CS224D Project Final Report
Summarizing Reviews and Predicting Rating for Yelp Dataset

Suhas Suresha
ICME
Stanford University
suhas17@stanford.edu

Abstract

The report explores the use of Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN) in summarising text reviews and predicting review rating for the Yelp dataset. I use the fact that the reviews are labelled (by rating) to extract important sentences/words, which are then used as the summary for the review. I use an interesting evaluation technique to measure the relevance of the summary by comparing the performance of the summary with a randomly extracted text in predicting the rating of a review.

Keywords: LSTM Recurrent Neural Network, Convolutional Neural Network, NLP

1. Introduction

The task of automatically summarizing a paragraph into a few important sentences or key words is a very important problem in Natural Language Processing. This task involves identifying the key phrases and words in a paragraph that captures the meaning/sentiment of the whole paragraph. Automatic summarization is extremely useful for companies like Yelp to provide a better service for their users.

Another important problem in Natural Language Processing is predicting the sentiment of a paragraph. In this report, I look at predicting the rating (a very good measure of sentiment) of a review from the text. This task also involves identifying key words/phrases that summarize the sentiment of the paragraph. Rating prediction is also really useful for Yelp to provide better personalized suggestions for its users by sensing the sentiments in their reviews.

It is obvious that solving one of the above problems can greatly assist in solving the other. I make use of this fact to use the labelling of a review to automatically summarize review text into key sentences/words. I use LSTM Recurrent Neural Networks and Convolutional Neural Networks for the task of predicting rating and also summarizing text from reviews. I explore an interesting technique to measure the relevance of a summary by comparing the performance of the summary with a randomly extracted text in predicting the rating of a review.

The rest of the paper is structured as follows. I first introduce the problem and the dataset being used in detail and provide some literature review. I then go into detail about

the neural network architecture used for solving the problem. I finally provide results, challenges encountered and possible future work.

2. Problem Description

My problem is to summarize text reviews and predict review rating for the reviews in the Yelp dataset. In this report, I follow the idea of Denil et. al. of using the fact that the review text is labelled to summarize sentences or words. I also, in the process, do the task of predicting ratings for the review text. The task of summarization that I am performing here is extractive summarization. Extractive summarization identifies the important text/words and throws away the rest, leaving the passage shorter.

I experiment with 2 methods to predict the ratings for the review text. In the first method, I extract **key words from the review text** while predicting the rating. In the second method, I extract **key sentences from the review text** while predicting the rating.

Typically, in most summarization tasks, the performance is evaluated using certain language metrics like ROGUE or BLEU. But I have tried to take a different path to evaluate performance of summarized sentences. I use the already trained model that we used for predicting ratings and feed in the summarized sentences (obtained from the model used to predict ratings) in a intermediate layer of the model. I also feed in a random text in the same intermediate layer. I now compare the performance of the summarized sentence against the random text in predicting the rating of the text correctly. I measure the percentage increase in the performance of the summarized text compared to the random text.

I use LSTM Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN) in my model architecture to perform the above mentioned tasks.

3. Data Extraction and Preprocessing

The Yelp datset consists of 2.2 million reviews and 591K tips for 77K businesses. For the first task in which I extract **key words from the review text**, I only use reviews having more than 35 words in their text. I extract exactly 40,000 such review texts along with their rating. I use 30,000 of these reviews as my training dataset, 5000 reviews as my validation set and the remaining 5000 as my test set.

For the second task in which I extract **key sentences from the review text**, I only use reviews having more than 10 sentences in their text. I extract exactly 100,000 such review texts along with their rating. I use 60,000 of these reviews as my training dataset, 20,000 reviews as my validation set and the remaining 20,000 as my test set.

For both tasks, I create a word to vector dictionary to provide a vectorized representation for all the words in the review texts. For this task, I first loaded the pre-trained word2vec model trained on part of Google News dataset. The model contains 300-dimensional vectors for 3 million words and phrases. I then created a dictionary linking all the words in the extracted dataset to a word vector. If the word was already present in the pre-trained word2vec model, I used the same vector representation. For new words which were not present in the Google News datset, I created random vectors of the same size as present in the pre-trained model.

