Quora Insincere Questions Classification

Abhishek Shambhu, Jeyamurugan Krishnakumar and Shreyans Singh
Department of Data Analytics Engineering George Mason University

ashambhu@gmu.edu, jkrishn@gmu.edu,ssingh@gmu.edu

***Abstract*— Quora is simply a question and answer website which can address any question or provide answers to question asked. However, even though many of the questions are sincere questions there any even many questions which are insincere. There are many problems present which makes a question insincere such as pornography, racism, use of abusive language in text, etc. So, we are going to address these issues with the help of various models implementation and looking at the experimental results achieved through each of the models on to how a question asked on the Quora website is classified as sincere or insincere.**

I. Introduction

Quora is a platform that empowers people to learn from each other. 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 [1]. The 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. For example, question having a non-neutral tone, whether a question is disparaging or inflammatory, isn't grounded, etc. To date, Quora has employed both machine learning and manual review to address this problem.

Recently, Quora has come up with a Kaggle challenge to handle this issue of toxic content in questions [2]. Most of the solutions focused mostly on the model building part then the pre-processing part, so we want forward into implementing some NLP techniques and using embeddings which would address this problem in an efficient manner from different perspectives. By doing so, we could distinguish genuine questions easily.

II. Related work

The questions asked on Quora were classified as insincere because of various reasons. Some of them are stated below:

- The question had an exaggerated tone
- The question is rhetorical and applies to a group of people
- Insulting / Bad / attacking statements against a group of people
- Statements having false information
- Statements with illogical assumptions
- Statements related to Sexual content, pornography, etc.

Previous attempts to solve this problem involved using various pre-processing techniques and building various machine learning and deep learning models and other different types of Neural Networks and frameworks such as Pytorch, TensorFlow, etc. where each method to solve this problem involves its own pros and cons such as processing speed, accessibility, run-time and flexibility.

Models such as deterministic neural networks using Pytorch, blend of LSTM and CNN, Single RNN with models, Naive Bayes Bernoulli, Logistic Regression and much more. In an attempt to solve this problem by using any classifier model, the first step is of text data cleaning like data duplication and text pre-processing and then to convert the text to a vector and apply one of the Machine learning algorithms on the text data. Cleaning can be done using the following features–

- Punctuation removal
- Tokenization
- Stemming
- Lemmatization
- Spelling correction

To convert a text to a vector following are the different approaches that could be used. We will be using TF-IDF as an approach to convert -

- BOW (Unigram and Bigram)
- TF-IDF
- word2Vec
- Avg word2Vec

III. Potential explorations

Quora provided a large amount of training and test data to identify a question as a sincere or insincere question. The training dataset consists of following three data fields:

1. qid – unique question identifier
2. question_text – Quora question text
3. target – a question labeled "insincere" has a value of 1, otherwise 0.

The train dataset is about 1.31 million, and the test data is 560.

Few of the examples from the train dataset which are sincere and insincere are as follows:

Sincere Questions:

- What is the best way to propose a girl without annoying her?
- What are some best college for aircraft propulsion(M.S)?

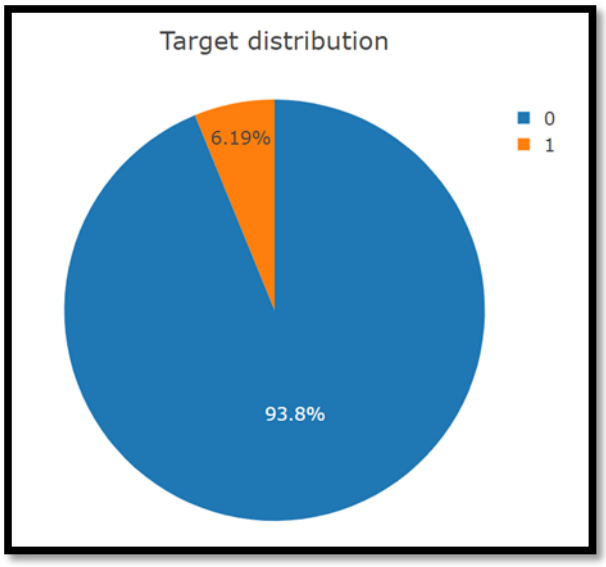
Insincere Questions:

- AreÂ JatÂ andÂ GujjarÂ girls beautiful?
- Why is Jinping the biggest serial religion murderer and rapist leader of the world after Mao?

