**University of Michigan - Dearborn**

**Sentiment Analysis Using Deep Learning**

**IMSE 514 - Multivariate Statistics**

**Fall 2023**

**Professor Abdallah Chehade**

**Maya Ravichandran**

**12/13/2023**

**Abstract**

Social media is more important than what we think. Nowadays, companies have access to data for customer feedback on social media platforms. They are using advanced technology to leverage the power of those platforms to improve their businesses. They are analyzing their streams and extracting information about what people think about their product or service. This is called “sentiment analysis”. Some of them have already moved beyond the overall sentiment and are doing comprehensive research using the latest AI tools to get more information about the users’ preferences and intentions. Today’s tools can handle large volumes of data consistently and accurately.

In this project, we conducted a sentiment analysis on a substantial Twitter dataset, utilizing Long Short-Term Memory (LSTM) networks and Recurrent Neural Networks (RNN) as our primary analytical tools. Our results indicated an accuracy around 75% in predicting tweets.

**Introduction**

Sentiment analysis is a way to evaluate a text in order to get information about the expression whether it is positive, negative, or neutral. It uses a mix of statistics, natural language processing, and machine learning. It can be applied to one sentence or to a whole document like an email, comment, or review. Although it gives accurate and unbiased results, sentiment Analysis is still a developing field so the results might not always be perfect especially when working with a small dataset.

There are multiple types of sentiment analysis. For example, subjectivity classification classifies a sentence as opinionated or not opinionated. In polarity classification, opinionated sentences are classified as positive or negative. Furthermore, aspect based sentiment analysis detects sentiments toward a specific component of a product or service. Lastly, fine grained sentiment analysis is conducted to define both the sentiment and its intensity at a sentence level. It is used to process comparative expressions.

Sentiment analysis is used for various purposes. One of its most important applications is brand monitoring where companies monitor what people say about an organization, product, or service and the reasons behind that. Other companies use it for competitive research to track how society evaluates their competitors in order to understand their weaknesses and strengths. Moreover, it is used to optimize customer care by analyzing customer reviews and fixing the issues before they escalate. Other companies gather early feedback on a product as a part of the product analysis to improve it after it is launched. They also track and study consumers behavior for market research to predict future trends. In addition to that, human resource managers analyze employee surveys to keep track of their satisfaction and improve what is reducing their performance at work.

Sentiment analysis can be conducted by the following steps: data collection, sentiment annotation, text cleansing, word embedding, and model training and testing. The recent advances in deep learning improved the ability of analyzing texts. It does not only show the user expression but also the aspects that they care about and their underlying intentions through sentiment analysis, intent analysis, and contextual semantic search.

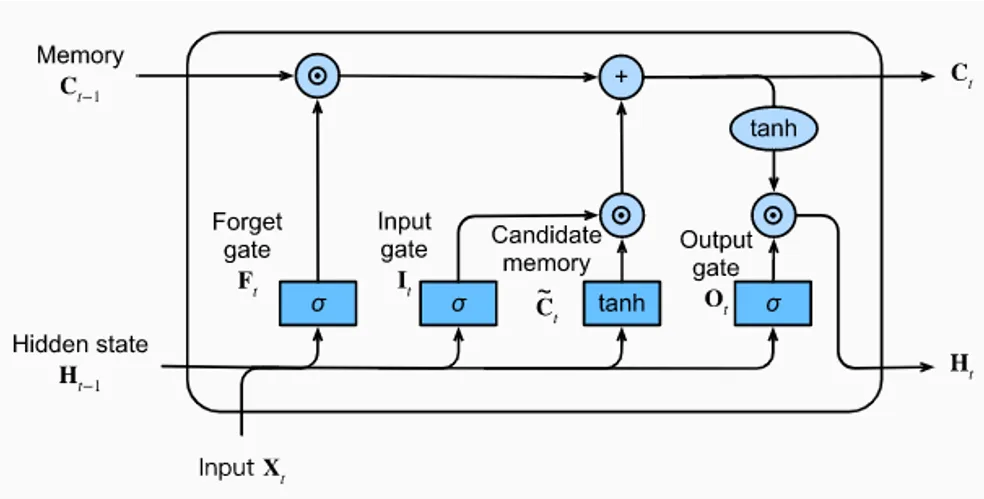
Uber is an example of a leading company that collects and analyzes huge amounts of data from the customers complaints, suggestions, and reviews on digital media. It has been using them to improve the quality of the service it provides. This explains why the employees care a lot about their customers' reviews and work hard to make them satisfied with their service and why people trust them. Uber is interested in following many categories in the news like: price, safety, payment, cancellation, and service. However, data analysis showed that the most popular topic that people care about is safety, the fact that made uber develop multiple safety features to keep the customers protected all the time.

