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.

Models

Naive Bayes

\[
\phi_{y=1} = \frac{1 + \sum_{i=1}^{n} I(y_i = 1)}{2 + \sum_{i=1}^{n} I(y_i = 1)} \\
\phi_{y=0} = \frac{1 + \sum_{i=1}^{n} I(y_i = 0)}{2 + \sum_{i=1}^{n} I(y_i = 0)} \\
\phi_y = \frac{\sum_{i=1}^{n} I(y_i = y)}{n}
\]

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)

\[
y = \sign\left(\sum_{k=1}^{K} e^{w_k^T x + b_k}\right)
\]

Concurrent Neural Network

\[
f_i = \sigma_i(W_i z_i + U_i h_{i-1} + b_i) \\
z_i = \sigma_i(W_i z_i + U_i h_{i-1} + b_i) \\
\eta_i = \sigma_i(W_i z_i + U_i h_{i-1} + b_i) \\
h_i = (1 - z_i) \odot h_{i-1} + z_i \odot \sigma_i(W_i z_i + U_i (\eta_i \odot h_{i-1}) + b_i) \\
\alpha_i = f_i \odot \alpha_{i-1} + i_i \odot \sigma_i(W_i z_i + U_i h_{i-1} + b_i) \\
h_{\text{avg}} = \alpha_i \odot \sigma_i(c_i)
\]

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.

References:

http://jalammar.github.io/illustrated-bert/

https://www.asimovinstitute.org/neural-network-zoo/

Results

<table>
<thead>
<tr>
<th>Models</th>
<th>Training Loss (750k samples)</th>
<th>Testing Loss (250k samples)</th>
<th>Accuracy</th>
<th>F1</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naive Bayes (w/ tf-idf)</td>
<td>0.165</td>
<td>0.172</td>
<td>0.940</td>
<td>0.102</td>
<td>0.647</td>
<td>0.055</td>
</tr>
<tr>
<td>Naive Bayes (w/ word counts)</td>
<td>0.204</td>
<td>0.232</td>
<td>0.936</td>
<td>0.552</td>
<td>0.483</td>
<td>0.645</td>
</tr>
<tr>
<td>Logistic with regularization</td>
<td>0.093</td>
<td>0.123</td>
<td>0.953</td>
<td>0.546</td>
<td>0.674</td>
<td>0.459</td>
</tr>
<tr>
<td>Logistic with regularization and boundary</td>
<td>0.093</td>
<td>0.123</td>
<td>0.947</td>
<td>0.603</td>
<td>0.565</td>
<td>0.647</td>
</tr>
<tr>
<td>CNN basic</td>
<td>0.148</td>
<td>0.130</td>
<td>0.947</td>
<td>0.351</td>
<td>0.725</td>
<td>0.231</td>
</tr>
<tr>
<td>CNN bidirectional</td>
<td>0.155</td>
<td>0.131</td>
<td>0.949</td>
<td>0.428</td>
<td>0.672</td>
<td>0.324</td>
</tr>
<tr>
<td>CNN bidirectional &amp; Adam optimization</td>
<td>0.135</td>
<td>0.116</td>
<td>0.956</td>
<td>0.575</td>
<td>0.689</td>
<td>0.509</td>
</tr>
<tr>
<td>CNN bidirectional &amp; Adam &amp; boundary</td>
<td>0.126</td>
<td>0.135</td>
<td>0.951</td>
<td>0.625</td>
<td>0.591</td>
<td>0.662</td>
</tr>
<tr>
<td>Averaged Perceptron (MCNet)</td>
<td>-</td>
<td>-</td>
<td>0.941</td>
<td>0.293</td>
<td>0.667</td>
<td>0.188</td>
</tr>
</tbody>
</table>

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.