INTERNATIONAL INSTITUTE OF
INFORMATION TECHNOLOGY

INTRO TO NLP - CS7.401.S23

NATURAL LANGUAGE INFERENCE

Team-12

Submitted By :
Ardhendu Banerjee - 2022201005
Mayank Gupta - 2022201012
Aman Motwani - 2022201077

Course Instructor :
Prof. Manish Shrivastava

ABSTRACT

- NLI is an important task in NLP that involves determining the relationship between two texts:
  - **Premise**
  - **Hypothesis**.
- The premise provides context for the hypothesis, which may or may not be true based on the premise.
- The goal is to determine if the hypothesis **entailed**, **contradicted**, or is **neutral** with respect to the premise.

DATASET OVERVIEW

SNLI Dataset

- ➢ 570152 English sentence pairs labeled with entailment, neutral, and contradiction categories.
- ➢ The SNLI dataset is based on the image captions from the Flickr30k corpus, where the image captions are used as premises. The hypothesis was created manually by the Mechanical Turk workers.

MultiNLI Dataset

- ➢ 400000 English sentence pairs labeled with entailment, contradiction, or neutral categories.
- ➢ These sentences are sourced from various genres of written and spoken text, including government reports, fiction, and telephone conversations.

EDA AND DATA PRE-PROCESSING

- The data was organized into dataframes, and relevant columns were selected.
- Entries labeled as '-' were excluded from the training, development, and testing datasets.
- Any rows with missing values were removed.
- Unnecessary punctuation was also removed.
- The gold labels, which indicate the truth value of each entry, were converted to numerical values where 0 represents entailment, 1 represents neutrality, and 2 represents contradiction.

APPROACHES

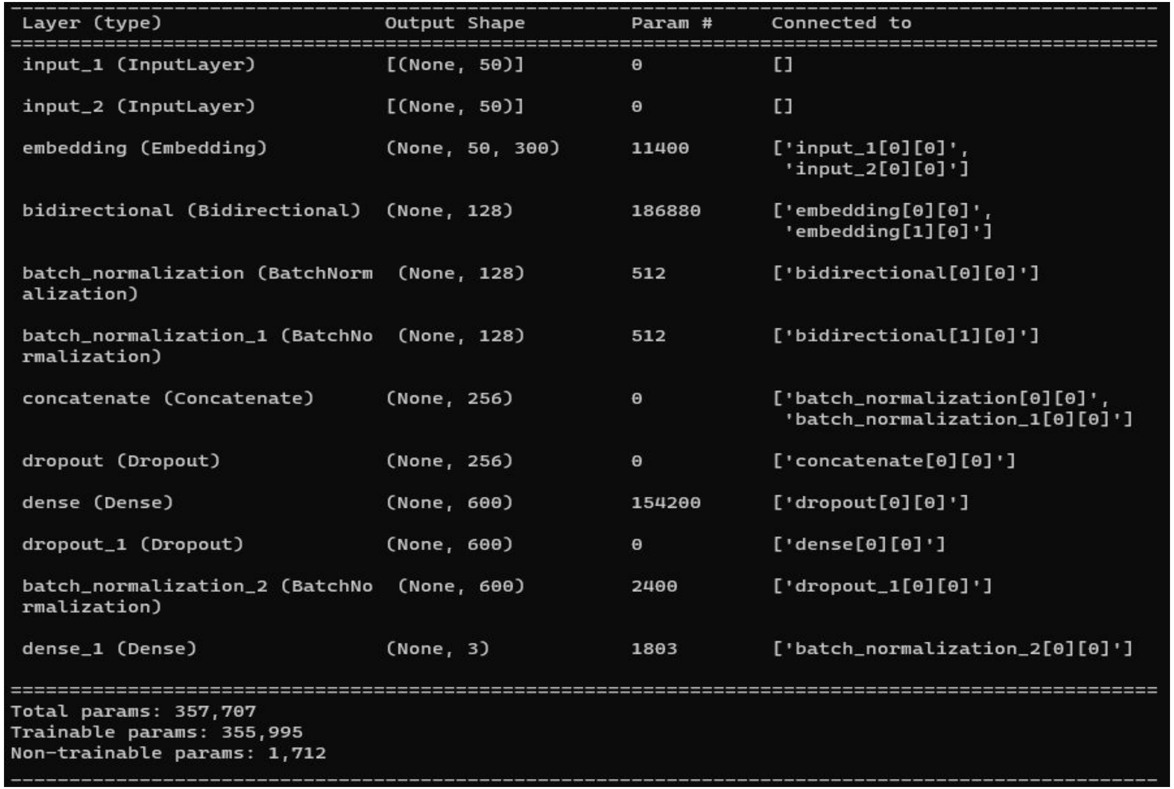
1. Logistic Regression
2. LSTM-based Neural Network with character-level embeddings
3. LSTM-based Neural Network with word-level embeddings
4. Transformer based Model with Pre-trained Transformer(BERT)

