ITCS 6156 MACHINE LEARNING

Final Project Report

Quora Insincere Questions Classification

Detect toxic content to improve online conversations

Team Members:

Member Name	Student ID
Akhil Morampudi	(801138186)
Bharadwaj Aryasomayajula	(801151165)
Mahanth Mukesh Dadisetty	(801034945)

Team Name: Invincibles

GITHUB LINK: https://github.com/bharadwaj995/Invincibles.git

Introduction

Summary

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.

Abstract

Machine learning models have been very successful when dealing with text classification problems. In this project we are tasked with classifying questions from a dataset provided by the popular website Quora, as 'sincere' or 'insincere'. The large-scale dataset, provided by the website Kaggle [https://www.kaggle.com/c/quora-insincere-questions-classification/data] contains over 1,300,000 questions with labels to train our models on. A separate test set that contains over 300,000 unlabeled questions is used by Kaggle to test our model. We implement three different models, logistic regression, Naive Bayes, and recurrent neural networks, and use different pre-trained word embedding's to achieve the best results

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:[3]

- Has a non-neutral tone
- Has an exaggerated tone to underscore a point about a group of people
- Is rhetorical and meant to imply a statement about a group of people
- Is disparaging or inflammatory
- Suggests a discriminatory idea against a protected class of people, or seeks confirmation of a stereotype
- Makes disparaging attacks/insults against a specific person or group of people
- Based on an outlandish premise about a group of people
- Disparages against a characteristic that is not fixable and not measurable
- Isn't grounded in reality
- Based on false information, or contains absurd assumptions
- Uses sexual content (incest, bestiality, pedophilia) for shock value, and not to seek genuine answers

The training data includes the question that was asked, and whether it was identified as insincere (target
= 1). The ground-truth labels contain some amount of noise: they are not guaranteed to be perfect.

Literature Survey

Abstract:
Sentiment analysis, also called opinion mining, is a form of information extraction from text of growing research and commercial interest. In this paper we present our machine learning experiments with regard to sentiment analysis in blog, review and forum texts found on the World Wide Web and written in English, Dutch and French. We train from a set of example sentences or statements that are manually annotated as positive, negative or neutral with regard to a certain entity. We are interested in the feelings that people express with regard to certain consumption products. We learn and evaluate several classification models that can be configured in a cascaded pipeline. We have to deal with several problems, being the noisy character of the input texts, the attribution of the sentiment to a particular entity and the small size of the training set. We succeed to identify positive, negative and neutral feelings to the entity under consideration with ca. 83% accuracy for English texts based on unigram features augmented with linguistic features. The accuracy results of processing the Dutch and French texts are ca. 70 and 68% respectively due to the larger variety of the linguistic expressions that more often diverge from standard language, thus demanding more training patterns. In addition, our experiments give us insights into the portability of the learned models across domains and languages. A substantial part of the article investigates the role of active learning techniques for reducing the number of examples to be manually annotated.

Methodology:
Machine learning techniques for sentiment classification gain interest because of their capability to model many features and in doing so, capturing context, their more easy adaptability to changing input, and the possibility to measure the degree of uncertainty by which a classification is made. Supervised methods that train from examples manually classified by humans are the most popular. The most common approaches here use the single lowercased words (unigrams) as features when describing training and test examples.

In opinion mining certain sentiments are expressed in two or more words, and the accurate detection of negation is important because it reverses the polarity. Pedersen showed that word n-grams are effective features for word sense disambiguation, while Dave et al. indicated that they are able to capture negation. In an alternative approach to negation, each word following a negation until the first punctuation receives a tag indicating negation to the learning algorithm.

Other approaches select only a subset of the words, often by considering solely adjectives detected with a part-of-speech (POS) recognizer.

Pros:

1. The supervised techniques are applied mostly for recognizing the sentiment of complete documents.

2. Step-wise approach for the classification of texts by first removing objective sentences from it (using a machine learning-backed minimal cut algorithm), and then classifying the remaining ones.)

