Project Final Report

CS7.401. Introduction to NLP

Project: **Textual Coherence**

Team: **ByteSpeak**

- Smruti Biswal - **2020112011**
- Pranali Jagdale - **202010198**
- Shiva Shankar - **2023202005**

I. Introduction

Textual coherence is the property that makes a text logically connected and semantically meaningful and unified as a whole. It is achieved through various linguistic features that create continuity of meaning and logical connections between the different parts of a text.

II. Datasets Used

- **Grammarly Corpus of Discourse Coherence (GCDC)**
  The Grammarly Corpus of Discourse Coherence (GCDC) is a benchmark dataset used for evaluating discourse coherence algorithms.
  It contains four different domains: Yahoo, Clinton, Enron, and Yelp. Each domain has two files, one for the training set and one for the test set.
  The dataset is annotated into classes 1,2 and 3 where 3 denotes the most coherent paragraph. It consists of four training datasets (1000 paragraphs each) and four testing datasets (200 paragraphs each). We have merged the four training datasets along with two testing datasets to train our model (4600 paragraphs) and used the remaining test data to test the accuracy of our model.

  The files are in CSV format, and each row represents a single text instance with the following columns:

  text_id: A unique identifier for each text instance.

subject (Clinton, Enron, Yelp): Additional context provided to the annotators, such as the subject line of an email, but not used for training models.
question_title and question (Yahoo only): Additional context provided to the annotators, but not used for training models.
text: The actual document or text content.
ratingA1, **ratingA2**, **ratingA3**: Three coherence ratings provided by expert annotators on a scale (e.g., 1-5).
labelA: The consensus label based on the three expert ratings, indicating whether the text is coherent or incoherent.
ratingM1, **ratingM2**, **ratingM3**, **ratingM4**, **ratingM5**: Five coherence ratings provided by Amazon Mechanical Turk (MTurk) annotators on a scale (e.g., 1-5).
labelM: The consensus label based on the five MTurk ratings, indicating whether the text is coherent or incoherent.

official_wikipedia.jsonl and official_cnn.jsonl are the data files for Wikipedia and CNN/Daily news sets, respectively. Files in the linguistic_prob_data directory are linguistic probe dataset.

For each item, there are

(1) file_id: the unique id for the file;

(2) ctx: the original file (untouched version);

(3) to_be_replaced: the sentence from the original file, which will be replaced by the "replace_with" sentence;

(4) replace_with: the newly introduced sentence to replace the 'to_be_replaced' sentence in the document;

(5) sen_position: indicating the position of the "to_be_replaced" sentence or the intruder sentence (-1 indicate that the document remain as it is, the coherent document);

(6) train: the flag to indicate whether the document belongs to the train/test set (1 indicates that it is in the train set; otherwise it is in the test set);

III. Models Used

RNN:
Recurrent Neural Networks (RNNs) are a class of neural networks designed to handle sequential data. Unlike traditional neural networks that process inputs in isolation, RNNs have internal loops allowing them to maintain information in 'hidden' layers across inputs. This makes them particularly suited for tasks where context or the temporal sequence of data is important, such as language modeling, speech recognition, and time series prediction.

LSTM:
Long Short-Term Memory (LSTM) is a type of Recurrent Neural Network (RNN) architecture used in the field of deep learning. LSTMs are designed to solve the problem of vanishing and exploding gradients that can occur in traditional RNNs, making them effective for learning dependencies in sequence data. Each LSTM unit contains a cell, an input gate, an output gate, and a forget gate. These gates control the flow of information into and out of the cell, allowing the network to retain or forget information dynamically over time.

GRU:
Gated Recurrent Units (GRUs) are a type of Recurrent Neural Network (RNN) designed to more efficiently capture dependencies in sequence data without the complexity of traditional LSTMs. GRUs simplify the architecture of LSTMs by combining the forget and input gates into a single "update gate" and merging the cell state and hidden state. This results in a model that is easier to compute and often performs as well as, if not better than, LSTMs on certain tasks.

