Washington University in St. Louis

OLIN BUSINESS SCHOOL

Deep Learning for Prediction of Business Outcomes

Final Project

Deep learning tool and Disneyland review rating analysis

Section Group 29

Group members:

Zixiao Wu

Shuangjia Li

Rentian Liu

May 10, 2023

presentation recording:

https://wustl.zoom.us/rec/share/l3zeu5p2ADhejtV_evLcDZLbGxe5LdCDmuqbqdh5kYtOzxuVYyyrMOnXBfG04rLw.-CXzGj69bLS3rn5W

Contents

Abstract

Predicting the numerical rating based on the text-format review is our project target. We realized the target using deep learning models and sentiment analysis methods. In the testing process, we choose mean absolute percentage error (mape) as the standard measurement.

We set up 3 models, a LSTM model, a CNN model and finally the rating prediction model. Then we optimized the model and tested it with the test model.

1 Introduction

Disneyland, the paradise of the kids, and one of the most popular and famous amusement park brands, has 6 branches all around the world. In Asia, they have Hongkong, Shanghai and Tokyo branches, in Europe, they have Paris branch, and in North America, they have Orlando and California branches. The parks serve millions of customers everyday and there are a lot of reviews, suggestions and ratings after trips.

Those reviews are valuable to Disneyland as they can improve their quality of service by analyzing them and then increase the customer satisfaction and loyalty.

The goal of our project is to build proper deep learning models and use sentiment analysis methods to help Dineyland make better predictions on the ratings based on the customers' reviews.

2 Literature review

2.1 Sentiment analysis in food delivery service

Prior research has shown that sentiment analysis of customer reviews is valid to be used to improve customer satisfaction and loyalty. According to Adak (2022), food delivery has been using this technique and has gained success. Compared to the traditional machine learning tools, deep learning models have a better output facing the customer review sentiment analysis quests. However, knowing that deep learning is a black box process in reality, the deep learning models still need support from other techniques. But, this article has proved that it is valid and recommended to use deep learning models and sentiment analysis tool in our project.

2.2 Deep learning for sentiment analysis on Google Play consumer review

Google Play, one of the pioneers of deep learning technology, has been using sentiment analysis for them to improve customer experience as well. (Day, M. Y., & Lin, Y. D. 2017) The usage of sentiment analysis has an accuracy of 89.42%. It proved that our choice of project method is proper.

3 Problem description

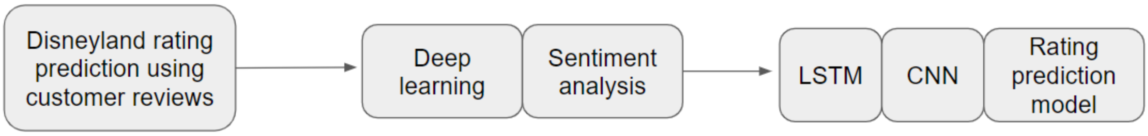

Figure 1 Problem framework

The target of our project is to make a prediction of Disneyland rating using the customers text reviews so that Disneyland can locate the area or process of their service that require improvement, and increase customer satisfaction. We chose the following deep learning models : LSTM model and CNN model. Then we made a rating prediction model to finish our task. A testing model is set in the end to test the accuracy.

4 Database introduction and data preprocessing

We get the data from Kaggle. The "Disneyland Reviews" dataset has been chosen as our data for the model, which contains more than 40,000 reviews of the Disneyland Park in California, USA. The reviews were scraped from TripAdvisor and cover a period of several years. Each review includes information about the reviewer's rating, title, text, date, and language. The variables measured include the rating (on a scale of 1-5) and the text of the review.