**Modeling Techniques Used**

We used LSTM which is a type of RNN for our analysis and modeling. RNN is a recurrent neural network and it is a bidirectional artificial neural network. The term "recurrent neural network" is used to refer to the class of networks with an infinite impulse response. Their ability to use internal state (memory) to process arbitrary sequences of inputs makes them applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition.

LSTM: Additional stored states and the storage under direct control by the network can be added to both infinite-impulse and finite-impulse networks. The storage can also be replaced by another network or graph if that incorporates time delays or has feedback loops. Such controlled states are referred to as gated state or gated memory, and are part of long short-term memory networks (LSTMs) and gated recurrent units. This is also called Feedforward Neural Network (FNN). Recurrent neural networks are theoretically Turing complete and can run arbitrary programs to process arbitrary sequences of inputs.

Below is the depiction of how the LSTM architecture works.

** *Figure 1.***Architecture of a LSTM Unit

**Executive summary**

The project was conducted with the following steps:

1. Import python packages: Natural Language Toolkit (NLTK) which is used for natural language processing, SK-Learn to implement various machine learning models, and Tensorflow for fast numerical computations.
2. Load tweets into a dataframe from the dataset.
3. Preprocessing: Set positive and negative labels, select a subset of positive and negative tweets, initialize variables and constants for preprocessing and model development, define some data preprocessing functions, execute data cleaning, tokenize words to numeric form, and prepare the training and the test set.
4. Build RNN model: Define RNN layers and train RNN Model.
5. Model testing and inference: Evaluate the model, test a few samples, and run inference.
6. Check the model’s performance: Create the confusion matrix and plot the ROC curve.

**Exploratory Data analysis**

The dataset contains 1,600,000 tweets extracted using the twitter api. The tweets have been annotated (0 = negative, 2 = neutral, 4 = positive) and they can be used to detect sentiment .

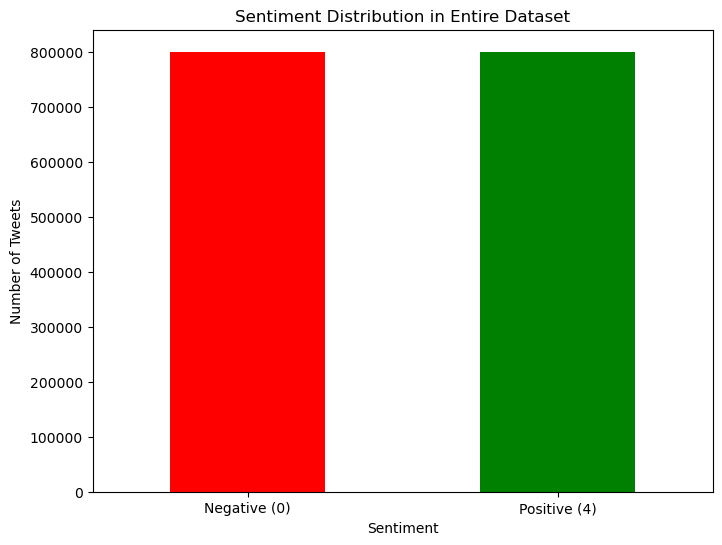
The content contains the following 6 fields:

1. target: the polarity of the tweet (*0* = negative, *2* = neutral, *4* = positive)
2. ids: The id of the tweet ( *2087*)
3. date: the date of the tweet (*Sat May 16 23:58:44 UTC 2009*)
4. flag: The query (*lyx*). If there is no query, then this value is NO\_QUERY.
5. user: the user that tweeted (*robotickilldozr*)
6. text: the text of the tweet (*Lyx is cool*)

Twitter data was analyzed by the following filtering techniques:

1. Emotions which had the following symbols were removed/ stripped off:
   1. Emoticons mapped to :),:-),: ) ,:D,=)
   2. Emoticons mapped to :( , :(, :-(, : (,
2. Any tweet containing both positive and negative emoticons are removed. This may happen if a tweet contains two subjects.
3. Retweets are removed. Retweeting usually happens if a user likes another user’s tweet. Retweets are commonly abbreviated with “RT.”
4. Tweets with “:P” are removed
5. Repeated tweets are removed