IV. Approaches Used

The following approaches were used on the GCDC corpus to fine tune the models:

LSTM:

We trained our model on the GCDC corpus using the default annotation to make a three-way multi-classifier. Later, we switched this approach, remodeled our data to binary labels- coherent and incoherent- and used this data on a binary classifier to observe better results

LSTM model split into training and testing sets (4400/400) for Yelp and Enron data

1. Multi-class Classification LSTM

Embedding Layer: Converts word indices to 64-dimensional vectors, handling up to 40,000 words.

LSTM Layers: Two layers with 64 units each, using a dropout of 0.3 to reduce overfitting. The first layer returns sequences; the second does not.

Output Layer: A dense layer with softmax activation, targeting four classes.

Training: The model is compiled with categorical cross-entropy loss and trained for 10 epochs.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 500, 64)           2560000

 lstm (LSTM)                 (None, 500, 64)           33024
 lstm_1 (LSTM)               (None, 64)                33024

 dense (Dense)               (None, 4)                 260

=================================================================
Total params: 2626308 (10.02 MB)
Trainable params: 2626308 (10.02 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
None
Epoch 1/10
138/138 [==============================] - 108s 748ms/step - loss: 1.0777 - accuracy: 0.4550
```

```
Epoch 2/10
138/138 [==============================] - 96s 698ms/step - loss: 0.9765 - accuracy: 0.5405
Epoch 3/10
138/138 [==============================] - 98s 710ms/step - loss: 0.7324 - accuracy: 0.6841
Epoch 4/10
138/138 [==============================] - 102s 742ms/step - loss: 0.4731 - accuracy: 0.7941
Epoch 5/10
138/138 [==============================] - 97s 705ms/step - loss: 0.2979 - accuracy: 0.8866
Epoch 6/10
138/138 [==============================] - 97s 706ms/step - loss: 0.1588 - accuracy: 0.9455
Epoch 7/10
138/138 [==============================] - 99s 713ms/step - loss: 0.0981 - accuracy: 0.9684
Epoch 8/10
138/138 [==============================] - 97s 701ms/step - loss: 0.0754 - accuracy: 0.9784
Epoch 9/10
138/138 [==============================] - 105s 761ms/step - loss: 0.0573 - accuracy: 0.9834
Epoch 10/10
138/138 [==============================] - 100s 723ms/step - loss: 0.0593 - accuracy: 0.9814


