|  |
| --- |
| Western Governors University |
| Advanced Data Analytics |
| D213 TASK 2: SENTIMENT ANALYSIS USING NEURAL NETWORKS |

|  |
| --- |
| Allison Casey  1-25-2025 |

**Part I: Research Question**

**A1: RESEARCH QUESTION**

Can consumer sentiment be predicted by using previous reviews from other users?

**A2: OBJECTIVES OR GOALS**

The goal of this analysis is to attempt to predict the sentiment of a user’s review based off their word choice.

**A3: PRESCRIBED NETWORK**  
A Recurrent Neural Network (RNN) takes information from previous inputs to influence the current input and output. This is useful for this scenario because this allows for sequential discrepancy in words and phrases to be taken into consideration.

**Part II: Data Preparation**

**B1: DATA EXPLORATION**

1. A regex pattern was used to detect special characters by noting all characters that did not match a-z, A-Z, 0-9, or space. The special characters found were:



1. The vocab size was determined to be 3315 unique words.
2. The proposed embedding length was determined by taking the fourth root of the vocab size which came out to 8.
3. The max sequence length was determined to be 32 as that was the longest word length of an individual review in the data frame so that no reviews are truncated, and the input data is preserved.

See D213\_Task2.ipynb for related code

**B2: TOKENIZATION**

The goal of the tokenization process is to separate the reviews into smaller units. This allows an index to be assigned to each unit which helps with training the model. Other goals include transforming the text into sequences and then padding these sequences to the maximum length. The packages used were “from tensorflow.keras.preprocessing.text import Tokenizer” and “from keras.preprocessing.sequence import pad\_sequences”

A screen shot of a computer program

Description automatically generated

See D213\_Task2.ipynb for full code.

**B3: PADDING PROCESS**

The padding process adds zeros onto the end of each sequence so that all of the sequences contain the maximum number of words which was determined to be 32.

A screen shot of a computer

Description automatically generated

**B4: CATEGORIES OF SENTIMENT**

Two categories of sentiment will be used for the analysis related to the score where 1 is positive and 2 is negative. The activation function to use will be sigmoid since the output is binary.

**B5: STEPS TO PREPARE THE DATA**

1. Data was read in from the Yelp and Amazon data sets and concatenated.
2. Data was checked for unusual or any abnormal characters and removed them in the process.
3. Data was checked and treated for duplicates and missing values.
4. The vocab size, maximum sequence embedding, and maximum sequence length were calculated to explore the data
5. Created and fit the tokenizer on the review data
6. The sequences were padded to match the maximum sequence length
7. Created a two-dimensional NumPy array with our ratings by encoding the Sentiment column of the data frame.
8. Split data into training and testing sets with an 80/20 split.
9. Exported the prepared training and testing data to csv

See D213\_Task2.ipynb for full code.

**B6: PREPARED DATA SET**

X\_train.csv

y\_train.csv

X\_test.csv

y\_test.csv

**Part III: Network Architecture**

**C1: MODEL SUMMARY**

A screenshot of a computer program

Description automatically generated

**C2: NETWORK ARCHITECTURE**

The model has five layers with a total of 27057 parameters:

1. Embedding layer with 26528 parameters
2. Global average pooling layer with 0 parameters
3. Dense layer with 216 parameters
4. Dense layer with 300 parameters
5. Dense layer with 13 parameters

