SENTIMENT ANALYSIS USING NEURAL NETWORKS

**ADVANCED DATA ANALYTICS — D213**

**PRFA — NLM2**

Jason Smith

Student ID: 293963

jsmi268@wgu.edu

WGU – MSDA

Program Mentor: Christiana Okhipo

Instructor: William Sewell

**Part I: Research Question**

**A1. Research question**

Can a user's opinion, of either positive or negative, be predicted based on preivious user's reviews?

**A2. Goal**

The objective of this analysis is to perform sentiment analysis based on customer reviews to determine if patterns exist.. If so, these patterns can be leveraged in future business decisions.

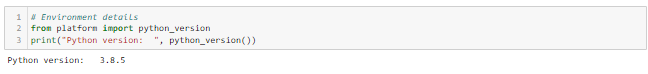
**A3. Neural Network Type**

TensorFlow combined with Keras will be used to perform text classification using a sequential model. TensorFlow is a deep learning framework (Brownlee, 2019). The dataset will be broken into two parts. The first part will be used for training the model and second part will be used to test model accuracy.

**Part II: Data Preparation: EDA and Data Cleaning**

**B1. Data exploration**

The IDE used is Jupyter notebooks with python



Libraries are imported and personal preferences are set.

Text

Description automatically generated

The Amazon customer tools review data is in .json.gz format. This data is imported into a data frame.

A picture containing text

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Identify the number of rows and columns using the shape function.



Remove the unnecessary columns and re-run the shape function.

A picture containing graphical user interface

Description automatically generated

Perform basic statistics on the numeric columns using the describe function.

Graphical user interface, application

Description automatically generated

Rename columns.



Change ratings data type to an integer.



Drop any null values.



Convert any non-string values to strings.



The customer ratings are based on 5 stars which will be convert to either a positive review with value of 1 or a negative review with a value of 0.

Graphical user interface, text, application, email

Description automatically generated

Text, application

Description automatically generated

Check the number of rows and columns using the shape function.



*Check and correct for:*

*• presence of unusual characters (e.g., emojis, non-English characters, etc.)*

*• vocabulary size*

*• proposed word embedding length*

*• statistical justification for the chosen maximum sequence length*

Graphical user interface, text, application, email

Description automatically generated

Text, Word

Description automatically generated

Precursor cleaning of the review data is necessary to achieve accurate NLP results. The existence of non-English characters and emojis were not detected in the review string data. Additional data cleaning steps were performed including the transforming the review text to lower case, the removal of numbers and punctuation.

Graphical user interface, text, application, email

Description automatically generated

**Removal of stop words**

**A picture containing text

Description automatically generated**

Stop words are words that help add clarity and understandability to humans. Stop words are not meaningful to NLP algorithms and should be remove prior to performing the analysis. Stop words were removed using NLTK.

Graphical user interface, text, application, email

Description automatically generated

**Vocabulary size**

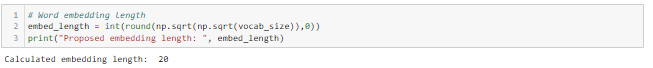
**Graphical user interface, text, application, email

Description automatically generated**

The vocabulary size metric indicates the number of unique words in the contained data. The review data has 174,558 words.

**Proposed Embedding length**

Word embedding is the"learned representaion of text where words that have the same meaning have a similar represntationis a process of rating how similar individual words are to one another"(Brownlee, 2017). Embedding length is determined by taking the 4th root of the vocabulary size.



**Statistical justification for the chosen maximum sequence length**

Graphical user interface, text, application, email

Description automatically generated

Neural networks require to have inputs that have the same size and shape (Caner, 2020). Sequence length, a related term, is the length of the longest input sentence and dictates the amount of padding that needs to be applied to the other inputs. The length is measured in the number of words. The longest review sequence contains 214 words. The remaining sequences must be brought to the same length through padding with zeros.

