Classifying the Insincere Questions using Transfer Learning

MSc Research Project
Data Analytics

Shriya Gandhi
Student ID: x19218079

School of Computing
National College of Ireland

Supervisor: Jorge Basilio

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Shriya Gandhi
Student ID:	x19218079
Programme:	Data Analytics
Year:	2021
Module:	MSc Research Project
Supervisor:	Jorge Basilio
Submission Due Date:	16-08-2021
Project Title:	Classifying the Insincere Questions using Transfer Learning
Word Count:	6147
Page Count:	22

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:	

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):	

Classifying the Insincere Questions using Transfer Learning

Shriya Gandhi
x19218079

Abstract

Hate speech and insincere content on social media and online communication forums are digitalized forms of personal attacks. Such content if left unattended, tamper the decorum of the forum and lead to a lack of trust by the users. Manual screening of content posted online is tedious and psychologically harmful for the people reviewing these posts. Developing a robust and scalable model to detect such content automatically is a pressing priority. This research project proposes using pre-trained language representation model based on transformer architecture to identify the insincere questions posted on Quora. The dataset for research work is extracted from the Kaggle data repository. To limit the use of high computational power, which is otherwise required for NLP problems, we have created three samples of data and trained the transformer-based BERT and XLNET models. Due to high imbalance in the dataset, macro f1-score is considered as the metric for model performance evaluation. The results show that both BERT and XLNET outperform the baseline model, logistic regression. Amongst BERT and XLNET, the XLNET model achieves a higher macro-f1 score and weighted f1-score of 0.84 and 0.96, respectively.

1 Introduction

1.1 Background & Motivation

The advent of internet and development of IT technologies has led to a lot of people spending a considerable amount of time online. The presence of social media and Question & Answering (Q&A) forums encourages people to connect and share knowledge. The social media platforms allow users to render individual opinions in a way to develop a constructive and safe ecosystem for everyone. Through these QA portals, users can post new questions and get a quick response. The questions posted are diverse and range from personal, professional to purely anything. While these QA forums are originally meant for knowledge-sharing, some people misuse these platforms to harass others by posting derogatory and harmful content. Thus, leading to bad user engagement for audiences genuinely seeking information through these platforms.

Quora is one such popular QA forum that has facilitated collaborative learning with around 300 million active users monthly (Paul et al.; 2012). Similar to any social or QA forum, Quora too has content ranging from high-quality to low-quality. The presence of insincere content on Quora puts the platform and the users at risk. With the massive volume of data, manually removing the insincere questions is challenging and psychologically harmful. Therefore, the need to develop an automated system to filter the incoming data before publishing arises.

Previously Quora employed the use of manual review and machine learning algorithms to filter the incoming data. With increased user base and complexity, the need for a more scalable and robust system for identifying the existing insincere questions along with filtering new ones surfaced. Therefore, the Quora team published a labeled dataset with questions and associated labels as sincere or insincere. The definition for labeling a question as insincere is outlined as well. The questions based on false assumptions, encourage rage, sexual content, or questions posted to target a particular audience are labeled as insincere questions.

The literature around toxic content identification show use of machine learning and deep learning models. The ML models such as random forest, logistic regression, SVM (Ranganathan et al.; 2019), KNN (Kajla et al.; 2020) (Mungekar et al.; 2019) are implemented with various feature extraction techniques. The deep learning models implemented include bi-directional LSTM, GRU (Sampath; 2019)(Mediratta and Oswal; 2019), MLCNN (Roy; 2020), and MLP (Priyambowo and Adriani; 2019). The recent developments of the pre-trained language representation models have proven to remember more profound context than any ML or DL models. These pre-trained models enable transfer learning and do not require any feature engineering or large corpus of data. In this research work, we will explore the pre-trained models BERT and XLNET for insincere question classification task with limited computational power.

1.2 Research Question & Objective

1.2.1 Research Question

1. How far can the classification of insincere questions on Quora be improved by leveraging pre-trained language representation models?

2. Can a transformer-based model produce competitive results with a smaller dataset size and limited computational power?

1.2.2 Research Objective & Contribution

The research around the identification of hate speech and toxic comments on online forums has geared only recently. Thus, there is a scope of building more robust and scalable models for the task at hand by using models that have proven to produce state-of-the-art results in the NLP domain. The primary contribution of this project is the implementation of BERT and XLNET transformer models for the insincere question classification task with different samples of data (10%, 50% and 70%).

Below are the objectives outlined to answer the research question poised above –

1. Critical review of studies performed for insincere content identification.
2. Data pre-processing to meet the input parameter requirement of the BERT and XLNET models.
3. Implementation of BERT model for text classification task.
4. Implementation of XLNET model for text classification task.
5. Evaluation of the performance of pre-trained models trained on samples of data to minimize the usage of computational power.
6. Comparison of the BERT and XLNET model with f1-score as the metric.
7. Comparison of trained models with a baseline model.

1.3 Roadmap

The remaining document is structured as follows. A brief review of literature is discussed in the second section. The project methodology and design flow diagram are discussed in the third and fourth section, respectively. The project implementation, result evaluation, discussion and conclusion are discussed in the subsequent section.

2 Related Work

In this section, we will discuss the different techniques explored by researchers for identification of toxic comments or hate speech on online portals. Andročec (2020) in his survey paper highlighted that the research around toxic content identification expediated recently with earliest work published in 2018.

2.1 Machine Learning

Several Machine learning techniques are leveraged to identify disparaging content posted online. The supervised machine learning algorithms such as random forest, decision tree, naïve bayes, logistic regression, and SVM are implemented on the Quora dataset by Mungekar et al. (2019). An ensemble of naïve bayes and logistic regression is implemented as well. The questions are pre-processed, combining lemmatization with TF-idf and bag of words as vectorization techniques for different models. The trained models are evaluated using f1-score metric, and the models trained with bag-of-words as vectorization technique have flared well. The best results are observed for logistic regression and its ensemble model with an f1-score of 0.63.

