shows the change in learning rate with respect to iterations numbers. This change of learning rate is called a slanted triangular learning rate that does not remain persistent throughout the model fine-tuning mechanism. For some iterations, it increases slowly, and then it can go high or low within a span of time, giving the state-of-the-art performance. Fig. 6c shows the learning rate momentum, which is set to 0.8 or 0.7 in our model. The cycle momentum starts with the maximum value and then decreases to 0.8 or 0.86 with the rise in learning rates. The best value of momentum chosen throughout the training sessions can help us achieve better results.

Min numerical gradient : $2.75E+00$

Min loss divided by 10 : $1.58E-02$

Fig. 6 shows the super convergence mechanism in which neural networks can be trained swiftly as compared to traditional training methods. The above figure shows that training is completed in 70000 iterations with a constant learning rate of 0.1, and it shows the training loss and validation loss for learning rate ranges. To track the loss function over epochs the learn.recorder.plot_losses () function is used.



(a) learning rate finder for numerical gradient 2.75E+00

(b) Change of Learning rate

(c) learning rate momentum

Figure 6: Batch loss over epochs

In Tab. 3 we started training the model with a learning rate recorder to minimum gradient learning using 4.79E+00 using fit_one_cycle and 30 epochs at a time. The training dataset is used to fit the model. The model will see the data and learn from it. Multiple set of iterations are shown in different tables to see that how accuracy reaches to its maximum level while going through the different passes.

Table 3: First set of iterations

Epoch	Train_loss	Valid_loss	Accuracy	Time
0	4.432514	4.115708	0.325435	01:10
1	4.165674	3.944358	0.337418	01:10
2	4.079137	3.934625	0.336518	01:10
3	4.079288	3.988007	0.333036	01:10
4	4.120203	4.064188	0.329944	01:10
5	4.174441	4.126517	0.326176	01:10
6	4.220677	4.184680	0.321964	01:10
7	4.250630	4.221560	0.324182	01:10
8	4.277025	4.245505	0.321953	01:10
9	4.241834	4.246242	0.321633	01:10
10	4.225307	4.250954	0.322879	01:10
11	4.216391	4.240699	0.323385	01:10
12	4.188015	4.236150	0.323527	01:10
13	4.144524	4.223304	0.324289	01:10
14	4.116143	4.205905	0.324773	01:10
15	4.072273	4.200931	0.325353	01:10
16	4.019020	4.174726	0.326369	01:10
17	3.964488	4.162907	0.328226	01:10
18	3.891470	4.151286	0.329408	01:10
19	3.860048	4.128718	0.330484	01:10
20	3.779978	4.115340	0.331953	01:10
21	3.729634	4.094161	0.333028	01:10
22	3.661896	4.082439	0.333869	01:10
23	3.602297	4.070265	0.334635	01:10
24	3.552823	4.061080	0.336231	01:10
25	3.496910	4.051255	0.336432	01:10
26	3.458700	4.045945	0.337418	01:10
27	3.419256	4.043776	0.337697	01:10
28	3.403183	4.043035	0.337638	01:10
29	3.382870	4.043025	0.337612	01:10

The series 0 to 29 in Tab. 3 shows the number of epochs, and it can be seen that after going through multiple epochs, the accuracy is around 33% in the beginning. The model recalibrated only the last layers while leaving the rest of the model as it is. We require to train the whole model, so we started unfreezing the whole model after tuning the last layers.

Tabs. 4 and 5 show the accuracy of around 89%, which is quite relatively high with a learning rate of 5e-3/2, 5e-3 and moms = (0.8, 0.7) in fit one cycle, which is the best approach as well as faster being compared to the other available approaches. In Tab. 6 set of iterations, accuracy reaches the maximum level of more than 90%.

Table 4: Second set of iterations

Epoch	Train_loss	Valid_loss	Accuracy	Time
0	45.553246	57.596161	0.666782	**03:20**
1	38.299252	4.597684	0.765283	**03:20**
2	23.187498	21.841812	0.728604	**03:21**
3	2.960057	3.963846	0.559439	**03:22**
4	0.477815	0.390162	0.845341	03:22

Table 5: Third set of iterations

Epoch	Train_loss	Valid_loss	Accuracy	Time
0	0.462717	0.335325	0.873178	**04:04**
1	0.453356	0.328290	0.881882	**04:05**
2	0.428039	0.300610	0.888306	**04:03**
3	0.403834	0.287824	0.894177	**04:12**
4	0.395884	0.280136	0.893970	**04:12**