Cons:

1. Few weakly supervised learning approaches in the literature. In such settings the manual labeling is limited. Clustering can be used to determine the semantic orientation of many adjectives present in a corpus; the obtained clusters are then manually labeled.

2. Identify in texts product properties (PPs) and their correlated opinion words (OWs) by using an iterative cross-training method, which starts from a small labeled part of the corpus. Two naïve Bayes classifiers (respectively for detecting PPs and OWs) are trained using contextual features.

3. Noisy character of the input texts, the attribution of the sentiment to a particular entity and the small size of the training set.

Quora Insincere Question Classification:

Neural network models have been proved to achieve remarkable performance in text sentiment classification. CNN trained on top of pre-trained word vectors got great results for sentence-level classification tasks [8]. Some combinations and modifications can get improvement. Gated RNN dramatically outperforms than standard RNN [9] in document modeling for sentiment classification. A unified model with CNN and LSTM called C-LSTM is able to capture both local features of phrases as well as global and temporal sentence semantics.[10] [11].

Method

In our approach we used the Quora Insincere dataset and analyzed it. This analysis labeled datasets using the unigram feature extraction technique. We used the framework where the preprocessor is applied to the raw sentences which make it more appropriate to understand. Further, the different machine learning techniques trains the dataset with feature vectors and then the semantic analysis offers a large set of synonyms and similarity which provides the polarity of the content. The complete description of the approach has been described in next sub sections and the block diagram of the same is graphically represented in the below figure.

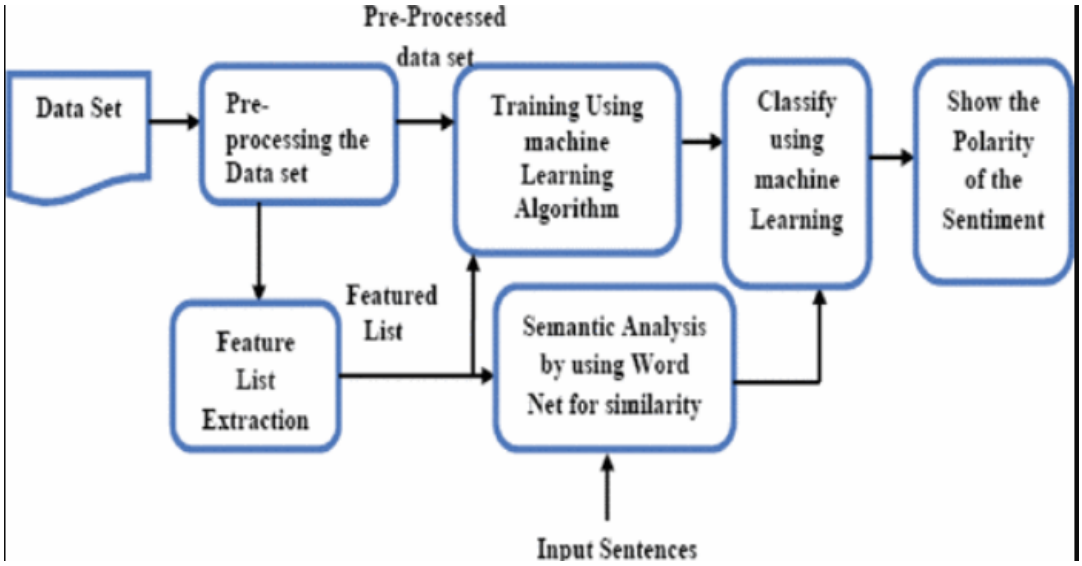


Fig : Flow diagram of the Implemented Model

Objective

The objective of this project is to implement machine learning algorithms on the training dataset provided by Kaggle, and then create a set of predictions for a separate unlabeled test set. In this project we test several models that have been known to perform well with text classification problems. We begin with two baseline models, logistic regression and naive Bayes, and then create a more advanced recurrent neural network model.