The role of feature engineering in machine learning is unparalleled. Proper feature extraction leads to an increase in model's performance. For the Quora insincere question classification task, Priyambowo and Adriani (2019) investigated the outcome of different feature selection techniques to form a baseline for future studies. The researcher undersampled the data to get an equal proportion of the target class and blended the syntax, semantic and lexical features for feature engineering. The different features extracted were fed into Random forest classifier, Decision tree, KNN, SVM, multinomial naïve bayes, and multilayer perceptron to find what combination of features and model produces the best results. The author conferred from the evaluation results that extraction of unigram features performs reasonably well with all the chosen models. Adding other feature engineering techniques such as POS increases the f1 score of some models. The effect of different pre-processing techniques and feature representation on model's performance is evaluated by Al-Ramahi and Alsmadi (2020) for the dataset released by Quora. The author performed two experiments considering smaller and balanced corpus of data with approximately 60k records for one experiment and 15k records for the other. For the first experiment, stemming and stop word removal are performed in the pre-processing stage, and bag of words (BoW) is used for feature space representation. Whereas for the second experiment, only the stop words are removed, and n-gram technique is used for feature space representation. To avoid overfitting caused by the large feature space generated by BoW and n-gram, the chi-square (X2) statistic is used. This technique helps select relevant features from the large feature space by measuring the chi-square statistic for each feature regarding the target class. The machine learning classifier models, Random Forest, Naïve bayes, Logistic regression, SVC, and Decision tree are trained on the pre-processed data. The trained models are evaluated using f1-score, and the results of both experiments are almost the same. Thus, highlighting that stemming is not an essential pre-processing step for such problems. Overall, Logistic regression produced the best results in comparison to other models.

On top of the binary classification dataset by Quora, a dataset with six labels for further granular classification of insincere questions was released in FIRE 2019 competition. The six labels included sincere, hate speech, rhetorical, hypothetical, objectionable, and other category. The TF-IDF vectorization technique with SGD optimized SVM is used by Ranganathan et al. (2019) to classify the questions into six labels. Additionally, the most commonly used hate speech and offensive words are filtered to refine the classification further. The model achieved accuracy close to 48%. The same dataset is used by Kajla et al. (2020) to train different machine learning models with evaluation metrics such as log-loss and hamming-loss. The researcher thoroughly pre-processed the data and trained Logistic regression, Naïve bayes, Random forest, SVM, Decision tree, and KNN models. Logistic regression performs the best with minimum hamming loss.

Overall, we can infer that logistic regression achieves good results for insincere question classification for most studies. However, the dependency on feature extraction and data pre-processing techniques can not be disregarded. Machine learning models perform well only with correctly extracted features. Also, they fail to retain the semantic meaning of sequences during vectorization. Wankmüller (2021) pointed out that domain-specific knowledge and multiple trials are required to extract features for ML models.

2.2 Neural Network & Deep Learning

In Natural Language Processing, contextual learning is enabled by using deep learning architectures. CNN and RNN are two commonly used deep learning frameworks based on neural networks. With introduction of Attention mechanism, transfer learning models based on transformer architecture are used as well for different NLP problems. In the subsequent sections, we have briefly discussed the work done by researchers using variations of CNN, RNN and pre-trained models for hate speech and insincere question identification.

2.2.1 Convolutional Neural Network (CNN)

The recent studies show application of CNN architecture in the NLP domain in contrast to the former trend of its use with image data. Roy (2020) addressed the quora insincere question classification problem by implementing a multilayer CNN (MLCNN). Firstly, the raw data is pre-processed, and a question matrix is created using two manually created embeddings (Skipgram and continuous BoW) and one pre-trained embedding (GloVe). All the questions are padded to be of uniform length. The convolution process with different kernel sizes starts once the question matrix is ready. A pooling layer is added to reduce the size of feature space. A dense layer is added at the end for target probability calculation. Various experiments with combinations of kernels (2-g, 3-g, and 4-g), dropout layer, and embeddings are performed to find the model with best performance. The experimental result proves that the model's performance improves with the dropout layer and larger kernel size.

Two datasets, namely, Adolescents on Twitter (ALONE) and FIRE'20 containing youth conversation on twitter and facebook are combined by Malik et al. (2021) for toxic content identification. The researcher trained various machine learning and deep learning models for the task at hand. Particularly for the DL model, BERT and fastText embeddings are used for vector representation of data which is then fed into LSTM, CNN, and multilayer perceptron. The results of the study show that CNN with BERT embedding produces the best results with an f1-score of 0.81.

2.2.2 Recurrent Neural Network (RNN)

RNNs are a common form of neural network capable of handling data sequences. LSTMs are slightly different from RNN and were developed to resolve the vanishing gradient problem faced by RNN. These architectures have been readily used for text classification use cases.

A deep learning framework incorporating bi-directional GRU and LSTM with corresponding attention layers is used by Sampath (2019) for insincere question classification. The researcher pre-processed the data and vectorized it by using a combination of GloVe and FastText embeddings. An undersampled version of data is used for model training. The model is evaluated using k-fold cross-validation using f1-score as the metric. Instead of using bi-directional GRU, Mediratta and Oswal (2019) used bi-directional CUDNNGRU for the Quora problem statement. Other models using bi-directional LSTM, Naïve Bayes, and SVM are trained as well to compare the performance. The author undersampled the data using a random undersampler. The results obtained for all the models are competent. Jain et al. (2020) tackled the insincere question problem by developing a novel deep refinement model. The deep refinement model included layers of bi-directional GRU, bi-directional LSTM, capsule and attention layer. Other parameters chosen include adam optimizer, log loss function and cyclic learning rate. The author compared the proposed model's performance with other machine learning and deep learning models. The comparison results prove that the deep refinement produces better results when compared to other baseline models. Though the model produces compelling results, it is worth noting that the introduction of capsule layers increases the training time and resources required.

A bi-directional LSTM model is implemented by Do et al. (2019) to identify hate speech in the Vietnamese shared task 2019 competition. The dataset is labeled with three target variables, namely, clean, hate, and offensive. Extensive pre-processing is performed on the dataset, and word2vec and fastText embedding layers are used for vectorization. Bi-LSTM with fastText embedding achieves an f1-score of 0.71.

2.2.3 Transformers

The field of Natural language processing has seen tremendous advancements with the introduction of Transformer architecture by Vaswani et al. (2017). Transformers solely use Attention mechanism for sequence modeling, thereby evading the use of recurrent neural networks. The attention mechanism in transformer architecture aids in solving complex text analytics tasks. The performance of models based on transformer architecture improves as it is designed to allow parallelization during model training. It comprises encoders and decoders with multi-headed self-attention and fully connected layers. Another critical aspect of the transformer architecture is its ability to encode the relative position of tokens of the input sequence.

Transformer architecture enables transfer learning in NLP. Several models have been developed based on transformer architecture to solve problems such as text classification, sentiment analysis, language inference, text generation, and more. The pre-trained models, BERT (Devlin et al.; 2018), XLNET (Yang et al.; 2019), RoBERTa (Liu et al.; 2019), GPT-2 (Radford et al.; 2019), and more, are fine-tuned with domain specific data for different NLP tasks.