There are also few questions in trainset which looks Sincere but has been classified as insincere because of noise in the dataset.

So, because of this increasing ambiguity in data classification based on text, we need to implement some methods which will at least reduce the error rate and will provide a higher F1 score i.e. a higher value of precision. Here, we look at the F1-score performance metric as we have a highly unbalanced dataset.

Also, after doing some basic exploratory analysis we see that there are no null values in the train data columns. Looking at the frequency distributions it was seen that the Sincere Questions count is 1225312 whereas for Insincere Questions count is 80810. This clearly shows an unbalanced dataset classification. Sincere being 93.8% of the data and insincere being merely 6.19%. We also thought of doing downsampling and upsampling on the train data but the issue is we have to test on the unseen test data. Once done with the submission on Kaggle, Quora tests it with its other test dataset of 376K test records and gives us a final score.



Also, to have a look at the most frequent words and n-grams of words we plot wordcloud to have a look at the most sincere and insincere words and their unigram, bigram and trigram counts. After having a look at the top 10 words of the unigram, we observed that some of the top words are common across both the classes like 'people', 'will', 'think', etc. The other top words in sincere questions after excluding the common ones at the very top are 'best', 'good', etc. The other top words in insincere questions after excluding the common ones are 'trump', 'women', 'white', etc. Then we looked in details of both train and test data based on frequency of Words, Unique words, Characters, Stop words, Punctuations, Upper case words, Title case words and Average length of the words. And then created a box plot for the same. We came to an inference that insincere questions have a larger number of words and characters compared to sincere questions. So, this might be an identifier in classifying text questions.

The method which we are planning to implement is also similar to those of the above-mentioned methods, but, by using NLTK/Keras in majority. By performing some Natural language processing techniques including Tokenization, Lemmatization, Part of Speech Tagging, n-grams, looking out for stopwords, vectorization and using some embeddings etc. in order to achieve a good accuracy as well as a higher F1-score in predicting the model which will determine whether the question could be classified as sincere or not. In our implementation till now, we have imported the Quora train dataset and tried to do some basic exploratory data analysis and visualizing various plots based on given text and wordcloud implementation which gives us an idea of words which are highly sincere and which are highly insincere by separating the train data based on target class as '1' and '0'.

Visualization of n-grams for Sincere and Insincere Questions:

Top 10 Frequent Words from Sincere Questions when doing Unigrams, Bigrams and Trigrams:

	word	wordcount
0	best	60816
1	will	45675
2	people	37960
3	good	34827
4	one	28840
5	make	25696
6	think	21641
7	many	20788
8	much	20108
9	someone	19728

	word	wordcount
0	best way	6973
1	year old	2972
2	will happen	2084
3	many people	1931
4	computer science	1870
5	even though	1859
6	known for?	1822
7	united states	1797
8	long take	1796
9	high school	1775

	word	wordcount
0	tips someone starting	716
1	someone starting work	713
2	useful tips someone	713
3	advice give someone	640
4	short-term business travelers	519
5	hotels short-term business	519
6	good hotels short-term	519
7	give someone moving	519
8	good bad neighborhoods	515
9	best known for?	400

Top 10 Frequent Words from Insincere Questions when doing Unigrams, Bigrams and Trigrams:

	word	wordcount
0	people	11036
1	trump	4893
2	women	4757
3	will	4590
4	think	3774
5	many	3552
6	white	3351
7	men	3152
8	indian	2984
9	muslims	2828

	word	wordcount
0	donald trump	1076
1	white people	673
2	black people	653
3	many people	383
4	united states	360
5	even though	335
6	trump supporters	335
7	year old	330
8	president trump	328
9	hillary clinton	305

	word	wordcount
0	will donald trump	43
1	black lives matter	42
2	long will take	38
3	kim jong un	36
4	12 year old	35
5	people still believe	33
6	14 year old	33
7	united states america	31
8	ask stupid questions	30
9	think donald trump	30

