Quora Insincere Questions Classification

Souvik Das
souvik.das@iiitb.org
IIIT Bangalore

Vaibhavi Tikone
vaibhavi.tikone@iiitb.org
IIIT Bangalore

Hritik Arora
hritik.arora@iiitb.org
IIIT Bangalore

Abstract

Quora allows one to ask any kind of questions and can have answers. Such freedom invites credibility issues such as some questions may target community or person or group of people and can impact insincere in terms of tone of questions and words used. An insincere question is defined as a question intended to make a statement rather than look for helpful answers.
Some characteristics that can signify that a question is insincere:1. Has a non-neutral tone, Is rhetorical and meant to imply a statement about a group of people. 2. Is disparaging or inflammatory 3. Makes disparaging attacks/insults against a specific person or group of people 4. Is not grounded in reality 5. Uses sexual content (incest, bestiality, pedophilia) for shock value, and not to seek genuine answers.
The report presents methodology for quantitative, large-scale analysis of a large corpus of online questions and classify as sincere or insincere question with the help of traditional machine learning methods. The corpus consists of questions from Quora with labels. We use this data to train a machine learning classifier and be able to give label 0 or 1 as sincere or insincere for test corpus.
We are presenting the model using classical machine learning methods that identifies and flag insincere questions. We have used publicly available data on Kaggle for inputs. We have used natural language processing tools and techniques to handle text data.

Index Terms: Logistic regression, TFIDF Vectorizer, Word2vec, Count Vectorizer, Naïve Bayes, Stacking, Preprocessing, Tokenizer, Ensemble

1 Problem Statement

An existential problem for any major website today is how to handle toxic and divisive content. Quora wants to tackle this problem head-on to keep their platform a place where users can feel safe sharing their knowledge with the world.

Quora is a platform that empowers people to learn from each other. On Quora, people can ask questions and connect with others who contribute unique insights and quality answers. A key challenge is to weed out insincere questions – those founded upon false premises, or that intend to make a statement rather than look for helpful answers.

2 Dataset Description

We have taken dataset from kaggle. Data set contains train.csv, test.csv, sample submission.csv and different word embeddings. The training data includes:

- Qid: Unique question id
- Question_text: Contains questions that were asked
- Target: sincere or insincere(Ground truth)

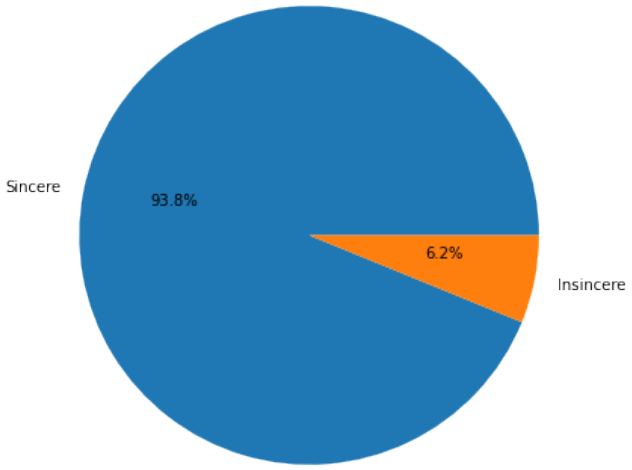

Figure 1: A visualization of the data distribution.

There are 48,451 questions that are insincere and other 7,35,222 questions are sincere. So about 6% of the training data are insincere questions (target=1)and rest of them are sincere.

3 Data Processing Tasks

3.1 Visualizations and Inferences

we used N-Gram analysis to see the phonemes, syllables, letters, words or base pairs according to the application.We also looked into most frequent insincere and sincere words, one of the plots is shown below.