Identifying hate speech on online portals is complicated as a considerable number of users are multilingual. Users tend to blend languages while conversing, and this leads to code-mixed text being generated. Banerjee et al. (2020) implemented BERT, RoBERTa, DistilBERT, and XLNet to identify hate speech in Hindi-English code-mixed text. The researcher highlighted that a limited amount of work is done regarding hate speech identification in code-mixed text. The dataset for this study is collected from Twitter and annotated. Along with individual model training, an ensemble with all the models is developed as well. The results show that models perform well for different metrics (macro f1, weighted f1, recall, and accuracy). Overall, XLNet performs well for the code-mixed text with a macro f1-score of 0.67. BERT and multilingual-BERT models are implemented by Dowlagar and Mamidi (2021) to classify hate speech for FIRE 2019 and FIRE 2020 datasets. The researcher considered the SVM model as the baseline model and fine-tuned BERT for the datasets chosen. A learning rate of 2e-5, drop out of 0.1, and batch size of 64 is chosen. A 5-6% increase in accuracy and macro f1-score is observed with the use of pre-trained models.

2.3 Conclusion

An overview of literature shows that machine learning and NLP techniques have made a breakthrough in identifying harmful or insincere content on online portals. Both machine learning and deep learning models have been rigorously used in the past for insincere question identification problem statement. However, as highlighted earlier, ML models heavily depend on feature engineering, and Deep learning models (RNN and CNN) rely on availability of large datasets. As the hidden layers of DL model increase, the model tends to overfit if the dataset size is not adequate. Also, these models fail to capture dependencies for multiple and extra-long sentences. The modern advances in transfer learning techniques overcome the shortcomings of ML and DL models by eliminating the need for feature extraction and large datasets.

Considering the current state of research for insincere question identification, it is clear that the transformer-based models are relatively unexplored. Thus, this research will use the pre-trained model, BERT and XLNET, for the task at hand and compare their performance with the baseline model.

3 Methodology

This research project focuses on building a robust model for classification of questions posted on Quora – sincere and insincere. The Cross-industry process for data mining (CRISP-DM) methodology is used to classify the questions. The CRISP-DM approach helps to effectively plan, organize and execute data analysis projects. Figure 1 represents different stages involved in CRISP-DM.

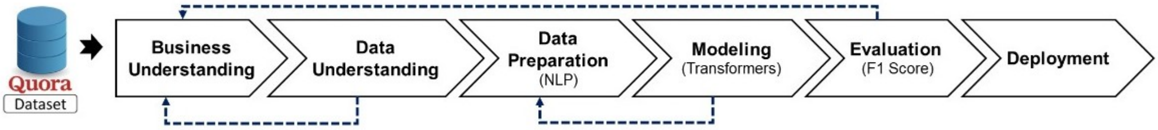

Figure 1: CRISP-DM Methodology

3.1 Business Understanding

The internet and social communication channels are great platforms to exchange knowledge promptly. People these days use community question and answer forums such as Quora to get feedback on their questions. Some people often misuse these knowledge-sharing platforms to spread hatred or even spread misleading information by posting personal opinions as questions rather than genuine questions. Questions posted to target a specific community, gender or race degrade the quality of content on Quora and are not safe for the audience who genuinely use this platform to seek information. Therefore, there is a dire need to identify these insincere questions before they are visible to a broader audience and take appropriate actions to maintain the safety of the online knowledge-sharing ecosystem.

3.2 Data Understanding

Earlier, the Quora team leveraged machine learning models and manual review processes to remove the insincere questions from the platform. In order to have more scalable and robust models for identifying toxic or misleading content, a dataset with approximately 1.3M records was published by the Quora team. This dataset is made publicly available for research on the Kaggle data repository under competitions[1]. There are three columns in the dataset: qid – the question id, question text – the actual questions posted by the users, and the target – binary variable indicating whether the question is sincere or insincere (0 or 1).

Exploratory data analysis (EDA) is performed to gain insights about the data distribution. Figure 2 indicates the target variable distribution. The plot clearly shows that the dataset is highly imbalanced with most questions being sincere. We are not disturbing the target variable distribution as performing undersampling might lead to loss of important information from the data and oversampling might cause model overfitting. Figure 3 shows a wordcloud with commonly used words in questions posted on Quora.

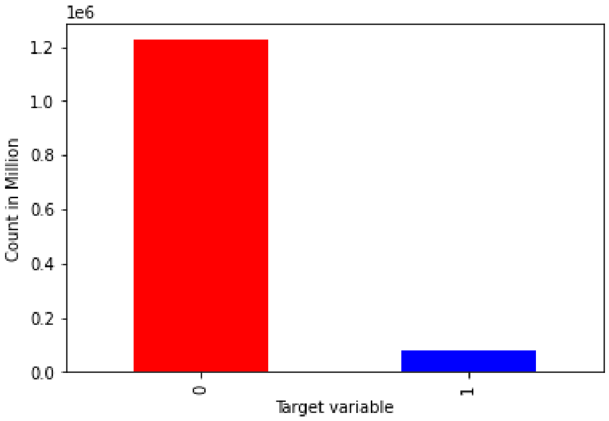

Figure 2: Target variable distribution

[1] https://www.kaggle.com/c/quora-insincere-questions-classification/data

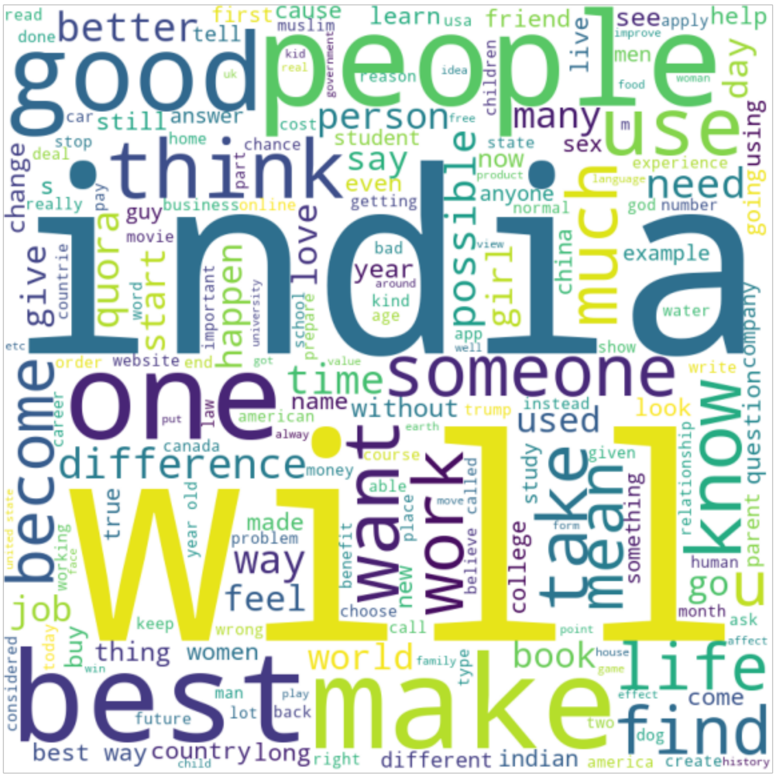

Figure 3: Word Cloud

3.3 Data Preparation

The questions in the dataset require pre-processing before feeding them into the model for training. The textual questions are pre-processed to remove special characters, numeric characters, punctuations, duplicate records, and the different contraction words are expanded using python libraries. The dataset is validated for presence of duplicate records. Once the pre-processing is done, the cleaned questions are tokenized and converted into vectors to meet the required input format of pre-trained transformer-based model.

