Text Based Sincerity Prediction for Quora Questions

Chunan Huang
A13716180
Department of Computer Science & Engineering
chh179@ucsd.edu

Etsu Nakahara
A14140903
Department of Computer Science & Engineering
etnakaha@ucsd.edu

Chi Gao
A13907181
Department of Mathematics
chg043@ucsd.edu

Abstract

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.

Our task is to build a predictor that can filter out those insincere questions based on the text.

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

Part 1: Dataset

1.1 Basic intro to our dataset:

We decide to study a dataset provided in one of the Kaggle competitions. This dataset contains Quora questions that are labeled as either sincere or insincere. The dataset includes 1306122 data with the following three fields: qid (id of the question), question_text (text content of the question), and target (boolean data, where 1 indicates the question is insincere and vice versa). The way Quora has classified a question as insincere is as the following:

- Has a non-neutral tone
- Is disparaging or inflammatory
- Isn't grounded in reality
- Uses sexual content (incest, bestiality, pedophilia) for shock value, and not to seek genuine answers

The more detailed description can be found here:
[1]https://www.kaggle.com/c/quora-insincere-questions-classification/data

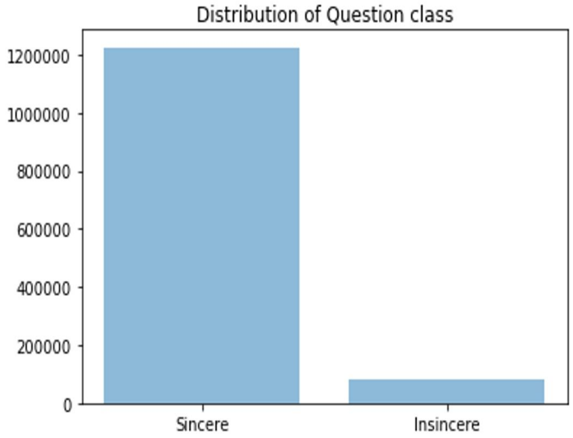
So far, the classification of sincerity, according to Quora, is done by a mix of machine learning and manual classification.

1.2 Exploratory analysis of data:

Basic Stats and Properties:

- All question text ends with a question mark since they are expected to be questions made in Quora. However, there are few exceptions where the question texts are simply statements appended by a question mark. (Since the users who entered these question texts do not intend to ask questions sincerely, these statements tend to be informal and hence most are labeled as insincere.
    - E.g. Fuck Obama all he did was screw this country up, with the Clintons.?

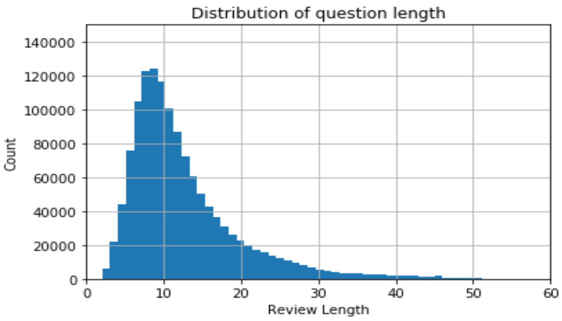
Target (= sincerity of the question)	Count in the dataset
0 (= sincere)	1225312
1 (= insincere)	80810