Table 6: Fourth set of iterations

Epoch	Train_loss	Valid_loss	Accuracy	Time
0	0.369418	0.277097	0.895628	**06:38**
1	0.390243	0.268706	0.902950	**06:36**
2	0.315766	0.262049	0.906403	**06:34**
3	0.284710	0.267073	0.907716	**06:33**
4	0.301545	0.256997	0.907508	**06:32**

4.2 Confusion Matrix

In Figs. 7 and 8, the confusion matrix shows the TP TN, FP and FN values. The TP true positive shows the number of sincere questions which are predicted correctly. As shown in Tab. 7 of the confusion matrix,

the actual questions in the data that were insincere and are predicted as insincere by the algorithm (True Negatives) are 5868. Similarly, the questions were insincere and are predicted as sincere by the algorithm (False Positives) are 761. In addition to this, the questions that were actually sincere and are categorized as sincere by the algorithm (True Positives) are 7254, and those which are categorized as sincere and marked as insincere by the algorithm (False Negatives) are 594.

	Predicted 0	Predicted 1
Actual 0	TN= 5868	FP=761
Actual 1	FN=594	TP=7254

Figure 7: Confusion matrix

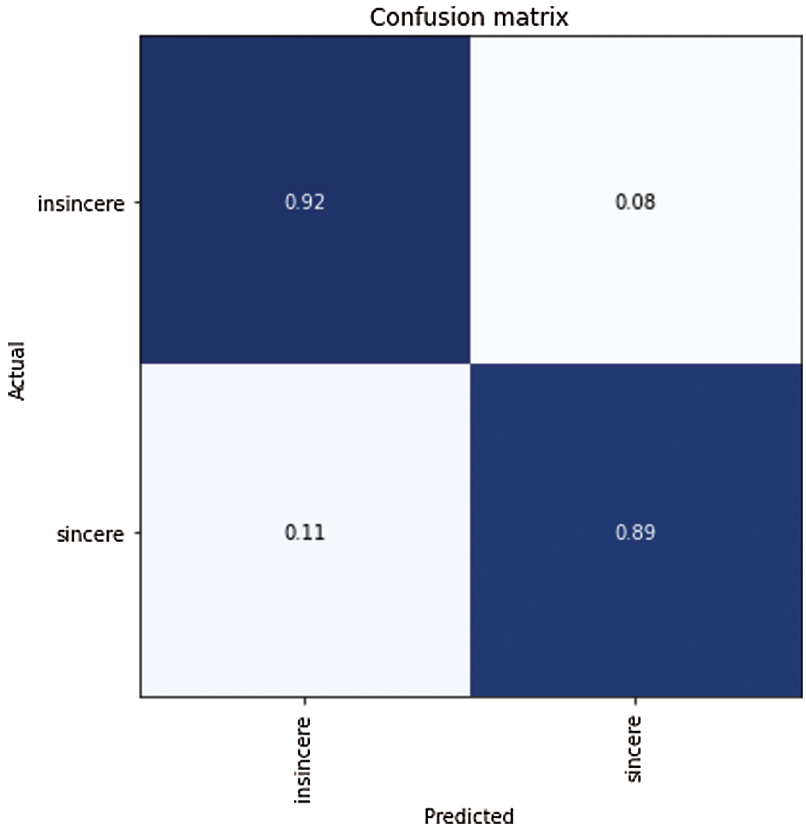


Figure 8: Confusion matrix with classified and misclassified labels

4.3 Performance Metrics

Methods are evaluated based on the following four accuracy measures. Performance parameters such as accuracy, sensitivity, specificity, and AUC are calculated for authentication of the proposed technique. The optical and parametric results of the offered technique are associated with the current works. The

performance parameters of the offered technique are computed as follows: Performance metrics, such as accuracy, area under curve (AUC), sensitivity and specificity are used for validation of the procedure. Performance metrics for the procedure are determined as follows.

