Sentiment Analysis of Restaurant Reviews using Deep Learning

Srivathsava Bathini, Chandhana krishna vardhana Chinthakindi, Siva Keerthik Reddy Tagira

*Dept. of Computer Science and Engineering*

University of Texas at Arlington  
sxb8962@mavs.uta.edu, cxc0550@mavs.uta.edu, [sxt6529@mavs.uta.edu](mailto:sxt6529@mavs.uta.edu)

***Abstract*— Sentiment analysis is contextual mining of text which identifies and extracts subjective information in source material, and helping a business to understand the social sentiment of their brand, product or service while monitoring online conversations Different web journals and Social Media (Facebook, Twitter, Instagram) are the most prevalent stage for the consumers and users where most of the time they express their judgement about trending topics, different brands, restaurant, films, books and so on. Analyzing sentiment is an exceptionally brilliant and viable way to discover people views about news, place, restaurant, film, book, brand. It is helpful for both the owners and sellers. In this study, we built a model using natural language processing techniques and deep learning algorithms to automate the approach of classifying a review posted on TripAdvisor website. In this paper, we developed two effective deep learning approaches to build a model that can predict the sentiment by analyzing the customer’s review of a restaurant. One of our models achieved an accuracy of 92% besides other classification models. The result from this research shows that these two methods get the customer response accurately and LSTM method is more accurate than CNN sentiment analysis with a different accuracy of 3%.**

**Index Terms—Sentiment analysis, CNN, LSTM, Text classification, Deep learning, Word embedding**

1. INTRODUCTION

Sentiment analysis (or opinion mining) is a natural language processing (NLP) technique used to analyze the product sentiment in customer feedback and to understand customer needs. Sentiment analysis is commonly performed on text data which includes reviews, feedback, blogs and this determines whether the data is positive, negative or neutral. The aim of our project is to perform sentiment analysis of reviews of restaurants. Restaurant is a business which is mainly based on customer satisfaction. Repeating customers and attracting new customers are two important strategies most of the restaurants follow in the earlier strategy, Withholding the customer base is as important as running the business. Customers express their views and opinion on restaurants through different means. One such means is reviews, Reviews are given to the owner and to the future customer through websites, apps, feedback forms etc. We are considering review through website and more specifically given on trip advisor website which is the popular website in review writing. In this project we are performing sentiment analysis using two deep learning methods CNN and LSTM.

1. DATASET DESCRIPTION

We performed web scraping on Trip advisor website and obtained the dataset. The dataset is primarily based on the reviews written by different customers. Our dataset contains a total of 907 reviews. The attributes of the dataset are review and rating. The attribute revies contains the original text review given by the customer whereas the attribute rating consists of two values 0 and 1 where 0 indicates that the review is negative and 1 indicates the review is positive. When analyzed we conclude that the dataset contains 500 positive reviews and 407 negative reviews. Our raw dataset contains redundant and unwanted information which has no significance in the analysis thus we perform data cleaning and preprocessing.

1. PROJECT DESCRIPTION
   1. *Description*

The main aim of the project is to perform sentimental analysis on restaurant reviews and to analyze the mood of the public. This helps the restaurant owners to address the public's response promptly and in a timely manner. In order to perform sentimental analysis firstly we need to load the dataset such that the dataset contains only that information which is useful for instance, a restaurant review which says “The food is amazing and ambience is great” here we would be only interested in words like food, amazing, ambience, great the rest of the words which makes the sentence understandable in real life plays no significant role in training the model thus we discard them. To achieve better results from the deep learning models the format of the data must be in a proper manner. The process of discarding the information in the dataset which is not useful to us is known as Data cleaning and preprocessing. In this process we are performing three important and major steps they are

1. *Remove everything (special characters) except alphabets*

For removing everything except the alphabets regular expression is used and is defined such that lowercase alphabets, upper case alphabets are part of the pattern, and all the words have single spacing. This removes symbols and non-alphabet expressions.