We have refrained from removing the stop words and performing stemming or lemmatization to keep the syntactical meaning of the questions intact. By omitting these steps, the valuable information of the data is preserved.

3.3.1 Tokenization & Padding

Tokenization is a fundamental step in NLP problems, involving breaking text corpus into smaller units known as tokens. BERT and XLNET models have their different individual tokenizers that were used while pre-training these models. BERT uses a WordPiece tokenizer to tokenize the input questions (Schuster and Nakajima; 2012). It is a data-driven tokenization approach that helps to strike a balance between vocabulary size and out of vocab (OOV) words. With WordPiece tokenizer, BERT stores 30522 words in its vocabulary and breaks the OOV words until it is found in the existing vocabulary. XLNET uses a SentencePiece tokenizer to convert input sequences into tokens (Kudo and Richardson; 2018). The SentencePiece tokenizer is designed to implement subword tokenization using byte-pair encoding (BPE) and unigram model.

Additionally, the special tokens [CLS] and [SEP] are added at start and end of statement, respectively, to enable the model to differentiate between different sets of input data. The last hidden state of [CLS] token is considered while deciding on the output of classification tasks. The BERT and XLNET models expect input sentences to be of a uniform length. Since that is not the case in real-life scenarios, i.e., different users post questions or comments of varying lengths, padding is performed. Padding ensures that all sentences are of the same length as specified by the max_seq_length parameter. If a sentence has lesser tokens, it is filled with zero values, and if a sentence is longer than max_seq_length, it is truncated.

3.3.2 BERT & XLNET Embedding

The BERT and XLNET models are pre-trained on large corpus of data and they provide embedding layers which can be used to fine-tune these models with task specific data. Typically, there are three layers in the embeddings, namely, token, segment and positional embeddings. The token embedding represents the tokenized inputs into vectors of fixed dimensions. The positional embeddings store the temporal positions of the input tokens with respect to each other. The segment encoding is crucial when the input data contains multiple sentences or sentence pairs. The model differentiates each sentence from another by its segment embeddings. XLNET uses relative segment encoding instead of segment encoding in its embedding layer. Figure 4 shows embedding layers of BERT model.

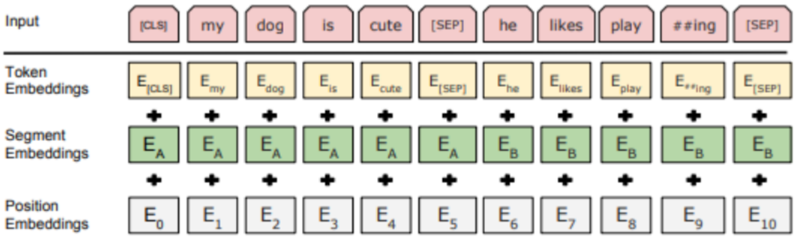
Input
[CLS]
my
dog
is
cute
[SEP]
he
likes
play
##ing
[SEP]
Token Embeddings
$E_{[CLS]}$ E_{my} E_{dog} E_{is} E_{cute} $E_{[SEP]}$ E_{he} E_{likes} E_{play} $E_{\#\#ing}$ $E_{[SEP]}$
Segment Embeddings
E_A E_A E_A E_A E_A E_A E_B E_B E_B E_B E_B
Position Embeddings
E_0 E_1 E_2 E_3 E_4 E_5 E_6 E_7 E_8 E_9 E_{10}

Figure 4: BERT input representation

3.4 Modeling

In this research project, BERT and XLNET models that enable transfer learning are chosen to classify insincere questions on Quora. These models are pre-trained with a large corpus of data and are made available for fine-tuning on domain-specific datasets. A comprehensive review of the literature around insincere/toxic speech identification shows that transfer learning techniques have not been explored for this particular domain. We are using the Simple Transformers library for easy implementation of complex transformer models. It is built on top of open source Hugging face library. We have considered Logistic Regression model as the baseline model for comparison.

3.4.1 BERT

BERT is developed using encoder of the transformer architecture and is a bi-directional pre-trained model. It has produced state-of-the-art results for text classification tasks on a number of metrics, SQuAD v1.1 F1-score, GLUE benchmark, and more. Devlin et al.

(2018) proposed two variations of BERT (BERT_base and BERT_large) in their original paper. These models differ in the number of layers, hidden vector size, and attention heads. BERT_base is smaller in size with 12 layers, 768 hidden vector size, 12 attention heads, and is trained with 110M parameters. On the other hand, BERT_large has 24 layers, 1024 hidden vector size, 16 attention heads, and is trained with 340M parameters. Each layer of the BERT model is represented by encoder block with multi-headed self-attention. BERT models are pre-trained using masked language modeling (MLM) and next sentence prediction (NSP). The pretraining of BERT model is computationally expensive. These pre-trained models can be fine-tuned for downstream tasks such as text classification, translation, and more by transferring the language modeling capabilities of the pre-trained model to domain specific data.

3.4.2 XLNET

XLNET is a generalized pre-trained autoregressive language model (Yang et al.; 2019). The XLNET model is designed to overcome the limitations of AR and AE models by incorporating permutation language modeling (PLM). In PLM, the tokens are predicted randomly for each combination of input by considering context from left-to-right and right-to-left. Using permutation to capture bidirectional context during pretraining helps XLNET model to overcome the limitations of AR and AE models. XLNET overcomes the constraint of fixed-length context learning by borrowing the idea of recurrence from Transformer-XL architecture (Dai et al.; 2019). The segment level recurrence in Transformer-XL helps XLNET model to learn from hidden state of previous segment, thus solving the problem of fixed-length context vector. The hidden states of previous segments are cached and passed on to the processing stage of current segment as keys or values. In order to efficiently use the previous hidden states, researchers suggested the use of relative position encoding. Using relative position encoding, the transformer-based models can differentiate between different segment inputs across different layers. The XLNET_base model is similar to BERT in terms of size, i.e., 12 layers, 768 hidden vector size, and 12 attention heads, and has surpassed the performance of BERT for 20 language tasks.