Table 7: Classification report for validation data

	Precision	Recall	F1 score	Support
Insincere	0.89	0.91	0.90	6462
Sincere	0.92	0.91	0.91	8015
Accuracy			0.91	14477
Macro Average	0.90	0.91	0.91	14477

4.3.1 Accuracy

In simple words, accuracy is the quality or condition of being correct, true, or precise. The mathematical representation of accuracy is given in Eq. (1). In our case it is a number of total truly predicted insincere and sincere questions:

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{1}$$

4.3.2 Sensitivity

Sensitivity is the capability of an examination to correctly recognize insincere and sincere questions. The mathematical representation of accuracy is given in Eq. (2).

$$Senstivity = \frac{TP}{TP + FN} \tag{2}$$

4.3.3 Specificity

Specificity is the capability of the examination to correctly recognize insincere and sincere questions. The mathematical representation of accuracy is given in Eq. (3).

$$Specificity = \frac{TN}{TN + FP} \tag{3}$$

False-Negative (FN): The feature extracted result is 0 and predictive power is not pre- sent.

True-Negative (TN): The feature extracted result is 1 and predictive power is absent.

False-positive (FP): The feature extracted result is 0 and predictive power is present.

True-Positive (TP): The feature extracted result is 1 and predictive power is present.

Precision Score: Precision score is the ratio of predicting the labels as positive or correct. In our case it is a value of c(m) of calculating the anomaly score as true positive. The formula is shown in Eq. (4).

$$Preceision = \frac{TP}{TP + FP} \tag{4}$$

F1 Score: F1 score is the harmonic mean between precision and recall score and can be calculated as follows using Eq. (5):

$$F\ score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{5}$$

Recall Score: The ratio of correctly predicted anomaly where it exists it can be calculated using Eq. (6):

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

4.4 Predictions with Language Model

The validation data gives an unbiased analysis of the model and regulate the model's hyper-parameters. The higher level hyper-parameters of the model will be updated using the validation set results. So now, the validation set indirectly affects the model. It is also called the Dev Set because it helps during the development stage of the model. Predictions and classification report on a validation dataset showing the different values for accuracy, recall, f1 score, and sensitivity, specificity, and MCC value.

4.5 Classification Report for Test Data

The test dataset also gives the unbiased assessment of the final model obtained from both the training dataset and the validation dataset. It generally provides the gold standard that is used to identify the model. In Tabs. 8 and 9 below is the classification of test data and validation data showing the different values for accuracy, recall, f1 score, sensitivity, specificity and MCC values.

Table 8: Classification report for test data

	Precision	Recall	F1 Score	Support
Insincere	0.88	0.91	0.90	**6511**
Sincere	0.92	0.90	0.91	**7966**
Accuracy			0.91	**14477**
Macro Average	**0.90**	**0.91**	**0.91**	**14477**

Table 9: Evaluation parameters

Evaluation metrics	Sensitivity	Specificity	Accuracy	MCC Value
Test Data	0.88	0.93	0.91	**0.807**
Validation Data	**0.887**	**0.9231**	**0.906**	**0.814**

4.6 Comparative Analysis of All Evaluation Measures against All Techniques

In Tab. 10, a comparative study is conducted that shows the precision of multiple machine learning and deep learning algorithms and its feature encoding techniques used in previous research. It can be seen that the ULMFiT model produced the highest F1 score of 0.91.

Table 10: Comparative analysis

Model	Encoder	F1 score
Naïve Bayes Network	Tf-idf	0.52939
Logistic Regression	Tf-idf	0.61593
RNN	Tf-idf	0.62816
RNN	Word embedding	0.67219
Decision Tree	Count vectorizer + Lancaster stemmer	0.46
Random Forest	Count vectorizer + lemmatization	0.41
RNN	Glove + paragram	0.69
ULMFiT	Pre-trained	0.91

Quora is a platform that promotes socialism among people by sharing knowledge on a common platform, and people tend to gain information from each other. Different members ask multiple questions of the community that is looking for some helpful answers. A key challenge for us was to filter out the insincere questions that are merely a statement rather than looking for any helpful material or a question founded upon false arguments. The challenge was addressed earlier by multiple people using many machine learning and deep learning methodologies such as Naïve Bayes, Logistic Regression, SVM, Random Forest and some deep learning frameworks i.e., RNN, LSTM, BiLSTM are implemented to achieve the maximum possible efficiencies. Such as with Naïve Bayes Model, the F1 score was 0.52, and with logistic regression, the F1 score is 0.61. The highest F1 score achieved so far is 69%, with RNN, and glove + program as a feature extraction technique. We aim to ascend the efficiency of the model by using the pre-trained language model ULMFiT as it is the state-of-the-art technique that includes transfer learning, handful optimization and regularization techniques and the model is already trained on wikitext 103, and we fine-tuned it on our dataset.