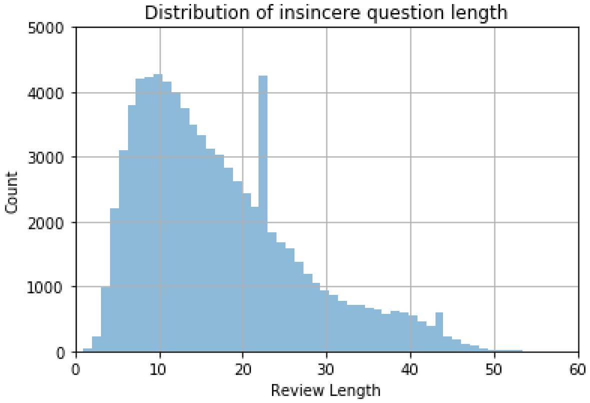


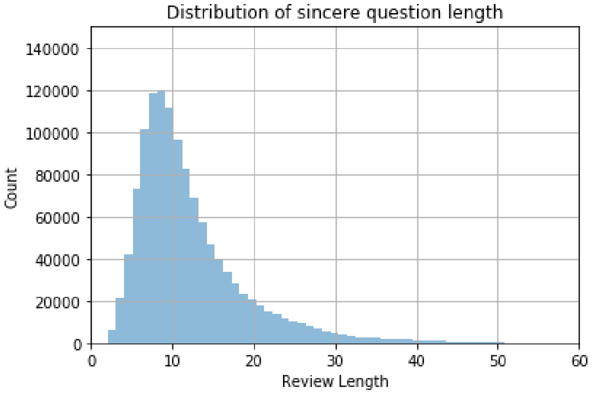
- Another challenge posed by the nature of this problem is that this dataset is extremely imbalanced. That is, most of the questions are sincere questions, and only a fraction (6.2%) of questions are insincere.

Discussion on the Length of the Question:

- The maximum length of the review is 134, the minimum length of the review is 1.
- To investigate the distribution of the review length:







- As we can see, both graphs peak around 10, and both graphs skew to the right;

however, insincere question length tend to spread out a bit wider thus have a more gradual decrease towards the right; whereas the distribution of sincere question length has a sharp drop right after the peak.

- Noted that for insincere review, there is an abnormal jerk right after 20. The reason is unknown.

	mean	variance
Insincere question length	17.27	91.55
Sincere question length	12.51	45.57

- The average length for insincere question length is significantly higher than the sincere question length. The variance is significantly higher than that of sincere question length due to their differences in sample size.
- There are few questions that are mislabelled, as the competition description has noticed. This problem was also addressed in a discussion thread in Kaggle. A possible cause to this is the error in Quora's classification method such as human error in the manual classification.
    - [2] https://www.kaggle.com/c/quora-insincere-questions-classification/discussion/72983

Observation & Interesting Findings:

- Many questions labeled as insincere includes words related to religion, race, country, and personal names. Some frequently used words are "Muslim," and "Trump."After all, all of these words are capitalized. Although there are also sincere questions, it is likely to be insincere when the question text includes more than 2 of such words.
- Many questions labeled as sincere also do includes seemingly offensive words such as e.g. ass and kill. These questions are usually asking facts rather opinions, and tend to have less personal feelings in it.
    - E.g. Why is scrolling such a pain in the ass on my 2017 MacBook?
    - E.g. How do I kill a flying beetle?
- Profane words, or offensive words, can be seen in both sincere and insincere questions but it is likely to be insincere when both the verb and object of the question are sensitive, or offensive, words. Similarly, when the subject and adjective of the question are sensitive words, the question tends to be insincere.
    - E.g. Do Germans dislike Black Americans than Africans?
    - E.g. Why are Australians not formal?
- Following the notion of capitalization mentioned earlier, questions with more capitalization tend to be insincere due to capitalized sensitive words and overly-emphasized word. A question might be insincere when wholly capitalized words (e.g. ASIAN) are used to emphasize some sensitive words.
- Stop words tend to have less effects on determining the sincerity. This is because the sincerity of a question is

usually determined by the topic or subject of the question.

- While some sensitive words are used in both sincere and insincere questions, questions that use multiple sensitive words tend to be insincere.
- Questions with “Quora” and offensive words are likely to be labeled insincere.
    - E.g. Why do I always think Quora is a piece of shit?

Part 2: Task & Models

2.1 Predictive Goal:

Since what we have in the data set are basically two fields: text of a question and an indicator that shows if the question is sincere or not, we would like to perform a task that, given a question, we predict if this question is sincere (1 represents NOT sincere and 0 represent sincere).

