Insincere Question Classification on Quora

Hao Mao (haomao@), Rekha Kumar (rekha123@), Jerry Chen (jchen98@) {stanford.edu}

Introduction

Quora is an online platform for people to ask and answer questions. While most of these questions are asked in good faith, a small percentage of bad actors post questions that are insincere or problematic. In this project we experimented with multiple classifiers in an attempt to automatically identify such problematic questions.

Data

Our dataset comes from Kaggle and consists of 1.3 million question text and label pairs. Examples include:

-"What a difference between religion and spirituality?" (0, sincere)

-"Why are religious folks not considered clinically insane?" (1, insincere)

Features

- Mostly we used word counts or tf-idf scores
- For CNN we used the sentence length, number of words in a sentence, the capital letter ratio, and unique words ratio features and combined the vectorized question text.

Models

Naive Bayes

$$\begin{aligned}\phi_{j|y=1} &= \frac{1+\sum_{i=1}^{n} 1\{x_j^{(i)}=1 \wedge y^{(i)}=1\}}{2+\sum_{i=1}^{n} 1\{y^{(i)}=1\}} \\ \phi_{j|y=0} &= \frac{1+\sum_{i=1}^{n} 1\{x_j^{(i)}=1 \wedge y^{(i)}=0\}}{2+\sum_{i=1}^{n} 1\{y^{(i)}=0\}} \\ \phi_y &= \frac{\sum_{i=1}^{n} 1\{y^{(i)}=1\}}{n}\end{aligned}$$

Logistic Regression

$$\min_{w,c} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \log(\exp(-y_i(X_i^T w + c)) + 1)$$

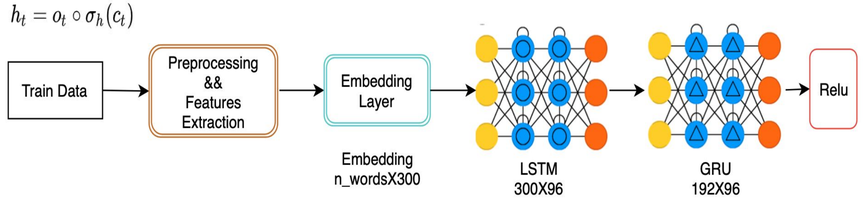
Averaged Perceptron (Perceptron with averaged weight vector)

$$\hat{y} = \operatorname{sign}\left(\sum_{k=1}^{K} c^{(k)} \left(w^{(k)} \cdot \hat{x} + b^{(k)}\right)\right)$$

Concurrent Neural Network

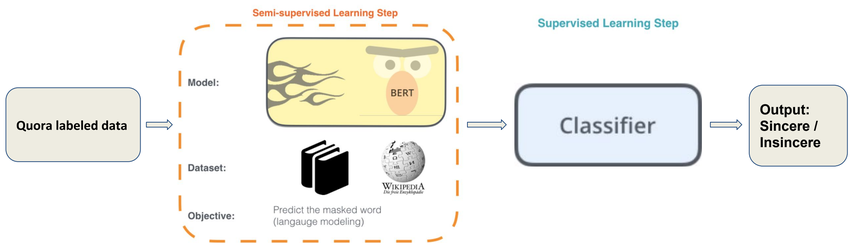
$$\begin{aligned}f_t &= \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \\ i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \\ o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \\ c_t &= f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \\ h_t &= o_t \circ \sigma_h(c_t)\end{aligned}$$

$$\begin{aligned}z_t &= \sigma_g(W_z x_t + U_z h_{t-1} + b_z) \\ r_t &= \sigma_g(W_r x_t + U_r h_{t-1} + b_r) \\ h_t &= (1-z_t) \circ h_{t-1} + z_t \circ \sigma_h(W_h x_t + U_h(r_t \circ h_{t-1}) + b_h)\end{aligned}$$



BERT

BERT is a NLP pretraining method developed by Google. We ran BERT on our dataset and ran classification on the outputs.



Results

	Training Loss (780k samples)	Testing Loss (260k samples)	Accuracy	F1	Precision	Recall
Naive Bayes (w/ tf-idf)	0.165	0.172	0.940	0.102	0.647	0.055
Naive Bayes (w word counts)	0.204	0.232	0.936	0.552	0.483	0.645
Logistic with regularization	0.093	0.123	0.953	0.546	0.674	0.459
Logistic with regularization and boundary	0.093	0.123	0.947	0.603	0.565	0.647
CNN basic	0.148	0.130	0.947	0.351	0.725	0.231
CNN bidirectional	0.155	0.131	0.949	0.428	0.672	0.324
CNN bidirectional & Adam optimization	0.135	0.116	0.956	0.575	0.689	0.509
CNN bidirectional & Adam & boundary	0.126	0.135	0.951	0.625	0.591	0.662
Averaged Perceptron (MLNet)	-	0.180	0.957	0.630	0.678	0.588
Bert(10k data)	-	-	0.941	0.293	0.667	0.188

Discussion

- Bert is powerful but time consuming.
- Average perceptron surprisingly performs better than logistic regression and naive bayes.
- Bidirectional RNN can help the model better understand the text.
- Adam optimization improves the RNN model a lot.
- Boundary condition change helped improving results for this unbalanced data to reduce the number of FN results.

Future Work

- Bert needs lot of time for training. We would get more powerful instances with gpus, optimize the code to run faster. Try newer techniques like fast-bert.
- We would like to try more features to see whether it improves the result in RNN.
- Given sufficient data we have, we could try k-fold cross validation to pick the best model to make the test data predictions.

References:
http://jalammar.github.io/illustrated-bert/
https://www.asimovinstitute.org/neural-network-zoo/