Accuracy:  37.74999976158142
```

2. Binary Classification LSTM

Input Length: Adjusted to 501 to include an additional feature.

Dropout Rate: Slightly reduced to 0.2.

Output Layer: Modified for binary output using softmax.

Loss Function: Binary cross-entropy.

Training: The model trains for 15 epochs, using the standard data augmented with binary labels.

3. Binary Classification LSTM with Feature Augmentation

Feature Augmentation: Cosine similarity scores are added as an extra input feature.

Training: Uses the same LSTM configuration as the binary classification model but now includes the augmented feature.

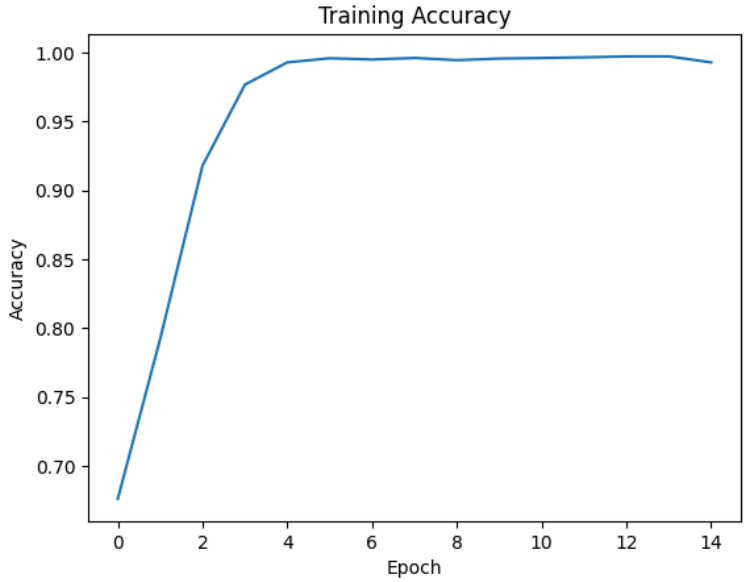
Implemented the compute_paragraph_similarity function to calculate similarity scores between consecutive sentences within paragraphs of the dataset. These scores were then added as a new column named 'similarity_scores' to the dataset.

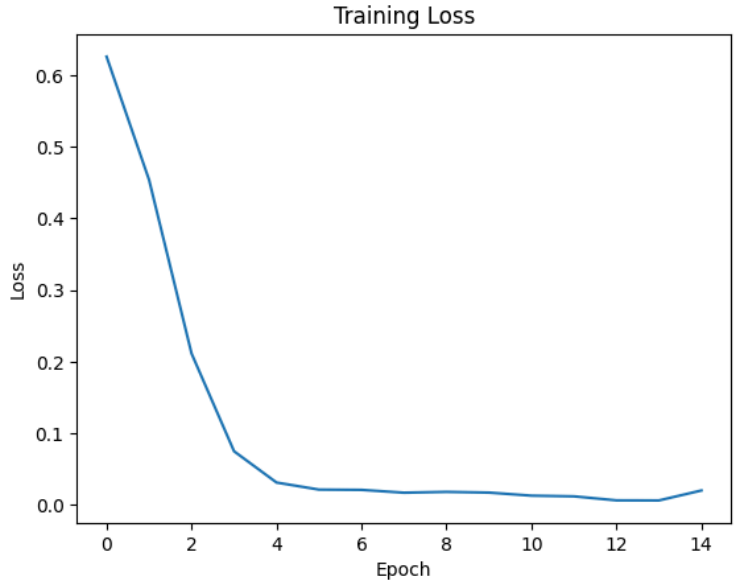
The compute_paragraph_similarity function processes a DataFrame containing text paragraphs by performing the following steps:

- Iterates through each paragraph, computing cosine similarity between consecutive sentences.
- Tokenizes sentences and removes stopwords to focus on keywords.
- Constructs binary vectors for each sentence based on word presence, then calculates the dot product.
- Determines the minimum cosine similarity score within each paragraph.
- Appends the minimum similarity score for each paragraph to the dataset as a new column.

```
Epoch 1/15
138/138 [==============================] - 22s 126ms/step - loss: 0.6256 - accuracy: 0.6764
Epoch 2/15
138/138 [==============================] - 13s 95ms/step - loss: 0.4537 - accuracy: 0.7923
Epoch 3/15
138/138 [==============================] - 11s 82ms/step - loss: 0.2115 - accuracy: 0.9180
Epoch 4/15
138/138 [==============================] - 8s 58ms/step - loss: 0.0752 - accuracy: 0.9766
Epoch 5/15
138/138 [==============================] - 9s 64ms/step - loss: 0.0315 - accuracy: 0.9927
Epoch 6/15
138/138 [==============================] - 9s 65ms/step - loss: 0.0217 - accuracy: 0.9957
Epoch 7/15
138/138 [==============================] - 7s 54ms/step - loss: 0.0213 - accuracy: 0.9948
Epoch 8/15
138/138 [==============================] - 6s 44ms/step - loss: 0.0173 - accuracy: 0.9959
Epoch 9/15
138/138 [==============================] - 7s 54ms/step - loss: 0.0185 - accuracy: 0.9943
Epoch 10/15
138/138 [==============================] - 5s 38ms/step - loss: 0.0175 - accuracy: 0.9955
Epoch 11/15
138/138 [==============================] - 6s 45ms/step - loss: 0.0132 - accuracy: 0.9959
Epoch 12/15
138/138 [==============================] - 6s 41ms/step - loss: 0.0122 - accuracy: 0.9964
Epoch 13/15
```

138/138 [==============================] - 5s 39ms/step - loss: 0.0067 - accuracy: 0.9970
Epoch 14/15
138/138 [==============================] - 7s 52ms/step - loss: 0.0066 - accuracy: 0.9970
Epoch 15/15
138/138 [==============================] - 5s 34ms/step - loss: 0.0204 - accuracy: 0.9927






