Qoura Insincere Question Classification (Kaggle Challenge)

Madhu Hegde   Rohit Aggarwal

mrhegde@stanford.edu   rohitagg@stanford.edu

Problem Definition

To build a machine learning model to predict whether a question asked on Quora is sincere or insincere. The sincerity detection is a Natural Language Understanding (NLU) task involving sentiment analysis and sentence/text classification.

Motivation

There is a real need to filter out toxic and divisive content on the internet to make it a safe place for knowledge sharing. The Quora challenge is one such attempt.

Challenges

The main challenge is to understand the semantics of imperfect real world data. The data is heavily unbalanced and difficult to generalize.

Approach

We propose a solution using a neural network taking advantage of various word embeddings and LSTM building blocks, to efficiently encode and extract useful features from the text input to be able to make a good prediction

Data

We used the labeled data provided by Quora for this project. Train data consists of 1.3 million rows and 3 features:

- **qid** — unique question identifier
- **question__text** — Quora question text
- **target** — a question labeled “insincere” has a value of, 1 otherwise 0

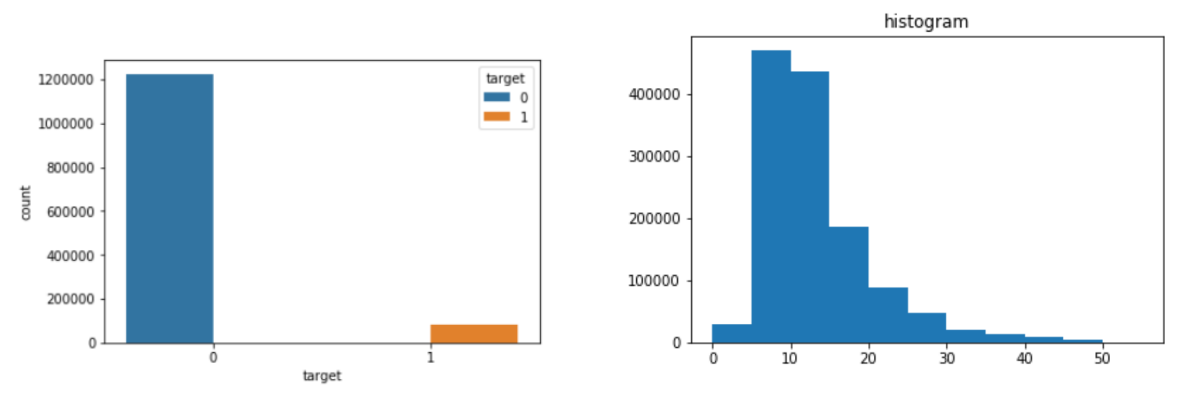
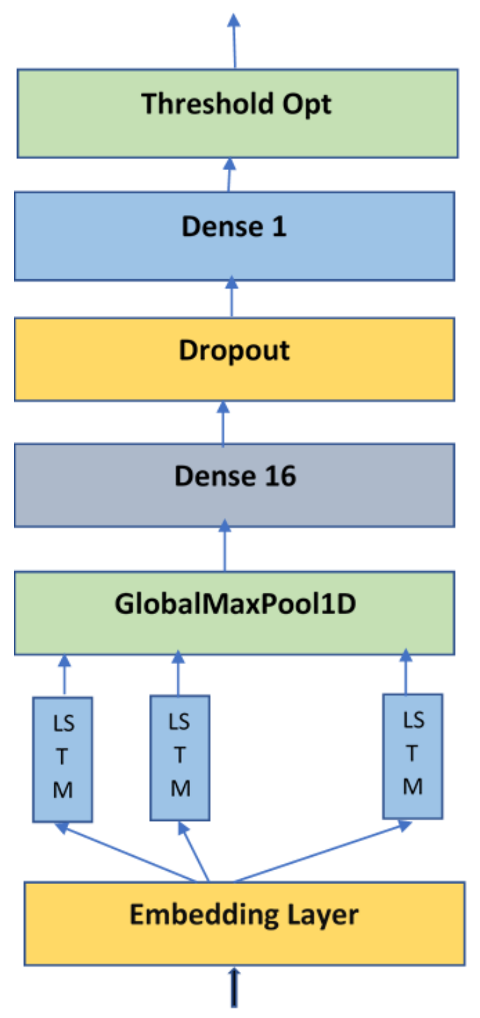


Figure 1:Left figure demonstrates the skew among the labels of the dataset. Right figure is the histogram showing distribution of data points by number of words per example.

Feature Extraction

- Unigrams are used as features. Aim at improving generalization between texts and labels.
- Tokenizer used to fit training data using a vocabulary size of 50,000 words.

