Yelp Rating Prediction with Global Vectors for Word Representation (GloVe) Embedding, Bidirectional Long Short-term Memory (LSTM) and Gated Recurrent Units(GRU) Layers

Lai Jiang, Wenhuan Sun
Carnegie Mellon University
5000 Forbes Ave, Pittsburgh, PA 1523
1:laij@andrew.cmu.edu 2:wenhuans@andrew.cmu.edu

Abstract

Sentiment analysis of business review texts and subsequent star rating prediction are trivial tasks for humans, yet remain more difficult for machines. In this project, we analyzed a provided Yelp dataset and identified features that influence the polarity of the reviews. Then, we treated the task of star rating prediction based on review texts and selected features as multi-class classification problem and built several traditional classification algorithms as baseline models. The performance of each baseline model was analyzed using metrics including accuracy, confusion matrix, precision and recall matrix. A Recurrent Neural Network (RNN) based classifier with LSTM and GRU layers were designed and implemented. Using the same dataset, this model achieved significant improvement in all metrics when compared with baseline models and achieved 89.4% test accuracy. The same model structure was trained and tested using a larger dataset (1 million reviews) from Yelp.com, and higher test accuracy was observed with increased data size.

1 Introduction

Online reviews and star ratings in websites, such as Yelp.com, have been playing an important role in influencing the decision making of both business owners and customers. For example, many will refer to the corresponding Yelp reviews and overall star ratings before walking into a new restaurant. Quite commonly, restaurants owners or managers need to extract information from the comments and use them to adapt business strategies. In this sense, being able to understand and interpret the underlying messages in the review contexts is critical. This task is trivial for humans, as we have been trained to perform sentiment analysis since a very young age. However, it has remained an extremely difficult challenges until not long ago.

Recent advance in natural language processing (NLP) brought about several techniques that convert text input into vectors that are compatible with machine calculation. Based on these different conversion schemes, efforts have been make to improve the computational efficiency and to represent the latent correlation between words that appear in different locations in the text input, such as the different types of word embedding datasets.

Predicting the corresponding rating scores (usually 1-5) based the raw review texts and/or selected features were an interesting and meaning tasks, which can be extended to many other applications where a numeric rating is not available, for example, predicting the quality of a Youtube video based on the associated comments.

2 Methods

2.1 Data Source

For the baseline model evaluation and comparison, as well as model selection experiments for the RNN based classifier, the provided Yelp data was used, which contains 36692 entries with features ranging from the review text, a sentiment score to the number of occurrence of some selected words. For final evaluation of the selected model structure, a larger dataset from Yelp was used (available at https://www.yelp.com/dataset/download). To test its ability to generalize to all types of business reviews beyond restaurant reviews, the first one million reviews from this dataset were used in the training and testing of the model. For each routine, 80% of the data were used as training data, and 20% were reserved for testing.

2.2 Baseline Models

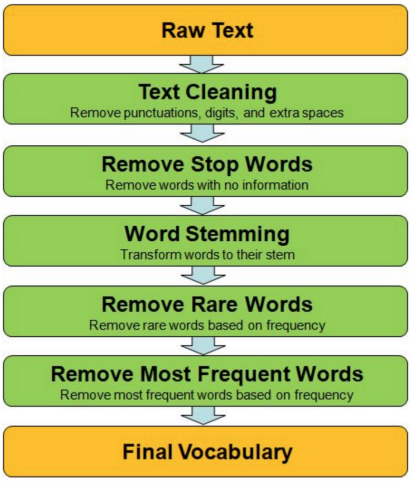

Figure 1: Texts pre-processing procedure for baseline models

Some common models were chosen for the baseline study, including K nearest neighbors, decision tree, random forest, logistic regression, multinomial naive Bayes, multilayer perceptron and gradient boosting.

The most important part for sentiment analysis is feature engineering. For text part, the raw text was firstly pre-processed according to the figure 1 procedure. During the process, words with no specific meaning (stop words), rare words and most frequent words were removed, because they don't contribute much to the classification. Then, the cleaned texts were vectorized based on the word occurrences in each review. Vader sentiment analysis was also performed and included in the non-text features in addition to provided data, which calculates the "positivity", "negativity", "neutrality" and "compound" of texts based on pre-assigned scores for each word. For non-text part, Chi-squared test were performed to rank the features, and only top important features were used for training. Finally, the vectorized text is combined with non-text features for training.