```
Accuracy:  57.499998807907104
```

GRU:

Similar to Long Short-Term Memory (LSTM) units, Gated Recurrent Unit (GRU) models are used for learning from sequence data.

Three models are implemented and trained: a standard GRU model, a binary-labeled GRU model, and a GRU model that incorporates similarity scores as features.

The GRU (Gated Recurrent Unit) models are configured with embedding layers for input word vectorization, GRU layers for learning from sequence data, and dense layers for output classification. The binary-labeled model and the one incorporating similarity scores differ slightly in output layer configuration and data input structure.

Models are trained with padded sequences of tokenized text, where padding ensures uniform input size. Training involves adjusting model weights to minimize a loss function (binary cross-entropy) over several epochs, and accuracy is monitored as a key performance metric.

Incorporation of Similarity Scores:

For the model using similarity scores, these scores are appended to the input feature matrix, increasing the input dimension by one. This allows the model to learn not only from the textual content but also from the structural similarity between sentences within the text.

standard GRU

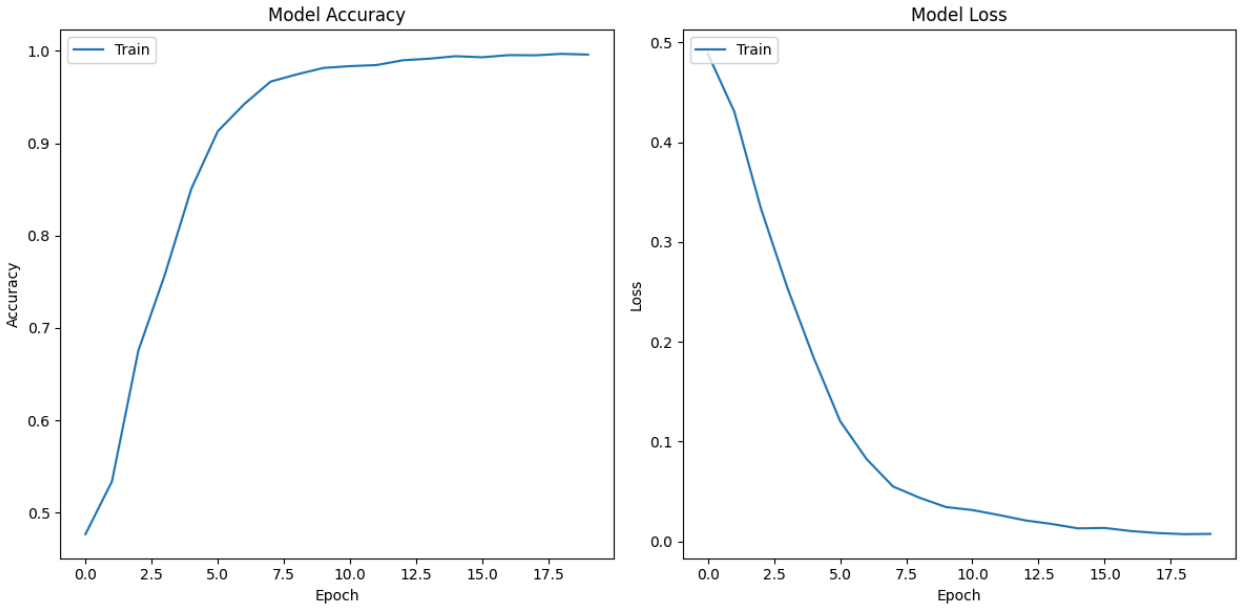
Model: "sequential"

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 500, 32)	1280000
gru (GRU)	(None, 500, 32)	6336
gru_1 (GRU)	(None, 32)	6336
dense (Dense)	(None, 4)	132

Total params: 1292804 (4.93 MB)
Trainable params: 1292804 (4.93 MB)
Non-trainable params: 0 (0.00 Byte)

