Data Exploration on Quora Insincere Questions Dataset

Vasin Srisupavanich
vs2n19@soton.ac.uk
ID: 31300162

Yu Hong Leung
yhl1u19@soton.ac.uk
ID: 31291279

Peng Chen
pc3n19@soton.ac.uk
ID: 31142354

1 INTRODUCTION

One of the major problems many online social platforms facing today is the difficulty to handle toxic and misleading contents. Quora is one of these websites where one can ask questions on any topics and receive answers from other users across the world [2]. The key challenge for Quora is to filter those insincere contents out from its platform. To alleviate this issue, Quora has created a competition on Kaggle called ***Quora Insincere Questions Classification*** [3], with the aim to encourage participants to create machine learning models to identify and flag insincere questions.

In order to create an accurate and unbiased model, we first need to start by exploring the dataset, and find useful insights from doing analysis on the data. In this paper, we will explain how we did the exploratory data analysis, and show interesting results and visualisations, which could potentially help us develop the model to classify the insincere questions more accurately. Discussions on evaluation metrics, word embeddings, and topic modelling are followed to provide a fuller scope of this project. All of the code and results of the analysis are available on Google Colab (https://colab.research.google.com/drive/1wXiJFhg8Sh76ilB2ShJgLObeu9ctL69y).

2 EXPLORATORY DATA ANALYSIS

2.1 Dataset

The dataset provided by Kaggle contains 1,306,122 labeled training examples, and a test set with 375,806 unlabeled examples in CSV format. The training data contains 3 columns: unique question ID, question text, and a target label of 0 and 1. The target of 0 represents sincere question, while the target of 1 represents insincere question. Figure 1 shows the structure of the CSV file along with the data sampled from the training set. The following shows examples of both sincere and insincere questions.
Example of insincere questions:

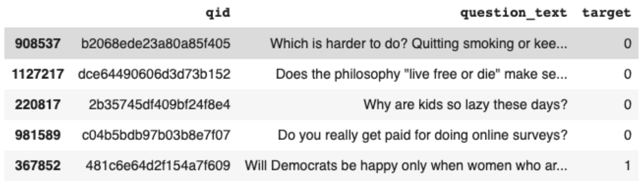
- Has the United States become the largest dictatorship in the world?
- Is there such thing as straight women like, do they exist?
- Was Adolf Hitler really bad?

Example of sincere questions:

- Which novels are good for amateur readers?
- What is more important between work and love?
- What are the pros and the cons of homeschooling?

As stated in the competition's page [3], an insincere question is defined as a question intended to make a statement rather than look for helpful answers. Some of the characteristics of insincere question has a non-neutral tone, is disparaging, isn't grounded in reality, and usually contains sexual contents.

Apart from the training and test data, Kaggle also provided us with 4 different pre-trained word embeddings to help with developing the model. These include Google's word2vec, Stanford's GloVe, Facebook's fastText, and Paragram word embeddings. We



	qid	question_text	target
908537	b2068ede23a80a85f405	Which is harder to do? Quitting smoking or kee...	0
1127217	dce64490606d3d73b152	Does the philosophy "live free or die" make se...	0
220817	2b35745df409bf24f8e4	Why are kids so lazy these days?	0
981589	c04b5bdb97b03b8e7f07	Do you really get paid for doing online surveys?	0
367852	481c6e64d2f154a7f609	Will Democrats be happy only when women who ar...	1

Figure 1: Sampled data from the training set

will discuss in detail the benefits and the difference of using these embeddings, as well as, how we might use them when we develop the classification model in section 4.

2.2 Target distributions

Although the training data is quite large in size, the distribution of the targets is highly imbalanced. From the training set, there are only 80,810 questions (6.2%) which are insincere, while there are 1,225,312 sincere questions (93.8%) as shown in Figure 2. This represents one of the biggest challenge from this dataset which we have to keep in mind when developing the machine learning model, and is the main reason why the competition chose F1 score as an evaluation metric.

