Quora Insincere Questions Classification Project - Report

Objective:

Quora is a knowledge sharing platform for asking questions and connecting with others who contribute unique insights and quality answers. Unfortunately, insincere questions are posted from time to time and these are the questions that are based upon false premises, intend to make a statement rather than look for helpful answers or even try to insult against a specific group of people. Such questions are against Quora's core values and might pose a threat to the Quora community. The goal of this project is to identify insincere questions based on the question wordings so that Quora can develop scalable methods to detect toxic and misleading content.

Research Question: Identify insincere questions based on how the questions are worded.

Data:

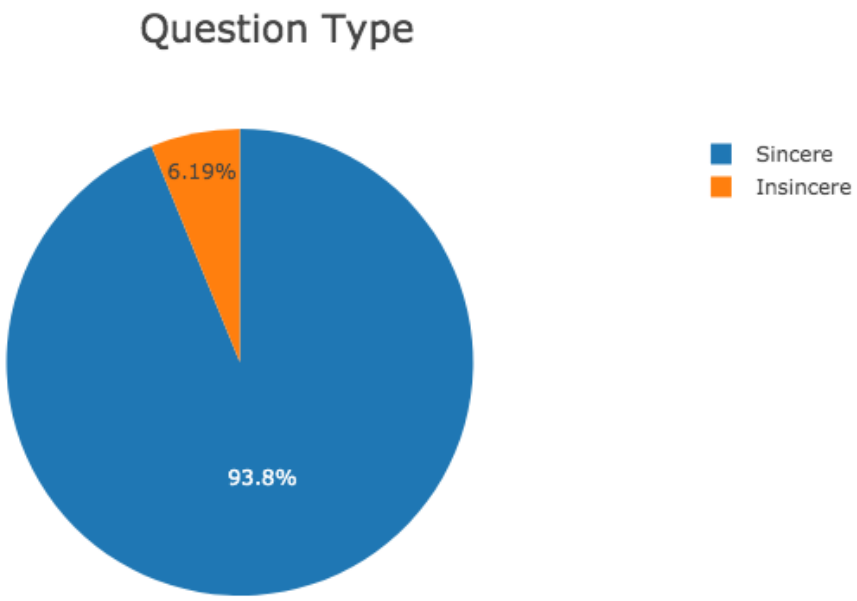
The datasets for this project are available on Kaggle (https://www.kaggle.com/c/quora-insincere-questions-classification/data)
These datasets are:

- train.csv: training data. For each question the unique question identifier, question text and label (whether the question is insincere or not) are provided.
- test.csv: testing data. It contains the same fields as train.csv's except that test_users.csv does not include the label.