```
Epoch 1/20
163/163 [==============================] - 39s 201ms/step - loss: 0.4886 - accuracy: 0.4767
Epoch 2/20
163/163 [==============================] - 23s 142ms/step - loss: 0.4307 - accuracy: 0.5338
Epoch 3/20
163/163 [==============================] - 15s 89ms/step - loss: 0.3341 - accuracy: 0.6756
Epoch 4/20
163/163 [==============================] - 14s 83ms/step - loss: 0.2546 - accuracy: 0.7577
Epoch 5/20
163/163 [==============================] - 10s 63ms/step - loss: 0.1841 - accuracy: 0.8506
Epoch 6/20
163/163 [==============================] - 10s 60ms/step - loss: 0.1205 - accuracy: 0.9129
Epoch 7/20
163/163 [==============================] - 9s 57ms/step - loss: 0.0825 - accuracy: 0.9423
Epoch 8/20
163/163 [==============================] - 7s 40ms/step - loss: 0.0550 - accuracy: 0.9667
Epoch 9/20
163/163 [==============================] - 9s 53ms/step - loss: 0.0437 - accuracy: 0.9746
Epoch 10/20
163/163 [==============================] - 8s 47ms/step - loss: 0.0344 - accuracy: 0.9815
Epoch 11/20
163/163 [==============================] - 7s 44ms/step - loss: 0.0314 - accuracy: 0.9835
Epoch 12/20
163/163 [==============================] - 8s 48ms/step - loss: 0.0264 - accuracy: 0.9846
Epoch 13/20
163/163 [==============================] - 7s 41ms/step - loss: 0.0210 - accuracy: 0.9898
Epoch 14/20
163/163 [==============================] - 7s 45ms/step - loss: 0.0174 - accuracy: 0.9915
Epoch 15/20