Initial Model


Threshold Opt

Dense 1

Dropout

Dense 16

GlobalMaxPool1D

LSTM

LSTM

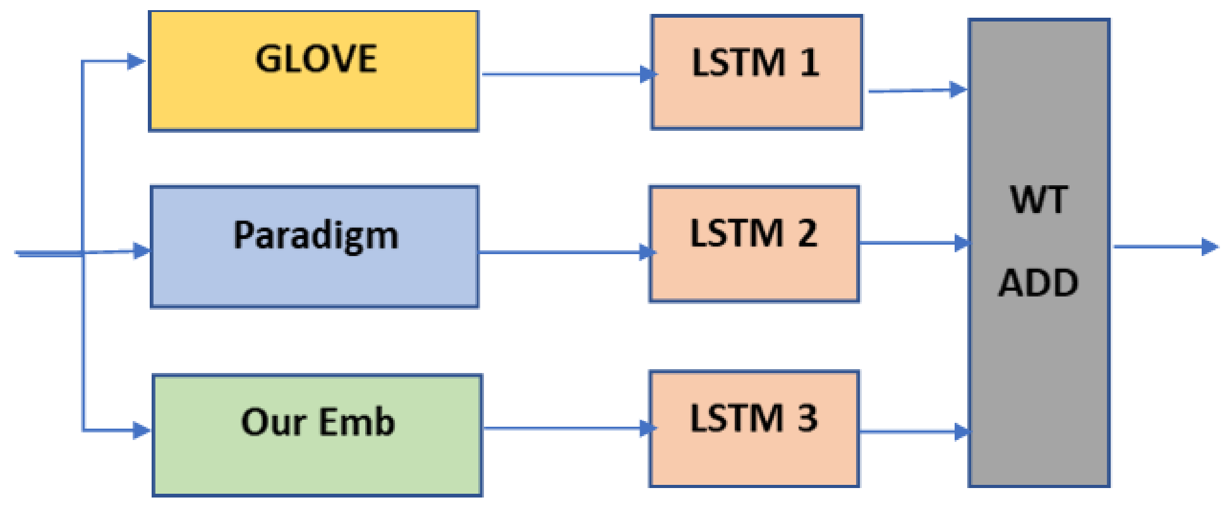
LSTM

Embedding Layer

Figure 2:Initial Model architecture consisting of

- Basic Model consists of word embeddings, followed by BiDirectional LSTM layer, and a few dense layers to make final prediction.
- GlobalMaxPooling layer helps to detect match at any time step instead of taking results from the last node.
- Used a dense layer of size 16, followed by a dropout layer to reduce overfitting.

Final Model


GLOVE

Paradigm

Our Emb

LSTM 1

LSTM 2

LSTM 3

WT ADD

- After experimenting with first model with different embeddings, we believe combining multiple embeddings might improve F1 score further.
- Such emsemble will help to model the diverse semantic content of Quora questions

Training Strategy

- **Oversampling and Undersampling Techniques** - We tried SMOTE but generation of synthetic samples of minority class did not improve the performance. We plan to try undersampling of majority class next.
- **Class weights** - Used class weights to help with skewed data. Minority class weighted higher in loss function compared to majority class, based on data distribution
- **Attention/BERT model** - We tried BERT from Tensorflow-Hub but didn’t see much improvement. Perhaps attention is not an issue when sentence length is small.
- **Threshold Optimization** - Optimized by tuning on CV set. Improves performance when dataset in unbalanced
- **Early Stopping** - Model tends to overfit and we use the model after 2 epochs with least CV loss for prediction
- **Hyper-parameter Tuning**
- **Loss function** - Binary Crossentropy + Class weights

$$\mathcal{L} = -\sum_{i\in n} w_{pos} * y * \ln \hat{y} + w_{neg} * (1-y) * \ln(1-\hat{y}), \quad (1)$$

where n is the total number of examples, y is the true label, $\hat{y}$ is predicted label, w_{pos} and w_{neg} are class weights for positive and negative classes respectively.

Results

Based on our best model using glove embeddings, the loss/accuracy curve, ROC curve, and confusion matrix are shown below.