The preprocessing mainly contains two parts: normalize and tokenize. In the normalize part, we set a function to normalize the data, the function contains several steps.

```
def normalize_text(text):
    text = normalize_accented_characters(text)
    text = html.unescape(text)
    text = strip_html(text)
    text = expand_contractions(text, contraction_mapping)
    text = tokenize_text(text)
    text = lemmatize_text(text)
    text = remove_special_characters(text)
    text = remove_stopwords(text)
    text = keep_text_characters(text)
    return text
```

Figure 2 normalize function

In the tokenize part, the tokenization tool was imported so that the data could be used by the model.

```
from keras.preprocessing.text import Tokenizer
from keras.utils import pad_sequences
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWar
  and should_run_async(code)
```

```
#initialize parameters
max_words = 11451
maxlen = 114
#batch_size = 32

tokenizer = Tokenizer(num_words=max_words)
tokenizer.fit_on_texts(x_train)
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWar
  and should_run_async(code)
```

```
#tokenize
X_train=tokenizer.texts_to_sequences(x_train)
X_test=tokenizer.texts_to_sequences(x_test)

print(len(X_train))
print(len(X_test))
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWar
  and should_run_async(code)
29859
12797
```

Figure 3 Tokenize function

5 Supervised Learning: A Satisfaction Rating Prediction Model

5.1 LSTM

LSTM networks are a type of recurrent neural network (RNN) that are designed to model temporal dependencies in sequential data. They are particularly effective at capturing long-term dependencies and handling vanishing gradient problems that arise in traditional

RNNs. Bidirectional LSTM networks combine the forward and backward processing of an input sequence, allowing the network to leverage both past and future context of each word.

The LSTM layer itself consists of a memory cell and several gates that regulate the flow of information through the cell. The memory cell is responsible for storing and updating the information over time, while the input gate, output gate, and forget gate control the flow of information into and out of the cell.

We totally establish 2 LSTM layers in the model. Firstly, a bidirectional LSTM layer is added to the model with 128 memory units, 20% dropout for input connections, 20% dropout for recurrent connections. Then the next LSTM layer has 64 memory units.

5.2 CNN

CNNs are commonly used for image recognition tasks, but can also be used for text classification tasks. They are effective at extracting local features from input sequences, which can then be used by subsequent layers to learn higher-level features. The max pooling operation reduces the output dimensionality and provides translation invariance to small shifts in the input sequence.

Our data set is a non-image data set. Therefore, we built a Conv1D model instead of the traditional Conv2D model. We use a sequential model with 1 Conv1D layers and 1 MaxPooling1D Layer.

The original data set has been split into training and testing sets with 29859 and 12797 observations. After several times of operations, we apply 128 filters, which determines the number of output channels produced by the layer, and the size of the filters is 5, which determines the number of adjacent time steps considered by each filter. We then choose the

ReLU activation function, because this activation function introduces non-linearity and can help the network learn more complex representations of the input data.

5.3 The Rating Prediction Model

We built the Rating Prediction Model that combines convolutional neural networks (CNN) and bidirectional long short-term memory (LSTM) networks for text classification tasks. The model is composed of an embedding layer, a one-dimensional CNN layer, a bidirectional LSTM layer with recurrent dropout and another bidirectional LSTM layer, and a fully connected layer.

The embedding layer maps each word in the input sequence to a dense vector representation. The CNN layer applies a set of filters to extract local features from the input sequence. The max pooling layer reduces the dimensionality of the output feature maps. The bidirectional LSTM layers leverage the temporal dependencies of the input sequence by processing the sequence in both forward and backward directions, which allows the model to capture both past and future context of each word. The recurrent dropout helps to prevent overfitting by randomly dropping out connections between LSTM units during training.

```
model = Sequential()
model.add(Embedding(max_words, 128, input_length=maxlen))
model.add(Conv1D(128, 5, activation='relu'))
model.add(MaxPooling1D(pool_size=4))
model.add(layers.Bidirectional(layers.LSTM(128,dropout=0.2,recurrent_dropout=0.2, return_sequences=True)))
model.add(layers.Bidirectional(layers.LSTM(64)))
model.add(layers.Dense(1))
```

Figure 4 CNN and LSTM Model

Overall, we build a model to leverage the strengths of both CNNs and bidirectional LSTMs to effectively capture both local and long-term dependencies in text data and achieve state-of-the-art performance in text classification tasks.

5.4 Results

We use Adam optimizer and set the epochs to 30 and batch size to 48. The results of this model are dissatisfied. We get the mape as 8.8572 and val_mape as 17.4140. Therefore, we try to optimize the model and fit another one with larger max_len and batch size.

5.5 Model Optimization

In the deep learning model, the selection of the maxlen hyperparameter should be based on the characteristics of the data set and the nature of the problem. Factors such as the length of the input sequences, available computational resources, and the potential presence of outliers should be taken into consideration.