163/163 [==============================] - 7s 42ms/step - loss: 0.0130 - accuracy: 0.9942
Epoch 16/20
163/163 [==============================] - 8s 49ms/step - loss: 0.0134 - accuracy: 0.9931
Epoch 17/20
163/163 [==============================] - 7s 40ms/step - loss: 0.0103 - accuracy: 0.9954
Epoch 18/20
163/163 [==============================] - 7s 44ms/step - loss: 0.0083 - accuracy: 0.9952
Epoch 19/20
163/163 [==============================] - 6s 39ms/step - loss: 0.0072 - accuracy: 0.9967
Epoch 20/20
163/163 [==============================] - 7s 43ms/step - loss: 0.0075 - accuracy: 0.9960
```

Accuracy: 34.00%



binary-labeled GRU

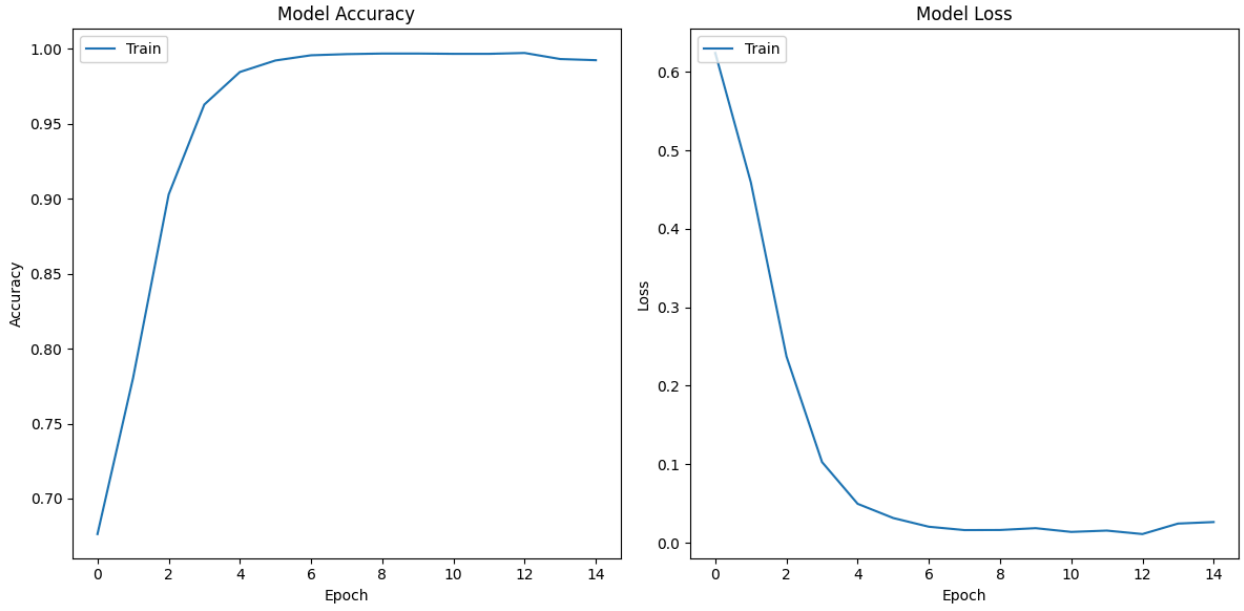
Model: "sequential_1"

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 500, 32)	1280000
gru_2 (GRU)	(None, 500, 32)	6336
gru_3 (GRU)	(None, 32)	6336
dense_1 (Dense)	(None, 2)	66

Total params: 1292738 (4.93 MB)
Trainable params: 1292738 (4.93 MB)
Non-trainable params: 0 (0.00 Byte)

Epoch 1/15
163/163 [==============================] - 39s 185ms/step - loss: 0.6243 - accuracy: 0.6763
Epoch 2/15
163/163 [==============================] - 16s 98ms/step - loss: 0.4592 - accuracy: 0.7804
Epoch 3/15
163/163 [==============================] - 12s 73ms/step - loss: 0.2375 - accuracy: 0.9027
Epoch 4/15
163/163 [==============================] - 10s 62ms/step - loss: 0.1027 - accuracy: 0.9627
Epoch 5/15

163/163 [==============================] - 8s 51ms/step - loss: 0.0495 - accuracy: 0.9844
Epoch 6/15