Basic Feature Engineering:

We can add some features as a part of feature engineering pipeline for Quora Insincere Questions Classification Challenge.
Some features that I have included are listed below:

Data Pre-processing:

The text data is not entirely clean, thus we need to apply some data pre-processing techniques.

Pre-processing techniques for Data Cleaning:

- Removing Punctuation
- Number of capital letters
- Number of special characters
- Number of unique words
- Number of numeric
- Number of characters
- Number of stop words

Proposed Model and Methodology:

Machine Learning Methods:

Advance NLP Text Processing:[4]

Tokenizing:

Tokenization is the act of breaking up a sequence of strings into pieces such as words, keywords, phrases, symbols and other elements called tokens. Tokens can be individual words, phrases or even whole sentences. Tokenization is a necessary first step in many natural language processing tasks, such as word counting, parsing, spell checking, corpus generation, and statistical analysis of text. Tokenizer is a compact Python module for tokenizing, which converts Python text strings to streams of token objects, where each token object is a separate word, punctuation sign, etc. It also segments the token stream into sentences, considering corner cases such as abbreviations and dates in the middle of sentences.
NLTK provides a number of tokenizers in the tokenize module

Word Embeddings:

To perform well on most natural language processing tasks we first need to have some notion of similarity and difference between words. In this project, we mainly use GloVe to encode word tokens.
The following are the word embeddings implemented in our model for text classification.

- **Global Vectors for Word Representation (GloVe)**
- **GoogleNews-vectors-negative300** (pre-trained **Google News** corpus (3 billion running words) word vector model (3 million 300-dimension English word vectors).
- **wiki-news-300d-1M** (1 million word vectors trained on Wikipedia 2017, UMBC
- **paragram_300_sl999** (webbase corpus and statmt.org news dataset (16B tokens). (Embed sentences into a low-dimensional space such that cosine similarity in the space corresponds to the strength of the paraphrase relationship between the sentences)

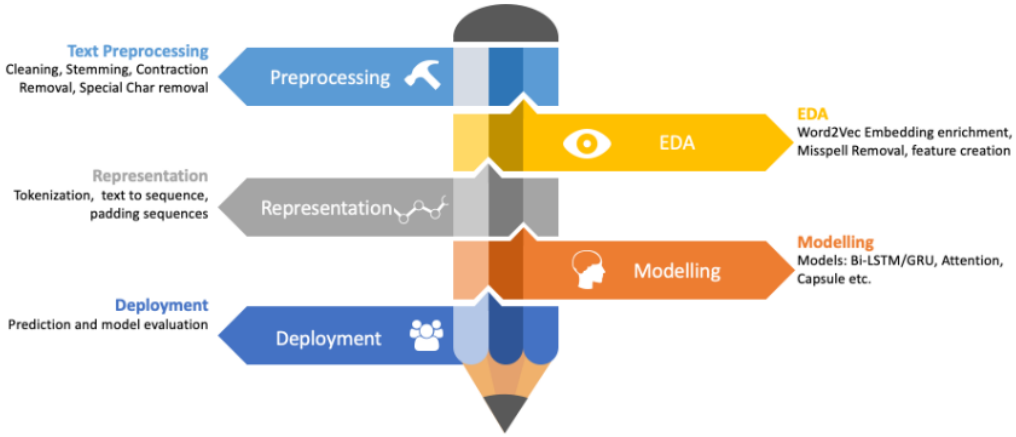
Global Vectors for Word Representation (GloVe)

GloVe is a new global log bilinear regression model that combines the advantages of the two major model families in the literature: global matrix factorization and local context window methods.[9] It efficiently leverages global statistical information by training only on the nonzero elements in a word-word co-occurrence matrix, and produces a vector space with meaningful sub-structure. It consistently outperforms word2vec on the word analogy task, given the same corpus, vocabulary, window size, and training time. It achieves better results faster, and also obtains the best results irrespective of speed. Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space.