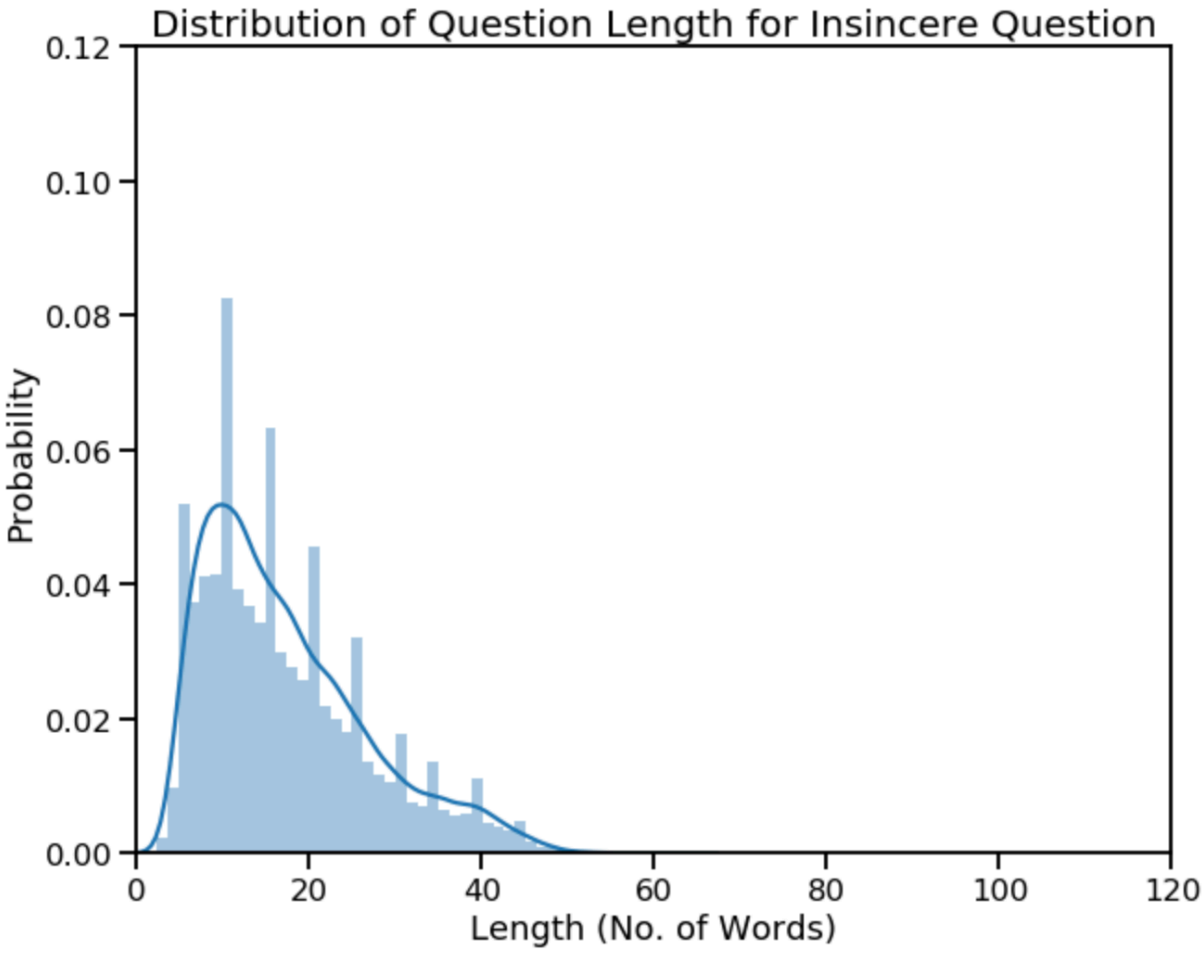
Data Wrangling:

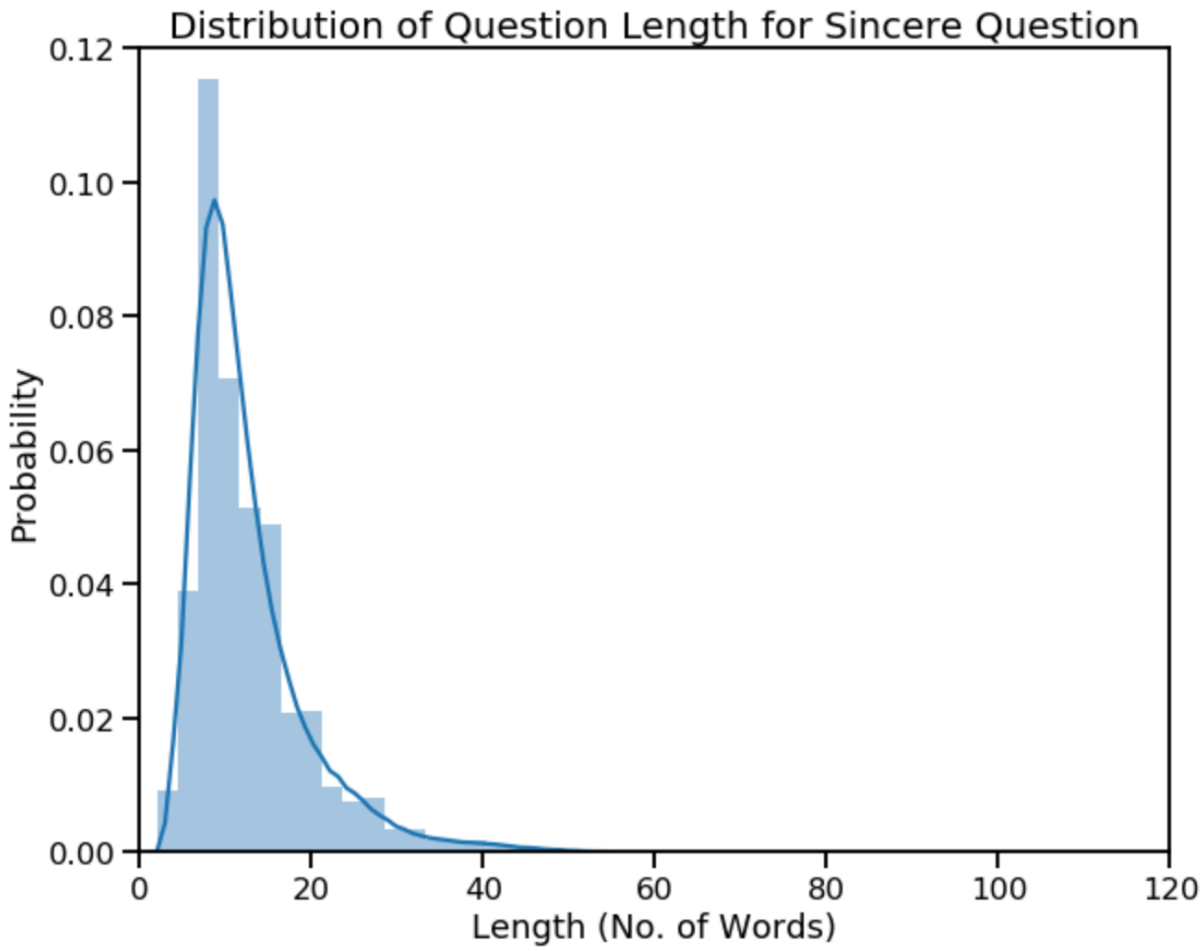
- Change contracted forms (e.g. I'm) to long forms (e.g. I am).
- Remove special characters, punctuations and stop words.
- Tokenize and Lemmatize the words.
- After removing special characters, punctuations and stop words, some questions become NaNs. Replace these NaNs with string 'na' for embedding purpose.

