Jiaxu He

ANLY-590

Final Project Writeup

A Close Look of Text Classification for Yelp and Amazon Reviews

Abstract

In this project, I investigated the effectiveness of using Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) to classify the online 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 and Naïve Bayes to compare the effectiveness. I found out that predicting usefulness is really hard, but CNN has relatively better results than the other methods. And logistic regression and CNN lead the results in classifying whether the reviews are positive and negative.

Introduction

As the main part of the connections between customers and sellers, reviews have given a lot of information and have 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 service or keep on it. 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.

Related Work

In the past related work, classifying the usefulness and attitude of the reviews has always been separated in two individual tasks. But in general, the main work of this project could be classified into “Text Classification” area, and there are a lot of people and researchers keep contributing in this area. The most famous work in this area was done by Yoon Kim in 2014. He provided a convolutional neural network method for sentence classification, which has been considered as the first person actually used CNN in this area. The CNN model that Yoon Kim built is a basic model, which contains input layer, convolutional layer, max-pooling layer and softmax layer. Since the input is texts instead of images, he used word embedding for input data

representation. In the training process, the model added some dropout layers. As the results Yoon Kim provided, his work achieved a great success. And most importantly, he provided a new thought to deal with text classification. As a traditional method dealing with images in deep learning, CNN and other neural networks have been discovered more possibilities after Yoon Kim's first step. Middle of this year, A universal language model claims to be the state of art in text classification. It uses transfer learning method that can be applied to any task in NLP. It also introduces the techniques that are key for fine-tuning a language model. their method significantly outperforms the state-of-the-art on six text classification tasks, reducing the error by 18-24% on the majority of datasets. Furthermore, with only 100 labeled examples, it matches the performance of training from scratch on 100x more data. Comparing with those researchers pioneers, my method seems to take a starting insight into this field and try to focus on the influences and results for different situations include years, categories, attitude and usefulness.

Datasets

The datasets that I used in this project come from Kaggle, and they contain the reviews from Yelp and Amazon. For Amazon data, I extracted the reviews only about the "electrics products". For Yelp, the reviews are all about the "food and restaurants". In this way, the two datasets represent two different categories, which will help to train model more specifically. Besides, the reviews come from recent four years. So, I separated the data into four years to create smaller datasets. Those separations will help to understand the influence of topics and years. In order to classify the data, I cleaned the data for "Stars" and "Helpful Votes". For the review that has "Stars" less or equal than 3, I marked it as negative review, otherwise marked it as positive review. For the review that has more or equal than one "Helpful Votes", I marked it as helpful review, otherwise marked it as not helpful review. The total dataset has more than 110000 reviews, and I cleaned and processed the data, so it has same number of positive reviews and negative reviews. Also, it has same number of helpful reviews and not helpful reviews. To set every element in the same situation, the datasets that were separated in four years also contain similar total amount and equal reviews for helpfulness and attitudes.

Methods

The first part of doing text classification is to pre-process the texts. There are a lot of ways to produce word embedding. In this project, I chose word2vec and bag of words, while word2vec creates a vector space of high dimensions and bag of words counts the frequency and changes the texts into multiset of words represented by numbers. Through these two pre-process text methods, the input data could be treated well and ready for training and testing part.

The models that were used to train and test in this project could be divided into two types. One is machine learning methods include logistic regression and Naïve Bayes, the other one is deep learning methods include CNN and RNN. While logistic regression and Naïve Bayes are the classic methods for classification, they are expecting to work well with texts after processing the texts into numeric input. And I built CNN and RNN models in simple architecture. My CNN model contains 378497 parameters in total, which include four convolutional layers followed by

max-pooling layers and two dropout layers. My RNN model contains 84001 parameters in total, which include two lstm layers and one dropout layer.

Results

First, I performed logistic regression and Naïve Bayes to predict the classification results. As Figure 1 shows, the logistic regression and Naïve Bayes methods showed strong power in classification area. Almost all the datasets about attitudes have accuracies that are higher than 0.77. However, the results about the reviews are helpful or not did not have good results for these two methods.

	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

Figure 1. (Table for ML Methods Results)

Moving into deep learning methods, I tried CNN first, and the results surprised me and pleased me in the same time. As Figure 2 shows, keeping year and prediction target same, two categories datasets showed two different results. Amazon, which represents “electric products” category, has around 0.75 prediction accuracy. And Yelp, which represents “food and restaurants” category, has around 0.85 prediction accuracy.

Figure 2. (CNN Results for Different Categories)

Figure 3 is showing the results of how years influence accuracy while keeping prediction target, categories same. As the graphs indicates, older year (2014) has the accuracy of 0.8, and the recent year (2017) has the accuracy of 0.85.