**Split into train and test**

The cleaned data will now be split into 80% training and 20% test datasets using the train\_test\_split function imported from sklearn.model\_selection.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

### B2. Goals of the tokenization process

Tokenization is the process of breaking larger things into smaller things and is one important step in natural language processing. The amazon review data is being broken into word tokens. Other tokens include characters or sub words such as "smart-er"(Pai,2020). Other goals of tokenization include(Elleh, 2022):

1. Replacing abnormal characters, formatting, and standardizing tests.

2. Lemmatizing words and transforming the text into sequence

3. Preparing the transformed sequences to a maximum sequence length by padding.

The packages used for preprocessing and tokenization include:

from nltk.corpus import stopwords # for stopword usage

from nltk import word\_tokenize

nltk.download ('stopwords')

from keras import preprocessing

from keras.models import Sequential

from tensorflow.keras.preprocessing.text import Tokenizer

from keras.layers.embeddings import Embedding

Word meanings that are unknown to Keras defined as "Out of Vocab", also known as OOV. OOV tokens can be assigned to unknown words which you can see in the following code block.

Text

Description automatically generated



**B3. Padding process**

Each word has been assigned a unique number in the word index. The cleaned and processed review text has been transformed into sequences for both train and test using the Keras tokenizer.texts\_to\_sequences. The words in the sequences have been replaced by the word index. This keeps the words in their intended order to maintain consistency and meaning. Observe below, the words have been replaced by numbers (Authors, 2020).

Text, letter

Description automatically generated

**Padding the pre-processed lines**

All the sequences need to be the same length in order for the NLP models to work accurately. This either achieved through truncation or padding the shorter sequences so they are all of the same length. The thensorflow.keras.preprocessing.sequence was used to perform padding and post padding with zeros was the chosen method to maintain meaning (Authors, 2020). The Tensorflow documentation recommends using "post padding when working with RNN layers in order to be able to use CuDNN implementation of the layers. Examples of padded text for train and test follow.

A picture containing application

Description automatically generated

**B4. TensorFlow categories and activation function**

Graphical user interface, text, application, email

Description automatically generated

**B5. Data preparation steps**

**Graphical user interface, text, application, Word, timeline

Description automatically generated**

**Graphical user interface, text, application, email

Description automatically generated**

**Graphical user interface, text, application, email

Description automatically generated**

**Text

Description automatically generated**

**A picture containing calendar

Description automatically generated**

**B6. Save the prepared dataset to CSV**

**Shape, rectangle

Description automatically generated with medium confidence**

**Part III: Network Architecture**

**C1. Keras model summary**

Graphical user interface, text

Description automatically generated

**C2. Keras model discussion:**

There are a total of 5 layers.

The first layer is an embedding layer which is an input type. This layer helps with dimension reduction and can relate similar word contexts. The longest sentence length 16038 words. the model used twenty embedding dimensions and 3,489,180 parameters.

The second layer is the Global Average Pooling 1D layer. This layer is type "flatten" and flattens out the vectors from the previous layer. No parameters are used here.

The next three layers are dense layers and are also referred as 'Fully connected' layers. These layers describes the neuron connections from one layer to the next. Dense\_4 has 100 neurons and contains 2,100 parameters and is of type "hidden". Dense\_5 has 50 neurons and is of type "hidden". Dense\_6 has 2 neurons and contains 102 parameters and is of type "output".

**C3. Justification of hyperparameters**

**Activation functions:**

The SoftMax function was used to activate the final dense layer and performs a multi-class logistic regression. It was used to convert vector values to probability distributions. The output vector values are in a range from 0 to 1.

The Rectified Linear Unit (ReLu) activation function was used for the dense layers and is common to use when 2 or more sentiments are being evaluated.

**Number of nodes**

One hundred nodes were used for dense\_4 layer. Fifty nodes were used for the dense\_5 layer. Two nodes were used for the final dense layer.

**Loss function**

The Cross\_Entropy loss function was used to measure the model's performance and was used because in compares the similarity of words, and then transforms the transforms the loss into a numeric values. The higher the value the higher the loss.

**Optimizer**

Adaptive Movement Estimation, Adam, is the chosen optimizer. It uses squared gradients to scal the learning rate. It is capable of creating individual learning rates for different parameters (I2 tutorials, 2019).

**Stopping criteria**

Early stopping with a patience level of 2 was used to determine a sufficient epoch setting, or number of training runs. Too much training leads to over fitting and tool little training leads to underfitting.