Exploratory Analysis:

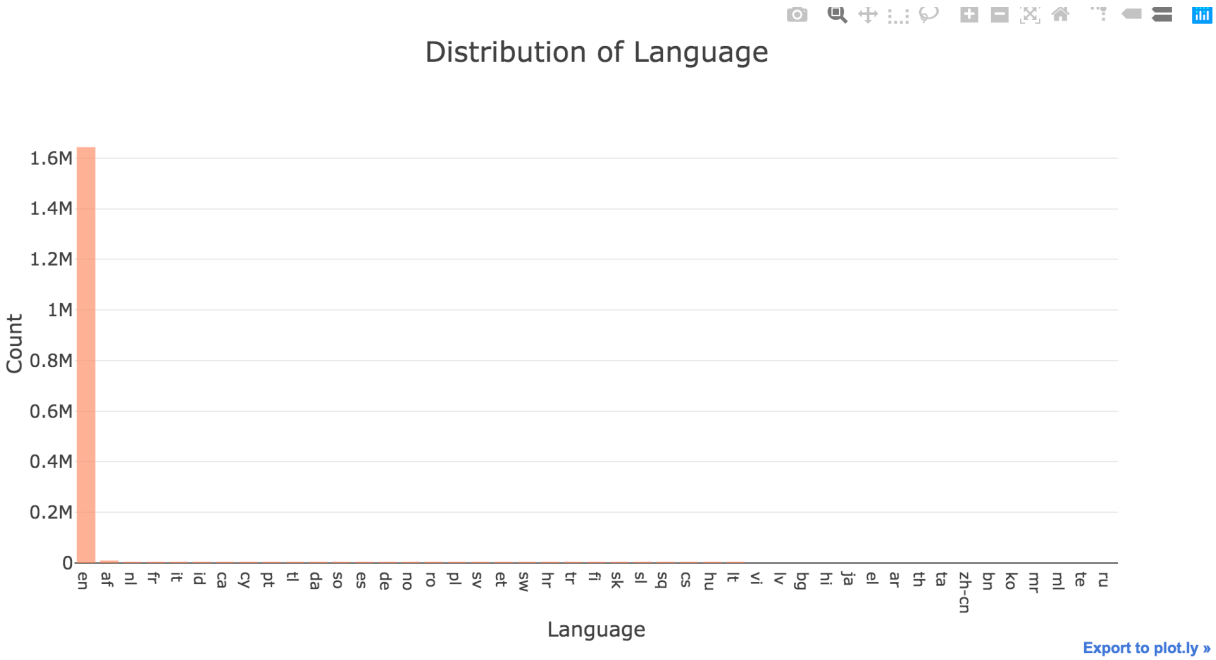


- We have an imbalance dataset with only 6% of questions labeled as ‘insincere’ and 94% as ‘sincere’.