163/163 [==============================] - 9s 58ms/step - loss: 0.0314 - accuracy: 0.9921
Epoch 7/15
163/163 [==============================] - 7s 44ms/step - loss: 0.0203 - accuracy: 0.9956
Epoch 8/15
163/163 [==============================] - 6s 37ms/step - loss: 0.0161 - accuracy: 0.9963
Epoch 9/15
163/163 [==============================] - 8s 49ms/step - loss: 0.0162 - accuracy: 0.9967
Epoch 10/15
163/163 [==============================] - 6s 37ms/step - loss: 0.0185 - accuracy: 0.9967
Epoch 11/15
163/163 [==============================] - 8s 48ms/step - loss: 0.0137 - accuracy: 0.9965
Epoch 12/15
163/163 [==============================] - 6s 37ms/step - loss: 0.0154 - accuracy: 0.9965
Epoch 13/15
163/163 [==============================] - 11s 67ms/step - loss: 0.0110 - accuracy: 0.9971
Epoch 14/15
163/163 [==============================] - 6s 39ms/step - loss: 0.0244 - accuracy: 0.9931
Epoch 15/15
163/163 [==============================] - 6s 40ms/step - loss: 0.0263 - accuracy: 0.9923



Accuracy: 63.25%

GRU model that incorporates similarity scores:

Model: "sequential_2"

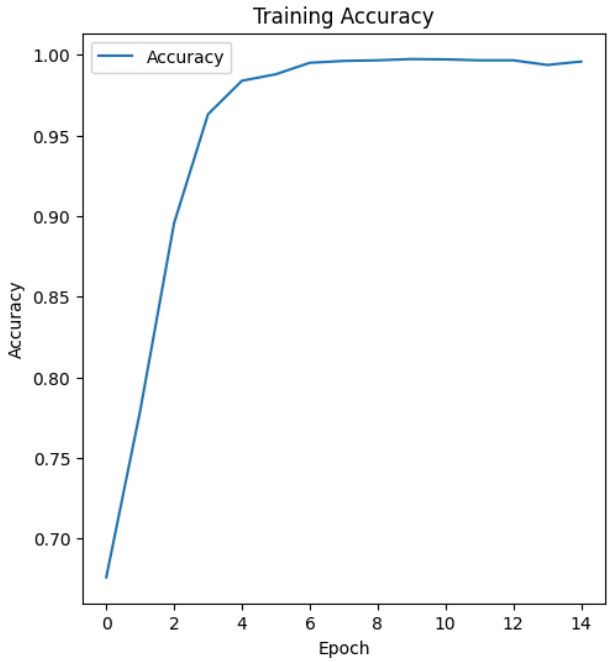
_____

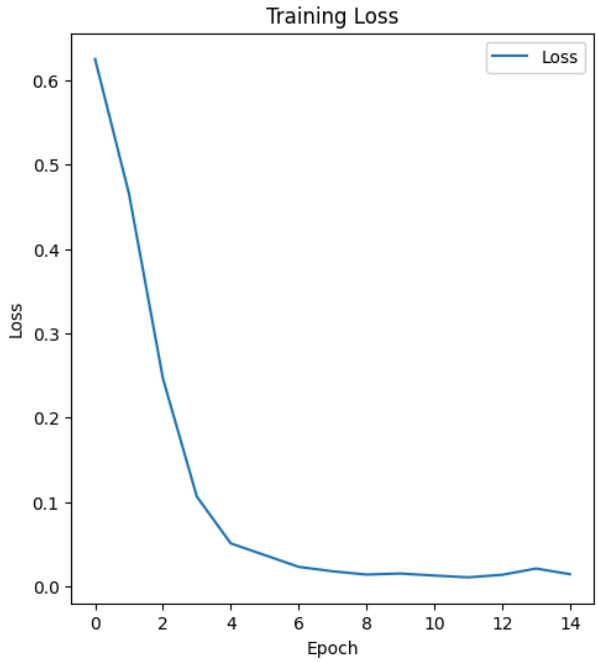
Layer (type)	Output Shape	Param #
embedding_2 (Embedding)	(None, 501, 32)	1280000
gru_4 (GRU)	(None, 501, 32)	6336
gru_5 (GRU)	(None, 32)	6336
dense_2 (Dense)	(None, 2)	66

Total params: 1292738 (4.93 MB)
Trainable params: 1292738 (4.93 MB)
Non-trainable params: 0 (0.00 Byte)