3.5 Evaluation

The Quora insincere questions dataset is highly imbalanced, with a majority of records belonging to the sincere (target - 0) category. To replicate a real-world scenario wherein the number of insincere or toxic questions is less than sincere, we have not performed any resampling technique. Owing to the imbalanced nature of our dataset, we have considered f1-score as primary metric for model evaluation. Classification report is obtained in python to show the model performance in terms of f1-score, macro avg f1-score, and accuracy (Dowlagar and Mamidi; 2021).

$$F1score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Macro avg f1-score is the average arithmetic mean of the f1 score per class.

4 Design flow diagram

Figure 5 represents the design flow diagram of our project. The detailed implementation is explained in the next section.

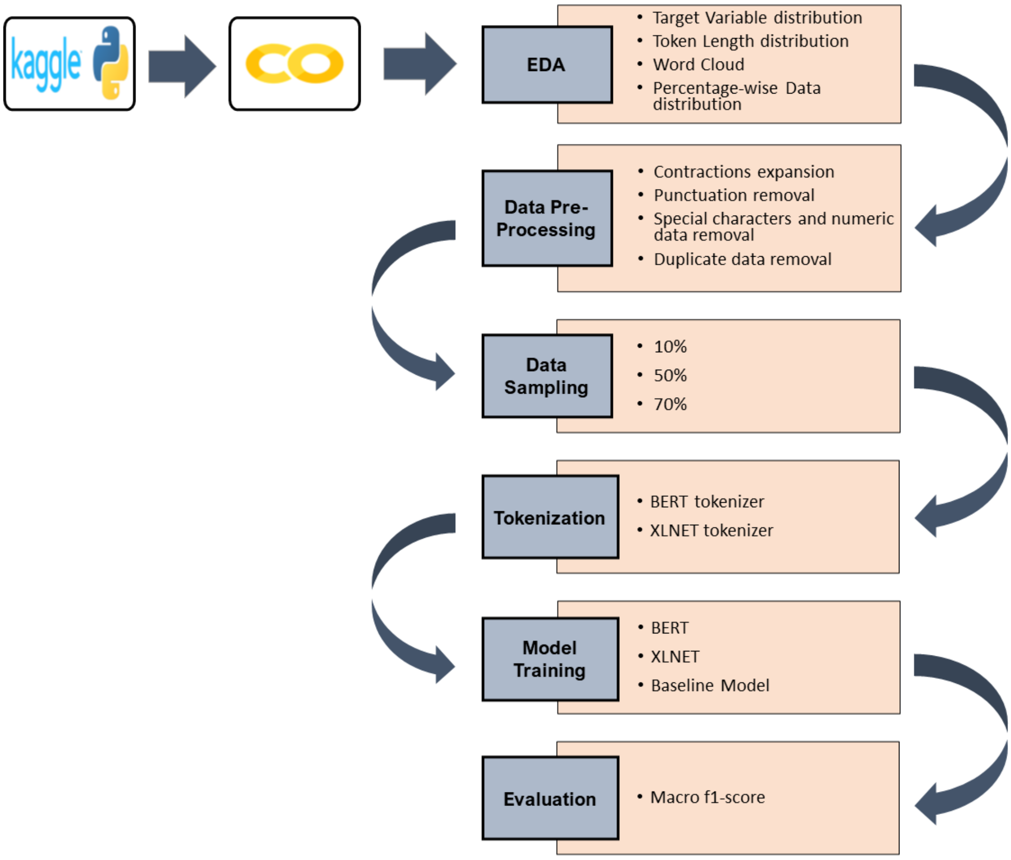

Figure 5: Project Design Flow

5 Implementation

For the implementation, we have used the SimpleTransformer library built on top of HuggingFace. The SimpleTransformer library provides a framework for easy implementation of pre-trained transformer-based models for various NLP tasks. We have used the Classification model from the SimpleTransformer library, with BERT and XLNET as the input. This section provides details about the technologies, tools, and libraries used for our research project. Furthermore, the implementation details of BERT and XLNET models are outlined.

Table 1: Technical Configuration

IDE	Google Colab Pro
Programming Language	Python
Computation	1 GPU (Tesla P100-PCIE-16GB)
Visualization Library	Matplotlib, WordCloud, Seaborn, Wandb
Modeling Library	SimpleTransformer, HuggingFace Transformer, Sklearn, Pandas, Numpy, NLTK, Regex
Framework	Pytorch

5.1 Exploratory Data Analysis

The dataset for insincere question classification is extracted from Kaggle and loaded into Google Colaboratory with GPU enabled. Exploratory data analysis and data preprocessing are performed using Python libraries. The dataset contains nearly 1.3M records with very few samples from the insincere class (6%).

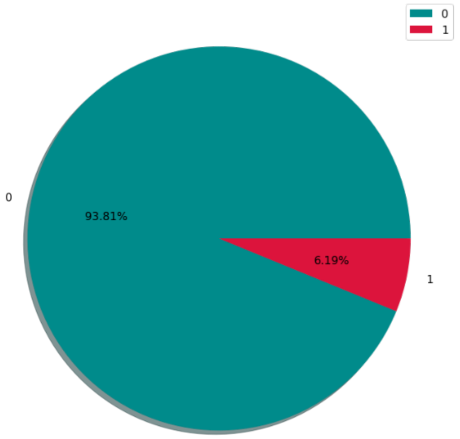

Figure 6: Percentage wise distribution of target variable

To visualize the token length of questions, we have plotted a dist plot which is a combination of histogram and kdeplot. The y-axis of Figure 7 indicates the probability density function of the kernel. On the x-axis is the length of the questions for Sincere and Insincere questions. The plots below indicate that the length of insincere questions is shorter than that of sincere questions, with maximum length being 134. Additionally, we calculated the mean and median length of questions in Python. The mean and median lengths are 13 and 11, respectively. Since the maximum length of questions in our dataset is not too large ($<$ 512) and mean length being too small, we have considered the max_seq_length as 134 for model training.