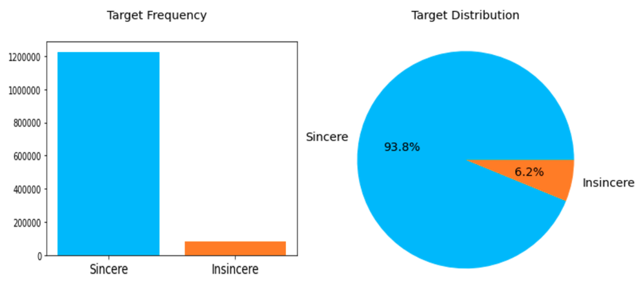


Figure 2: Target distribution from the training set

2.3 N-grams analysis

Performing an analysis on N-grams, or continuous sequences of words, can help us understand which words tend to be used in each type of the questions. In order to create the N-grams and visualise the results, there are several pre-processing steps which we performed. First, we needed to split the question text into tokens, and remove the stop words or commonly used words, such as 'the', 'and' and 'is'. Then we obtain the N-grams from NLTK's function [5], and maintain the count of each N-gram in Python's dictionary.

For this task, we performed 3 types of N-grams including unigram, bigram, and trigram on both type of questions. The result from performing bigram analysis is shown in Figure 3. In addition, we also created word cloud for unigram to visualise the frequency of the word in the insincere questions as shown in Figure 4. All of the results and visualisations for N-grams analysis and word cloud can be seen in our Google Colab notebook.

The key insights we gained from doing this analysis are as follow. First, we can see that insincere questions are dominated by words like Donald Trump, black people, white people, Indian, Muslim, and woman. On the other hand, words that appear frequently in the sincere questions are good, best, will, and people. In general, we can see that insincere questions are related to age, race and hypothetical scenarios, while sincere questions are related to advice, tips, suggestions and facts.

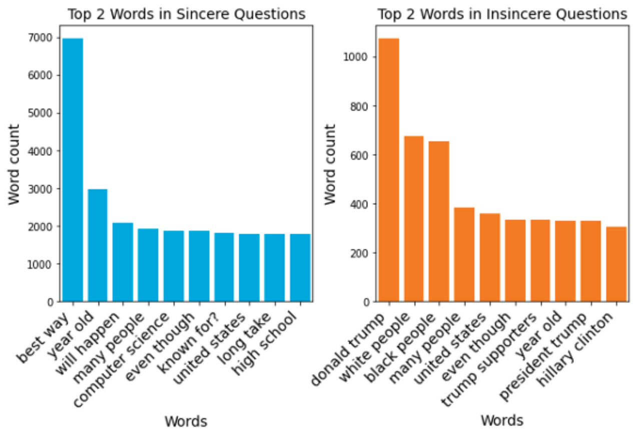


Figure 3: Top 10 bigrams in both type of questions

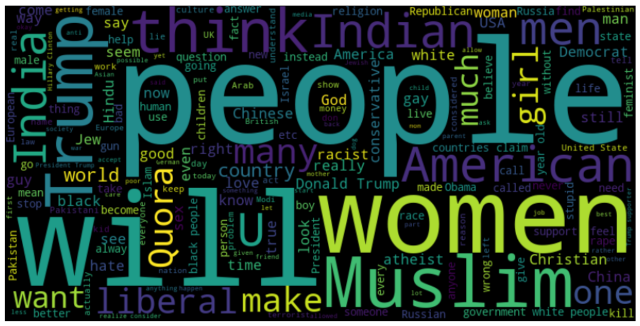


Figure 4: Word cloud from insincere questions

2.4 Metafeature analysis

To gain additional insight, we also created several meta features from the question texts, as well as, visualisations of the statistical distribution between two classes. These features include: number of words, number of unique words, number of characters, number of stopwords, number of punctuation, number of upper case words, and average length of the words for each question. The distributions of number of words and number of characters are shown as box plot in Figure 5. We can see that in general insincere questions tend to have higher number of words and characters compared to sincere questions, so these features could potentially be useful when we develop the machine learning model.