2.2 Evaluation method:

In order to evaluate the prediction, we decide to use F1 score, which is also used in the Kaggle competition. The main reason of using F1 score instead of accuracy is that the dataset is highly unbalanced with most of the questions labeled as sincere; therefore, even a model that simply outputs all 0 will achieve accuracy higher than 90%. Since our goal is to label insincere questions with few error, we need to achieve high F1 score:

F1 = 2 * (precision * recall) / (precision + recall), where

Precision = TP / (TP + FP); recall = TP / (TP + FN)

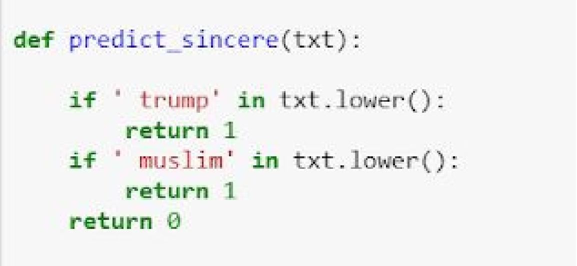
F1 score does not simply check for the overall accuracy; rather, it finds a balanced measurement to assess how accurate are the positive predictions. Thus, F1 score is a genuine reflector of how successful the model predicts positive result, which is the desired goal.

2.3 Baseline:

Since the purpose of the task is to find out the questions that are insincere, we first consider what can make a question insincere. We filter out all the insincere questions in the training set and quickly go over it; then we realize there are many “joking” or “racist” questions against some specific figures or races.

The most common ones we find are “Trump” and “Muslim.” After filter out all questions that contain “Trump” or “Muslim”, we find out that if a question contains either of these two words, there is a nearly 57% chance that this question is insincere.

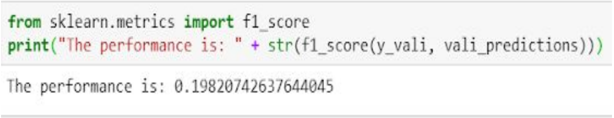
Base on that, we build our baseline model. It simply produces ‘1’ (insincere) if a question contains either “Trump” or “Muslim”, and return 0 otherwise:



```
def predict_sincere(txt):

    if ' trump' in txt.lower():
        return 1
    if ' muslim' in txt.lower():
        return 1
    return 0
```

With this model, we check its performance in our validating set.



```
from sklearn.metrics import f1_score
print("The performance is: " + str(f1_score(y_vali, vali_predictions)))
```

```
The performance is: 0.19820742637644045
```

2.4 Models outline:

Based on the observation and knowledge learned from class, we have designed 3 models with several minor variations to improve the F1 score. We will SVM for all of our models, so the features of each model will all be fed into SVM to produce our prediction. The following is the outline of the models and minor variations of them:

1. Text mining
    a. SVM on tf-idf of top 500 frequent words appearing in the questions in the training set
    b. SVM on word count of top 500 frequent words appearing in the questions in the training set
    c. SVM on the more predictive model between a and b but with stemming and stop words excluded from the features
2. Search for bad words
    a. SVM on tf-idf of "bad words"
    b. SVM on word count of "bad words"
    c. SVM on the more predictive model between a and b with an additional feature of total count of bad words
3. Text and bad word mining
    a. Combine the model 1 and 2 into one feature matrix and SVM on the combined feature matrix

2.5 Feature Engineering Process for Each Model

Model 1: Frequent Word Text Mining

In this model, we will use the words that are frequently appearing in the question texts as our feature. Based on all the question text in the training set, we count the word count of all appearing words and order them in decreasing order. Then, by taking the first 500 words we obtain the top 500 frequently seen words. Using these 500 words, we create a feature matrix and feed it into SVM.

Variation 1:

In the first variation of this model, we will use the word count of each word in the 500-word set that appears in each question text as the entries of the feature matrix.

Variation 2:

In the second variation, instead of word count, we use the tf-idf value of the top 500 words that have appeared in each question text. Specifically, for each question text, the feature entries of the words in the top 500 that appeared in a that question text will be the word's tf-idf value; the feature entries of the words in the top 500 that don't appeared in a that question text will be simply zero. The tf-idf value is calculated based only on the training set.

Variation 3:

In this variation, we will use the approach that produces higher F1 score between variation 1 and 2. However, in this variation, we modify the top 500 popular word set but excluding the stop words. Hence, from the list of decreasing order of word count, we exclude the stop words and produce a set of the top 500 frequently seen non-stop words. Therefore, the feature of this model will be either word count or tf-idf value of the top 500 non-stop words.

Model 2: Search for bad words

In this model, instead of the words frequently seen in the training set, we get the words from outer sources. As we know offensive words are likely to be used in insincere questions, we obtained a list of bad words from a CMU research group. ([3]https://www.cs.cmu.edu/~biglou/resources/bad-words.txt) We will use the occurrence of these bad words in each question text as the features.

