**INTERNSHIP: PROJECT REPORT**

------------------------------------------------------------------------------------------------------------------------------------------

| | |
|---|---|
| Internship Project Title | TCS iON RIO 125 - Automate sentiment analysis of textual comments and feedback |
| Project Title | Automate sentiment analysis of textual comments and feedback |
| Name of the Company | ICT Academy of Kerala |
| Name of the Industry Mentor | Debashis Roy |
| Name of the Institute | ICT Academy of Kerala |

| Start Date | End Date | Total Effort (hrs.) | Project Environment | Tools used |
|---|---|---|---|---|
| 10-02-2023 | 23-03-2023 | 169 | Windows, Python, python libraries, colab | Keras, NLTK, Scikit-learn, colab, jupyter notebook etc. |

Project Synopsis:

The project aims to develop deep learning algorithms to detect different types of sentiment contained in a large paragraph, including positive, negative, and neutral sentiments. The project involves identifying and finalizing a collection of English sentences or a large paragraph with contradictory statements. A deep learning model is developed for the detection and segmentation of sentiments. The model is then enhanced to accurately predict the overall sentiment of the paragraph, even if it contains contradictory statements. Finally, the model is tested for accuracy to evaluate its performance.

Solution Approach:

1. Data Collection: The first step is to collect a dataset of English sentences or paragraphs that have contradictory statements and include a mix of positive, negative, and neutral sentiments.
2. Data Preprocessing: Preprocess the collected data by cleaning the text, removing stop words, and tokenizing the sentences. Convert the text data into numerical vectors that can be used as input for a deep learning model.
3. Model Development: Develop a deep learning model to detect and segment the different types of sentiments in the text. Train the model on the preprocessed data and optimize the model parameters to improve performance.
4. Model Enhancement: Enhance the model to predict the overall sentiment of the paragraph, even if it contains contradictory statements.
5. Model Testing: Test the accuracy of the model on a separate test dataset that the model has not seen before.

Assumptions:

1. The English language is the only language used in the dataset and the model is designed to work specifically with this language.
2. The model is able to correctly identify and segment different sentiments within a single paragraph, without confusing them with one another.
3. The model is able to handle and process large paragraphs with a high degree of accuracy.
4. The model is expected to perform well on new data and real-world scenarios, not just the data used for training and testing.

Project Diagrams:

1. A line plot of model accuracy (train and validation) over epochs

2. A bar chart of the number of positive and negative sentences in the input paragraph
3. A pie chart of the proportion of positive and negative sentences in the input paragraph

Algorithms:

1. LSTM (Long Short-Term Memory) - a type of recurrent neural network architecture used for sequential data analysis, particularly in natural language processing (NLP) tasks.
2. Tokenizer - a utility class in the Keras API used to preprocess text data by converting text to sequences of integers (i.e., tokens) based on word frequency.
3. Embedding - a layer in a neural network used for learning a dense representation of words in a text corpus, which can be used to map words to vectors in a high-dimensional space.
4. LabelEncoder - a utility class in scikit-learn used to encode categorical labels as integers.
5. SpatialDropout1D - a type of dropout layer used in neural networks for text data, which randomly drops out entire 1D feature maps in order to improve model robustness and prevent overfitting.

Outcome:

This project automates sentiment analysis on movie reviews from the IMDB dataset using a deep learning model. It trains a sequential model with an embedding layer, a spatial dropout layer, an LSTM layer, and a dense layer with softmax activation. The model is trained on 90% of the data and tested on the remaining 10%. After training, the model is tested on three example sentences and then on a paragraph of text to determine the overall sentiment. Finally, the model is evaluated on the test set and the test accuracy is printed along with a bar chart of the training and validation accuracy over each epoch.

Exceptions considered:

1. model.fit() raised various exceptions such as ValueError and RuntimeError because the GPU ran out of memory.
2. The number of epochs had to be reduced and activation functions had to be changed several times due to the large amount of time it took to train the model.
3. The model accuracy suffers due to the lack of a diverse range of sentiments in the training data and because the deep learning model is trained on movie review data it may not perform best when tested on different types of data.

Enhancement Scope:

1. Fine-tuning the model: Fine-tuning the deep learning model on specific text data can further improve the model's accuracy. By retraining the pre-trained model on a specific dataset, the model can learn domain-specific features and better understand the nuances of the language used in the data.
2. Incorporating context and tone: Sentiment analysis is not only about identifying positive, negative, or neutral sentiments in a sentence, but also about understanding the context and tone of the text. By incorporating contextual features and tone analysis, the model can better differentiate between different types of sentiments and make more accurate predictions.
3. Handling contradictory statements: One of the main challenges in sentiment analysis is dealing with contradictory statements. To improve the model's accuracy in this area, techniques such as multi-task learning, ensemble learning, or adversarial training can be employed.
4. Incorporating human feedback: Incorporating human feedback through active learning can help the model to learn from its mistakes and improve its accuracy over time.

Link to Code and executable file: https://github.com/ManuDavis/TCS-Internship