2.3 Effects of Selected Features on The Review Polarity

For each of the selected features, the data were grouped by manually selected ranges/bins from the corresponding parameters (e.g. sentiment score), and one-way ANOVA was performed to study the effect of those features on the review score.

2.4 GloVe Embeddings Based Tokenization

A preliminary check was performed and it was found that the review texts contain a large number of non-standard English words, which is close to the way people write twitter posts. Therefore, a

200-dimension GloVe embeddings from Twitter corpus were used to convert the raw review text into vectors for training and testing (tokenization). The twitter corpus was preferred over other ones, such as the Wikipedia corpus, for the affinity of language style between the Yelp reviews and Twitter posts.

2.5 Neural Network Based Model Selection

For simplicity, only review texts were used as input for neural network (NN) parameter tuning. NNs with 1, 2 and 3 layers were evaluated. The default setup parameters are: 64 units per layer for all the hidden layers, 'relu' as the activation function. The output layers used 'sigmoid' function and had 5 nodes that correspond to the five possible rating class. 3-layer NNs with varied activation functions and/or node number were also evaluated to study the effects of those architecture parameters. Features other than the raw texts were later included as auxiliary input, and the effects of their inclusion on the model prediction performance were also analyzed.

2.6 Introducing LSTM and GRU Layers

Traditional RNN models face the issue of vanishing gradients. Two techniques, LSTM and GRU, were used to combat the vanishing gradient problem and to learn the long-term dependency amongst the input texts. LSTM and GRU layers were added to standard default NNs from the model selection step respectively, and their contributions were evaluated (LSTM only, GRU only and LSTM + GRU).

3 Results

3.1 Effects of Selected Features on The Review Polarity

Among the selected features, sentiment score was shown to have the most statistically significant effects on the star rating. When grouped by star ratings, the sentiment-score distribution shifted towards the maximum score with increasing star ratings and vise versa. Similarly, when groped by sentiment score, the star ratings shifted towards the maximum with increasing sentiment scores (Fig. 2). These results showed that the sentiment-score may be a strong and effective predictor of star rating.

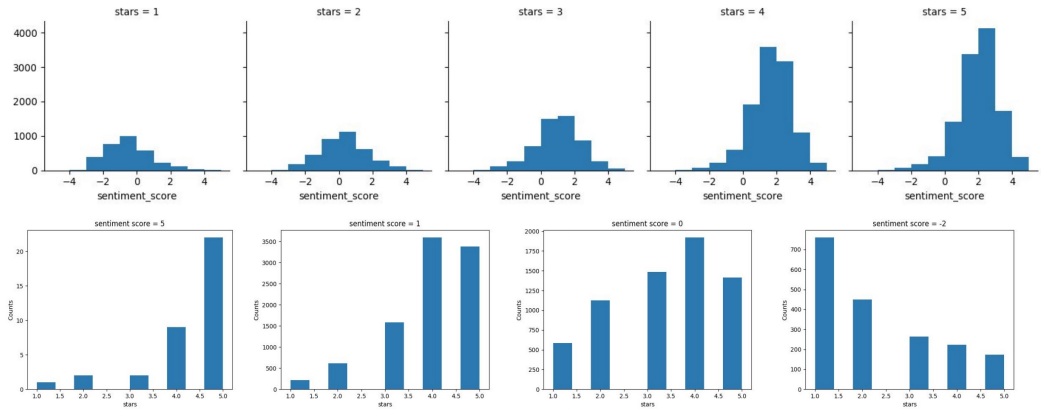

Figure 2: Sentiment score affects review rating distribution

3.2 Baseline Models

The results of polarity analysis and 5-star classification are tabulated in table 1. The polarity analysis is a binary classification which classifies 1, 2 and 3 stars as negative, and 4 and 5 stars as positive. Overall, the baseline classifiers work much better on polarity tests compared to 5-star classifications. For 5-star classification, they performed better on more extreme reviews such as 1 star and 5 stars, and F1-scores start to decay when texts become more neutral.