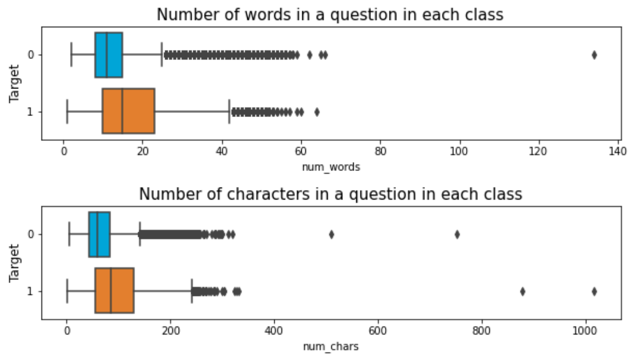


Figure 5: Number of words and characters in each class

3 EVALUATION METRICS

As mentioned in section 2.2, the main reason for using F1 score is because of the imbalance of the training set. If a classifier predicts all results to be sincere, the accuracy is going to be 93.8% on this training set. The accuracy shows that this classifier performs well, however, obviously such a classifier isn't what we want. Therefore, we need to use F1 metric for handling the imbalanced data [1].

The precision of a class defines how trustable is the result when the model predicts a data point which belongs to that class. It measures the proportion of correct answers in all positive predictions.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

The recall of a class expresses how well the model is able to detect a class. It indicates the proportion of correct answers in all true positive examples.

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

In order to consider both precision and recall at the same time, the F1 score is given by the harmonic mean of precision and recall.

$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{3}$$

4 WORD EMBEDDINGS

Word embeddings provide a way to represent words in numeric vectors that are able to capture the context, semantics, and syntactic property of the vocabularies in documents. Words that share similar meaning or context would be 'similar' within the vector space (by computing the cosine similarity between the word embeddings of two words).

Kaggle provides the competitors four kinds of word embeddings along with the dataset that can be used in the models. These are as follows:

- GoogleNews-vectors-negative300
- glove.840B.300d

- paragram_300_sl999
- wiki-news-300d-1M

The "GoogleNews-vectors-negative300" are pre-trained vectors using Word2vec with negative sampling which are trained on the Google News dataset with about 100 billion words. The main idea of Word2vec is to create word embeddings based on the local context of each word so that similar words will be close in the vector space. Word2vec can be trained using CBOW or the skip-gram model; Each gives different advantages and speed of training. The resulting vectors are of 300 dimensions for 3 million words and phrases.

The "glove.840B.300d" are vector representations using GloVe, which trains on the aggregated global word-word co-occurrence statistics (how often do words co-occur together). The resulting vectors are of 300 dimensions as well for 2.2 million vocabularies.

The "paragram_300_sl999" are trained on word pairs from a lexical portion of the Paraphrase Database (PPDB), which contains 169597 paraphrase pairs, and the hyperparameters are then tuned on SimLex-999 (SL999) word similarity task. The resulting vectors are of 300 dimensions.

The "wiki-news-300d-1M" are trained using fastText, in which the main idea is to use a bag of character n-grams to train and represent each word. The main benefit is that fastText can build better word representation for out-of-vocabulary words by averaging the vector representation of its character n-grams. The resulting vectors are of 300 dimensions for 1 million words.

4.1 Word Embedding Analysis

Before any preprocessing of the training data, the coverage of the pre-trained word embeddings is poor because of noisy data. Improving the coverage of the pre-trained word embeddings is important because out-of-vocabulary words can be represented less meaningfully in models like Word2vec, which will lead to poorer performance in subsequent tasks. The best coverage before preprocessing is the GloVe embeddings, with 32.77% of unique words and 88.75% of the whole training set covered. The Paragram word embeddings have the worst performance, with only 19.37% and 72.21% coverage for unique words and the whole training set respectively.

A number of preprocessing steps can be done in the training set to improve the coverage substantially. First, there are embeddings for words with upper letters, but no embeddings for the same word with lower letters only. Thus, it can be reasonable to embed the lowercase word identically if the lowercase word does not have an embedding already. In addition, there are missing embeddings for words which contain contractions, e.g. ain't, what's, or punctuation such as question mark and exclamation mark. Especially for fastText embedding, it contains no embeddings for contractions at all, which could affect the model's performance significantly. In this case, for contractions, we can create a lookup list to replace the contractions with other vocabularies which should be identical or extremely similar in meaning, for instance "he's" as "he is". Furthermore, there are special characters in text which could be messy because of encoding schemes. From inspection, GloVe and Paragram embeddings have the most unknown punctuation and special characters. These special characters can be replaced with characters which are similar in look or actual meaning, for example converting all kinds of apostrophes to the same standard. Finally, we can also include a lookup list for some frequently mispelled words or British/American spelling difference in the dataset, for example "colour" and "centre".

