Project Report: Text Similarity Evaluation with Machine Learning

# Objective

The objective of this project is to develop a machine learning model capable of evaluating the similarity between pairs of text inputs. This involves processing natural language data, transforming sentences into numerical representations, and predicting a similarity score that reflects the degree of similarity between the two text inputs.

# Data Preparation

The project begins with importing necessary libraries, including pandas for data manipulation, numpy for numerical operations, and tensorflow along with tensorflow\_hub for leveraging pre-trained models and building the neural network. The input data, consisting of pairs of sentences with their corresponding similarity scores, is loaded into a dataframe from a CSV file for initial exploration and understanding.

# Text Encoding

A crucial step in the project is the encoding of text data into tensors using a pre-trained Universal Sentence Encoder from TensorFlow Hub. This encoder converts sentences into high-dimensional vectors (embeddings) of fixed size (512 dimensions in this case), effectively capturing the semantic meaning of the sentences in a form that can be processed by machine learning models.

# Model Architecture

The model architecture is designed to compare pairs of sentence embeddings and predict their similarity. It involves the following steps:  
- Taking two input tensors corresponding to the embeddings of two sentences.  
- Calculating the difference and the absolute difference between these embeddings.  
- Concatenating these differences to form a composite feature vector.  
- Passing this feature vector through a series of dense layers with ReLU activation functions, culminating in a single output neuron with a sigmoid activation function to predict the similarity score.  
The model is compiled with the Adam optimizer, mean squared error as the loss function, and mean absolute error as a performance metric.

# Training

The model is trained using the embeddings of sentence pairs and their corresponding similarity scores. A validation split is employed to monitor the model's performance on unseen data during the training process.

# Application and Model Persistence

After training, the model is saved for future use. A function is defined to predict the similarity score for any given pair of sentences by first encoding them into embeddings, then passing these embeddings through the trained model to obtain a similarity score. This process can done by the flask API with ‘POST’ method on URL “ <https://testing20123.pythonanywhere.com/predict> ” .

The Request body format :

{“text1”: ”nuclear body seeks new tech .......”, ”text2”: ”terror suspects face arrest ......”}

And the Response will be like

Response body: {“similarity score”: 0.2 }

# Conclusion

This project demonstrates the application of neural networks and natural language processing techniques to assess the similarity between pairs of sentences. By leveraging the Universal Sentence Encoder for text representation and a custom neural network architecture for prediction, the project provides a robust solution for quantifying text similarity, with potential applications in various domains such as content comparison, plagiarism detection, and semantic search.