DEEP LEARNING MODELS USED:

Feature Extraction:

Feature extraction from text is done to feed as input to a machine learning model.



Finally applying the processed data to the below Machine Learning Algorithm.

CNN: Use convolutional neural networks can be used as a recurrent structure to capture contextual information as far as possible, which may introduce considerably less noise compared to traditional window based neural networks.

Convolutional Neural Network (CNN) to perform text classification with word embeddings

CNN has been successful in various text classification tasks. a simple CNN with little hyperparameter tuning and static vectors achieves excellent results on multiple benchmarks – improving upon the state of the art on 4 out of 7 tasks [13].

However, when learning to apply CNN on word embeddings, keeping track of the dimensions of the matrices can be confusing.

For example, let us take an instance where a sentence comprising of 7 words is given as an input to the CNN for text classification.

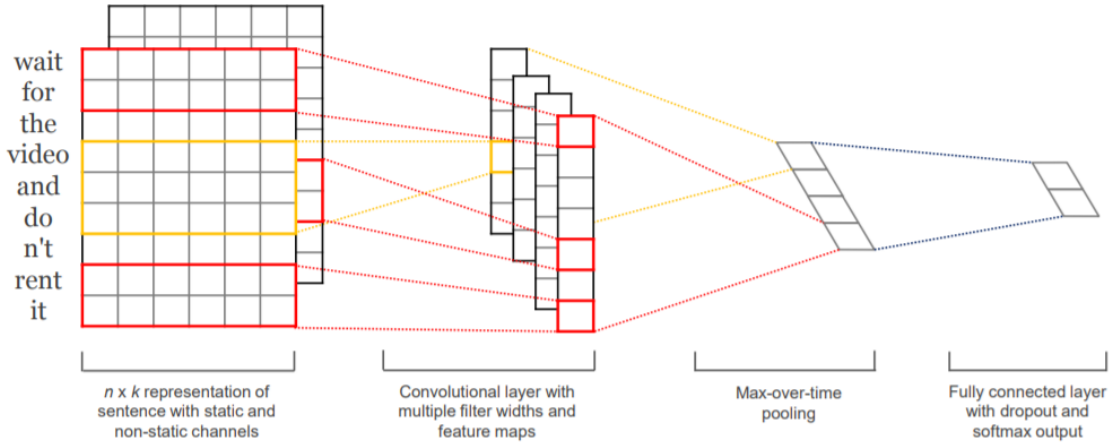


Figure 1 : Model architecture with two channels for an example sentence [13]

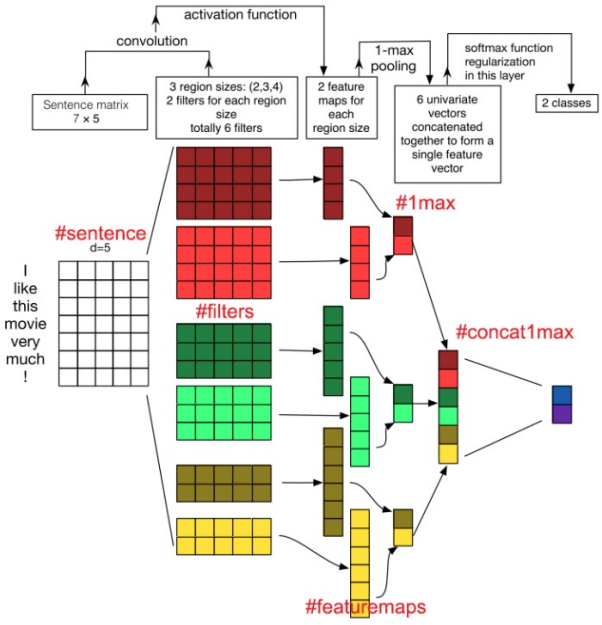


Figure 2: Illustration of a CNN architecture for sentence classification. [12]

We depict three filter region sizes: 2,3,4, each of which has 2 filters. Filters perform convolutions on the sentence matrix and generate (variable-length) feature maps; 1-max pooling is performed over each map, i.e., the largest number from each feature map is recorded. Thus, a univariate feature vector is generated from all six maps, and these 6 features are concatenated to form a feature vector for the penultimate layer. The final

SoftMax later then receives this feature vector as input and uses it to classify the sentence; here we assume binary classification and hence depict two possible output states."