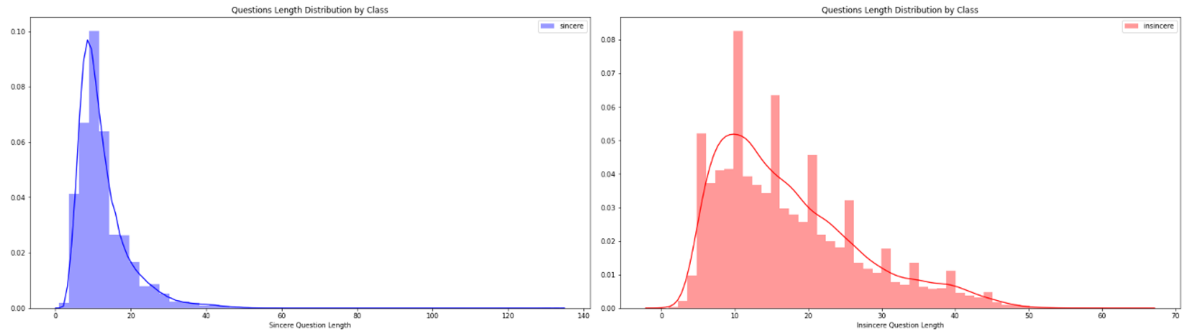

Figure 7: Plot to find the token length of questions

5.2 Data Cleaning

BERT and XLNET models are trained on a large corpus of textual data and can learn the features of the data on their own due to the self-attention mechanism. Hence, we have performed data cleaning and omitted the feature engineering steps.

The data cleaning steps are discussed below -

i. Expansion of contractions

Contractions refer to the shortened version of words in English, for example – didn't. We have expanded the contractions present in the dataset to ensure text standardization. The Python contractions library is imported and used to expand contractions from the questions of the dataset.

ii. Removal of punctuations and special characters

The questions in the dataset contain various emojis, special characters, and punctuations. They are removed by using regular expressions (regex). Lambda expression is used to apply regex filtering to all the questions.

iii. Removal of numbers

After expanding the contractions and removing the special characters, we could find numeric data in the questions of the dataset. Lastly, the numbers are removed from the questions by filtering out the numeric characters using lambda expressions.

Once the data cleaning step is done, the cleaned question column and target variable are retained in the final dataframe.

5.3 Data sampling for optimum computational power usage

The dataset provided by Quora is large and contains different files for training and testing samples. In order to meet our objective of achieving state-of-the-art results with limited computational power, we have trained the transformer models by considering different size (in %) of data for each iteration. For this, we have taken different samples of data from the original dataset while keeping the imbalance ratio constant. The sample() function with frac as an argument extracts 10%, 50%, and 70% of data for each iteration.

Table 2: Data Samples

Sr. No.	Data %	Number of Samples
1	10%	130533
2	50%	652664
3	70%	913730

5.4 Data splitting for training and testing

The dataset is split into Train-Validate-Test with a ratio of 80-10-10. Since the k-fold cross-validation is computationally expensive, we have utilized the train_test_split function from sklearn library in Python to split the dataset.

5.5 BERT Model Training

We have fine-tuned the bert-base-uncased model with our dataset. The pre-processed training data is tokenized first by using the bert tokenizer. The bert tokenizer encompasses word piece tokenizer to convert word sequences into tokens. The textual data is converted into tokens and later represented as vectors, representing the embedding layer. The max_seq_length is initialized as 134 and is passed as a parameter to the model training function. The model is trained for two epochs with the training and evaluation batch size as 16 and 32, respectively. The learning rate is initialized as 1e-5, and the AdamW optimizer with an epsilon value of 1e-8 is used. Binary cross entropy is the loss function, and dropout is set as 0.3.

We have used the ClassificationModel module provided by the SimpleTransformers library during fine-tuning[2]. It takes in bert, bert-tokenizer, and other hyper-parameters as its inputs. Additionally, a wandb_project[3] is passed as an input to the model. This wandb_project is a part of the Weight and Bias framework supported by SimpleTransformers. Model training and evaluation details are logged inside this project clearly with interactive visualizations for training and eval losses, GPU utilization, and model performance. The dense layer and softmax layer for binary classification is implicit with this model. All the steps mentioned above are implemented for each dataset sample (10, 50, and 70%).

5.6 XLNET Model Training

The xlnet-base-cased model of the Simple Transformers library is used. The training data is tokenized using xlnet-tokenizer, and the model is fine-tuned using vector representations of the input sequences. We have trained the model with two different learning rates, 1e-5 and 2e-5, for two epochs. Additionally, the maximum sequence length, batch size,

[2] https://simpletransformers.ai/docs/classification-models/

[3] https://docs.wandb.ai/

optimizer, dropout, and the loss function are kept the same as that of the BERT model to enable a fair comparison of model performance.

5.7 Choice of Hyperparameters

After careful consideration, the value for each hyperparameter is chosen. The rationale behind selecting each hyperparameter is listed below-

1. Epoch - The transformer models are pre-trained on a large corpus of data, and fine-tuning the model with 2, 3 or 4 epochs produces exemplary results. Also, Devlin et al. (2018) highlighted that when fine-tuning with 100k+ records, training with larger epochs does not substantially improve the results. We initially trained our models for 3 epochs and observed that the models started to overfit after the 2nd epoch. Thus, we trained our models with epoch = 2.

2. Learning Rate - We selected learning rate as 1e-5 2e-5 for the model training as suggested by researchers to avoid model overshooting from local minima for text classification tasks (Devlin et al.; 2018) (Yang et al.; 2019). For our dataset, models generalized better with a learning rate of 1e-5.

3. Dropout - The dropout value is chosen by reviewing the literature around text classification (Hande et al.; 2021) (Puranik et al.; 2021). Initially, the models were trained with dropout as 0.1. The models were observed to overfit the training data. In order to prevent overfitting, the dropout value was increased from 0.1 to 0.3. On implementation, we observed that model with dropout set to 0.3 with 2 epochs generalized better. Hence, 0.3 is chosen as dropout for BERT and XLNET models.

4. Train Batch Size - Optimal batch size is chosen as a very small batch size leads to longer training time and larger batch size leads to non-convergence to global minima and running out of memory issue. As we have chosen epoch as 2, we chose the train_batch size as 16 for a fair trade-off with training time and generalized performance.

6 Evaluation & Discussion

6.1 Evaluation of Trained Models

Post model training, the fine-tuned model is evaluated using the validation data and tested on the test data using the predict function. The data is highly imbalanced; therefore, we are not relying on accuracy alone for the performance evaluation. Since, macro f1-score provides per class average arithmetic mean, we will compare the performance of models by using the same.

6.1.1 XLNET Results

Table 3 shows the test results for XLNET model trained with 10, 50 and 70% data. The results shown are for the model trained using a learning rate of 1e-5 as it produced better results than 2e-5. We can see from the results that with only 10% of the data, XLNET model achieved a macro f1-score of 0.81. The macro f1-score increases marginally for 50% and 70% data samples.