Distribution of Question Length for Insincere Question
Probability
0.12
0.10
0.08
0.06
0.04
0.02
0.00
0
20
40
60
80
100
120
Length (No. of Words)



- The distributions of question length between insincere and sincere questions are quite different; insincere questions have a lower and broader peak.



- Most questions are in English. Since we only have very few questions in other languages, it is safe to ignore them.

Word-cloud for Insincere Questions

Word-cloud for Sincere Questions

- The word-cloud for insincere questions is mainly made up of words related to nationality/ race and gender. On the other hand, the word-cloud for insincere questions is more diverse and consists of different topics like knowledge, career, relationships, etc.

Modeling:

The goal of the project is to identify insincere questions based on question wordings. To achieve this the raw text data are transformed into feature vectors which can be used in a machine learning model. The following methodologies are implemented to transform text data into feature matrix: Term Frequency (TF), Term Frequency-Inverse Document Frequency (TF-IDF) and word embeddings. Different machine learning/ deep learning models are trained using different feature matrices. Predictions based on the test data are submitted to Kaggle and are evaluated using F1 Score.

Model:

TF Model

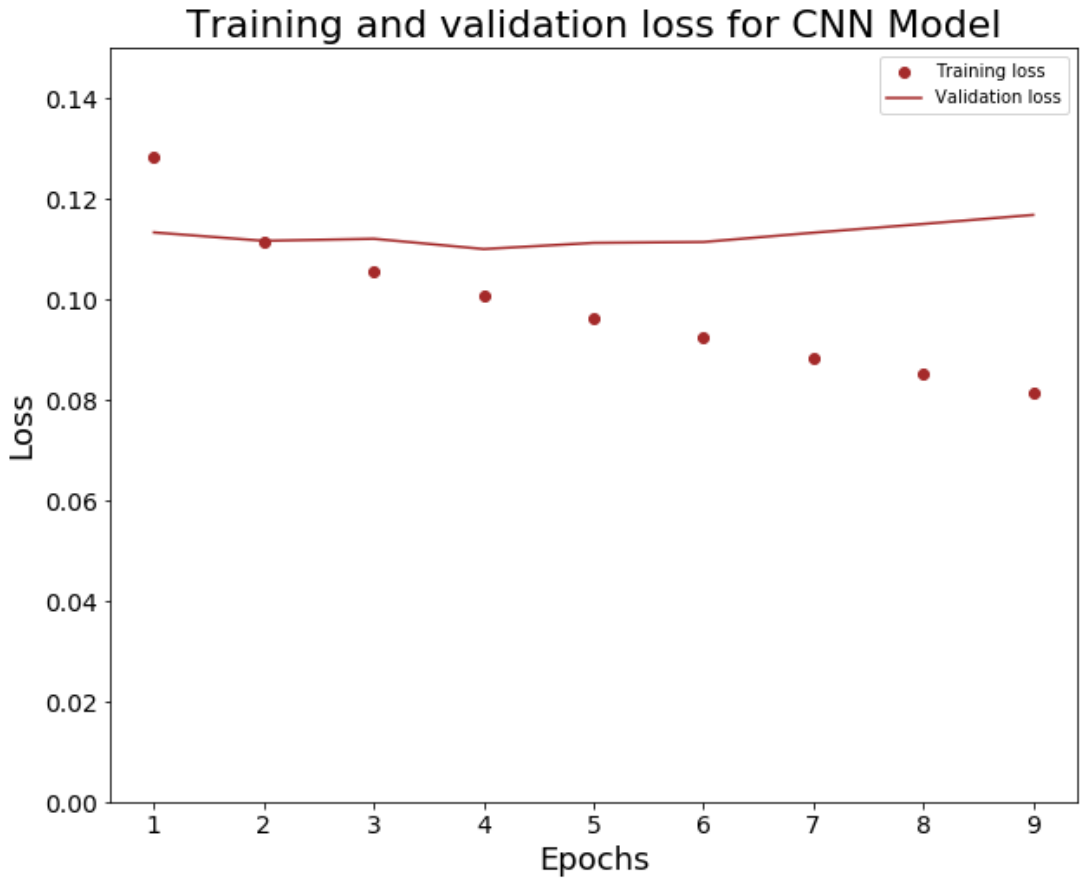
- A logistic regression model, a multinomial naïve bayes model and a random forest model are trained respectively using the TF matrix.
- Although the multinomial naïve bayes model gives the best out-of-sample F1 score (0.5360), the model is not really useful in identifying insincere questions as the F1 score is close to 0.5.