In our opinion, the main cause of the discrepancy between binary and 5-star classifications is the data loss during text vectorization. For example, a text like "Awesome!" and "AWESOME!" show totally

Classifier	Polarity Accuracy	F1-score Negative	F1-score Positive	5-Star Accuracy	F1-score for 1, 2, 3, 4, 5 stars				
Gradient Boosting	82.9%	0.83	0.83	54.45%	0.66	0.47	0.46	0.45	0.65
Multilayer Perceptron	86.8%	0.87	0.87	50.45%	0.66	0.43	0.40	0.43	0.59
Multinomial Naïve Bayes	82.65%	0.81	0.84	53.4%	0.66	0.41	0.43	0.44	0.66
Logistic Regression	83.95%	0.84	0.84	45.3%	0.62	0.36	0.40	0.30	0.55
Random Forest	76.75%	0.79	0.75	41.2%	0.56	0.32	0.30	0.32	0.53
Decision Tree	75.85%	0.76	0.76	40.3%	0.54	0.38	0.32	0.32	0.46
K Nearest Neighbor	66.05%	0.67	0.66	32.3%	0.42	0.27	0.24	0.29	0.40

Table 1: Accuracy and F1-score of baseline models

different emotional feelings, but the baseline text pre-processing treat them as the same. In addition, because non-standard English is commonly used in Yelp reviews, text like "yummmmmmy" can't be recognized by dictionary-based vectorizer, so it will be removed during the pre-processing. Therefore, this cleaning process yields a big information loss, which is not suitable for sentiment analysis.

3.3 Neural Network Structure Tuning

Given the same input data, NNs with larger depth were shown to have higher test accuracy (Fig. 3). Increasing layers adds to the model complexity and enables the model to capture more complex underlying correlations. However, continuing increasing layers might lead to over-fitting and increase in computation cost. The red lines in Fig. 3 represent a test accuracy of 84%. The number of hidden layers nodes per layer were also varied from 16 to 128 and the effects of this variation were evaluated. As shown in Fig. 3, the test accuracy increased and decreased when the number of nodes per layer increased from 16 to 128 and peaked at 64. Increasing the number of nodes per layer has similar effects with increasing the depth of the NN. As a result, for subsequent analysis, NNs with 3 layers and 64 nodes per layer were used as the default NN structure. The activation function was changed to tanh and the performance was evaluated in Fig. 4, where the test accuracy decreased with the change of activation function.

Three cases were tested to differentiate and gauge the contribution of LSTM and GRU layers (RNN + LSTM, RNN + GRU, and RNN + LSTM + GRU), and the training and test accuracy were evaluated in Fig. 4.

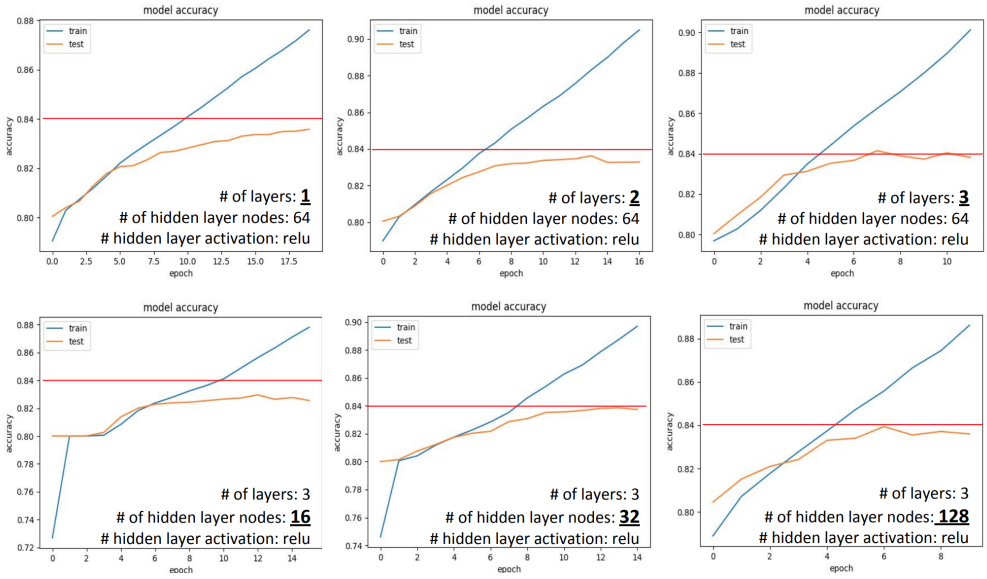

Figure 3: Performance comparison among different NN architectures: training and validation accuracy

