A Close Look of Text Classification for Yelp and Amazon Reviews

Jiaxu He
Analytics Program of Georgetown University

Abstract

In this project, I investigate the effectiveness of using Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) to classify the reviews and gain insights of whether the reviews are helpful or not. To classify the reviews and predict whether the reviews are helpful is a big challenge, but the good results will help both customers and consumers to improve the effectiveness of business. I also tried several methods besides neural networks, include logistic regression ,Naïve Bayes and so on to compare and contrast the effectiveness. I find out that predicting usefulness is really hard, but CNN and RNN have better results than other methods. Classifying the reviews have much better results in all the methods.

Introduction

As the main part of the connection between customers and sellers, reviews have given a lot of information and been used in a lot of ways. However, the classification of the reviews has been a topic for long time, not only because it can show the positive or negative attitude, but also it can determine the mistaken reviews. Meanwhile, the quality of the reviews varies significantly. While poor reviews do not provide accurate information and useful advice, high quality reviews received a lot of agreements overtimes and proved useful information not only for other customers, but also for the sellers to improve or keep working. In that way, getting the quality and fresh review provides important commercial value to business. On the other way, knowing which review are the fresh and quality one helps customers to gain insights into the services and products.

Data

The data that I use in this project come from sources, Amazon and Yelp. For Amazon data, I extract the reviews only about the "electrics products". For Yelp, the reviews are all about "food and restaurants". In this way, the two data represents two different categories, which will help to train model more specifically. Besides, the reviews come from recent four years. So, I separate the data into years to create smaller datasets. Those separations will help to understand the influence of topics and years.

In order to classify the data, I clean the data for "Stars" and "Helpful Votes". For the review that has "Stars" less or equal than 3, I mark it as negative review, otherwise mark it as positive. For the review that has "Helpful Votes" more or equal than 1, I mark it as helpful, otherwise mark it as not helpful.

The total dataset have more than 110000 reviews, and the numbers of reviews of "Positive" and "Negative" are the same, and the numbers of reviews of "Helpful" and "Not Helpful" are the same.

Methods

The methods that I use could be divided into two parts:
The first part includes using bag of words and word2vec to processing the data, and using logistic regression and Naïve Bayes to do the prediction.
The second part is using CNN and RNN to do the prediction. The models of CNN and RNN are:

CNN:

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d_5 (Conv1D)            (None, 4463, 128)         17408
_________________________________________________________________
max_pooling1d_5 (MaxPooling1 (None, 892, 128)          0
_________________________________________________________________
conv1d_6 (Conv1D)            (None, 892, 128)          82048
_________________________________________________________________
max_pooling1d_6 (MaxPooling1 (None, 178, 128)          0
_________________________________________________________________
conv1d_7 (Conv1D)            (None, 178, 128)          82048
_________________________________________________________________
max_pooling1d_7 (MaxPooling1 (None, 35, 128)           0
_________________________________________________________________
conv1d_8 (Conv1D)            (None, 35, 128)           82048
_________________________________________________________________
max_pooling1d_8 (MaxPooling1 (None, 7, 128)            0
_________________________________________________________________
dropout_3 (Dropout)          (None, 7, 128)            0
_________________________________________________________________
flatten_2 (Flatten)          (None, 896)               0
_________________________________________________________________
dense_3 (Dense)              (None, 128)               114816
_________________________________________________________________
dropout_4 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_4 (Dense)              (None, 1)                 129
=================================================================
Total params: 378,497
Trainable params: 378,497
Non-trainable params: 0
```

RNN:

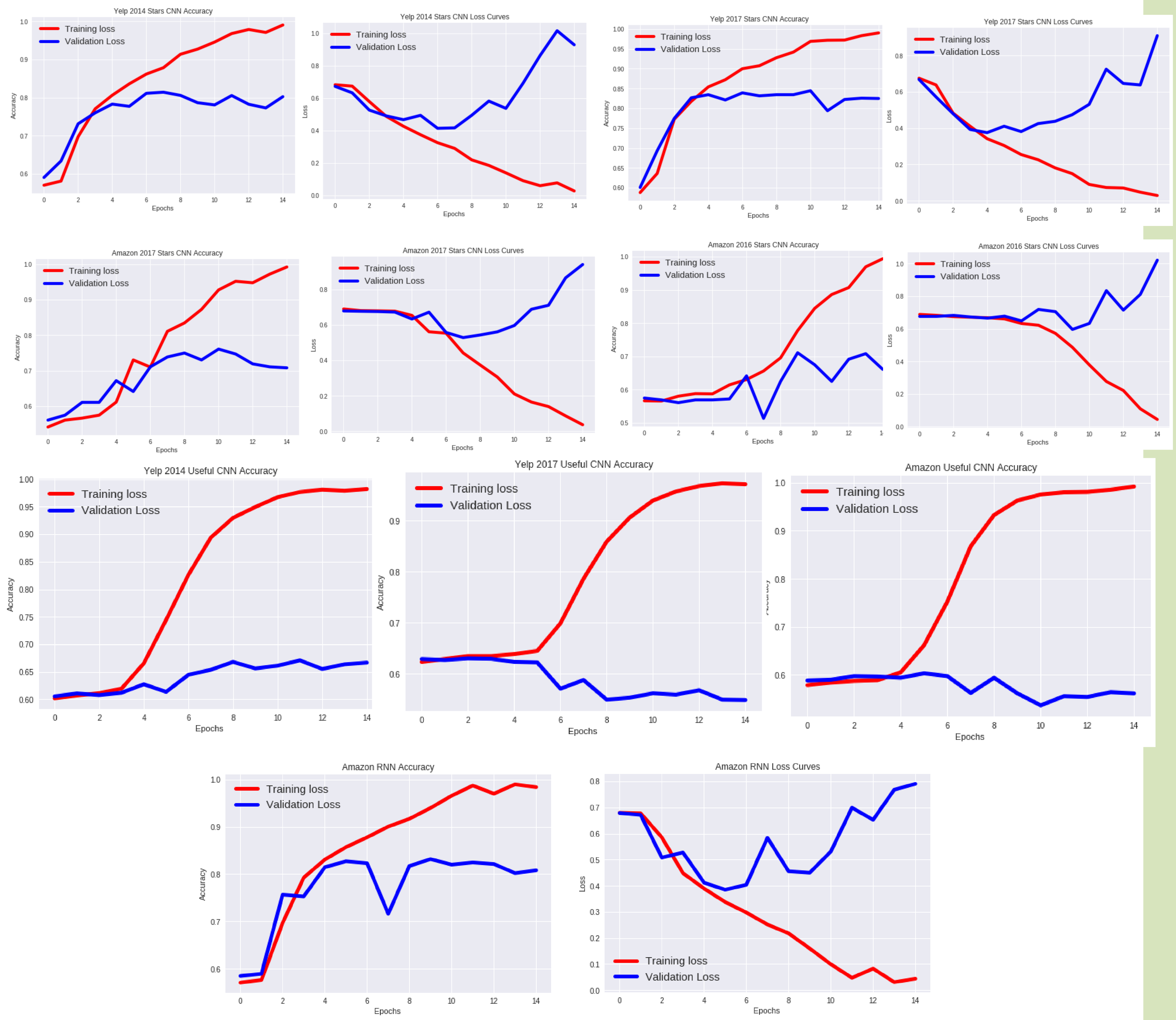
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_1 (LSTM)                (None, 4271, 100)         51200
_________________________________________________________________
lstm_2 (LSTM)                (None, 50)                30200
_________________________________________________________________
dropout_1 (Dropout)          (None, 50)                0
_________________________________________________________________
dense_1 (Dense)              (None, 50)                2550
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 51
=================================================================
Total params: 84,001
Trainable params: 84,001
Non-trainable params: 0
```

Results

Part1:

	Logistics Regression	BernoulliNB
Amazon Classify	0.8128	0.8010
Amazon Helpful	0.5527	0.5320
Yelp 2014 Classify	0.7734	0.8122
Yelp 2017 Classify	0.8467	0.8192
Yelp 2014 Helpful	0.5069	0.5062
Yelp 2017 Helpful	0.5228	0.5117

Part2:



Discussion

The methods in part1 have good results on classifying the reviews to positive or negative, however, the prediction of whether the review is helpful is meaningless since the results are around 0.5.
The similar results happened in Part2, while CNN have great results in classifying reviews with the accuracy more than 80% for most cases. But the helpful review prediction keeps around 60%, which is a little bit higher but could not help too much.

There are also other insights from the results. Generally, amazon have better results than yelp, which means electric products' reviews are easier to classify than food and restaurants' reviews. Also, by comparing the reviews by years. It is easily to find out that recent reviews have better results than the older reviews. That may caused by the changing of culture.

There are definitely a lot more things to improve this project. The first thing is model structures, because results is not perfect. And also there could include more topics data. And last may discover a little bit more about the relationship between helpful votes and reviews.

Contact

Jiaxu He
Georgetown University
Email: jh2045@Georgetown.edu
Phone: 3193839417

References

1. https://www.kaggle.com/yelp-dataset/yelp-dataset/version/4
2. https://www.kaggle.com/datafiniti/consumer-reviews-of-amazon-products
3. http://nlp.fast.ai/classification/2018/05/15/introducting-ulmfit.html
4. https://taylorwhitten.github.io/blog/word2vec