Variation 1 & 2:

The first and second variations are the same as model 1; we use word count of each bad word as feature entries in the first variation and use tf-idf trained by the training set in the second variation. The word counts and tf-idf values are obtained in the same way as model 1, but using the bad word set instead of the top 500 word set.

Variation 3:

Similar to model 1, the third variation will use the better approach of variation 1 and 2, and add to it an additional feature of the total count of bad words in a question text. The total count of bad words is the total number of any bad words in a question text, which can be obtained by counting all the bad words appeared in each question text.

Model 3: Text and bad word mining

This model will use the features in model 1 and 2. The feature is obtained simply by combining the feature matrix of the best variation of model 1 with that of model 2.

Part 3: Models & Details & Analysis

Model 1: Text mining

This model employs the basic text mining techniques we have learned from the course material. Based on all the question text in the training set, we produce a list of the top 500 frequently seen words. Using these 500 words, we create a feature matrix and feed it into SVM. To optimize this model, we will compare three variations and use the one that results in the highest F1 score, and also tune the SVM by inputting different C values.

In the first variation, we will use the word count of each word in the 500-word set that appears in each question text as the entries of the feature matrix.

In the second variation, instead of word count, we use the tf-idf value trained by the training set.

After determining which of the two above variations result in a higher F1 score, using that approach, we use a different 500-word set created similarly from top 500 frequently seen words but excluding the stop words.

Justification:

This model is suggested due to our knowledge from the course material and experience from past assignments. We know that using simply words in the training set can result in category prediction that is accurate to some extent. In this case, the category will be just “sincere” and “insincere.” While the first two variations are certainly based on course materials, the third variation is based on one of the observation. As what we have observed, stop words might have less effect on sincerity. We thought that excluding those words might bring up the more effective words to the feature matrix, and hence result in better result.

Limitation:

A possible weakness of this model might be that it cannot consider the words outside of the top 500-word set. Some insincere words are specific and rare, and hence might not appear in the top 500 ranking. This model cannot take into account such rare words. However, the next model can solve this problem, and hence lead to the third model where model 1 and 2 are combined.

Another weakness, specifically of the third variation where stop words are excluded, is that the model will consider less of the grammar of each question. While stop words have less effect on the topic of the questions, stop words can still affect the nuance and opinion of the questions which might influence the sincerity. Nevertheless, this model with features of individual words is not expected to perform well in capturing the grammar of each question from the first place.

Model 2: Search for bad words

In this model, instead of fetching the words to be used in the feature from the training set, we get the words from outer sources, which is the list of bad words produced by a CMU research group as explained earlier. Similar to model 1, we used the words in the bad word list to create the feature matrix and used three variations to optimize the model and tune the SVMs.

The first and second variations are the same as model 1; we use word count of each bad word as feature entries in the first variation and use tf-idf trained by the training set in the second variation.

The third variation is to add an additional feature of the total count of bad words in a question text to one of the feature matrices that results in a higher F1 score from the first and second variation. The total count of bad words is the total number of any bad words in a question text.

Justification:

This model is created based on our observation that insincere questions tend to use offensive words. However, these offensive words tend to be rarely used in the whole training set, where the number of sincere questions greatly outbalances that of insincere questions. After some of the offensive words might not be even used in the training set. Hence, we decide to use outer source, or a list of possible bad words.

The third variation is also based on another observation relating to usage of bad words. This variation is suggested due to the observation where insincere questions tend to use more bad words regardless of what kind of bad words.

Limitation:

This model's weakness is the lack of consideration of the non-bad words, which were taken into consideration in model 1. Some non-bad words might have effect on making a question insincere. In fact, in the observation, we learned that some bad words can be observed in both sincere and insincere questions. Possibly the non-bad words are also having an effect on the sincerity when accompanied with those half-insincere words. After all, we observed some pairing of normal words with bad words that resulted in very unethical topics, and hence make the questions labeled as insincere.

Another weakness of this model is that the bad word list might still not be comprehensive. As we know sensitive words can be really specific, such as religion-related concepts and country names. It is hard to obtain such comprehensive list of bad words and also even we do obtain it, the list will be too big for the model to predict in a reasonable amount of time.