Text Classification Using Recurrent Neural Network (RNN) :

A recurrent neural network (RNN) is a class of artificial neural network where connections between nodes form a directed graph along a sequence. This allows it to exhibit dynamic temporal behaviour for a time sequence. Using the knowledge from an external embedding can enhance the precision of your RNN because it integrates new information (lexical and semantic) about the words, an information that has been trained and distilled on a very large corpus of data[13]. The pre-trained embedding we'll be using is GloVe.

Bi-directional RNN (LSTM/GRU):

Text CNN works well for Text Classification. It takes care of words in close range. However, it still can't take care of all the context provided in a particular text sequence. It still does not learn the sequential structure of the data, where every word is dependent on the previous word or a word in the previous sentence.

RNN help us with that. They can remember previous information using hidden states and connect it to the current task. Long Short Term Memory networks (LSTM) are a subclass of RNN, specialized in remembering information for an extended period. Moreover, the Bidirectional LSTM keeps the contextual information in both directions which is pretty useful in text classification task

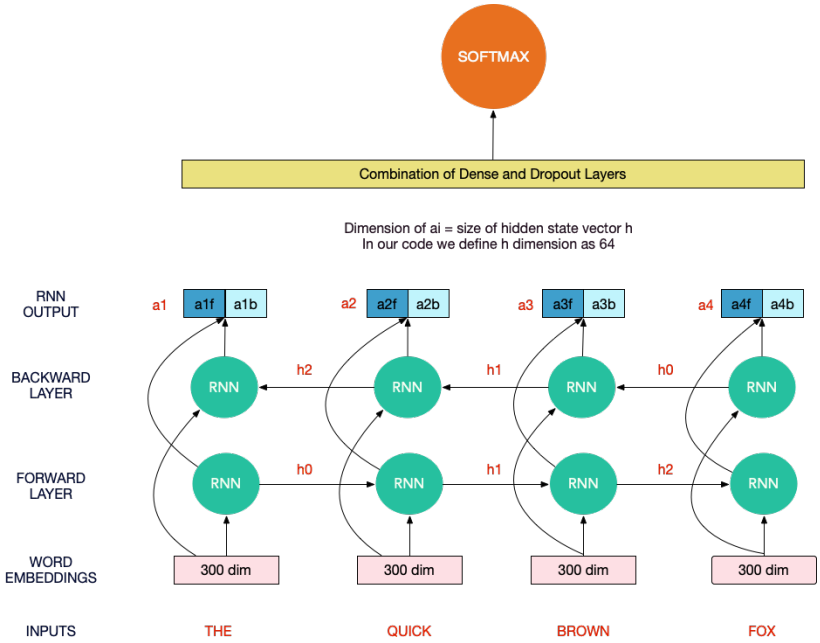


Fig: Bi-Directional RNN on Text Classification [15]

Long Short-Term Memory networks (LSTM) are a subclass of RNN, specialized in remembering information for an extended period. Moreover, the Bidirectional LSTM keeps the contextual information in both directions which is pretty useful in text classification task.