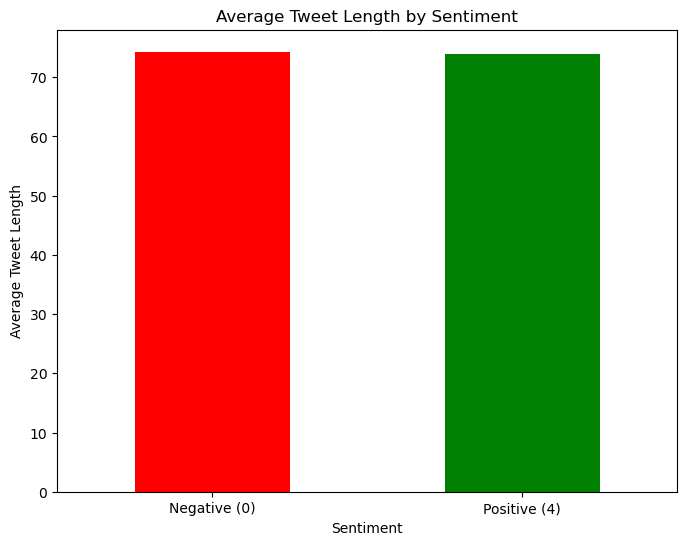
After post-processing the data, we took the first 800,000 tweets with positive emoticons, and 800,000 tweets with negative emoticons, for a total of 1,600,000 training tweets. The test data is manually collected, using the web application. A set of 177 negative tweets and 182 positive tweets were manually marked. Not all the test data has emoticons.



***Figure 2.*** *Sentiment Distribution in the Entire Dataset*

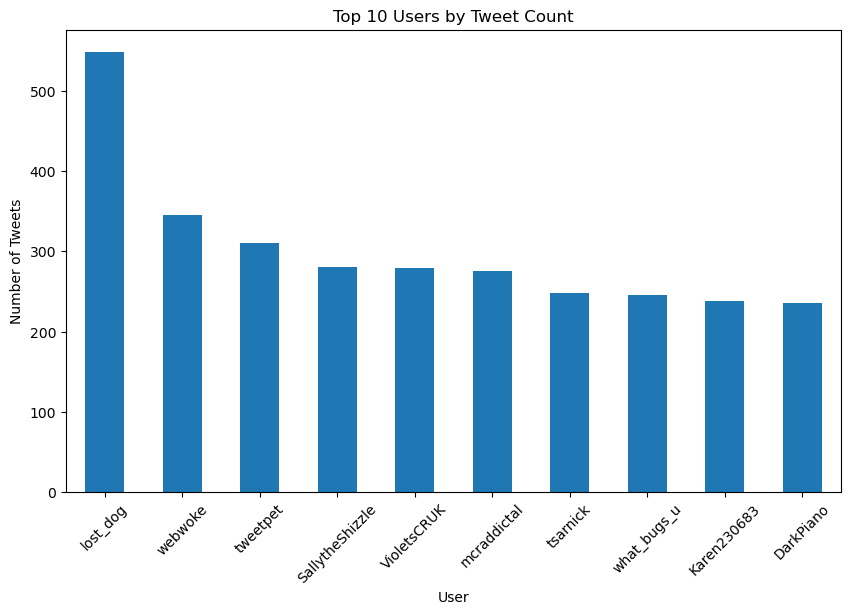
Here are a few exploratory analysis points that we observed:

1. The average tweet length by sentiment is analyzed as shown below:



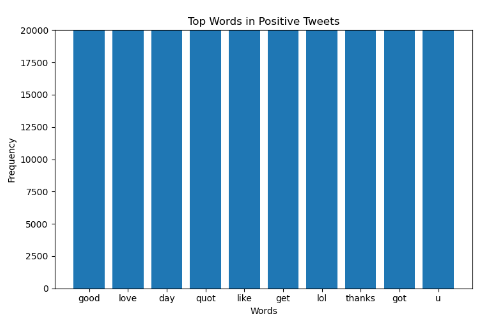
***Figure 3.*** *The Average Tweet Length by Sentiment*

2.The top 10 users by tweet count:

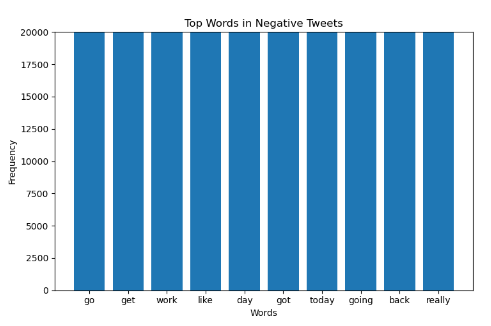


***Figure 4.*** *The Top 10 Users by Tweet Count*

3. The top words in each sentiment:



***Figure 5.*** *The Top 10 Words in Positive Tweets*



***Figure 6.*** *The Top 10 Words in Negative Tweets*

**Method**

Before building our model, we performed data cleaning to remove the unwanted features from the text which might cause noise and incorrect analysis. We removed irrelevant and duplicate data, punctuation marks, special characters, stopwords, Urls, emails, and numbers. Then we applied tokenization to separate the sentences into words. Then, we splitted the data into 70% training and 30% testing.