Table 3: XLNET Results

Sr. No.	Data	Accuracy	Macro Precision	Macro Recall	Macro f1-score	Weighted f1-score
1	10%	0.96	0.84	0.79	0.81	0.96
2	50%	0.96	0.85	0.83	0.83	0.96
3	70%	0.96	0.86	0.82	0.84	0.96

6.1.2 BERT Results

Table 4 shows the test results for BERT model trained with 10, 50 and 70% data. The model produces marco f1-score of 0.79 for 10% data. The macro f1-score for 50% and 70% is same with slight change in the macro Precision.

Table 4: BERT Results

Sr. No.	Data	Accuracy	Macro Precision	Macro Recall	Macro f1-score	Weighted f1-score
1	10%	0.95	0.83	0.76	0.79	0.95
2	50%	0.96	0.84	0.78	0.81	0.96
3	70%	0.96	0.83	0.79	0.81	0.96

6.1.3 Baseline Model

We have trained a logistic regression model as a baseline model with bag of words (BoW) and TF-IDF vectorization techniques. In addition to the pre-processing done earlier, we have removed the stopwords and performed lemmatization for the logistic regression model. The model achieves an f1-score of 0.58 and 0.59 with BoW and TF-IDF, respectively.

6.2 Discussion

6.2.1 Comparison based on model's performance

On observing the results in Table 3 & 4, we noted that both XLNET and BERT models produce state-of-the-art results with just 10% data. There is only a marginal increase in the macro-f1 score with increase in the size of data. The confusion matrix for both the models with 10% data is shown in Figure 8 & 9. From the true positive (12069) and true negative (471) values of the XLNET confusion matrix, we can infer that our trained models have correctly predicted most of the records. The high rate of false positives, i.e., 324, can be attributed to the imbalance in the dataset. Overall, the XLNET model has performed well compared to BERT for all three iterations (10, 50, and 70%).

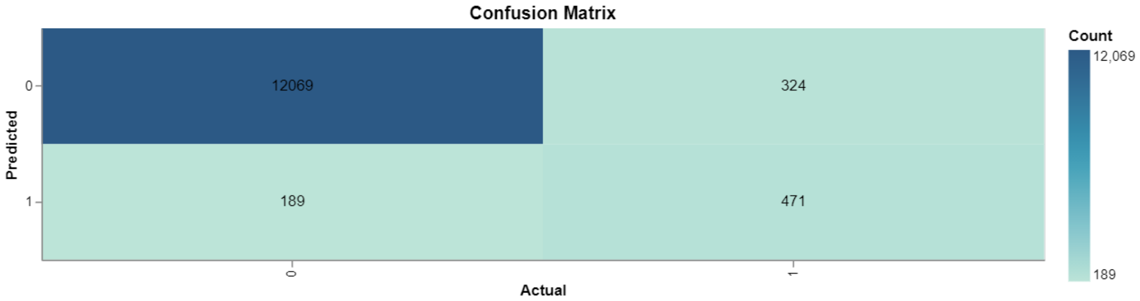

Figure 8: XLNET (10%) Confusion Matrix

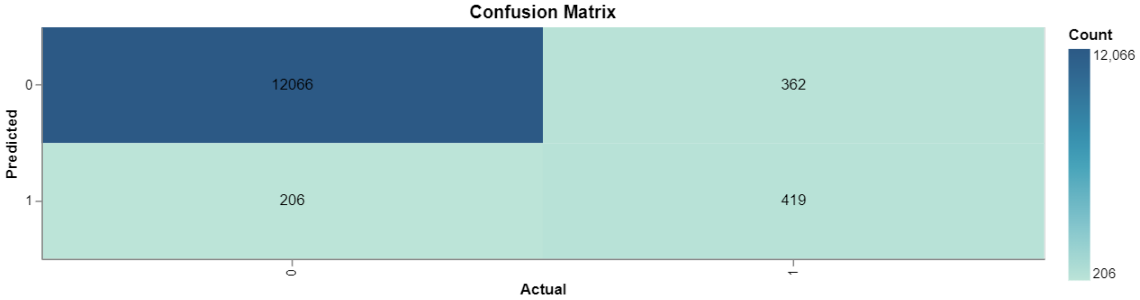

Figure 9: BERT (10%) Confusion Matrix

6.2.2 Comparison with the baseline model

From the literature, we could see that Logistic Regression model flared well for our dataset. We, therefore, trained an LR model with two different embeddings (BoW and TF-IDF). On comparing the results, we can clearly see that transformer models, BERT and XLNET, outperform Logistic Regression in terms of both accuracy and macro-f1 score. The macro-f1 score improved by 0.22 while using Transformer models.

Also, the BERT and XLNET models outperform the deep learning model implemented by Do et al. (2019) and Mediratta and Oswal (2019) with considerable difference across all the metrics. Table 5 shows the f1-scores obtained in the literature and our current implementation.

Table 5: Comparison of Results with Baseline Model

Sr. No.	Model	F1-score
1	Logistic Regression + BoW	0.58
2	Logistic Regression + TF-IDF	0.59
3	BiLSTM + FastText (Do et al.; 2019)	0.71
4	CUDNNGRU (Mediratta and Oswal; 2019)	0.72
5	BERT 10%	0.78
6	XLNET 10%	0.81

6.2.3 Comparison based on training time and GPU utilization

We have trained the BERT and XLNET models on a single GPU. Figure 10 shows the training time taken by BERT and XLNET for each iteration. The time taken for fine-tuning the XLNET model is more than BERT. The plot for GPU utilization has been extracted from the wandb project for each model. Figure 11 & 12 shows the overall GPU utilization for XLNET and BERT models with 50% data. With increased training time, the computation resources required also increases. Since we want to minimize the use of high computation power, XLNET with 10% data can be considered the best model with a macro f1-score of 0.81 and training time of approximately 1 hour.