4. Model Architecture

I now describe the model architecture I used for key word and sentence extraction tasks and in the process, predicting the rating.

4.1. Key Word Extraction

For key word extraction, I use 40,000 review texts and their corresponding ratings. All the extracted reviews have more than 35 words in their text. Using the word to vector dictionary, I obtain an initial vector representation for all the words in the review texts (each vector of length 300). For each of the text, I fixed the length of my input review sentence to be exactly 35 words. If the review was longer than 35 words, I would simply neglect the other words and keep the first 35 words. My task is to extract **3 key words** for each of the reviews which signifies its sentiment. So my input matrix is of size (Number of reviews, 35, 300). I do a train-validation-test split as explained in the previous section.

I now feed the training input data to a 2-layer LSTM RNN architecture as shown in figure 1. At the end of the 2 LSTM layers, I get a new trained vector representation of vector size 30 for each of the 35 words in a review. I pass these word vectors of size (35,30) through an affine layer to obtain a vector of size (3,30) (3 vectors of size 30). Let's call this vector as the key word vector. I then do a softmax regression on the key word vector to obtain a prediction for the rating. I use categorical cross-entropy loss (along with regularization) for training the model.

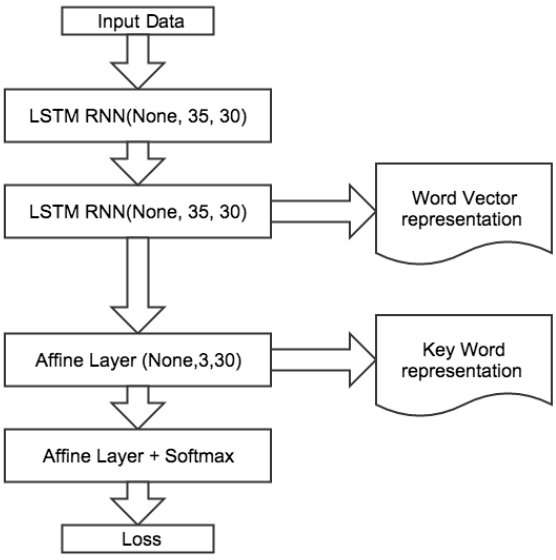


Figure 1: Architecture for Key Word Extraction

Now once I train the model to obtain better predictions for the rating, I look to extract 3 key words from each of the reviews. In this step, I use the trained word vectors after the LSTM layers as my input (35 vectors of size 30). I feed this input in the trained

model to obtain the key word vector (3 vectors of size 30) as my output. Now, I choose all possible combinations of 3 words from the trained word vectors and compare their word vector representation with the 3 key word vectors and determine the error. Then I select the 3 words with the least error as my **three key words** for the given review.

To measure the performance of my word summarization, I pass the 3 key words through the softmax regression to obtain a rating. I also pass 3 random words from the review throgh the softmax to obatin a rating. I evaluate performance of the 3 key words with the 3 random words to evaluate the performance of my word summarization. I measure the percentage increase in the performance of the summarized text compared to the random text.

Now, the reason I think this method of evaluation works is because the model will train in such a way that it chooses the 3 most important words that impacts the rating. These 3 words are likely to be the most important words when writing a summary of the review as well.

4.2. Key Sentence Extraction

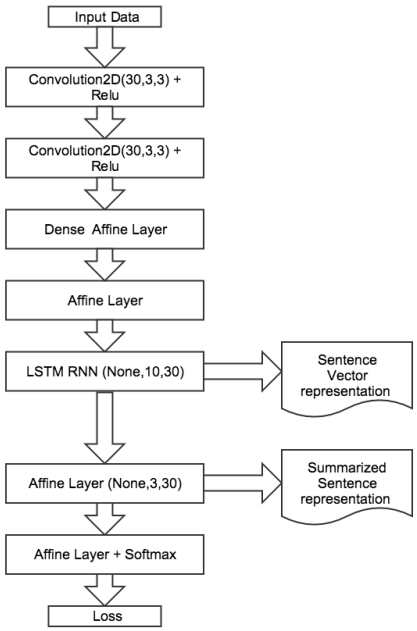
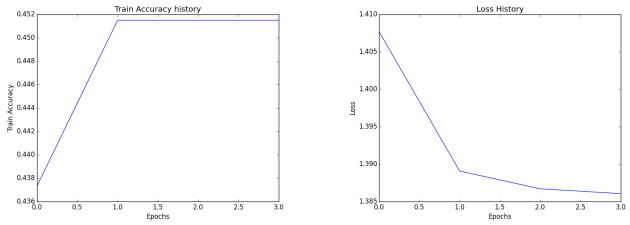