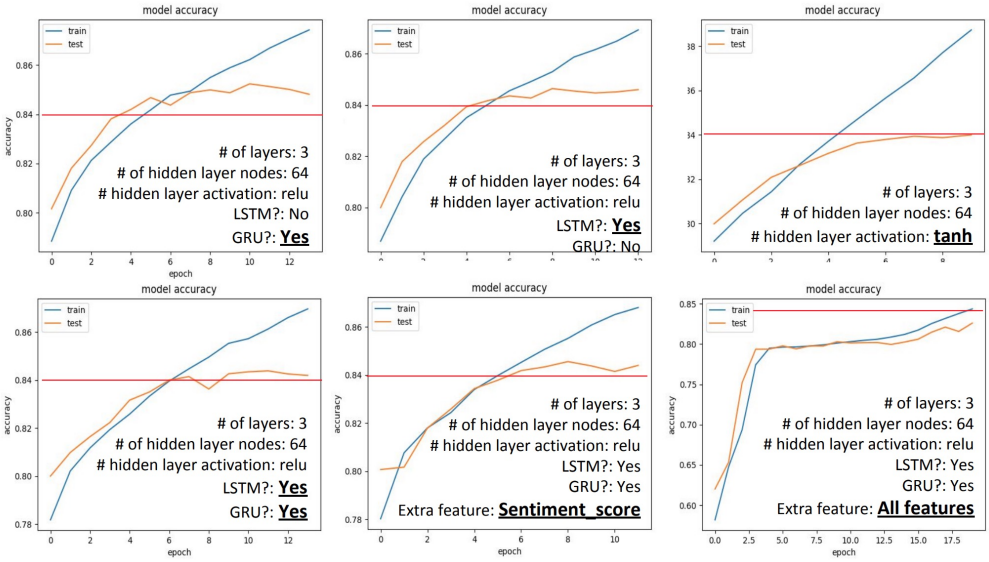

Figure 4: Performance comparison among different NN architectures (LSTM and GRU layers): training and validation accuracy.

3.4 Effects of Introducing LSTM and GRU Layer

As shown in Fig. 4, adding GRU or LSTM layers significantly improved the test accuracy from the RNN only models. However, the difference in the performance between LSTM and GRU layer is not significant. When the two types of layers were packed in the pipeline, the performance was not significantly further improved from LSTM-only or GRU-only models. Therefore, in subsequent model, a RNN followed by LSTM and GRU was used.

3.5 Effects of Introducing Selected Features

Another model structure was implemented where the selected features (sentiment-score or all-features) were introduced in the auxiliary input layer. The auxiliary layer was concatenated with the tokenized text input ad fed into the subsequent RNN (Fig. 5). As shown in Fig. 4, the introduction of sentiment-score did not significantly improved the model test accuracy. And the addition of all features in to the auxiliary input layer significantly decreased the performance, which is probably due to excessive model complexity.

3.6 Training and Testing with Larger Dataset

To test the model's ability to generalize to all types of business reviews beyond restaurant reviews, the first 100,000, 300,000 and one million reviews from a larger dataset from Yelp.com were used in the training and testing of the model (for each case, 80% used for training and 20% used for testing). As shown in Fig. 6, the test accuracy increased with the increase in training dataset size, and the test accuracy were significantly improved from 0.84 to 0.895. In addition, the loss decreased with increasing training dataset. The confusion matrix revealed that most of the data were correctly classified, and the classification error was lower with more extreme data.