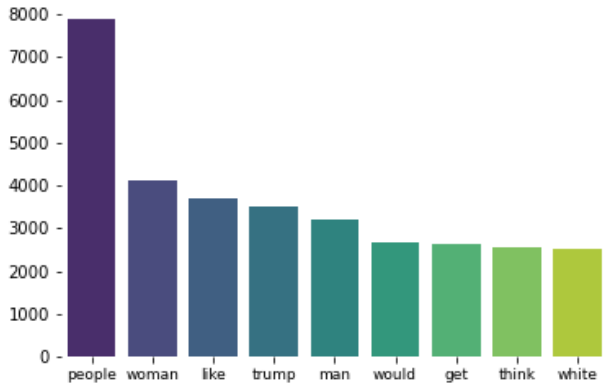

Figure 2: Frequent Insincere words

3.2 Data Cleaning and Preprocessing

We have applied some preprocessing, namely:

1. Text lowering: Converting all characters to lower case.

2. Remove punctuations: Punctuations like [;,',?,/,,,%] were not adding anything to the model, so we removed them.

3. Contraction mapping: Converting the words like wouldn't,can't to 'would not' and 'can not' respectively.

4. Spelling correction: There were many incorrectly spelled words so we corrected them using a certain mapping that was obtained from a huge corpus. Example- colour-¿color

5. Dealing with foreign language: There were some chinese and other foreign languages that need to be changed to english language to contibute to the model

6. Remove non-ascii characters: Unicode characters like 'â€™' , 'â€' which are not needed and were not contributing to the model were removed.

7. Tokenization: The sentences are split into words. The Natural Language Toolkit (NLTK) package is used for this purpose. It divides the words along with special characters treated as a word.After tokenization the further steps will be done like lemmatization

8. Lemmatization: Converting words to their root words.Root word are the words from which other words are made. Like dance is the root word for dancing, dances, danced etc. Using root words instead of the words itself result in a better accuracy rate.

3.3 Feature Engineering

We did some Exploratory Data Analysis in our dataset and found some new features.