5 Conclusions

The F1 score obtained from the pre-trained language model ULMFiT has out-performed other models on Quora dataset, giving a F1 score of 0.91. For the classification task, dataset is splitted into training, validation and test data, after that, necessary data preprocessing and cleaning such as tokenization, string cleaning, converting a document to some sequence of words, and removing stop words was performed. Then, a pre-trained language model ULMFiT which is already trained on the WikiText dictionary was fine-tuned on the training data. Language model with the help of some hyper-parameters such as learning rate parameter and dropout technique controlled over fitting and under fitting. A dropout value of 0.65 is used to control the over-fitting of the model. Finally, a multi-layer neural network was used as a to achieve the desired results. The 0.88 sensitivity value for test data shows the measure of proportion of actual positive cases that got predicted as positive and the value 0.93 for specificity shows the measure of proportion of actual negative cases that got predicted as negative. The 0.807 MCC value for test data shows the efficiency of the model taking all the confusion matrix parameters into consideration.

6 Future Work

The major challenge we have faced in our study, was to deal with the highly imbalanced dataset. In future, we can make the categories of the questions to understand the questions ambiguity. We can also

take datasets from other question answering websites and compare those using different SOTA models. The dataset should also be large enough to work on all the possible instances of the questions. To better achieve the efficiency of the proposed framework, we can use some other pre-trained language models such as the BERT and OpenAI's GPT-2 and achieve better performance.

Funding Statement: This work was supported by the Institute for Big Data Analytics and Artificial Intelligence (IBDAAI), Universtiti Teknologi Mara, Shah Alam, Selangor. Malaysia.

Conflicts of Interest: The authors declare that they have no conflicts of interest in reporting regarding the present study.

References

[1] N. Ansari and R. Sharma, "Identifying semantically duplicate questions using data science approach: A quora case study," arXiv preprint arXiv:2004.11694, 2004.

[2] P. S. Nishant, B. G. K. Mohan, B. S. Chandra, Y. Lokesh, G. Devaraju *et al.,* "Lexicon-based text analysis for twitter and quora," in *Int. Conf. on Innovative Data Communication Technologies and Application*, Cham, Springer, pp. 276–283, 2019.

[3] S. Patil and K. Lee, "Detecting experts on Quora: by their activity, quality of answers, linguistic characteristics and temporal behaviors," *Social Network Analysis and Mining*, vol. 6, no. 1, pp. 5, 2016.

[4] L. Sharma, L. Graesser, N. Nangia and U. Evci, "Natural language understanding with the quora question pairs dataset," arXiv preprint arXiv: 1907.01041, 2019.

[5] D. Yogeshwaran and N. Yuvaraj, "Text classification using recurrent neural network in quora," *International Research Journal of Engineering and Technology (IRJET)*, vol. 6, no. 2, pp. 638–642, 2019.

[6] A. Mungekar, N. Parab, P. Nima and S. Pereira, *Quora Insincere Question Classification*, National College of Ireland, 2019, accessed 2 Jan 2022, https://www.researchgate.net/profile/Prateek-Nima-2/publication/334549103_Quora_Insincere_Questions_Classification/links/5d30d8fb92851cf440900e16/Quora-Insincere-Questions-Classification.pdf.

[7] E. Dadashov, S. Sakshuwong and K. Yu, *Quora Question Duplication*, Stanford University, 2017. accessed 2 Jan 2022, https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1174/reports/2761178.pdf.

[8] M. A. Al-Ramahi and I. Alsmadi, "Using data analytics to filter insincere posts from online social networks. A case study: Quora insincere questions," in *Proceedings of the 53rd Hawaii Int. Conf. on System Sciences*, pp. 2489–2497, 2020.

[9] S. Somasundaran, T. Wilson, J. Wiebe and V. Stoyanov, "QA with attitude: Exploiting opinion type analysis for improving question answering in on-line discussions and the news," in *Int. Conf. on Weblogs and Social Media*, Boulder, USA, pp. 1–8, 2007.

[10] P. T. Ngoc and M. Yoo, "The lexicon-based sentiment analysis for fan page ranking in Facebook," in *The Int. Conf. on Information Networking 2014 (ICOIN2014)*, Phuket, Thailand, pp. 444–448, 2014.

[11] I. Smeureanu and C. Bucur, "Applying supervised opinion mining techniques on online user reviews," *Informatica Economica*, vol. 16, no. 2, pp. 81, 2012.