In terms of optimizing the model, we mainly adjusted the hyperparameters of the model, increasing the maximum text length, and increasing the number of training epochs. The initial maxlen is set to 114, so all text sequences will be truncated to a length of 114 words. We then change max_len to 200, epochs to 35. These adjustments can improve the accuracy of the model predictions and reduce the occurrence of overfitting. Additionally, batch size and optimizer type were also adjusted to improve training efficiency and performance. We set the batch size to 72 and find that Adam is the best optimizer for this question.

```
max_words = 11451
maxlen = 200
#batch_size = 72

tokenizer = Tokenizer(num_words=max_words)
tokenizer.fit_on_texts(x_train)
```

Figure 5 Optimization for maxlen

```
model.compile(optimizer='adam', loss='mae', metrics='mape')
model2 = model.fit(X_train, y_train, epochs=35, batch_size=72, validation_data=(X_test,y_test))
```

Figure 6 Optimization for epochs

The model was optimized by implementing these adjustments, as evidenced by the improved results obtained from training and validation.

```
Epoch 34/35
415/415 [==============================] - 446s 1s/step - loss: 0.1446 - mape: 4.8925 - val_loss: 0.4810 - val_mape: 16.9206
Epoch 35/35
415/415 [==============================] - 446s 1s/step - loss: 0.1423 - mape: 4.7644 - val_loss: 0.4916 - val_mape: 16.6817
```

Figure 7 Results

We obtain a value of 4.89 for mape and 16 for val_mape. We get a better answer, but we find that there is an overfitting problem in our new model.

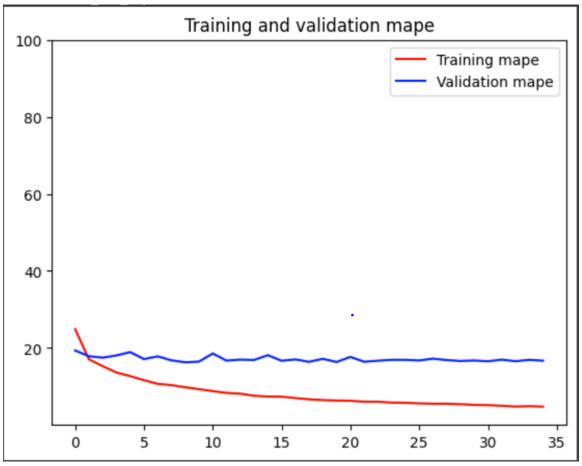

Figure 8 Plot of Mape

6 Conclusion, Discussion and Future work

According to our test result, the outputs of our model are:

Accuracy: 0.589

Precision: 0.580

Recall: 0.589

F1-score: 0.584

This means that with our model, Disneyland has a high probability of predicting the rating using customer reviews. We have mainly achieved our goal.

Future work: We will focus on adding more models to our process, and try to solve overfitting problems and increase our model accuracy. And, the uniqueness of our model is that its idea can fit the whole amusement park dataset well. The range is not limited to California Disneyland itself; it can also be extended to the Disneyland located in Shanghai, in Hong Kong, in Tokyo, in Paris, and even other amusement parks that have a similar review scale. By accurately predicting the rating of a review using deep learning, they can quickly identify areas that need improvement and prioritize their efforts accordingly. Additionally, the insights obtained from this analysis can be used to develop marketing and advertising strategies that are tailored to the needs and preferences of their customers, further increasing profitability and brand loyalty. Our model can be tried on similar datasets from other amusement parks in the future.

Reference

Adak, A., Pradhan, B., & Shukla, N. (2022). Sentiment analysis of customer reviews of food delivery services using deep learning and explainable artificial intelligence: Systematic review. Foods, 11(10), 1500.

https://www.mdpi.com/2304-8158/11/10/1500

Day, M. Y., & Lin, Y. D. (2017, August). Deep learning for sentiment analysis on google play consumer review. In 2017 IEEE international conference on information reuse and integration (IRI) (pp. 382-388). IEEE.

https://ieeexplore.ieee.org/abstract/document/9115602/

Disneyland Reviews Dataset

https://www.kaggle.com/datasets/arushchillar/disneyland-reviews