3.3.1 Bad words count

This plot shows the comparision of number of bad words between taget 0 and 1

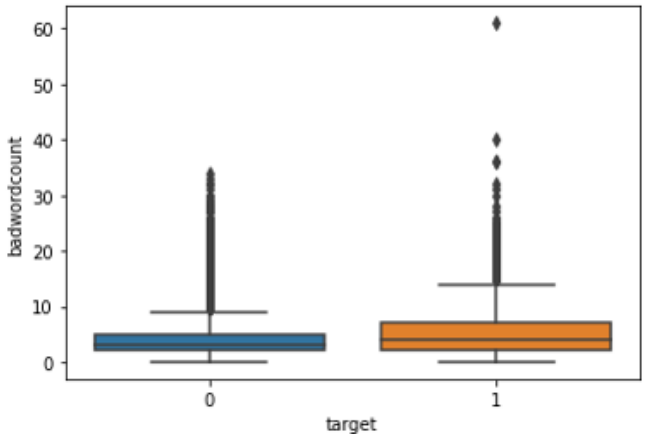

Figure 3: Bad words count

3.3.2 Characters count

This plot shows the comparison of number of characters between target 0 and 1

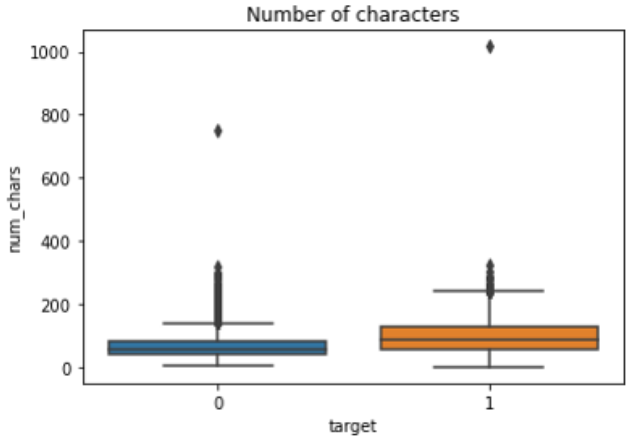

Figure 4: Character count

3.3.3 Unique word count

This plot shows the comparison of number of unique words between target 0 and 1

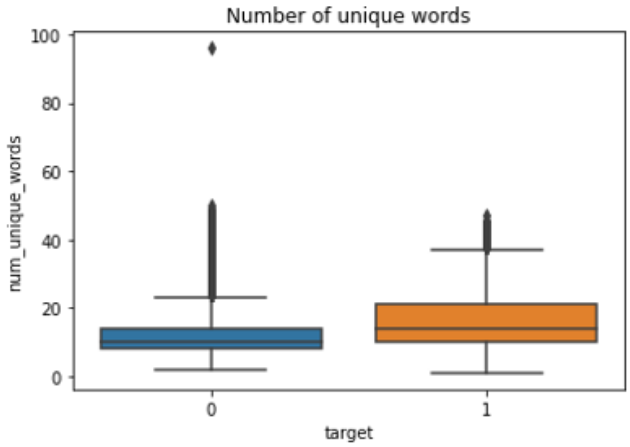

Figure 5: Unique words count

4 Splitting

Once we are done with preprocessing and EDA, we have to split the training data into 2 parts, one for training purpose and other for testing purpose.This is done using train_test_split under sklearn library.

5 Vectorization

Text Vectorization helps in converting textual data into a series of numeric factors for processing it further.We used two types of vectorizer:

5.1 Count Vectorizer

It is also called as bag of words representation of text. In this a document term matrix is created in which rows represent the sentences while the columns represent the words.It is a package in python called sklearn with a function of CountVectorizer which provides DTM. It has many parameters for tuning.We basically tried n-gram parameter in count vectorizer with logistic regression to find out which ngram is working best for our dataset.Below is plot representing the comparison between various n-grams.

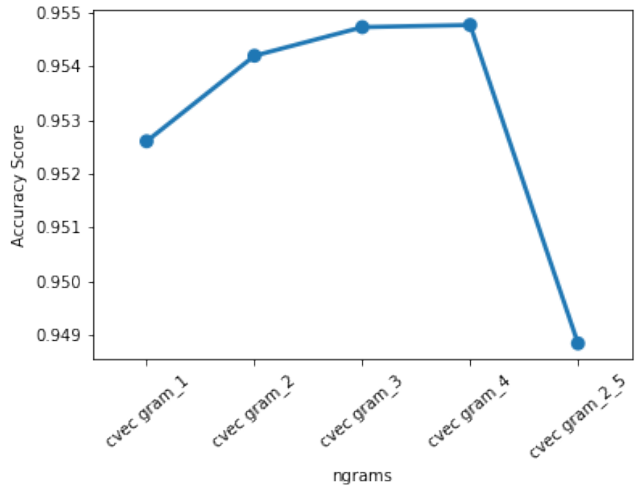

Figure 6: n-gram comparison

5.2 TFIDF Vectorizer

Term Frequency- Inverse Document Frequency are the two parts of the TF-IDF algorithm.Term Frequency is the frequency of word in the text and Inverse Document Frequency (TDF) is the importance of a word denoted in degrees. So we have used TFIDF Vectorizer with max_features=50000 and 4-grams.

5.3 Word Embeddings

We have also used word embeddings like glove,word2vec,wikicrawl etc. Word embeddings are a way to find relation between the words.

Firstly, all the words and their corresponding vector representations are extracted from the embeddings file and stored in a dictionary. Then we parsed our questions and searched for the common words in the dictionary. Then we averaged all the vectors for words in a particular sentence. That averaged vector is the vector corresponding to that particular sentence.

So here we used word embeddings to convert our text into a 300 dimension vector and then applied models to train these vectors.

6 Model Training

After vectorization,model training is done. We have used various models such as Decision trees, Random forest, Naive Bayes and Logistic Regression.Among these the best that worked for us is logistic regression.We have tried various combinations of vectorizer+models such as TFIDF+Logistic regression, TFIDF+Naive bayes, TFIDF+decision trees and Count vectorizer+logistic regression,Count Vectorizer+naive bayes etc.We have also tried logistic regession after vectorization with embeddings but it didnt gave us good score.

In **Decision Trees classifier**, At a time only two outcomes are tested, it is passed to the one it fits in. In this project, decision tree will be used to identify if a question is sincere or not based on its texual content.

Random forests classifier is a group of decision trees working together to give output. In this multiple decision trees are developed during training. These trees are then used to produce output using the mode of the decision trees. In this project various multiple trees are developed using various combinations of words. The mode of these trees is taken to produce the output of random forest.

Logistic Regression classifier does not assume anything. It gives the probability estimates of all classes.Using some threshold values we can get the best threshold for which the model gives highest score.