TF-IDF Model

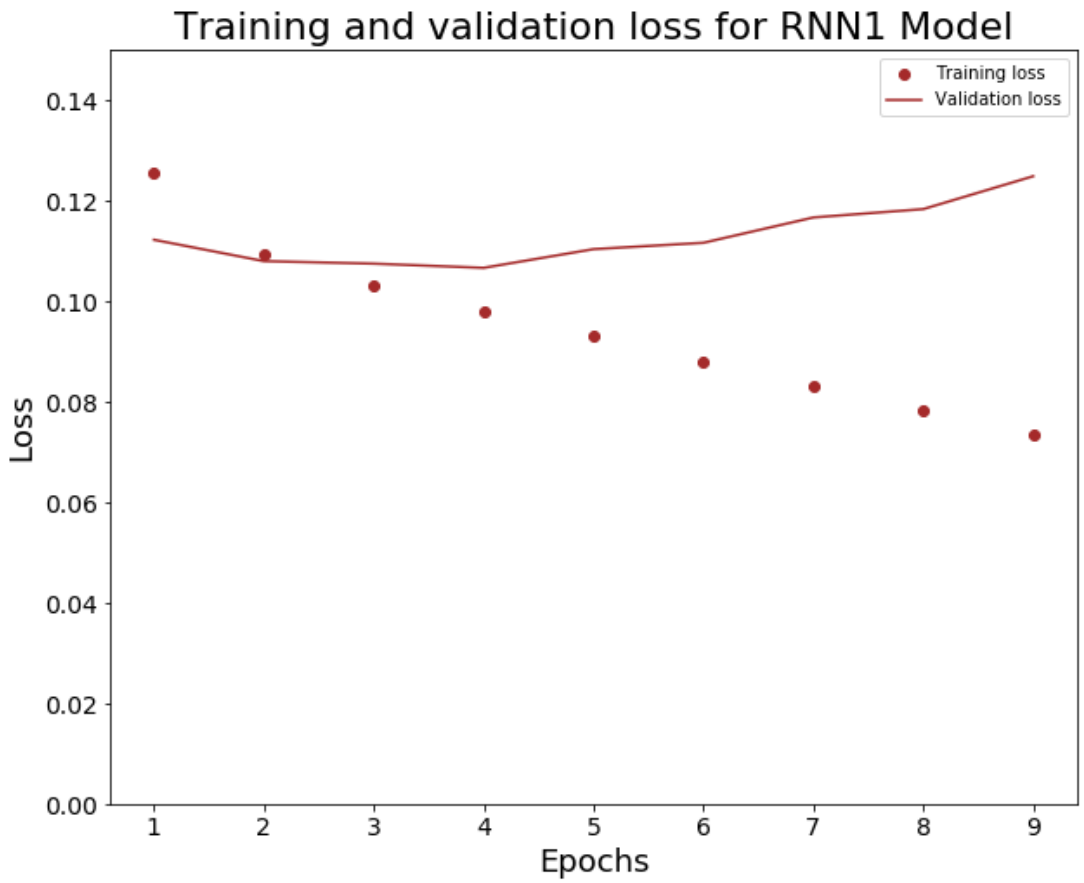
- A logistic regression model and a random forest model are built using the TF-IDF matrix.
- Yet, both models perform even worse than a random model (logistic regression: 0.4603) (random forest: 0.2476).