After all the preprocessing steps, the pre-trained embeddings achieved a coverage of around 99.5% of all text, and 60 to 70% of all unique words in the training corpus. In addition, we also created visualisation for each pre-trained embedding to compare how different words are located in the embedding space. The visualisation is created using t-SNE, a nonlinear dimensionality reduction technique, with common words selected from unigram analysis along with each of their top 5 most similar words from both type of questions. Figure 6 shows the visualisation of the GloVe embedding. Among all the embeddings, GloVe seems to give the best performance, in terms of how similar words are clustered together.

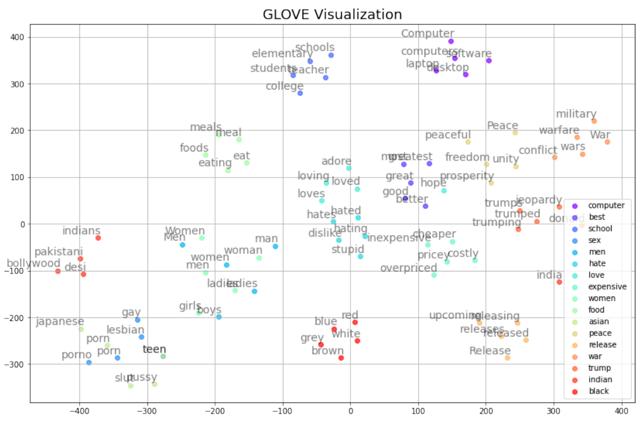


Figure 6: GloVe embedding visualisation using t-SNE

5 TOPIC MODELLING

As mentioned in section 2, insincere questions have the characteristics like exaggeration, discrimination, or sexual content. These characteristics could be reflected through words and phrases. As an important part of data exploration, we need to find the link between words and underlying features of insincere questions. One of the methods to map the text and sentiment is topic modelling which uses topic as a representation of characteristics.

Topic modelling belongs to unsupervised learning and aims to detect words and phrases patterns within a document. It assumes a single document can have several topics and gives the probability distribution of those topics on the document. For example, an article talking about pets may contain 50% dog topic and 50% cat topic. There are many approaches for topic modelling such as TF-IDF (Term Frequency and Inverse Document Frequency), NMF (Non-negative Matrix Factorisation), and LDA (Latent Dirichlet Allocation). Among them, LDA is the most popular topic modelling technique.

5.1 The Principle of LDA

LDA was first proposed by David Blei and Andrew Ng in 2003 [4]. In essence, it's a generative model and treats topic as unobserved latent variables that control the words. From our point of view, LDA simulates the process of creating an article by humans. When human writes, we first construct the central idea and paragraphs' general topic, and then select the proper words to complete the whole passage. LDA does the same thing.

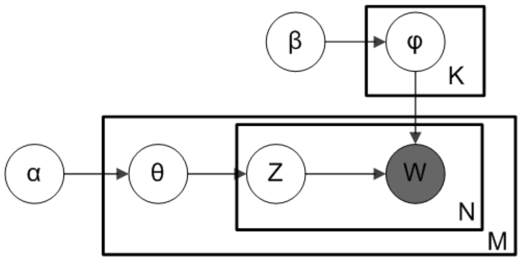


Figure 7: The graphical model of LDA

Figure 7 is the graphical model of the LDA, where α, β are hyperparameters, which represent the number of topics to be extracted from the corpus and the number of words composed in a single topic, respectively. θ, ϕ, z are latent variables and w represents visible words. The direction of arrows means the control or flow of conditional probability. θ is the distribution on topics for each document which is generated from a Dirichlet distribution with parameter α. Then, z represents the topic for each word with a discrete distribution given the parameter θ. Similarly, β is a parameter of Dirichlet distribution, which controls ϕ. ϕ is a Dirichlet distribution on words for each topic. It's much like bag of words, but it's in a probability form. Finally, LDA generate the words from the selected topic with parameters ϕ and z.