LOGISTIC REGRESSION

- We used logistic regression model from the sklearn library.
- Input : Concatenated feature vectors of premise and hypothesis.
- Output : one of three classes (entailment (0), neutral (1), and contradiction (2)).
- For creating feature vectors of premise and hypothesis, we used TfidfVectorizer from sklearn library.

LSTM with Character Level Embeddings

MODEL SUMMARY



```
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_1 (InputLayer)           [(None, 50)]         0           []

 input_2 (InputLayer)           [(None, 50)]         0           []

 embedding (Embedding)          (None, 50, 300)      11400       ['input_1[0][0]',
                                                                  'input_2[0][0]']

 bidirectional (Bidirectional)  (None, 128)          186880      ['embedding[0][0]',
                                                                  'embedding[1][0]']

 batch_normalization (BatchNorm  (None, 128)         512         ['bidirectional[0][0]']
 alization)

 batch_normalization_1 (BatchNo  (None, 128)         512         ['bidirectional[1][0]']
 rmalization)

 concatenate (Concatenate)      (None, 256)          0           ['batch_normalization[0][0]',
                                                                  'batch_normalization_1[0][0]']

 dropout (Dropout)              (None, 256)          0           ['concatenate[0][0]']

 dense (Dense)                  (None, 600)          154200      ['dropout[0][0]']

 dropout_1 (Dropout)            (None, 600)          0           ['dense[0][0]']

 batch_normalization_2 (BatchNo  (None, 600)         2400        ['dropout_1[0][0]']
 rmalization)

 dense_1 (Dense)                (None, 3)            1803        ['batch_normalization_2[0][0]']

==================================================================================================
Total params: 357,707
Trainable params: 355,995
Non-trainable params: 1,712
__________________________________________________________________________________________________
```