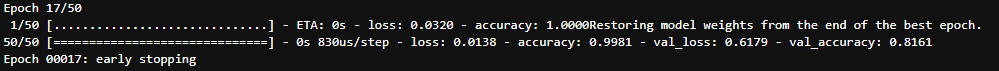
**C3: HYPERPARAMETERS**

* Activation functions
  + ReLu was used because it is simple and efficient for hidden layers
  + Sigmoid was used for the output layer because it is good for binary classification
* Number of nodes per layer
  + This was determined by starting with 50 at the highest and then increasing until there didn’t seem to be much improvement in the model accuracy with adding more layers. There wasn’t much improvement past 50 but decreased below 50 so this was kept as the number of nodes for the initial dense layer. The next layer halved this, and the output layer was set to 1 to be the right shape for the output.
* Loss function
  + Binary crossentropy was used because it is good for classification analysis
* Optimizer
  + The “adam” optimizer was used because it is easy and straightforward to implement and is commonly used (Elleh, 2025).
* Stopping criteria
  + An early stopping monitor was created to stop the model if it was no longer improving after two iterations based on accuracy
* Evaluation metric
  + The evaluation metric is accuracy to simply evaluate how well the model can classify comments based on sentiment

**Part IV: Model Evaluation**

**D1: STOPPING CRITERIA**

The number of epochs used was 50 which was paired with an early stopping monitor so that the model had plenty of epochs to improve beyond the standard 10 but would also be stopped as soon as there was no more indication of improvement. In this case the model was stopped at 17 epochs.



**D2: FITNESS**

The model scores well for accuracy with an accuracy of approximately .82 which indicates that it isn’t too overfit to the training data. To avoid overfitting the model utilized an early stopping monitor and as the model was being built it was started first with a smaller network and then increasing rather than immediately using a large network.

A computer screen shot of a program

Description automatically generated

**D3: TRAINING PROCESS**

A graph of a number of data

Description automatically generated with medium confidence

**D4: PREDICTIVE ACCURACY**

The evaluation metric chosen was accuracy which was determined to be approximately .82 or 82%. This is a high enough accuracy that it could potentially be useful though there may be room for improvement.

**Part V: Summary and Recommendations**

**E: CODE**

D213\_Task2.ipynb

**F: FUNCTIONALITY**

The network architecture was built in mind with trying to create a model well suited for binary classification which included incorporating things such as the sigmoid activation function for the output layer and the binary crossentropy loss function. The initial embedding layer sets the parameters for what the model can expect to process which includes the maximum size of the vocab, sequence embedding, and sequence length. The next layer is GlobalAveragePooling1D which is used to reduce the dimension of the sequences which helps simplify things for the next layers without losing essential information. The next two layers incorporate a dense layer with ReLu activation which allows the model to build non-linear, complex relationships as it narrows down the data the output layer to eventually reach a classification. The final layer is the output layer which is a dense layer with a sigmoid activation function which was ideal because it simplifies the probabilities into a range of 0 to 1 to predict which option the sequence will be classified into. This resulted in a model that can accurately classify the sentiment of reviews 82% of the time so the model is not perfect, but it is functional. This indicates that consumer sentiment can be predicted by using previous reviews from other users.

**G: RECOMMENDATIONS**

Based on these results there are a couple of advisable actions. First it would be good to continue to gather data and increase the amount of data for training and testing to improve the model. It could also be good to further improve the model by further exploring and improving the hyperparameters. In terms of using the model itself it could be helpful in better understanding customers and what is and isn’t working about a product and getting this classified information to the correct parties to help improve the overall product or service.

**Part VI: Reporting**

**H: REPORTING**

D213\_Task2.html

**Sources**

Brownlee, Jason. “How to Use Word Embedding Layers for Deep Learning with Keras.” *MachineLearningMastery.Com*, 1 Feb. 2021, machinelearningmastery.com/use-word-embedding-layers-deep-learning-keras/. Accessed 25 Jan. 2025.

Elleh, Festus. “Advanced Data Analytics - Task 1.” D213 Task 2 Cohort Webinar. 2025

GeeksforGeeks. “Natural Language Processing (NLP) Tutorial.” *GeeksforGeeks*, 17 Dec. 2024, www.geeksforgeeks.org/natural-language-processing-nlp-tutorial/. Accessed 25 Jan. 2025.

“Tf.Keras.Model  :  Tensorflow V2.16.1.” *TensorFlow*, www.tensorflow.org/api\_docs/python/tf/keras/Model. Accessed 25 Jan. 2025.