```
Epoch 1/15
163/163 [==============================] - 25s 130ms/step - loss: 0.6248 - accuracy: 0.6760
Epoch 2/15
163/163 [==============================] - 16s 97ms/step - loss: 0.4653 - accuracy: 0.7792
Epoch 3/15
163/163 [==============================] - 14s 88ms/step - loss: 0.2466 - accuracy: 0.8956
Epoch 4/15
163/163 [==============================] - 10s 58ms/step - loss: 0.1067 - accuracy: 0.9631
Epoch 5/15
163/163 [==============================] - 9s 56ms/step - loss: 0.0512 - accuracy: 0.9838
Epoch 6/15
163/163 [==============================] - 10s 59ms/step - loss: 0.0373 - accuracy: 0.9879
Epoch 7/15
163/163 [==============================] - 7s 42ms/step - loss: 0.0234 - accuracy: 0.9950
Epoch 8/15
163/163 [==============================] - 7s 46ms/step - loss: 0.0181 - accuracy: 0.9962
Epoch 9/15
163/163 [==============================] - 6s 37ms/step - loss: 0.0143 - accuracy: 0.9965
Epoch 10/15
163/163 [==============================] - 7s 45ms/step - loss: 0.0155 - accuracy: 0.9973
Epoch 11/15
163/163 [==============================] - 7s 42ms/step - loss: 0.0131 - accuracy: 0.9971
Epoch 12/15
163/163 [==============================] - 6s 40ms/step - loss: 0.0110 - accuracy: 0.9965
Epoch 13/15
163/163 [==============================] - 7s 40ms/step - loss: 0.0140 - accuracy: 0.9965
Epoch 14/15
163/163 [==============================] - 7s 42ms/step - loss: 0.0214 - accuracy: 0.9937
Epoch 15/15
```

163/163 [==============================] - 6s 36ms/step - loss: 0.0147 - accuracy: 0.9958





Accuracy: 66.25%

IV. Results and Analysis

1. **LSTM-GCDC**

   Multi-Class Classification LSTM Model:
   Dataset: Yelp and Enron.
   Training Performance: High accuracy of 98.14%.
   Testing Performance: Much lower at 37.75%, indicating overfitting.

   Binary Classification LSTM Model with Minimum Similarity:
   Dataset: Yelp and Enron, with additional preprocessing for binary labels and similarity scores.
   Training Performance: Very high accuracy of 99.70%.
   Testing Performance: Also shows overfitting with accuracy at 57.50%.

2. **GRU-GCDC**
    a. Standard GRU Model (gru_model):
        i. Accuracy: 34.00%
        ii. Loss: Improved over 20 epochs, indicating good learning.
        iii. Issue: The final accuracy is relatively low, which might suggest overfitting, insufficient model complexity, or issues with the training data.
    b. Binary Classification GRU Model (binary_gru_model):
        i. Accuracy: 63.25%
        ii. Loss: Decreased consistently over 15 epochs, which is a good sign of learning.
        iii. Note: Better performance compared to the standard GRU model, possibly due to the binary nature simplifying the task.
    c. GRU Model with Adjusted Input Length (model with similarity scores):
        i. Accuracy: 66.25%
        ii. Loss: Consistent improvement over 15 epochs.
        iii. Improvement: using similarity measurement improved accuracy

3. **GRU_WikiCNN**

- Without using any similarity parameter: approx 63,75%
- Using minimum similarity parameter: approx 62.92%

4.RNN_GCDC

Model RNN (Multi-Class): Achieved very high training accuracy (up to 99%) but a low test accuracy of 38%, indicating overfitting.
Model RNN_B (Binary Classification): Also showed high training accuracy but a test accuracy of 55%, again suggesting overfitting.
Model RNN_C (Binary with Additional Features): Performed slightly better with a test accuracy of 60.5%, yet still displayed a gap between training and testing results.
General Observations: All models exhibit signs of overfitting, as their high training accuracies did not translate well to the testing phase. This implies a need for

strategies to improve model generalization, such as enhanced regularization or more robust validation methods.

V. Observations

- Using Minimum Similarity as a parameter, we observe a significant rise of around 2-5% in the accuracy. This is in line with the definition of coherence that demands a level of similarity in the flow of the text.
- While varying the test data in the GCDC corpus on the LSTM Binary classifier with minimum similarity as a parameter we observe that the Enron dataset has the most accuracy, which implies that it probably has the most closed domain, while Yahoo has the most open domain.
- Expanding our dataset by including some test data into training increased the accuracy of our model by 6-7%. Using a much larger dataset, such as the Wikipedia CNN corpus (100x larger than GCDC) showed an even larger increase in the accuracy.