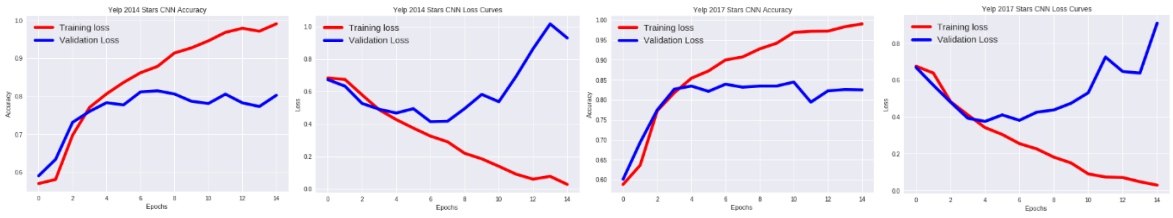

Figure 3. (CNN Results for Different Years)

Figure 4 is the results of two prediction targets comparison while keep year and categories same. The results surprised me because the prediction accuracy for usefulness is much lower than the prediction accuracy for attitudes.

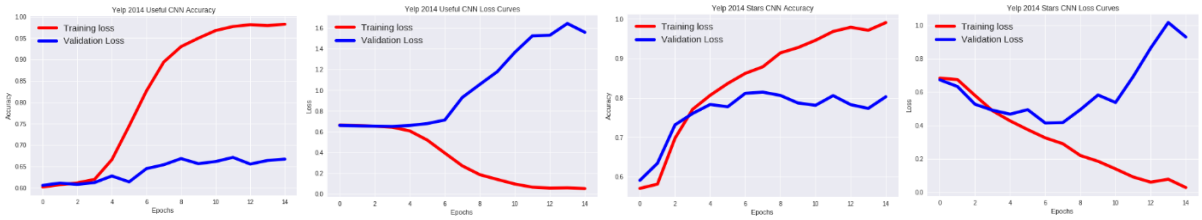

Figure 4. (CNN Results for Different Prediction Targets)

Next, I tried RNN to perform the tests whole over again. However, the results were not very good. I ran the tests a lot of times, the highest accuracy I got is around 0.6 no matter the year, prediction targets and categories. Finally, I chose a small dataset from Amazon reviews by hand, and one lucky run gave me figure 5.

Figure 5. (RNN One Lucky Result for Amazon Reviews)

Discussion

Logistic Regression and Naïve Bayes outperformed than I expected, they have good results in classifying whether the reviews are positive or negative. However, to predict the helpfulness of the reviews is hard for them. For CNN, it also did a good job in classifying the attitudes of the reviews. More importantly, it has better result in classify the usefulness of the reviews since the accuracy was very close to 0.7. On the other hand, RNN surprisingly performed very bad. The results were around 0.6 for either attitudes and helpfulness. The only one good result I got is to choose a small dataset from amazon reviews by hand. And after ran several times, one got a close accuracy of 0.8 with obviously changing tend in graph. Besides, recent years' reviews have better results than older years. And "food and restaurants" topic has better result than the topic of "electric products". Also, predicting the attitudes of the reviews is much easier than predicting the usefulness of the reviews in general.

Conclusions

After performing different methods for different datasets, I found that classifying attitudes of the reviews is relatively easy since the words in reviews would reflect attitudes more or less. However, the usefulness of the reviews may be decided by other elements like other people's experiences and their special. I also found that older reviews are gentler, which made them harder to predict the attitudes than the recent reviews. Also, "food and restaurant" topic has stronger attitudes showing in reviews than "electric products" topic.

References

Brown, W, Taylor. “Introduction to Word Embedding Models with Word2Vec”. *SOCIOCOMP*. 10 Jul 2016. https://taylorwhitten.github.io/blog/word2vec

Datafiniti. “Consumer Reviews of Amazon Products”. *Kaggle*. Nov 2018. https://www.kaggle.com/datafiniti/consumer-reviews-of-amazon-products

D’Souza, Jocelyn. “An Introduction to Bag-of-Words in NLP”. *A Medium Corporation*. 3 Apr 2018. https://medium.com/greyatom/an-introduction-to-bag-of-words-in-nlp-ac967d43b428

Howard, Jeremy and Ruder, Sebastian. “Introducing state of the art text classification with universal language models”. *NLP fast.ai.* 15 May 2018. http://nlp.fast.ai/

Kim, Yoon. “Convolutional Neural Networks for Sentence Classification”. *New York University*. 25 Aug 2014. https://www.aclweb.org/anthology/D14-1181

Reddy, Maryada Krisha. “Simple LSTM for text classification”. *Kaggle.* Apr 2018. https://www.kaggle.com/kredy10/simple-lstm-for-text-classification

Yelp, Inc. “Yelp Dataset”. *Kaggle*. 2017. https://www.kaggle.com/yelp-dataset/yelp-dataset/version/4