1. *Convert the data to lowercase*

The dataset contains a mixture of upper case and lower case. All the data is converted to lowercase.

1. *Stemming and Removing stop words from data*

Stemming is the process of producing morphological variants of a root/base word. For example, stemming algorithm reduces the words “retrieval,” “retrieved,” “retrieves” to the root word “retrieve.” Porter Stemmer Package is used from NLTK Library for stemming data.

NLTK is a standard python library which contains symbolic and statisticial natural language processing of English. A stop word is a commonly used word (such as “the,” “a,” “an,” “in”) that are useless and are filtered in preprocessing. The stop words package is used from NLTK Library for removing stop words.

In the next step we perform splitting of the dataset in to test and train data, we have split the data into 80 percent of test data and 20 percent of train data. For this purpose, we have used train\_test\_split package from sklearn library. After the dataset is split the statistical information of the splitting process indicates that the test data contains 725 values and the train data contains 182 values. Once the data set is split, we perform vectorization through tokenization, Tokenization is the process in which the sentence/text is split into array of words called tokens. We used Tokenizer to vectorize the text and convert it into sequence of integers after restricting the tokenizer to use only topmost common 2500 words. We used pad\_sequences to convert the sequences into a 2D NumPy array. We performed tokenization on test and trained data set separately using keras. Keras is a open source python inter face used for artificial neural network and developing evaluating deep learning models.

After the Vectorization process, we implement LSTM, LSTM stands for long short term-memory is an artificial recurrent neural network used in the d=field of deep learning, unlike feed forward neural networks LSTM has a feedback network. LSTM has a special architecture which enables it to forget unnecessary information. This is achieved because the recurring module of the model has a combination of four layers. It is an ideal choice to model sequential data and hence used to learn complex dynamics of human activity. The long-term memory is called the cell state. Due to the recursive nature of the cells, previous information is stored within it. The forget gate placed below the cell state is used to modify the cell states. The forget gate outputs values saying which information to forget by multiplying 0 to a position in the matrix. If the output of the forget gate is 1, the information is kept in the cell. The input gates determine which information should enter the cell states. Finally, the output gate tells which information should be passed on to the next hidden state. The architecture of the LSTM layer consists of one embedding layer, three LSTM layers, and 2 dense layers.

**LSTM Layers**

Diagram

Description automatically generated

Here, we are using relu activation and sigmoid activation functions and as optimizer we are using RMS Prop, for loss criterion we are using binary entropy loss We are training this model with 100 epochs and with a call back. We are also saving accuracy, loss, precision, recall at each point in history. Using this architecture, we were able to attain an accuracy of 92 percent.

In the next step we are implementing CNN, CNN which stands for convolution neural network is a popular technique in image processing, there the input is an image mostly with two dimensions, but our data set contains text data, hence we are considering text to be a 1 dimension and using conv1d layer. A CNN, in general, can be thought of as an artificial neural network with some type of specialization for being able to pick out or detect patterns.

Convolutional layers, which are hidden layers in CNNs, CNN takes advantage of the so-called convolutional filters that automatically learn features suitable for the given task. For example, if we use the CNN for the sentiment classification, the convolutional filters may capture inherent syntactic and semantic features of sentimental expressions, it has been shown that a single convolutional layer, a combination of convolutional filters, might achieve comparable performance even without any special hyperparameter adjustment. Furthermore, CNN does not require expert knowledge about the linguistic structure of a target language. The architecture of the CNN consists of one embedding layer, two conv1d layers, one flatten layer and one dense layer.

**CNN Layers**

Diagram

Description automatically generated

We are using 1 embedding layer, 2 conv1D layers, flatten and dense layers. Relu activation and sigmoid activations are being used. For loss criterion we are using binary entropy loss. We are using model checkpoint to save and reuse the model later for higher validation accuracies. Using early stopping, we stop training when a monitored metric has stopped improving. We will train the model for 70 epochs with the callback. We stored the accuracy, loss, precision, recall at each epoch in the history. Using the architecture, we have attained an accuracy of 89 percent.