IV. Model Approaches

1. Baseline Model

The objective of the model is to predict the class of the question asked, we decided to start by selecting logistic regression as it is the base model in classifying texts for many text classification problems as it is the most common and widely used classification data model. It will help us to classify the questions into sincere and insincere. Logistic Regression is one of the most simple and commonly used Machine Learning algorithms for two-class classification. It is easy to implement and can be used as the baseline for any binary classification problem. Its basic fundamental concepts are also constructive in deep learning. Logistic regression describes and estimates the relationship between one dependent binary variable and independent variables. [3]

For our baseline model, we implemented simple Logistic Regression. Performing k=5 splits or 5-folds CV and applying the Logistic Regression Classifier. We pay more attention to the F-score since we have more than 94% as sincere and remaining as insincere which will give a higher accuracy even though all the values are classified as sincere.

So, we look at the confusion matrix, accuracy and F1-score of the validation dataset:

```
array([[1138011,   87301],
       [  56613,   24197]], dtype=int64)
```

F1-score: 0.2516 and Accuracy: 0.8898.

So, we can see that we have higher false positives and true negatives count compared to train dataset. Moreover, setting the threshold value is very important for better F1 and accuracy.

Examples where the Baseline Model worked well and not well:

As the F1-score for the Baseline Model was just 25%, many were classified incorrectly yet almost similar questions were classified correctly.

The following questions were classified correctly as sincere:

- Do you have an adopted dog, how would you encourage people to adopt and not shop?
- Why does velocity affect time? Does velocity affect space geometry?

Although the following questions looked sincere, they were classified as insincere.

- What star were you born under?
- What is the greatest amount of blood you can donate?

What went wrong?

1. This dataset is highly unbiased dataset and so we look at the F-score to monitor the performance of our model.
2. The one thing which was not considered in the baseline model was the addition of features such as n-grams which could improve the accuracy and F1-score.
3. Here, our model shows high precision and low recall making it comparatively accurate.
4. Maybe its neighbouring words might have been classified as insincere or sincere. i.e. more number of sincere or insincere words in a question classifying the question as sincere or insincere.

Next Steps:

1. We can look at different hyper-params to improve performance.
2. Also, addition of word vectorization features, ngram features and other word embedding we plan to increase the F1-score.
3. Also, to look for F1-score with different combinations of models and embeddings and to see which model works better.

The main issue with using Logistic Regression is that it classifies the record as either sincere or insincere (binary classification), but it does not give us the probability of a question being classified to either of the class. Probability would have helped us to better have a look at the chances of question being sincere or insincere. Moreover, a look at the neighbouring words is something which it needs to take care off. Also, the interpretation is more difficult because the interpretation of the weight is multiplicative and not additive. Logistic regression can suffer from complete separation. If there is a feature that would perfectly separate the two classes, the logistic regression model can no longer be trained. This is because the weight for that feature would not converge, because the optimal weight would be infinite.[4]

Although Logistic regression will be the go-to baseline model for any machine learning algorithm, it does exhibit some disadvantages such as its reliance of proper presentation of data and is well known for its vulnerability to overfitting.
A bag-of-words representation is simple to generate but far from perfect. If we count all words equally, then some words end up being emphasized more than we need.The main characters do not stand out by simple frequency count alone. This is problematic.So, ideally, we'd like a representation that highlights meaningful words.

Improvement over Baseline Model for Logistic Regression by using n-grams and TF-IDF features:

Term Frequency – Inverse Document are the components of the resulting scores assigned to each word. Term Frequency summarizes how often a given word appears within a document.

Inverse Document Frequency downscales words that appear a lot across documents. TF-IDF are word frequency scores that try to highlight words that are more interesting, e.g. frequent in a document but not across documents. It is a simple twist on the bag-of-words approach. Instead of looking at the raw counts of each word in each document in a dataset, it looks at a normalized count where each word count is divided by the number of documents this word appears in.