Model 3: Text and bad word mining

As explained earlier, this model is introduced in order to compensate both model 1 and 2's weakness. This model will use the feature matrices obtained from model 1 and model 2. By combining the two feature matrices, we obtain a bigger feature matrix and feed it to SVM in this model.

Justification:

As explained earlier, model 1 is limited in its ability to capture the bad words that are not frequently seen in the training set. On the other hand, model 2 is limited in its ability to consider the words that are not offensive but might have effect on sincerity. Therefore, we design model 3 to compensate model 1 and 2's weaknesses by using themselves. Since the feature matrix is a fusion of that of model 1 and 2, model 3 is expected to consider both bad words and normal but important words.

Limitation:

A weakness of this model is that it only considers what words are used but not how they are used. That is, grammar and context are overlooked. Since it uses individual words separately as its features, it cannot build connection between each words and hence capture the grammar and context of each question text, which might be a main factor of determining sincerity.

Also, another weakness of this model is that it cannot detect Proper nouns such as human names. Some insincere questions deal with specific persons whose names are likely not included in the feature list. This is because human names are neither frequently used words nor bad words. It is likely that this model cannot label correctly of those questions related to specific persons.

Part 4: Related Literature

This dataset is posted on Kaggle by the Quora team, and the purpose of posting this dataset is to help the Quora team to solve the problem of marking insincere question using a more efficient and accurate algorithm.

Some similar questions are on rise these days. These questions belongs to a large category of text categorization. And under this umbrella term, hate speech detection problem arrests most attention. Hate speech detection problem are dealing with online speech that viciously targets on specific group, usually with discriminative tone. For instance, in Burnap and Williams (2015) research, they studied hate speech on the platform of Twitter. Some models and approaches that has been used to solve this problem including the following, which are characterized into two classes:

4.1 Approach/model in their research applied in our project:

Model:

- Support Vector machine: A classification methods that separates the yes and no boundary to achieve maximum separation between classes

Approach:

- Use word count & tf-idf to create feature

Feature Matrix:

- Bag of Word
- Popular words
- Bad words (racial slur)

4.2 Approach/model in their research NOT applied in our project:

Model:

- Bayesian Logistic Regression: Used to test which features are statistically significant
- Random Forest Decision Tree: Rule-based approach to classification
- Embedding word representation: Numerical representation of the text vector, to extract the relationship between words

Approach:

- Using context-free lexical parsing model
    - Consider typed dependencies as a feature to exploit grammar relationships in a sentence
- Using n-grams as bag of word model

As shown above, some basic approaches of our project coincide with the state-or-art research, such as bag of words and support vector machine. However, our project lack some of the more sophisticated tools that could extract information from the grammatical structure of sentences. We have mentioned this weakness in the discussion on our model. The F1 score (0.89) of Burnap and Williams demonstrates that lexical parsing is a more effective approach to categorize hateful speech.

[4]*Burnap, Pete, and Matthew L. Williams. "Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making." Policy & Internet 7.2 (2015): 223-242.*

Part 5: Results & Conclusions

The following table reports the performance of each model we build, with different variations (as specified in previously).

As one can observe, all of our models have higher F1 scores than the baseline, except for one variation (model 2 variation 3) which is an unsuccessful attempt. Also, as expected, model 3 has achieved the higher F1 score.

F-score	Variation 1: Using word count	Variation 2: Using tf-idf value	Variation 3: With total bad word count (for Model 1) / With stopword stemming (for Model 2)
Baseline: Using “Muslim/Trump”	0.1982 (no variations for Baseline)		
Model 1: Text mining	0.37293046357616	0.39129695733469	0.365145228216
Model 2: Bad word mining	0.23872794598127	0.25900805426028	0.19745365199910656
Model 3: Text and bad word mining	**0.42902469306111185** (no variations for Model 3)		

5.1 Conclusion for Model 1 (Text mining):

The optimization result shows that the second variation has higher F1 score than first, and third variation has no improvement in F1 score. In other words, text mining on the word set where stop words are included, and with the use of tf-idf instead of word count, resulted in the best prediction.

From this result, we can say that tf-idf value is favorable over word count. A possible reason is that tf-idf also accounts for the rarity of a given word, and adjusts the weight of word accordingly. Thus tf-idf approach exploits more information from the corpus.