In **Naive Bayes classifier**,the probability of a single word is calculated first. Then using matrix multiplication, we calculate the probability of the question as a whole.

7 Results

Various evaluation metrics are considered as the data is highly imbalanced. The accuracy level cannot be considered for judging the best model as even if all questions are to be considered sincere the accuracy will be above 90%. Hence, F1 score acts as the main metric for evaluating the performance of the models.

At first we used feature extraction and using those features and applying logistic regression we got f1 score of 0.3 which is very poor. Then we used preprocessing and vectorization with some classical models that improved our f1 score to a great extent.

As seen from the table below, Logistic regression performed using TF-IDF, Logistic Regression performed using Count Vectorizer and their ensemble method gives the highest F1 score. The most poor performance is given by glove embedding and logistic regression which has an F1 score of just 0.37.

Table 1: F1 score comparison

Model	Vectorizer	f1 score
Decision Tree	Count Vectorizer	0.45
Naive Bayes	Count Vectorizer	0.40
Logistic Regression	Glove embedding	0.37
Logistic Regression	word2vec	0.45
Logistic Regression	wikicrawl	0.43
Bernoulli Naive Bayes	TFIDF Vectorizer	0.42
Gaussian Naive Bayes	TFIDF Vectorizer	0.53
Logistic Regression	TFIDF Vectorizer	0.62
Logistic Regression	Count Vectorizer	0.63
Logistic Regression	ensemble(count+tfidf)	0.64

Confusion Matrix
(For one of our best score)

[[142790 4297]
[3301 6347]]

8 Conclusion

Overall it can be said that the project is able to classify between sincere and insincere questions. It used multiple models which follow supervised learning algorithm. An ensemble model is also implemented in the project which is implemented using logistic regression with two vectorizers(count vectorizer and tfidf vectorizer) which gave the best score.

One reason behind the failure of embeddings could be the absence of words in the dictionary of embeddings.For this we could make a huge dictionary of words and then try to find out the relation, that will for sure improve the score by some extent.

Future Work
For future implementation, this project can be implemented using various deep learning models, LSTM etc. The problem of data

imbalance can be solved by first training the data with equal number of sincere and insincere questions.Also meta embeddings can be created.

REFERENCES

[1] C. Liu, Y. Sheng, Z. Wei, and Y. Yang, "Research of text classification based on improved tf-idf algorithm," in 2018 IEEE International Conference of Intelligent Robotic and Control Engineering (IRCE), Aug 2018, pp. 218–222.

[2] O. Aborisade and M. Anwar, "Classification for authorship of tweets by comparing logistic regression and naive bayes classi- fiers," in 2018 IEEE International Conference on Information Reuse and Integration (IRI), July 2018, pp. 269–276

[3] "A comparison of several ensemble methods for text categorization," in IEEE International Conference onServices Computing, 2004. (SCC 2004). Proceedings. 2004, Sep. 2004, pp. 419–422

[4] C. Alfaro, J. Cano-Montero, J. G ´omez, J. Moguerza, and F. Ortega, "A multi-stage method for content classification and opinion mining on weblog comments." Annals of Operations Research, vol. 236, no. 1, pp. 197 – 213, 2016.

[5] Pennington, Jeffrey Socher, Richard Manning, Christopher. (2014). Glove: Global Vectors for Word Representation. EMNLP. 14. 1532-1543. 10.3115/v1/D14-1162.

[6] Kumari, Khushbu Yadav, Suniti. (2018). Linear regression analysis study. Journal of the Practice of Cardiovascular Sciences. 4. 33. 10.4103/jpcs.jpcs_8_18

[7] Ma, Long Zhang, Yanqing. (2015). Using Word2Vec to process big text data. 10.1109/BigData.2015.7364114.

[8] Qaiser, Shahzad Ali, Ramsha. (2018). Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents. International Journal of Computer Applications. 181. 10.5120/ijca2018917395.

[9] Nima, Prateek. (2019). Quora Insincere Questions Classification.

[10] Kaviani, Pouria Dhotre, Sunita. (2017). Short Survey on Naive Bayes Algorithm. International Journal of Advance Research in Computer Science and Management. 04.

[11] Kaggle competition link: https://www.kaggle.com/c/quora