The TF-IDF Vectorizer will tokenize documents, learn the vocabulary and inverse document frequency weightings, and allow you to encode new documents. [5]
We implemented Logistic Regression with TF-IDF Vectorization and n-grams and other word vectorization features. The model seems to be better than the normal Baseline Model. Using n-grams it helped us to increase the overall accuracy and F1 score of the model looking out for previous one and two words and deciding whether it is sincere or insincere.
Performing k=5 splits or 5-folds CV and applying the Logistic Regression classifier. So, we look at the confusion matrix, accuracy and F1-score of the validation dataset:

F1-score: 0.561 and Accuracy: 0.937.

Implementation of Naive Bayes Classifier:

As the objective of the model is to predict the label of the question asked I decided to go forward with most simple and straightforward approach of applying the Naive Bayes classification algorithm on the above data to predict the insincere questions.

The Naive Bayes algorithm is called "naive" because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. It is a Probabilistic Algorithm which calculates the conditional probability based on previous knowledge.

We implemented Bernoulli Naive Bayes Classifier with TF-IDF Vectorization and other word vectorization features. It gave us the following results:

F1-score: 0.525 and Accuracy: 0.918.

It can be clearly seen that even though Naive Bayes Classifier worked better than the Baseline Model, it showed a lesser F1-score (Accuracy) when compared to Logistic Model with n-grams.

NBSVM Model :

NBSVM is an approach to text classification proposed by Wang and Manning[1] that takes a linear model such as SVM (or logistic regression) and infuses it with Bayesian probabilities by replacing word count features with Naive Bayes log-count ratios. Despite its simplicity, NBSVM models have been shown to be both fast and powerful across a wide range of different text classification datasets. This proves that NBSVM is a robust performer.

For smaller snippets Naïve Bayes model performs better and for longer snippets, SVM works well. The NBSVM Model have been shown to be both fast and powerful across a wide range of different text classification datasets and performs well on longer documents, for sentiment, topic and subjectivity classification.

On using the model along with TFIDF vectors, we could see that the F1 score that could be achieved with this model was 0.62 which was better compared to previous models. However, we were still in the top 20% on the leaderboard and had to improve a lot. So, further we decided to use a CNN with embeddings provided to us in the dataset to see how it works.

CNN Model with Pre-Trained Embeddings:

CNN is a class of deep, feed-forward artificial neural networks (where connections between nodes do not form a cycle) & use a variation of multilayer perceptrons designed to require minimal preprocessing. Earlier CNN was used for Computer Vision Tasks however they have recently been applied to various NLP tasks and t surprise they gave better results then the other ML models. So, we decided to move forward with using the Neural Network Model Implementation to our problem to evaluate the results and to see if there is any improvement. In this solution, we also used the given 'Glove' embedding to get better results. We used Filter sizes as 1,2,3 and 5 and applied to each of the four convolutional 2D layers, number of filters as 42, max features being 40000, max length – 70 and embedding size – 300. We used the Tanh activation function for each of the four convolutional layers and passed on to the four Maxpooling layers and used Dropout of 10% and Dense with activation function 'sigmoid'. We used the 'adam' optimizer for our model. To perform testing on the test model, we used Batch size as 256 and epoch being considered is 2. So, after submitting on the Kaggle Competition we got the following scores:

F1 Score: 0.673

So, the cnn model with the Glove embedding worked best compared to the other three embeddings to our problem.

Model Evaluation:

Models	F1 Score
Baseline Model - Logistic Regression	0.251
Logistic Regression with TF-IDF Vectorization and n-grams	0.561
Naive Bayes with TF-IDF Vectorization	0.551
NBSVM Model with TF-IDF Vectorization	0.62
CNN Model with Glove Embedding	0.673

Although Logistic regression is the go-to baseline model for any machine learning algorithm, it does exhibit some disadvantages such as its reliance of proper presentation of data and is well known for its vulnerability to overfitting.

A bag-of-words representation is simple to generate but far from perfect. If we count all words equally, then some words end up being emphasized more than we need. The main characters do not stand out by simple frequency count alone. This is problematic. So, ideally, we'd like a representation that highlights meaningful words. The same when tried on Naïve Bayes Model and NBSVM Model, Naïve Bayes with TF-IDF Vectorization didn't showed better on the F1-score than the Logistic Regression with TF-IDF Vectorization. However, a combination of Naïve Bayes and SVM Model worked better with an F1 score of 0.62. Further, we tried to use all the four different embeddings and saw that the Glove Embedding with CNN was the best and gave an F1 score of 0.673.