[12] A. Agarwal, B. Xie, I. Vovsha, O. Rambow and R. J. Passonneau, "Sentiment analysis of twitter data," in *Proc. of the Workshop on Language in Social Media (LSM 2011)*, Portland, Oregon, pp. 30–38, 2011.

[13] O. Kucuktunc, B. B. Cambazoglu, I. Weber and H. Ferhatosmanoglu, "A large-scale sentiment analysis for Yahoo! answers," in *Proc. of the Fifth ACM Int. Conf. on Web Search and Data Mining*, Seattle, WA, USA, pp. 633–642, 2012.

[14] R. Parikh and M. Movassate, "Sentiment analysis of user-generated twitter updates using various classification techniques," Stanford University, 2009, accessed 2 Jan 2022, https://nlp.stanford.edu/courses/cs224n/2009/fp/19.pdf.

[15] Y. Li, Q. Pan, T. Yang, S. Wang, J. Tang *et al.,* "Learning word representations for sentiment analysis," *Cognitive Computation*, vol. 9, no. 6, pp. 843–851, 2017.

[16] X. Fang and J. Zhan, "Sentiment analysis using product review data," *Journal of Big Data*, vol. 2, no. 1, pp. 1–14, 2015.

[17] P. Liu, X. Qiu and X. Huang, "Recurrent neural network for text classification with multi-task learning," arXiv preprint arXiv:1605.05101, 2016.

[18] H. Wang, D. Can, A. Kazemzadeh, F. Bar and S. Narayanan, "A system for real-time twitter sentiment analysis of 2012 us presidential election cycle," in *Proc. of the ACL, 2012 system demonstrations*, Jeju Island, Korea, pp. 115–120, 2012.

[19] G. A. Buntoro, "Sentiments Analysis for Governor of East Java 2018 in Twitter," *Sinkron: Jurnal dan Penelitian Teknik Informatika*, vol. 3, no. 2, pp. 49–55, 2019.

[20] L. Zhang, R. Ghosh, M. Dekhil, M. Hsu and B. Liu, "Combining lexicon-based and learning-based methods for Twitter sentiment analysis," *HP Laboratories, Technical Report*, 2011, accessed 2 Jan 2022, https://www.hpl.hp.com/techreports/2011/HPL-2011-89.pdf .

[21] A. Go, R. Bhayani and L. Huang, "Twitter sentiment classification using distant supervision," *Stanford University*, 2009, accessed 2 Jan 2022, https://wwwcs.stanford.edu/people/alecmgo/papers/TwitterDistantSupervision09.pdf.

[22] T. Niu, S. Zhu, L. Pang and A. El Saddik, "Sentiment analysis on multi-view social data," in *Int. Conf. on Multimedia Modeling*, Miami FL USA, pp. 15–27, 2016.

[23] B. Gaire, B. Rijal, D. Gautam, S. Sharma and N. Lamichhane, "Insincere question classification using deep learning," *International Journal of Scientific & Engineering Research*, vol. 10, no. 7, pp. 2001–2004, 2019.

[24] V. Mujadia, P. Mishra and D. M. Sharma, "Classification of Insincere Questions with ML and Neural Approaches," in *FIRE (Working Notes)*. Kolkata, India, 451–455, 2019.

[25] Y. Xu, L. Mou, G. Li, Y. Chen, H. Peng *et al.,* "Classifying relations via long short term memory networks along shortest dependency paths," in *Proc. of the 2015 Conf. on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, pp. 1785–1794, 2015.

[26] D. Y. Kim, X. Li, S. Wang, Y. Zhuo and R. K. W. Lee, "Topic enhanced word embedding for toxic content detection in Q&A sites," in *Proc. of the 2019 IEEE/ACM Int. Conf. on Advances in Social Networks Analysis and Mining*, Vancouver, BC, Canada, pp. 1064–1071, 2019.

[27] V. Maslej-Krešňáková, M. Sarnovský, P. Butka and K. Machová, "Comparison of deep learning models and various text pre-processing techniques for the toxic comments classification," *Applied Sciences*, vol. 10, no. 23, pp. 8631, 2020.

[28] H. Fan, W. Du, A. Dahou, A. A. Ewees, D. Yousri *et al.,* "Social media toxicity classification using deep learning: Real-world application UK Brexit," *Electronics*, vol. 10, no. 11, pp. 1332, 2021.