For a most simplistic explanation of Bidirectional RNN, we think of an RNN cell as a black box taking as input a hidden state (a vector) and a word vector and giving out an output vector and

the next hidden state. This box has some weights which are to be tuned using Back propagation of the losses. Also, the same cell is applied to all the words so that the weights are shared across the words in the sentence. This phenomenon is called **weight-sharing.**

Notable Difference:

1. Instead of implementing Bag of words method we used various word embeddings used by the users across various social media platforms and concatenated all the embeddings. This concatenated embedding matrix is given as an input to the neural network.
2. Rather than using the conventional methods for sentence classification we had used Deep learning methodologies like Text CNN and Bi-LSTM. We focus on one dimensional CNN (to the exclusion of more complex models) due to their comparative simplicity and strong empirical performance, which makes it a modern standard baseline method.
3. Data pre-processing and feature extraction is key in sentiment analysis and we are using the combination of Tokenization and using 4 Word embeddings together which was not performed earlier.
4. Selecting the features in an iterative process and deciding on the best suitable features which gives better performance.
5. Applying the algorithms related CNN and RNN together to our model.
6. When randomly initializing words not in word2vec, we obtained slight improvements by sampling each dimension such that the randomly initialized vectors have the same variance as the pre-trained ones. It would be interesting to see if employing more sophisticated methods to mirror the distribution of pre-trained vectors.

Updated Timetable and Work Assignment:

Member Name	Responsibilities
Mahanth Mukesh (801034945)	1. Project Report 2. Infra Setup 3. Data Cleaning & Pre-processing 4. Modelling 5. Training & Prediction 6. Performance Analysis
Bharadwaj (801151165)	1. Project Report 2. Data Cleaning & Pre-processing 3. Feature Engineering 4. Modelling 5. Performance Analysis 6. Poster
Akhil Morampudi (801138186)	1. Project Report 2. Infra Setup 3. Data Cleaning & Preprocessing 4. Feature Engineering 5. Modelling 6. Training & Prediction

Project Timeline

Step 1	Owner	Status	Timeline	Priority
Project Proposal	B +2	Done	Sep 23 - Oct 3	High
Infrastructure Setup	B +2	Done	Oct 4 - 11	High
Data Cleaning & Pre-Processing	B +2	Done	Oct 12 - 18	High
Exploratory Data Analysis	B +2	Done	Oct 19 - 25	Medium
Feature Engineering	B +2	Done	Oct 26 - Nov 8	Medium
+ Add				

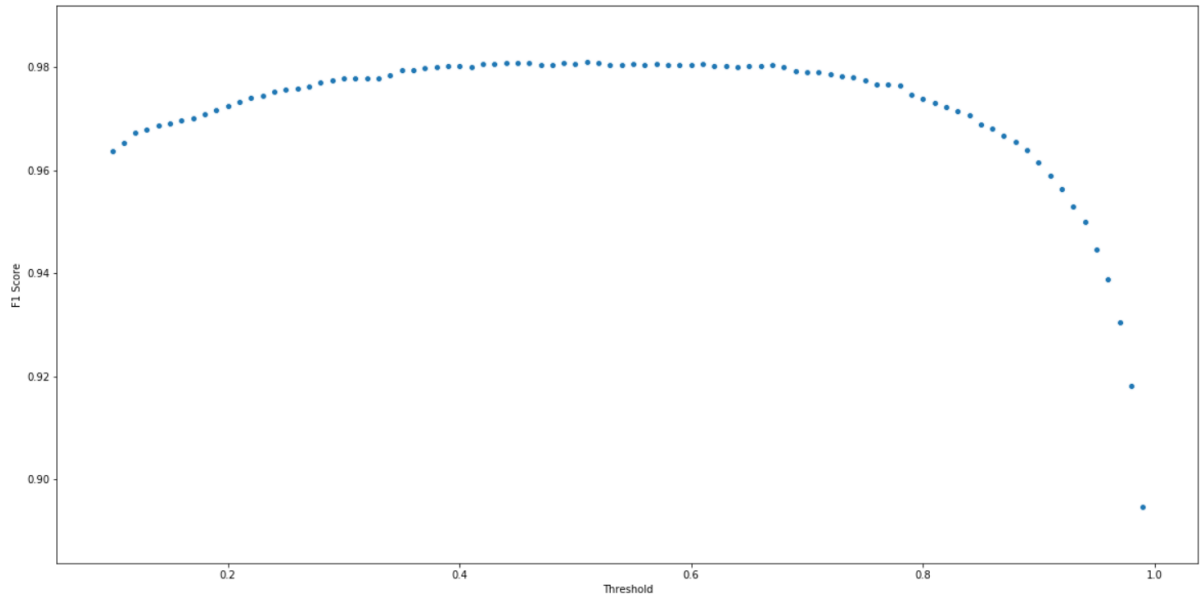
Step 2	Owner	Status	Timeline	Priority
Modelling	A +2	Done	Nov 9 - 15	Very High
Training & Prediction	B +2	Done	Nov 16 - 22	Very High
Performance Analysis and Metrics	B +2	Done	Nov 23 - 29	High
Final report & Poster Presentation	B +2	Done	Nov 30 - Dec 5	Medium
+ Add				