The training part is a process of statistical inference which used the given text to find the distribution of topic that maximise the likelihood. When training the parameters, two algorithms are mainly used for inference of latent variables, variational inference via EM (Expectation and Maximisation) algorithm and Gibbs Sampling.

5.2 Implementation and Result

During the implementation, we used the pre-processed text because it could create a smaller corpus, thus reducing the computation amount. After pre-processing, we first converted the insincere questions to vectors by sklearn's `CountVectorizer`. This function inputs the text and then outputs an matrix whose row represents the vector of each question. Secondly, we called sklearn's `LatentDirichletAllocation` to build a model. In the model, `n_components` corresponds to α in the LDA model in Figure 7. In our implementation, we set $\alpha = 8$. After training, the LDA outputs a probability distribution of words for each topic. Finally, the top 40 words in the 1st and 2nd topic were plotted in two WordClouds, as shown in Figure 8.

We can summarize that the words represented in the insincere questions could be possibly grouped into the following topics:

- **Gender equality**: women, men, male, gender, old;
- **Racial discrimination**: Indian, Hindus, Asian, Pakistani;
- **Faith conflict & religion**: Christian, Muslims;
- **Violence**: guns, terrorists, killed;
- **Pornography**: sexual, penis;

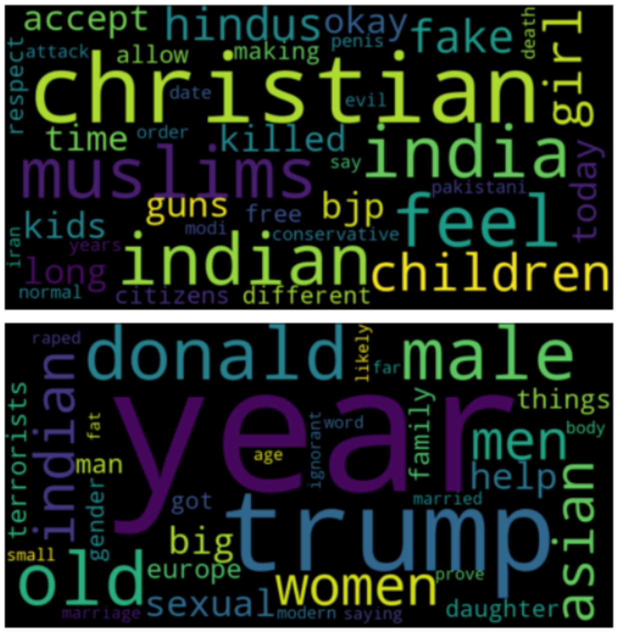


Figure 8: The WordCloud of 1st topic and 2nd topic

- **Non-neutral tone**: Donald Trump, Modi;

6 CONCLUSION

The results from the analysis show us several useful insights and features which we plan to leverage when we develop the learning machine. For instance, we learnt that the dataset is highly imbalanced, which could make our model biased toward one class, hence we would need to deal with this problem with techniques, such as data resampling and synthetic data generation. Moreover, from the n-gram analysis and topic modelling, we have seen that there are certain word patterns and topics which are very common in the insincere questions, therefore word frequency features, such as bag of words or TF-IDF could be a sensible baseline feature. Another potentially useful feature is the number of words or characters in the question, because they tend to be longer in an insincere question, as seen in the metafeature analysis. In addition, after exploring different pretrained word embeddings, we have seen that GloVe embedding provides us with the highest vocabulary coverage, and seems to cluster similar words relatively better as shown in t-SNE visualisation, so it would be reasonable to build our initial model with the GloVe embedding. Lastly, our analysis reveals that insincere questions tend to be related to topic about racial discrimination, faith and religion conflict, and political figures.

REFERENCES

[1] [n.d.]. Precision and recall. https://en.wikipedia.org/wiki/Precision_and_recall.
[2] [n.d.]. Quora. https://www.quora.com/.
[3] [n.d.]. Quora Insincere Questions Classification. https://www.kaggle.com/c/quora-insincere-questions-classification.
[4] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.
[5] NLTK [n.d.]. Natural Language Toolkit. http://www.nltk.org/.