**Epochs**

An epoch defines the number of training iterations. Typically, epochs is set to many hundreds of times. I chose a maximum epoch value of 20 for this analysis.

**Evaluation metric**

The Keras accuracy metric was used to measure model accuracy. This metric calculates the percentage of predicted sentiment values that match actual sentiment values(Dommaraju, 2020). Model accuracy for test is 92% with a loss of 28%.

Graphical user interface, text, application, email

Description automatically generated

**Part IV: Model Identification and Analysis**

**Table

Description automatically generated with medium confidence**

**D1. Impact of using stopping criteria**

Choosing the number of epochs is not an exact science. Choosing too many epochs can lead to overfitting and choosing too few can lead to an underfitted model(Brownlee, 2018). Conversely, Early Stopping will stop using epochs once performance stops improving. Early stopping can take the guessing out of designating epochs and lead to better model fit.

For this model, I used Early Stopping with an initial epoch of 20 and with a patience set to 2.

**D2. Visualizations**

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Chart, treemap chart, box and whisker chart

Description automatically generated

Graphical user interface, application, Word

Description automatically generated

**D3. Model fitness**

The model accuracy for my analysis is 92% which means there likely was no overfitting or underfitting.

Overfitting occurs when a model performs well with the training data but performs poorly with the testing dataset.

Stopping criteria was used to determine an appropriate number of epochs.

Early stopping was used to determine a lesser number of epochs which reduces the chance of overfitting.

In addition, reducing the number of layer and increasing the number of product reviews in training potentially could reduce overfitting as well.

**D4. Predictive Accuracy of the trained network.**

The model accuracy is 92% and model loss is 28% for the trained network. This shows the model has a good ability to predict sentiment based on review data. The code for the accuracy test can be found in section C3.

**Part V: Summary and Recommendations**

**E. Code for the trained network**

The code for the trained network is contained in attached file: 'SentimentAnalysisModel.h5'



### **F. Functionality and Impact**

The following table provides the number of reviews in the different datasets.

Number of reviews: Original dataset: 134,476 Training: 98965 Test: 24742

This study focused on Amazon customer review and sentiment data. Sentiment data was categorized as 0 for negative sentiment and 1 for positive sentiment. NLP was performed on the training dataset with the goal to predict negative or positive sentiment when new review data is applied to the model. This analysis is using a Sequential-NLP model consisting of 5 layers and was used to build a text classification model. The model displayed 92% accuracy which shows a strong predictive ability.

**G. Recommendations.**

Based on a high model accuracy, this model can be applied to new data and should yield a strong predictive ability of sentiment related to reviews of tools. Stakeholders at Amazon can actively apply this model to new tool review data and be confident it will correctly predict sentiment a majority of time. Further testing is required to determine whether this model can apply to other types of reviews such as reviews of movies or dresses.

**Part VI: Reporting**

**H. Source code**

This analysis has been performed using a Jupyter notebook. An html file of the code and output has also been uploaded as well.

**I. Third-Party Code Sources**

1. Coding Discuss(Mar, 2021). Detect strings with non English characters in Python

https://discuss.dizzycoding.com/detect-strings-with-non-english-characters-in-python/

(Discuss, 2021)

2. Palah Sharma(Jan, 2021). Keras Tokenizer Tutorial with Examples for Beginners

https://machinelearningknowledge.ai/keras-tokenizer-tutorial-with-examples-for-fit\_on\_texts-texts\_to\_sequences-texts\_to\_matrix-sequences\_to\_matrix/

(Sharma, 2021)

3. Detro(Jan, 2021). Remove Stop Words from Text in DataFrame Column.

https://www.datasnips.com/58/remove-stop-words-from-text-in-dataframe-column/

(Detro, 2021)

4. stackoverflow.com.IOPub data rate exceeded in Jupyter notebook.

https://stackoverflow.com/questions/43288550/iopub-data-rate-exceeded-in-jupyter-notebook-when-viewing-image

5. Hamish(2018). Using Keras OOV Tokens.

https://www.kaggle.com/code/hamishdickson/using-keras-oov-tokens/notebook

(Hamish, 2018)

**J. In-Line Sources**

1. Jason Brownlee (Dec, 2019). TensorFlow 2 Tutorial: Get Started in Deep Learning With tf.keras

https://machinelearningmastery.com/tensorflow-tutorial-deep-learning-with-tf-keras/#:~:text=TensorFlow%20is%20the%20premier%20open,Keras%20to%20the%20TensorFlow%20project.