**Flow chart of our project**

Diagram

Description automatically generated

* 1. *Main references used for our project*

1. https://ieeexplore.ieee.org/document/8970976
2. https://ieeexplore.ieee.org/document/8850982
3. https://ieeexplore.ieee.org/document/9300098
4. https://towardsdatascience.com/understanding-lstm-and-its-quick-implementation-in-keras-for-sentiment-analysis-af410fd85b47
5. https://medium.com/@dtuk81/confusion-matrix-visualization-fc31e3f30fea
6. https://medium.com/@BishnoiAmit/convolution-nets-for-sentiment-analysis-dfda50768dfc
   1. *Difference in APPROACH/METHOD between our project and the main projects of your references*

We used two deep learning techniques LSTM and CNN.

1. *Difference in approach for LSTM*

LSTM stands for long short-term memory. It is an artificial recurrent neural network architecture. LSTM has a special architecture which enables it to forget unnecessary information. This is achieved because the recurring module of the model has a combination of four layers.

According to reference 1 Embedded Layer, 1 Lstm layer, 1 Dense Layer are used. Softmax activation is used in dense layer. Categorial cross entropy is used for loss criterion. Adam optimizer is used. 1 Epoch is used.

In our project we used an Embedded Layer, 3 Lstm layers and 2 Dense layers. We used two activation layers relu and sigmoid as dense layers. For loss criterion we used binary cross entropy. Rmsprop is used as an optimizer. We trained model on 100 Epochs.

1. *Difference in approach for CNN*

CNN stands for convolution neural network. In general, it can be thought defined as an artificial neural network with some type of specialization for being able to pick out or detect patterns. Convolutional layers, which are hidden layers in CNNs, are what make a CNN.

In reference an Embedded Layer, ConvId layer, Maxpooling Layer, Dense Layer are used. Softmax activation is used in dense layer. Categorial cross entropy is used for loss criterion. Adam optimizer is used. 50 Epochs are used.

In our project we used an Embedded Layer, 3 Lstm layers and 2 Dense layers. We used two activation layers relu and sigmoid as dense layers. For loss criterion we used binary cross entropy. Rmsprop is used as an optimizer. We trained model on 70 Epochs.

For both models we used model checkpoint to save and reuse the model later for higher validation accuracies.

We also used the Early Stopping method to stop training when a monitored metric has stopped improving.

* 1. *Difference in ACCURACY/PERFORMANCE between our project and the main projects of your references*

1. *Difference in performance of LSTM model*

We achieved 6% more accurate results compared to the reference, as our accuracy is 92% whereas reference accuracy is 86%

**Confusion Matrix for our LSTM model**

Chart

Description automatically generated with medium confidence

1. *Difference in performance of CNN model*

We achieved 2% better accuracy compared to the reference, as our accuracy is 89% whereas reference accuracy is 87%

**Confusion Matrix for our CNN model**

A picture containing chart

Description automatically generated

1. ANALYSIS
   1. *What did we do well?*

We redesigned the model architecture of LSTM and CNN for better performance. We made various trials changing the architecture to achieve more accurate results

* 1. *What could we have done better?*

Time complexity for LSTM model can be reduced as it takes more time to run compared to CNN.

Trials can be made to optimize CNN so that it can give more accurate results than LSTM.

* 1. *What is left for future work?*

Trials can be made to achieve better accuracy using Bidirectional LSTM model on the dataset.

1. CONCLUSION

To conclude, the aim of the project is satisfied with optimal accuracies. The models built can predict the sentiment with comparatively high accuracies. LSTM model gives accuracy of 92% and CNN model gives the accuracy of 89%. We can conclude that depending on the conditions local to our problem LSTM can predict with better accuracy than CNN. However, CNN lacks with 3% less accuracy its runtime is exceptionally low.