Reflections:

1. We have described more about the existing problem and discussed about how differently we are tackling the problem.
2. Removed the reference paper titles from the report and included them in the references.
3. As per the previous feedback, we have understood the problem more and discovered new techniques to handle this problem and mentioned those in the method explanation.
4. The novel approaches to the problem have been updated clearly by doing a lot of research on the existing approaches and adding our way of solving this problem.
5. Added captions to help understand the figures in the report.
6. Citing references has been updated.

Evaluations and Performance metrics:

As this dataset is highly imbalanced, we will use F1 score as a metric for this dataset. Metric is F1 Score between the predicted and the observed targets. There are just two classes, but the positive class makes just over 6% of the total. So the target is highly imbalanced, which is why a metric such as F1 seems appropriate for this kind of problem as it considers both precision and recall of the test to compute the score. Our model got the best F1 score 0.9810100339995024 at threshold 0.51 which is a huge improvement compared to our base paper F1 score 0.67. [3] [6]



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 classifiers,” in 2018 IEEE International Conference on Information Reuse and Integration (IRI), July 2018, pp. 269–276.

[3] Mungekar, Akshay, et al. "Quora Insincere Questions Classification."

[4] https://towardsdatascience.com/a-gentle-introduction-to-natural-language-processing- e716ed3c0863

[5] https://towardsdatascience.com/quora-insincere-questions-classification-d5a655370c47

[6] Quora Insincere Questions Classification.‖ [Online]. Available: https://kaggle.com/c/quora- insincere-questions-classification. [Accessed: 29-Jul-2019]

[7] Boiy, Erik, and Marie-Francine Moens. "A machine learning approach to sentiment analysis in multilingual Web texts." *Information retrieval* 12.5 (2009): 526-558.

[8] Yoon Kim. 2014. Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882 (2014).

[9] Duyu Tang, Bing Qin, and Ting Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In Proceedings of the 2015 conference on empirical methods in natural language processing. 1422–1432.

[10] Chunting Zhou, Chonglin Sun, Zhiyuan Liu, and Francis Lau. 2015. A C-LSTM neural network for text classification. arXiv preprint arXiv:1511.08630 (2015).

[11] Gabbard, Samuel, Jinrui Yang, and Jingshi Liu. "Quora Insincere Question Classification."

[12]http://www.joshuakim.io/understanding-how-convolutional-neural-network-cnn-perform-text-classification-with-word-embeddings/

[13] Kim, Yoon. "Convolutional neural networks for sentence classification." *arXiv preprint arXiv:1408.5882* (2014).

[14] Zhang, Ye, and Byron Wallace. "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification." *arXiv preprint arXiv:1510.03820* (2015).

[15] https://towardsdatascience.com/nlp-learning-series-part-3-attention-cnn-and-what-not-for-text-classification-4313930ed566

GITHUB LINK: https://github.com/bharadwaj995/Invincibles.git