2. Ali Hamza (Jan, 2019). Effectively Pre-processing the Text Data Part 1: Text Cleaning

https://towardsdatascience.com/effectively-pre-processing-the-text-data-part-1-text-cleaning-9ecae119cb3e

3. Kristin H. Huseby(June, 2020). Word embeddings, what are they really?

https://towardsdatascience.com/word-embeddings-what-are-they-really-f106e1ff0874#:~:text=With%20word%20embeddings%20we%20assign,with%20the%20most%20useful%20results.

(Huseby, 2020)

4. Jeffrey Pennington, Richard Socher, Christopher D. Manning. Computer Science Department, Stanford University.

GloVe: Global Vectors for Word Representation.

https://nlp.stanford.edu/pubs/glove.pdf

(Pennington, Socher, Manning)

5. Jason Brownlee(Oct, 2017). How to Use Word Embedding Layers for Deep Learning with Keras.

https://machinelearningmastery.com/use-word-embedding-layers-deep-learning-keras/

6. Caner(Apr, 2020). Padding for NLP

https://medium.com/@canerkilinc/padding-for-nlp-7dd8598c916a

(Caner, 2020)

7. Aravindpai Pai (May, 2020). What is Tokenization in NLP? Here’s All You Need To Know

https://www.analyticsvidhya.com/blog/2020/05/what-is-tokenization-nlp/

(Pai,2020)

8. Festus Elleh(2022). Cohort Recorded Event.

https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=cedbd86a-2543-4d9d-9b0e-aec4011a606d

(Elleh, 2022)

9. TensorFlow Authors(2020). Preparing text to use with TensorFlow models

https://colab.research.google.com/github/tensorflow/examples/blob/master/courses/udacity\_intro\_to\_tensorflow\_for\_deep\_learning/l09c02\_nlp\_padding.ipynb#scrollTo=RX9Yx50TUies

(Authors, 2020)

10. Dwarampudi Mahidhar Reddy @ N V Subba Reddy(Mar, 2019). EFFECTS OF PADDING ON LSTMS AND CNNS

https://arxiv.org/pdf/1903.07288.pdf

(Reddy, 2019)

11. Tensorflow Core Guide. Masking and padding with Keras

https://www.tensorflow.org/guide/keras/masking\_and\_padding

(Masking and padding with Keras)

12. Keras. Softmax layer

https://keras.io/api/layers/activation\_layers/softmax/

(Keras, Softmax layer)

13. I2 tutorials (Sep, 2019). Home / Deep Learning Interview questions and answers / Explain about Adam Optimization Function?

https://www.i2tutorials.com/explain-about-adam-optimization-function/

(I2 tutorials, 2019)

14. Goutham Dommaraju, (May, 2020). Keras’ Accuracy Metrics

https://towardsdatascience.com/keras-accuracy-metrics-8572eb479ec7#:~:text=If%20%281%29%20and%20%282%29%20concur%2C%20attribute%20the%20logical,to%20the%20actual%20value%2C%20it%20is%20considered%20accurate.

15. Jason Brownlee (Dec, 2018). Use Early Stopping to Halt the Training of Neural Networks At the Right Time

https://machinelearningmastery.com/how-to-stop-training-deep-neural-networks-at-the-right-time-using-early-stopping/

(Brownlee, 2018)

6. TensorFlow Authors(2020). Preparing text to use with TensorFlow models

https://colab.research.google.com/github/tensorflow/examples/blob/master/courses/udacity\_intro\_to\_tensorflow\_for\_deep\_learning/l09c02\_nlp\_padding.ipynb#scrollTo=RX9Yx50TUies

(Authors, 2020)

7. Keras documentaion. Layer activation functions

https://keras.io/api/layers/activations/

(Keras documentaion)

8. Newdev(2021). What is the difference between sparse\_categorical\_crossentropy and categorical\_crossentropy?

https://newbedev.com/what-is-the-difference-between-sparse-categorical-crossentropy-and-categorical-crossentropy

(Newdev2021)