horizontal line

Sprint 5 Results

Sprint Created: 11.10.2019

Sprint End: 11.17.2019

Sprint Captain: Yirou Ge

Sprint Co-captain: Hongying Liu

**─**

# Sprint Goal

## Optimize Neural Nets and the Train Function:

* Modify the train function for better performance: **on track**

1. Read more papers and try to modify the baseline model
2. Implement stratified k-fold cross validation to choose train and validation datasets for neural network model

* Add more layers into the neural nets **on track**
* Generate additional feature - POS tags and add them into the neural network model
* Try to fine tune RoBERTa

## Feature Engineering and Label Model Optimization:

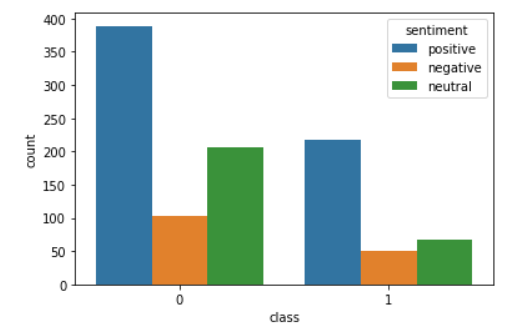
* Continue adding more features to train neural network models. This week we will focus on POS tags, adding topic modeling results to RNN model, and Automated Feature Engineering. **on track**
* Literature review on using snorkel labeling functions, combining with the manual labelling rules for gold\_standard dataset
* Use case demo of BabbleLabble model on gold\_standard dataset

# Results

1. Feature Engineering

For the past week, we generated 4 more features to add into the neural network:

* N-Gram: we added a unigram feature by counting whether the sentence include a certain unigram and selected the top 200 features to use
* POS tags: after inspecting the most common tags in the dataset, we decided to add the counts of POS tags 'NN', 'IN', 'JJ', 'NNS', 'DT', 'CC', 'CD', 'VB' as the new POS tag features
* Sentiment Score：
  + We first tried to get the sentiment score by counting positive words and negative words in each sentence, but the result is not good(the sentiment of more than 99% of sentences are neutral).
  + Then we tried another method for sentiment analysis, TextBlob, which is actually a high level library built over top of NLTK library. It uses a Movies Reviews dataset for training, and a Naive Bayes Classifier is used. The sentiment score got from TextBlob was added to our model.
  + We also explored the sentiment distribution of gold standard data, and found that in the 'positive' class, the percentage of sentences with positive sentiment is apparently larger than that in the ‘nonpositive’ class. (1=positive class)



1. RoBERTa Encoding Transformation

The original roBERTa encoding results return a (1, 50, 1024) vector for each sentence, where each row represents a token. In order to concatenate sentence vectors to the embedding results, we reduced its dimension to (1, 1024) by averaging the encoding results.

1. Neural Network Training

We transformed the Dataset function in the original codes to concatenate the new features we generated with the roBERTa embedding results.

The metrics for the neural network are as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Metrics/Models | Baseline Model | Noisy Model | Add New Features | Add New Features and layers |
| F1 Score | 0.509 | 0.40 | 0.377 | 0.383 |
| Accuracy | 0.58 | 0.63 | 0.606 | 0.62 |
| ROC AUC Score | 0.525 | 0.51 | 0.484 | 0.495 |

After adding new features (POS tags, n-gram, sentiment score), the model’s performance became worse but was improved after adding more layers.