Analysis of why the solutions succeeded over the Baseline:

Firstly, the main reason for being successful was because of n-grams and word vectorization features. N-grams were implementation upto tri-grams and added in the list of feature word vectorizations and applied to Logistic Regression Model to predict the sincerity of Questions. Moreover, other features such as 'num_words', 'num_unique_words', 'num_chars', 'num_stopwords', 'num_punctuations', 'num_words_upper', 'num_words_title' and 'mean_word_len' were added to the list of features to improve overall score. Moreover, adding to this was the TF-IDF word vectorization features.

Errors that remain open and what can be done:

For the records which are still not classified, we need to use a better word vectorization features or a combination of word embeddings and models to increase the f1-score and Precision. Also, a train data with more questions and a balanced data would be useful to classify it as sincere or insincere. We will try to look and add in more features studying the False Positives and True Negatives questions from the predictions.

On running the test file on the model, we could see that the questions were classified as sincere and insincere in a more refined sense.

The following questions were classified as Sincere:

- Why do customs areas at airports not allow pictures?
- What are the weakest body parts when fighting?

The following were the Insincere questions.

- Why do so many women become so rude and arrogant when they get just a little bit of wealth and power?
- Shouldnâ€™t we blame those who knew how bad Trump was and didnâ€™t go to vote for todayâ€™s American situation?

Although it works perfectly, we could see that there are certain improvements that could be made to the model. Even after running the model on our dataset, In one of the questions - "Do Armenians consider themselves Europeans, western Asians/middle eastern or both?", it was classified as Insincere. Although it seems to be a perfectly genuine question (to which people have actually given wonderful answers), the model does not perceive it as a sincere one. This proves that the model needs some improvement.

We could see from the above examples that the way the questions were classified are becoming logically coherent. This shows that there is a definitive improvement in the classification model. This resulted in a higher F1 score compared to the previous model although the model can be trained further for a higher F1 score.

Conclusion:

Since the dataset at hand is highly biased, a fairly large training dataset with just 6% of the data being classified as Insincere, the level of F1 score that could be achieved was not very high with most of the models since there was always some sort of performance degradation. The best being in our case the CNN Model with Glove Embedding. So, our proposed final model with embedding features allowed us to discover the optimal combination of simple text processing options with machine learning features for sentiment classification and to utilize them in order to establish stronger results.

In the future we aim to make use of our dataset-building approach in other domains, including LSTM models with embedding features, and to thereby increase the amount of data that can be used for sentiment analysis. This could potentially enable the successful application of sentiment analysis models in classifying the questions as Sincere or otherwise. We are also interested in tackling the specificities of short-text and long-text processing to identify Verbs, nouns etc., which could prove useful in building the model.

References

[1] C, V. (2018). Quora Insincere Questions Classification Problem solving Using Naive Bayes. [online] Medium. Available at: https://medium.com/@vbch.bi/quora-insince re-questions-classification-problem-solving-using-naive-bayes-bf3f97f9e7b8 [Accessed 6 Mar. 2019].

[2] Kaggle.com. (2019). Quora Insincere Questions Classification | Kaggle. [online] Available at: https://www.kaggle.com/c/quora-insincere-questions-classification/data [Accessed 6 Mar. 2019].

[3] Singh, R. (2018). Logistic Regression in Python. [online] Go Algorithm. Available at: https://goalgorithm.wordpress.com/2018/10/07/logistic-regression-in-python/ [Accessed 3 May 2019].

[4] Molnar, C. (2019). 4.2 Logistic Regression | Interpretable Machine Learning. [online] Christophm.github.io. Available at: https://christophm.github.io/interpretable-ml-book/logistic.html [Accessed 27 Mar. 2019].

[5] Casari, A. and Zheng, A. (2015). *Feature Engineering for Machine Learning*.[online] O'Reilly | Safari. Available at: https://www.oreilly.com/library/view/feature -engineering-for/9781491953235/ch04.html [Accessed 9 Apr. 2019].