**DS675 - Machine Learning Final Project**

-JASWANTH KUMAR KANDIPILLI(jk755)

-VENKATA SAI NEERAJ JANNELA(vj254)

-VISHNU PRIYANKA YARLAGADDA(vy57)

**Sentiment Analysis Using Word Embedding**

**Video:**

**Code:**[**Sentiment Analysis using Word Embedding .ipynb**](https://colab.research.google.com/drive/1EmJ_weaELtOXi_GmIZdzYGMCmhiJYQh3#scrollTo=IyWxobXIVzDG)

**Dataset:**https://www.kaggle.com/datasets/cosmos98/twitter-and-reddit-sentimental-analysis-dataset

### **Description**

The Twitter dataset includes approximately 163,000 tweets with sentiment labels, while the Reddit dataset comprises around 37,000 comments with sentiment scores. The data was collected using the Tweepy and Reddit APIs and primarily focuses on discussions related to political leaders, including Narendra Modi, and public sentiments regarding the next Prime Minister of India during the 2019 general elections. The sentiment labels range from -1 (negative) to 0 (neutral) to 1 (positive). Each dataset contains two columns, one for the cleaned tweets or comments and another for the corresponding sentiment labels.

### **Introduction**

This project involves sentiment analysis on social media text data from Twitter and Reddit. The goal is to build separate LSTM models for Twitter and Reddit data, predicting sentiment categories using deep learning techniques.

### **Principles Used in Model**

The principle behind using word embeddings in a sentiment analysis model involves representing words as continuous vectors in a high-dimensional space. This approach is rooted in the idea that words with similar meanings or contexts should have similar vector representations. Word embeddings capture semantic relationships between words, allowing the model to understand the contextual nuances of the language.

**1.** **Word Embeddings:**

· Semantic Similarity: The purpose of word embeddings is to capture semantic similarity between words. Words with comparable meanings or usage patterns are mapped to nearby vectors in the embedding space.

· Contextual Information: Unlike traditional approaches that represent words as discrete symbols, word embeddings consider the context in which words appear. Words that appear in similar contexts are embedded into similar vectors.

**2.** **Embedding Layer in Neural Networks:**

· Input Representation: An Embedding layer is used in neural networks for natural language processing applications such as sentiment analysis.

· Word -Vector Mapping: The Embedding layer maps each word in the input sequence to its corresponding dense vector (embedding) representation.

· Learnable Parameters: During training, the weights of the Embedding layer are adjusted based on the task at hand, optimizing the representation of words for the specific problem.

**3.** **Sentiment Analysis:**

· Context Understanding: Sentiment analysis involves understanding the sentiment expressed in a piece of text. The continuous vector representations of words help the model understand the contextual relationships between words and phrases.

· Learning from Data: The model learns from labeled data where the sentiment of the text is provided. It adjusts its parameters, including word embeddings, to optimize sentiment prediction.

**4.** **t-SNE Visualization:**

· Dimensionality Reduction: t-SNE is used for visualizing high-dimensional data in a lower-dimensional space, in this case, reducing the dimensionality of word embeddings to 2D.

· Insights into Relationships: The resulting visualization allows for insights into how words are positioned relative to each other in the embedding space, providing qualitative information about semantic relationships.

#### **Data Loading and Exploration**

**Twitter Data:**

· Loaded Twitter data from '/content/drive/MyDrive/Twitter\_Data.csv'.

· Displayed basic information about the Twitter dataset.

**Reddit Data:**

· Loaded Reddit data from '/content/drive/MyDrive/Reddit\_Data.csv'.

· Displayed basic information about the Reddit dataset.

**Handling Missing Values:**

· Dropped missing values from both Twitter and Reddit datasets.

**Text Cleaning:**

· Lowercase the text in both Twitter and Reddit datasets.

#### **Text Tokenization and Padding**

**Tokenization:**

· Tokenized Twitter text using the Keras Tokenizer.

· Tokenized Reddit text using the Keras Tokenizer.

**Padding Sequences:**

· Padded Twitter sequences to a maximum length of 250.

· Padded Reddit sequences to a maximum length of 250.

#### **Model Architecture**

**Twitter Model**:

· Created an LSTM model for Twitter data.

· Compiled the model using categorical cross entropy loss and Adam optimizer.

**Reddit Model:**

· Created an LSTM model for Reddit data.

· Compiled the model using categorical cross entropy loss and Adam optimizer.

#### **Data Splitting and Label Encoding**

**Reddit Data:**

· Converted Reddit labels to categorical format.

· Split Reddit data into training, validation, and test sets.

**Twitter Data:**

· Converted Twitter labels to categorical format.

· Split Twitter data into training, validation, and test sets.

#### **Model Training**

**Reddit Model:**

· Trained the Reddit model on the training set for 5 epochs.

**Twitter Model:**

· Trained the Twitter model on the training set for 5 epochs.

#### **t-SNE Visualization of Word Embeddings**

**Twitter Model:**

· Extracted embedding weights from the Twitter model.

· Applied t-SNE to reduce the dimensionality of the embeddings.

· Visualized embeddings for all sentiments, positive, and negative sentiments.

**Reddit Model:**

· Extracted embedding weights from the Reddit model.

· Applied t-SNE to reduce the dimensionality of the embeddings.

· Visualized embeddings for all sentiments, positive, and negative sentiments.

#### 

#### **Conclusion**

In conclusion, we successfully implemented sentiment analysis on social media text data. Two LSTM models were trained on Twitter and Reddit datasets, demonstrating their predictive capabilities. The t-SNE visualizations provided insights into the word embeddings' distribution for different sentiments. Further improvements and analysis can be explored in future iterations, including hyperparameter tuning, model evaluation, and potentially incorporating more advanced techniques for sentiment analysis. We created two models where the results for both the datasets showed remarkable results where accuracy lies between 90 to 95 epochs and batch size mainly influenced it. But the problem with this model is that time time-consuming because for the Reddit dataset it takes up to 30 minutes for each epoch whereas for the Twitter dataset, it takes 2 hr 30 minutes for each epoch which is quite time-consuming.

#### **Differences made from previous work**

In previous work, they implemented and compared using machine learning algorithms, their project aims to enhance the accuracy and efficiency of predicting sentiments from social media text. The proposed methodology, utilizing Spark and Python for scalable and parallelized processing, addresses the limitations of traditional model training methods. Their ultimate goal is to contribute to more accurate and resource-efficient sentiment analysis in the context of elections.

Our project highlights how important social media is as a valuable resource for forecasting voter sentiment. The utilization of sophisticated machine learning methods, specifically LSTM-based word embedding in sentiment analysis, improves the precision and comprehensiveness of comprehending public sentiments. The sentiment distribution can be understood more easily when word embeddings are visualized using t-SNE. The methodology demonstrates how sophisticated techniques have evolved from traditional sentiment analysis, leading to more accurate and nuanced predictions in the context of elections.