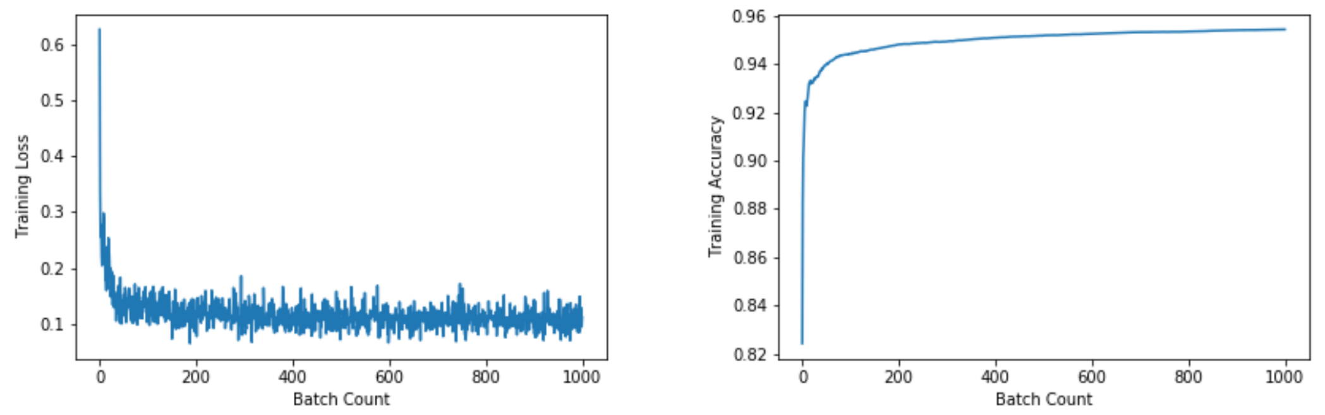


Figure 3:Loss/Accuracy Curve

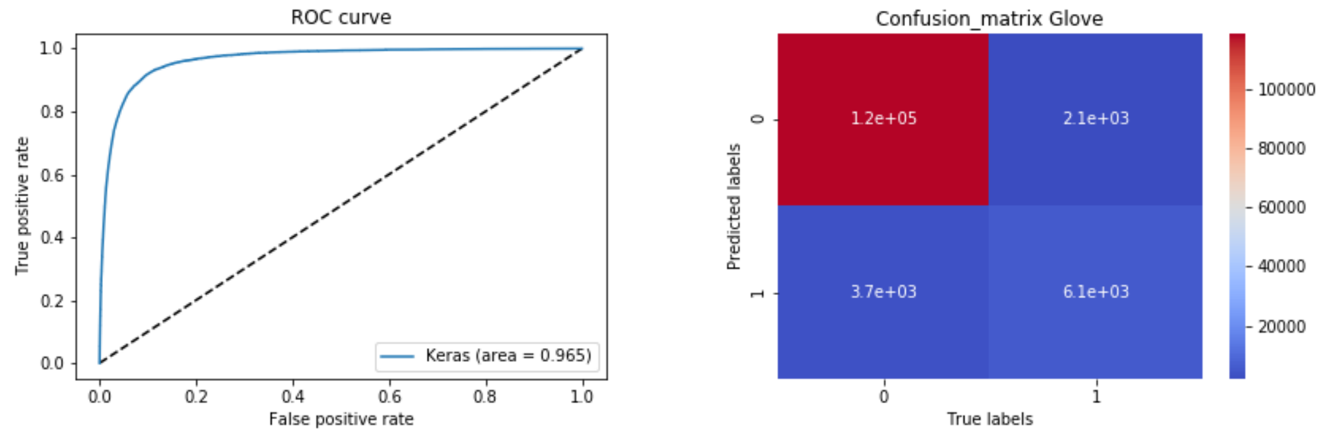


Figure 4:ROC curve on the left, and Confusion matrix on right

Results (Contd.)

Experiment results for tuning classification threshold are show below. We achieved an F1 score close to 0.68 by using the threshold of 0.28.

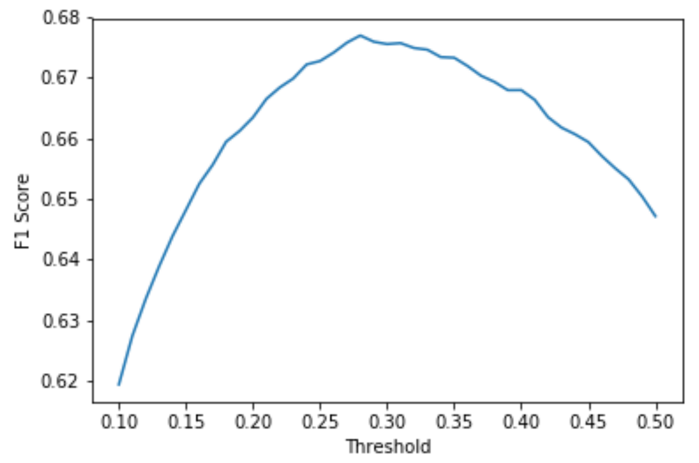


Figure 5:The variation in CV set F1 score as we vary the threshold for classification.

Table 1:F1 Score on Test Set for different Architectures

Model	Dataset F1 Score
	Test Set
Baseline (Naive Bayes)	0.53
Initial Architecture	0.607
Updated Arch (with Glove)	**0.677**
Updated Arch (with Paragram)	0.676
Updated Arch (with wiki)	0.665
Updated Arch (with Self Trained Emb.)	0.653

Discussion

Our current model achieves close to 0.68 F1 score on test set. Though it is still well below the human performance of 0.80, we are still trying some improvements. We will try to match the performance of competition winners of 0.72.

Future improvements

- Hyper parameter tuning of Final Model with Combining Word Embeddings/Model Ensembling
- Using Bigram and Trigram features

Acknowledgements

- Thanks to CS221 TA Sharman W Tan for counseling us on our project
- Thanks to Professor Percy Liang, Dorsa Sadigh, and the SCPD for this research project opportunity
- For our implementation, see `https://github.com/madhuhegde/CS221_Project`
- Video Presentation: `https://youtu.be/kAjWRM1po1E`