Figure 2: Architecture for Key Sentence Extraction

The methodology followed here is very similar to the above method. For key sentence extraction, I use 100,000 review texts and their corresponding ratings. All the extracted reviews have more than 10 sentences in their text. Using the word to vector dictionary, I obtain an initial vector representation for all the words in the review texts (each vector of length 300). For each of the text, I fixed the length of my input to be exactly 10 sentences. If the review was longer than 10 sentences, I would simply neglect the other words and keep the first 10 sentences. In each sentence, I only keep the first 7 words in the sentence. My task is to extract **3 key sentences** for each of the reviews which signifies its sentiment. So



(a) (b)

Figure 3: History of Loss and Train Accuracy for Key Word Extraction model

my input matrix is of size (Number of reviews, 10, 7, 300). I do a train-validation-test split as explained in the previous section.

I now first try to obtain a vector representation for each of the 10 sentences in each review. For this task, I pass the input through a Conventional Neural Network as shown in figure 2. The output of the CNN is then passed through a LSTM RNN to obtain a sentence vector representation for each of the 10 sentences (of size 30). Now I follow the same method as before to obtain a key sentence vector (3 vectors of size 30). I then do a softmax regression on the key sentence vector to obtain a prediction for the rating. I use categorical cross-entropy loss (along with regularization) for training the model.

Now once I train the model to obtain better predictions for the rating, I look to extract 3 key sentences from each of the reviews. In this step, I use the trained sentence vectors after the LSTM+CNN layer as my input (35 vectors of size 30). I feed this input in the trained model to obtain the key sentence vector (3 vectors of size 30) as my output. Now, I choose all possible combinations of 3 sentences from the trained sentence vectors and compare their vector representation with the 3 key sentence vectors and determine the error. Then I select the 3 sentences with the least error as my **three key words** for the given review.

To measure the performance of my sentence summarization, I follow the exact same procedure as before for word summarization.

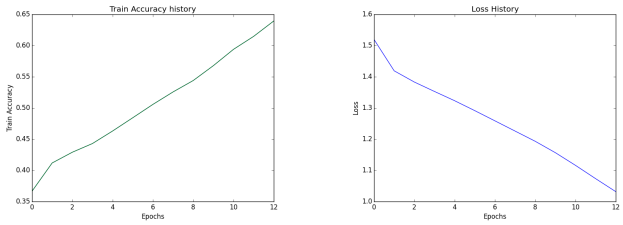
5. Results

I discuss the key results for both rating prediction and word/sentence summary performance in this section.

5.1. Key Word Extraction

The final key word extraction model test accuracy for rating prediction is **39.14%**. The final training and validation accuracy for rating prediction were **45.15%** and **40.18%** respectively. The loss and accuracy history for rating prediction is shown in figure 3.

Looking at the word summary performance, I observed that summary sentences gave a **35.67%** rating prediction accuracy on the training set and **34.35%** accuracy on the test set. Whereas, the random words gave a **33.65%** rating prediction accuracy on the training set and **34.14%** accuracy on the test set. We can observe that the summary words perform better compared to the randomly chosen words.