4 Discussion and Future Work

From the experiments explored in this project, it as shown that:

1. the optimal RNN architecture for this classification problem is: text-only input, tokenization based on GloVe embedding, LSTM and GRU layers and 3-layer NN (activation: relu, number of nodes per hidden layer: 64)

2. higher training data yielded higher testing accuracy and less over-fitting

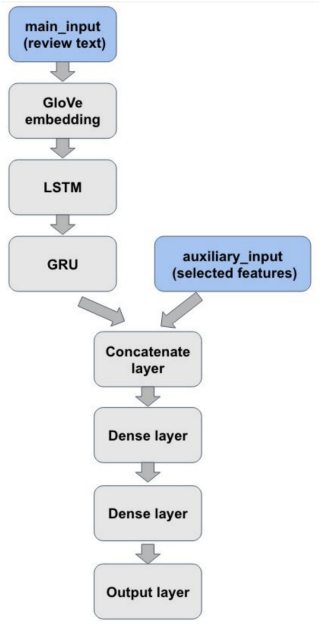

Figure 5: Network structure with auxiliary input

3. Achieved an overall classification accuracy of 89.4%, which is a significant improve from the baseline models.

The following strategies may be implemented to further boost the model performance:

1. ensembles of this RNN and other models

2. use combination of embedding instead of a single embedding

3. perform text cleaning

4. Implement train-time augmentation (TTA) to improve model robustness.

Acknowledgments

The authors would like to thank the teaching team for the help during the brain-storming, implementation and testing of the projects.

References

[1] Ganu, Gayatree, Noemie Elhadad, and Amelie Marian. ”Beyond the Stars: Improving Rating Predictions using Review Text Content.” WebDB. Vol. 9. 2009.

[2] Li, Chen, and Jin Zhang. ”Prediction of Yelp Review Star Rating using Sentiment Analysis.”

[3] Sabnis, Omkar, "Sentiment Analysis on the Yelp Reviews Dataset"

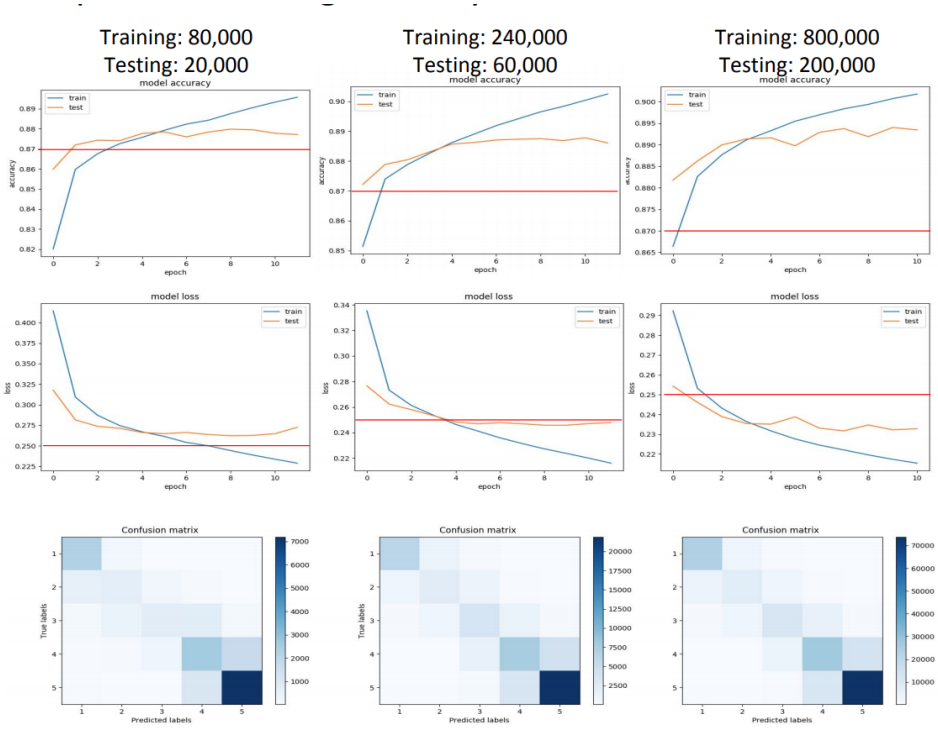

Figure 6: Training and testing using different amount of data