Prediction of Useful Votes on Yelp Reviews

Christina Pan (capan@stanford.edu), John Pothokaran (johnmp@stanford.edu) | Stanford University CS 230

Motivation

Yelp ratings and recommendations help users to search for different restaurants and pick restaurants to visit. However, not all Yelp reviews are useful, and identifying the most helpful reviews would add to the customer experience and prevent the website from being overly cluttered.

Objective

This project aims to build a model that predict the number of useful votes a review receives based on an analysis of the text, using state of the art natural language processing (NLP) and machine learning (ML) techniques.

Dataset

The dataset consists of the text of reviews gleaned from Yelp's Challenge dataset. It consists of a total of 1.5 million reviews, where the test set has 1.35 M reviews, the dev set has 75k, and test set 75k reviews. All reviews contain at most 50 words for memory reasons. 75% of reviews have 0 useful votes, 15% have 1 vote, and the maximum number of votes is 1341. This set of reviews contains a vocabulary of 170k words. The 20 most frequent words – aside from words like 'the' or 'a' - are shown in the graph below.

Approach

We decided to pursue a variety of NLP architectures of increasing complexity and assess their performances to find our most optimal model. All models receive the review's text converted into a word embedding representation (initialized with the GloVe vectors) and output the number of predicted votes. In addition, all models used in the experiments use the RMSE loss function and an Adam Optimizer.

Algorithms

Dense
The baseline model consists of an embedding lookup layer, followed by a simple, dense ReLu layer that outputs a prediction for the number of useful votes.

LSTM
Given the great success and sequential nature of information relayed over reviews, we tried a single layer LSTM.

Modified Bi-attentive Classification Network (BCN)
Since new work in encoder-decoder LSTM, attention, and CNNs have improved NLP classification tasks, we decided to modify the BCN used by McCann et al, with a self matching attention mechanism. (See equations below.) Due to overfitting and memory issues, the 'integration' LSTM was removed. Below is a diagram showing the layers.

Modified BCN Architecture:

Self-Attention Equations:

\[ s^i = \text{tanh}(W^s x^i + W^i s^i) \]
\[ s' = \exp(s')/\sum_{i} \exp(s') \]
\[ a_i = \text{softmax}(s') \]

Results & Discussion

Results Between Algorithms:

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Loss (MSE)</th>
<th>Average Absolute Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dense</td>
<td>0.098</td>
<td>0.313</td>
</tr>
<tr>
<td>LSTM</td>
<td>0.096</td>
<td>0.310</td>
</tr>
<tr>
<td>Modified BCN</td>
<td>0.000</td>
<td>1.86</td>
</tr>
</tbody>
</table>

As you can see, the LSTM does the best job on the dev set. Given that the modified BCN is a more complex model, it seems most likely that the modified BCN needs more hyper-parameter tuning.

Future Work

The predictions of the model could be improved by providing other contextual information from the Yelp Dataset and passing the features through different neural architectures. Identifying the usefulness of reviews could be approached as a NLP classification task as well.

Works Cited