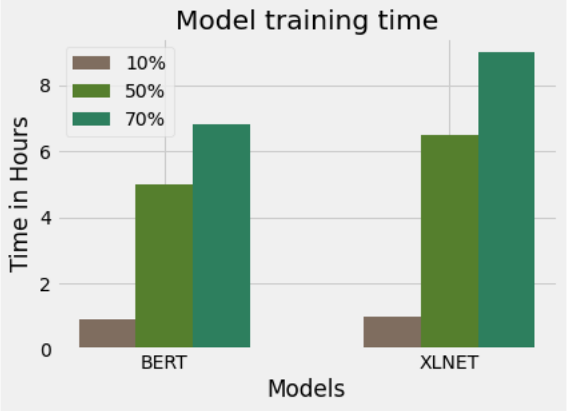

Figure 10: Model training time for BERT & XLNET

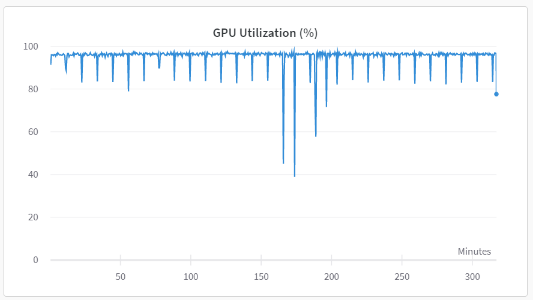

Figure 11: GPU utilization for XLNET with 50% data

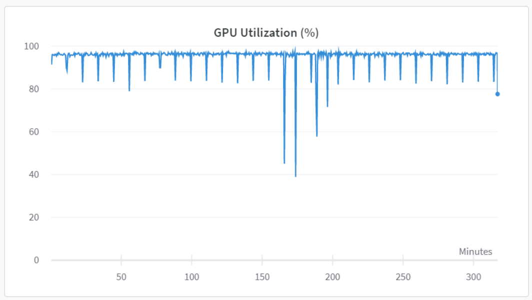

Figure 12: GPU utilization for BERT with 50% data

7 Conclusion

The identification of hate speech or toxic content on online portals is an ongoing area of research. Both machine learning and deep learning neural network models have been successfully implemented in the past to classify toxic content on social media platforms automatically. However, the machine learning and deep learning models depend on feature extraction and a large volume of data, respectively. In this research work, we have implemented the pre-trained BERT and XLNET models to classify the insincere questions on Quora. These pre-trained models have the capability of enabling transfer learning in

NLP. The data containing questions posted on Quora is extracted from Kaggle and pre-processed in Python. Thereafter, we took three data samples (10, 50, and 70%) while keeping the imbalance ratio constant. The model hyper-parameters are carefully selected, and the models are trained using the pre-processed data in three iterations. A logistic regression model is trained and considered as a baseline model. Since the dataset is highly imbalanced, macro f1-score is considered as the primary evaluation metric. The model evaluation results show that BERT and XLNET models outperform logistic regression and other deep learning models implemented in the past. Our objective of producing state-of-the-art results with limited use of computational power is fulfilled as with only 10% data, both BERT and XLNET models produce remarkable results. The comparison between results of BERT and XLNET shows that XLNET outperforms BERT for our problem statement.

For future work, we suggest creating an ensemble of these pre-trained language models with a small corpus of data to achieve more remarkable results. Also, instead of using a linear layer at the end of pre-trained model, future researchers can use a LSTM layer and evaluate the results.

References

Al-Ramahi, M. A. and Alsmadi, I. (2020). Using data analytics to filter insincere posts from online social networks. a case study: Quora insincere questions.

Andročec, D. (2020). Machine learning methods for toxic comment classification: a systematic review, *Acta Universitatis Sapientiae, Informatica* **12**(2): 205–216.

Banerjee, S., Chakravarthi, B. R. and McCrae, J. P. (2020). Comparison of pretrained embeddings to identify hate speech in indian code-mixed text, *2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, IEEE, pp. 21–25.

Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q. V. and Salakhutdinov, R. (2019). Transformer-xl: Attentive language models beyond a fixed-length context, *arXiv preprint arXiv:1901.02860* .

Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding, *arXiv preprint arXiv:1810.04805* .

Do, H. T.-T., Huynh, H. D., Van Nguyen, K., Nguyen, N. L.-T. and Nguyen, A. G.-T. (2019). Hate speech detection on vietnamese social media text using the bidirectional-lstm model, *arXiv preprint arXiv:1911.03648* .

Dowlagar, S. and Mamidi, R. (2021). Hasocone@ fire-hasoc2020: Using bert and multilingual bert models for hate speech detection, *arXiv preprint arXiv:2101.09007* .

Hande, A., Puranik, K., Priyadharshini, R., Thavareesan, S. and Chakravarthi, B. R. (2021). Evaluating pretrained transformer-based models for covid-19 fake news detection, *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, IEEE, pp. 766–772.

Jain, D. K., Jain, R., Upadhyay, Y., Kathuria, A. and Lan, X. (2020). Deep refinement: capsule network with attention mechanism-based system for text classification, *Neural Computing and Applications* **32**(7): 1839–1856.

Kajla, H., Hooda, J., Saini, G. et al. (2020). Classification of online toxic comments using machine learning algorithms, *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, IEEE, pp. 1119–1123.

Kudo, T. and Richardson, J. (2018). Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing, *arXiv preprint arXiv:1808.06226* .

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach, *arXiv preprint arXiv:1907.11692* .

Malik, P., Aggrawal, A. and Vishwakarma, D. K. (2021). Toxic speech detection using traditional machine learning models and bert and fasttext embedding with deep neural networks, *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, IEEE, pp. 1254–1259.

Mediratta, D. and Oswal, N. (2019). Detect toxic content to improve online conversations, *arXiv preprint arXiv:1911.01217* .

Mungekar, A., Parab, N., Nima, P. and Pereira, S. (2019). Quora insincere question classification, *National College of Ireland* .

Paul, S. A., Hong, L. and Chi, E. H. (2012). Who is authoritative? understanding reputation mechanisms in quora, *arXiv preprint arXiv:1204.3724* .

Priyambowo, H. and Adriani, M. (2019). Insincere question classification on question answering forum, *2019 International Conference on Electrical Engineering and Informatics (ICEEI)*, IEEE, pp. 390–394.

Puranik, K., Hande, A., Priyadharshini, R., Thavareesan, S. and Chakravarthi, B. R. (2021). Iiitt@ lt-edi-eacl2021-hope speech detection: There is always hope in transformers, *arXiv preprint arXiv:2104.09066* .

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I. et al. (2019). Language models are unsupervised multitask learners, *OpenAI blog* **1**(8): 9.

Ranganathan, A., Ananthakrishnan, H., Thenmozhi, D. and Aravindan, C. (2019). Classification of insincere questions using sgd optimization and svm classifiers., *FIRE (Working Notes)*, pp. 463–467.

Roy, P. K. (2020). Multilayer convolutional neural network to filter low quality content from quora, *Neural Processing Letters* **52**(1): 805–821.

Sampath, M. (2019). *Toxic Question Classification in Question & Answer Forum Using Deep Learning*, PhD thesis, Dublin, National College of Ireland.

Schuster, M. and Nakajima, K. (2012). Japanese and korean voice search, *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, pp. 5149–5152.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. and Polosukhin, I. (2017). Attention is all you need, *arXiv preprint arXiv:1706.03762* .

Wankmüller, S. (2021). Neural transfer learning with transformers for social science text analysis, *arXiv preprint arXiv:2102.02111* .

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. and Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding, *arXiv preprint arXiv:1906.08237* .