Furthermore, we can infer from this result that stop words do have effect on sincerity since the omission of them resulted in lower F1 score. As suggested earlier as variation 3’s weakness, this is possibly because the stop words can affect the grammar and context of the question text. This result tells us that grammar and context do contribute to the measurement of sincerity in addition to the subject or topic of the questions. This also means, with non-sensitive question topic, a question still can be insincere due its nuance.

The result also highlights the limitation of this model, specifically of the second variation of using tf-idf, which is that the question texts are too short. Most of the question texts are 1 to 2 sentences long, and the average length of the questions are around 15 words as explained earlier. This causes the text frequency to be limited to a small number, meaning it is unlikely for a word to appear more than 2 times in just one or two sentences. After all, the sincerity of a question depends more on the type of the word used rather than the frequency of the words used.

Another problem of this model suggested by the result is that it cannot consider the words outside of the top 500-word set. Some insincere words are specific and rare, and hence might not appear in the top 500 ranking. This model cannot take into account such rare words. However, the next model can solve this problem, and hence lead to the third model where model 1 and 2 are combined, which resulted in the best F1 score..

5.2 Conclusion for Model 2 (Bad word mining):

The optimization result shows that tf-idf variation has higher F1 score than the other two variations, more specifically, the third variation did not result in improvement of the F1 score. In other words, text mining on the word set solely with the use of tf-idf instead of word count resulted in the best prediction.

The result also tells that the third variation is an unsuccessful attempt. This means the total word count of bad words appearing in a question text is a redundant feature. A possible reason is that the insincerity of a question doesn't actually depend greatly on the number of bad words used but more on the type of bad words. Also, the value of text frequency in the tf-idf calculation has already taken into account of the bad words counts, which cause this additional field becoming redundant. This redundancy of information leads to an increase in variance, thus reduces the accuracy in prediction. While the total bad word count also attempted to allow bad words which are not seen in the training set and hence has tf-idf value of zero to have effect in the prediction, the result tells us that this is not a big concern.

Comparing to model 1, although model 2 resulted in a lower F1 score, it solves one weakness of model 1 which is the consideration of rare words. If our bad word list is thorough, this model will consider most of the rare bad words that weren't considered in model 1 due to the ranking.

On the other hand, the result also shows this model lacks consideration of the non-bad words, which were taken into consideration in model 1. From the fact that model 1 has much higher F1 score, we can tell non-bad words are crucial in determining the sincerity of a question. In fact, in the observation, we learned that some bad words can be observed in both sincere and insincere questions. Possibly the non-bad words are having an effect on the sincerity when accompanied with those half-insincere words. After all, the combination of normal words with bad words can result in unethical questions.

5.3 Conclusion for Model 3 (Text and bad word mining):

The F1 score is higher than the previous models, and certainly is the highest that we obtained. As we predicted, possibly model 3 has succeeded in solving model 1's weakness with model 2's features and vice versa.

However, as we compare our model 3's F1 score to other team's, such as the one introduced in Part 4, this model is not sufficient to be said as a successful predictor. This is certainly due to some limitation that still exists in this improved model.

One limitation of this model is that it cannot detect Proper nouns such as human names, as predicted in the model design stage. Some insincere questions deal with specific persons whose names are both rare words and non-bad words. It is likely that this model cannot label correctly of those questions related to specific persons.

Another major weakness of this model is that it only considers the tf-idf value of the remarkable words, or frequently seen words and words in the bad word list. This model only considers what words are used but not how they are used. That is, grammar and context are overlooked. In fact, in the approach introduced in Part 4, grammar and context are taken into account greatly and hence resulted in F1 score higher than our model 3. This result certainly tells us the importance of grammar, or how words are used, in determining insincerity.

A possible improvement to model 3 that can be done in the future is to consider specifically what types of words are used as the subject, verb, object, and adjective. This way the grammar of each question can be captured in the feature to an extent and hence result in a more reliable predictor of insincerity.

Part 6: References

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

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

[2] https://www.kaggle.com/c/quora-insincere-questions-classification/discussion/72983

[3] https://www.cs.cmu.edu/~biglou/resources/bad-words.txt

[4] *Burnap, Pete, and Matthew L. Williams. "Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making." Policy & Internet 7.2 (2015): 223-242.*