(a) (b)

Figure 4: History of Loss and Train Accuracy for Key Sentence Extraction model

A few examples of word summary for review texts is given below:

Review: Trying to book New Year's Eve dinner for four at the last minute anywhere is going to be a pain in the arse.You will get your pocketbook emptied at most places.I ended up making reservations here because it looked like a place we could afford.16.25 for a main entree).The restaurant is tucked inside a mall in the middle of downtown.Half of the restaurant is glass and overlooks the city.Most of the tables line the windows.The food itself was decent.I had their "spots".which was a crusted and pan fried white fish.It was good but was it $20 good.Not so much.The salad that accompanies the meal had a housemade balsamic vinaigrette to die for.I think I could've just eaten the balsamic vinaigrette for dinner and been good.Other diners in our party had the carbonara.proclaimed good.but not great).a shrimp with clams and garlic butter sauce pasta dish.proclaimed very tasty.

Word Summary: 'pain', 'good', 'housemade'

Review: The steaks here are very good.It's the same quality as the other high end places.prime cuts.aged.etc.However.I love the fact that they serve them on platters sizzling with butter.That's something somewhat unique compared to some of the other shops.I love the sound of the sizzle as the plates approach the table.I had dinner a couple of nights ago and while the steaks are top notch the bar isn't that good.The bartenders are competent.but the atmosphere is pretty lame.It's kind of an afterthought it seems.Oh yeah.and they don't have Guiness on tap.I'm going to start a petition that no liquor licenses should be issued without a guarantee that the proprietor have guiness on tap.I mean.how do you not at a steakhouse.If you go here for dinner do your happy hour elsewhere.The service is very good.

Word Summary: 'good', 'quality', 'sizzle'

5.2. Key Sentence Extraction

The final key sentence extraction model test accuracy for rating prediction is **37.05%**. The final training and validation accuracy for rating prediction were **63.95%** and **37.25%** respectively. The loss and accuracy history for rating prediction is shown in figure 4.

Looking at the sentence summary performance, I observed that summary sentences gave a **33.27%** rating prediction accuracy on the training set and **32.92%** accuracy on the test set. Whereas, the random sentences gave a **30.65%** rating prediction accuracy on the

training set and **31.54%** accuracy on the test set. We can clearly observe that the summary sentences perform better compared to the randomly chosen sentences.

A few examples of sentence summary for review texts is given below:

Review: I like it here.I've only visited in the past for happy hour or drinks out with friends.Recently.both times I came late in the evening after work functions and arrived hungry.Coincidentally.both nights I was there was also salsa night.Itś an interesting crowd.all ages.just gettinǵheir salsa on.I enjoy the music.but will sit back and let the "pros" do the dancing.They have friendly service.a few outdoor tables and a huge bar.Itś a pretty lounge.y vibe.There is a decent wine/beer list.and also specialty cocktails.Both times.I tried the sushi.which was not bad at all.

Sentence Summary: I've only visited in the past for happy hour or drinks out with friends. I like it here. It's an interesting crowd.

Review: 'Olive or Twist is the historic site of my VERY FIRST MARTINI when I turned 21.many years ago.It's been a long.happy union.While Olive or Twist is NOT a five star restaurant.it gets 5 stars from me because I love it.Its one of the few places I know of that has been CONSISTANTLY good.for years.The interior is warm and a little bit industrial.The downstairs is the same as it has always been and itś still very nice.The drinks are great although the first martini I ever had.a girl doesn't forget.[well.this one didn't]) was a Pineapple Upside Down martini and I don't think they offer that anymore but they do have a very expansive martini menu.I tried the chipotle martini and it was so excellent that when I took my first sip I sighed in a way my bf turned to me and sarcastically said."Oh.you're going to be Yelping about that.aren't you.Then.I refused to let him even try it bc it was that good.

Sentence Summary: many years ago.While Olive or Twist is NOT a five star restaurant.Its one of the few places I know of that has been CONSISTANTLY good

6. Final Discussion

The task of combining both the summarization problem and the rating prediction problem seems to work pretty well. As a future work, I would like to experiment with other model architectures to see if we have improved performance.

7. References

1. Extraction of Salient Sentences from Labelled Documents, Misha Denil, Alban Demiraj, Nando de Freitas1. Under review as a conference paper at ICLR 2015
2. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Effcient Esti- mation of Word Representations in Vector Space. In Proceedings of Workshop at ICLR, 2013.