We built a recurrent neural network (RNN) model and we defined its layers. Layer 1 for embeddings that provide the presentation of words and their relative meaning. Layer 2 for the spatial dropout that drops some neurons from the previous layer. Layer 3 is for Long Short Term Memory (LSTM) that saves the words and predicts the next words based on the previous words. Layer 4 is a dense layer that ensures that the final prediction is a probability distribution.

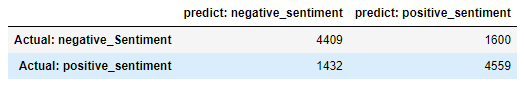
After that, we defined the loss function that measures the difference between the predicted output and the actual output of the model using sparse categorical cross entropy. We used Adam as an optimization method to update the weights during training. We set our metrics = accuracy to calculate the percentage of correct predictions on the validation set.

For the training part, we trained our model by feeding the data and using 15% of the training set for validation. We used a batch size of 80 which means that the model takes 80 tweets in each iteration and trains them. We also chose epochs = 5 which means that our model will be trained 5 times.

Finally, we tested the model by getting predictions on the test data and comparing them to the labels.

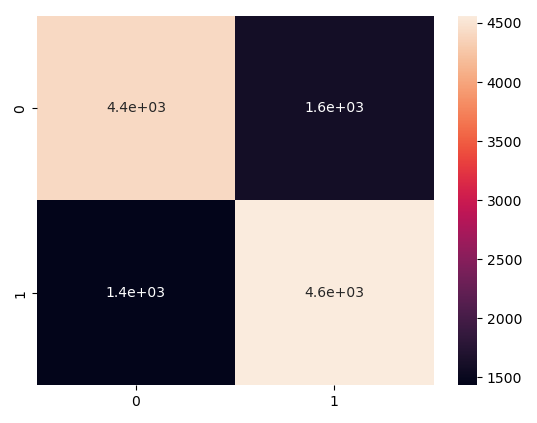
**Results**

The testing results indicated an accuracy of 0.7473 which means that 74.73% of the predicted tweets have been predicted correctly. The results also show that the number of true positives (TP) is 4559 tweets, the number of true negatives (TN) is 4409 tweets, the number of false positives (FP) is 1600 tweets, and the number of false negatives (FN) is 1432 tweets.

****

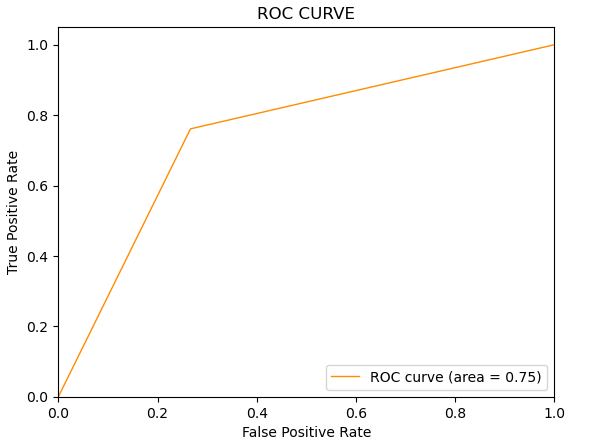
***Figure 7.*** *Table Showing the Testing Results*

The following is the confusion matrix. The pink boxes show the correct predictions and the navy blue boxes show the wrong predictions.

****

***Figure 8.*** *The Confusion Matrix*

The following ROC curve of our model. It is a performance measurement for the model at different thresholds. The area under the curve is 0.75 which means that there is a 75% chance that the model will be able to distinguish between the positive tweets and the negative tweets.

****

***Figure 9.*** *The ROC Curve*

**Conclusion**

To conclude, in this project we tried to classify the tweets sentiments on twitter as positive or negative by using RNN and LSTM. We prepared the data, trained the model, and tested its performance. The results showed a good accuracy of 75% for a batch size of 80 and epochs of 5. For future work, we can try parameter tuning by changing the batch size and epochs until we get better results.

**References**

<https://www.qualtrics.com/experience-management/research/sentiment-analysis/>

<https://towardsdatascience.com/sentiment-analysis-concept-analysis-and-applications-6c94d6f58c17>

<https://www.altexsoft.com/blog/sentiment-analysis-types-tools-and-use-cases/>

<https://d2l.ai/chapter_recurrent-modern/lstm.html>