Anubhav Kundu
Final Project
CIS 4526

In this project, we were to develop a multi-layer perceptron **(MLP)** model to determine if two given sentences are paraphrases of one another. I developed an **MLP** using the PyTorch library, as well as designed features such as:

- **Length Difference**
- **Levenshtein's Distance**
- **METEOR Score**
- **BLEU-1, 2, 3, 4 Scores**
- **Cosine Similarity**
- **2 & 3 NGrams Overlap**
- **Jaccard Similarity**
- **Sorenson's Dice**
- **Jaro-Winkler Distance**
- **# Overlapping Words**

Data Preprocessing:

- Data preprocessing was done using the Pandas and NumPy libraries, and the development files were read through the pd.read_csv() function
- Preprocessing functions were created to manage and add features to the training, development, and test dataframes
- The X values and y values were converted to PyTorch Tensors, to be usable in the neural network

Algorithms & Libraries:

- Libraries include: **NLTK, SKLearn, Pandas, NumPy, PyTorch, Re**
- The PyTorch library was used to implement a custom multi-layer perceptron model, with:
  - 3 Hidden Layers **(Tanh, ReLu Activation Functions)**
  - Sigmoid Output Layer
  - Xavier Uniform Distribution was used to randomly initialize weights and biases for the layers
  - PyTorch DataLoaders were used to enumerate through the training, validation, and test data

Experiences & Lessons:

- ❖ This project allowed for the experience of learning a solve a new world NLP problem using machine learning libraries
- ❖ Many hours were spent trying to develop the multi-layer perceptron model using PyTorch
- ❖ Familiarized using feature scaling to improve performance, as well as tuning hyperparameters **(batch size, learning rate, epochs)** for higher accuracy. Used weight decay to avoid the overfitting of training data
- ❖ At each epoch/step, the best performing model on the validation set **i.e. the lowest validation loss** was saved using PyTorch, and loaded when time to create test results