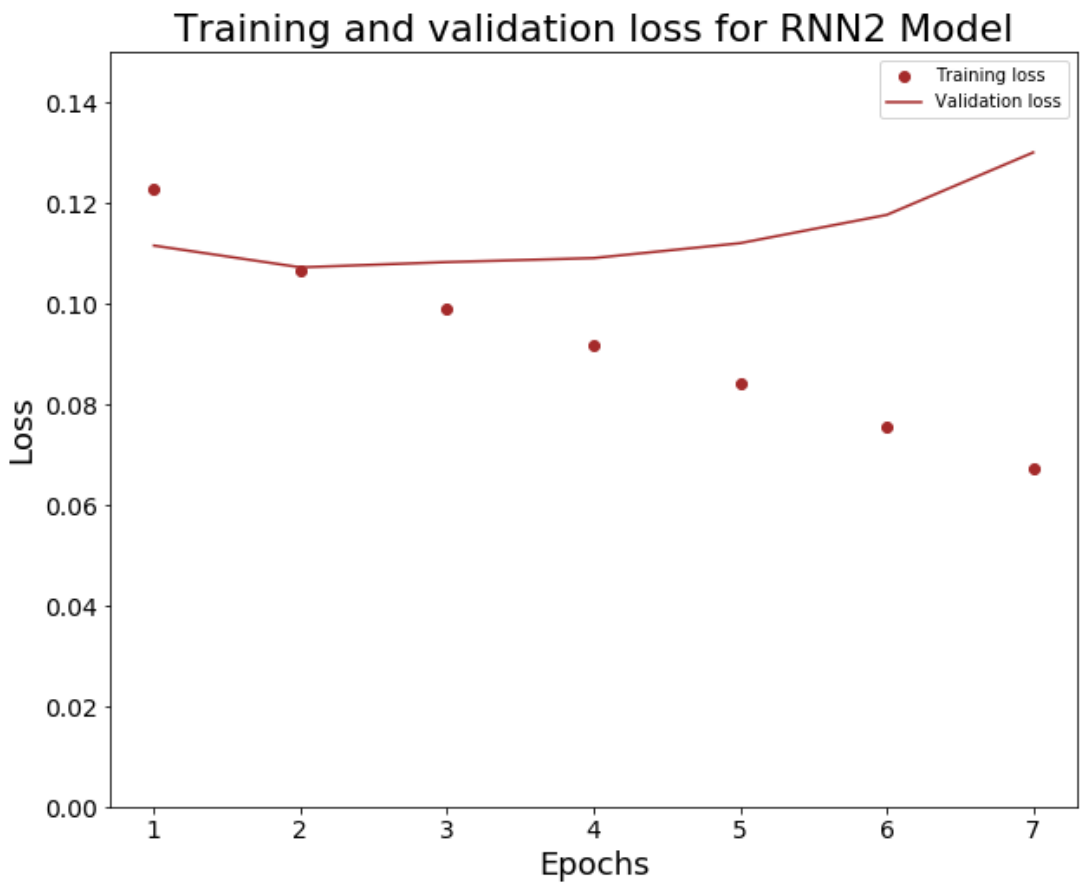
Word Embeddings

- Pre-trained Glove and Paragram word embeddings are utilized to transform text into dense vectors, and the transformed matrix is then fed into different Neural Networks. Early stopping is implemented to avoid overfitting.
- Different F1 scores are calculated based on different classification thresholds. The threshold which gives the highest validation F1 score will be chosen.
- A Convoluted Neural Network with 1-Dimensional convolution layers, maxpooling layers and dropout is built. A threshold of 0.4 gives an out-of-sample F1 score of 0.6642.



- Two Recurrent Neural Networks are constructed. One of the models (RNN1) consists of LSTM and dropout while the other one (RNN2) has bidirectional LSTM layers and dropout. Again, a threshold of 0.4 gives the best F1 score - RNN1 gives a score of 0.6689 while RNN2 gives a slightly higher F1 score of 0.6697.





My best performing model is RNN2 Model (F1 score: 0.6697).

Future Research Recommendation:

- Set up GPU: If given more time and resources I will try setting up GPU for faster training of my Neural Networks.
- Try more RNN models: If given more time and resources I will also try implementing more RNN models with different layers such as the attention layer and the Spatial Dropout.