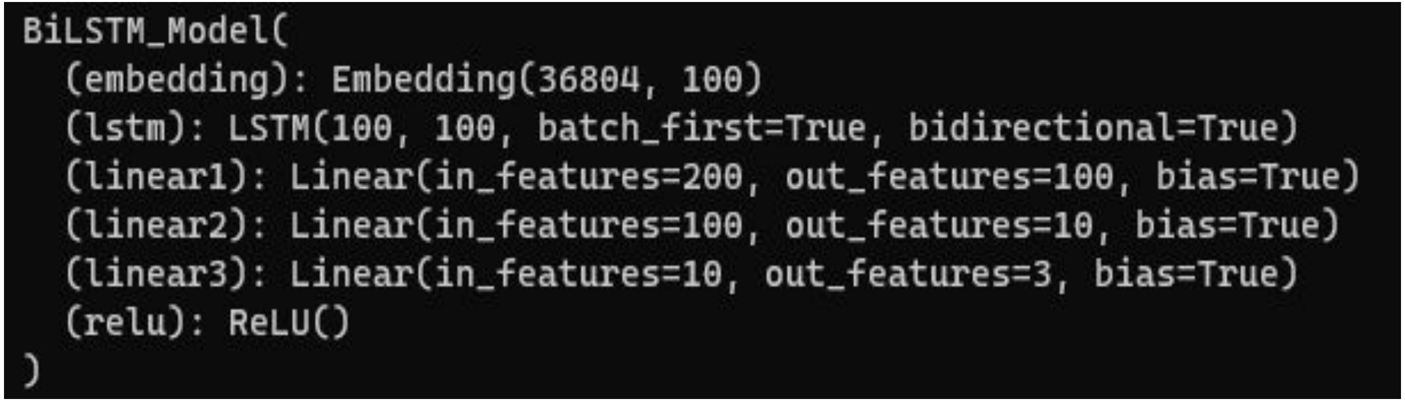
LSTM with Character Level Embeddings

Modeling of Layers

- The text sentences was transformed into integers at the character level and fed separately to the embedding layer.
- BiLSTM layer with ReLU activation functions was used to process the embedded data.
- Batch Normalization was applied to normalize the values, and the concatenated outputs were passed through a series of Linear, Dropout, and Normalization layers.
- A final Linear Layer with softmax function was applied to obtain the output for the 3 labels.

LSTM with Word Level Embeddings

MODEL SUMMARY



```
BiLSTM_Model(
  (embedding): Embedding(36804, 100)
  (lstm): LSTM(100, 100, batch_first=True, bidirectional=True)
  (linear1): Linear(in_features=200, out_features=100, bias=True)
  (linear2): Linear(in_features=100, out_features=10, bias=True)
  (linear3): Linear(in_features=10, out_features=3, bias=True)
  (relu): ReLU()
)
```

LSTM with Word Level Embeddings

Modeling of Layers

- ➢ The sentences are first converted into indexes using a word index dictionary and fed to the model as integer arrays.
- ➢ The model's first layer is an Embedding layer that tries to get embeddings of the words from the input sentences.
- ➢ The output of the embedding layer for both the premise and hypothesis is passed through a Bi-directional LSTM layer individually and concatenated.
- ➢ Finally, the concatenated output is passed through a series of 3 Linear layers, reducing the data dimension gradually to the size of the number of labels (3).

Transformer Based BERT Model

Motivation

- Transformers are effective at processing sequential data such as text.
- The self-attention mechanism in transformers enables them to selectively attend to various segments of the input sequence, allowing them to capture long-range dependencies and perform effectively in tasks that require contextual comprehension of words and phrases.
- Pre-trained BERT model captures contextual dependencies in text, and fine-tuning it for specific NLP tasks such as sentiment analysis or question answering leads to significant performance improvements, enabling new applications previously challenging.

Transformer Based BERT Model

Modeling of Layers

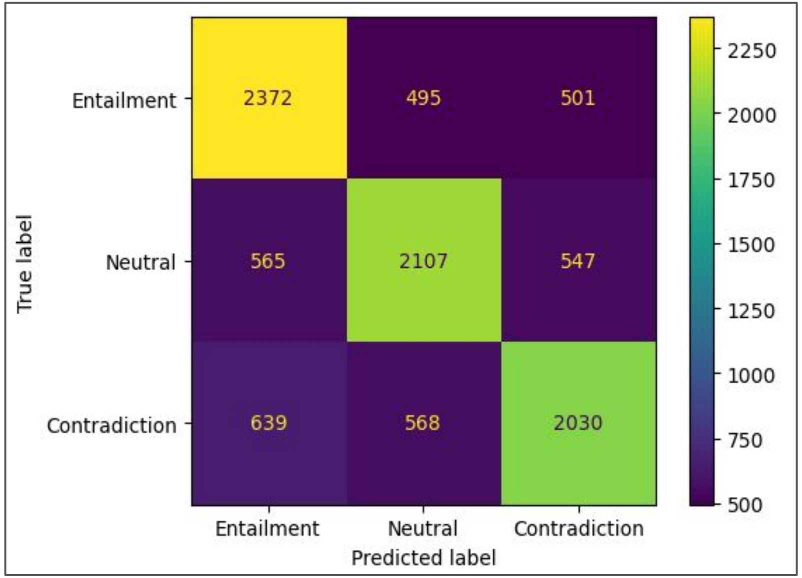
- The BERT model is used to generate a vector from sentences, which is transformed using linear layers.
- The model is fine-tuned with attention masks and token sequences. Sentences are padded to the same length, and the token sequence differentiates between the premise and hypothesis.
- The model takes in integer format padded sentences, attention masks, and token sequences, returning a [hidden dim] vector.
- This is used by a linear layer to predict the class of the input.

RESULTS

Test Accuracy On Different Models/Datasets

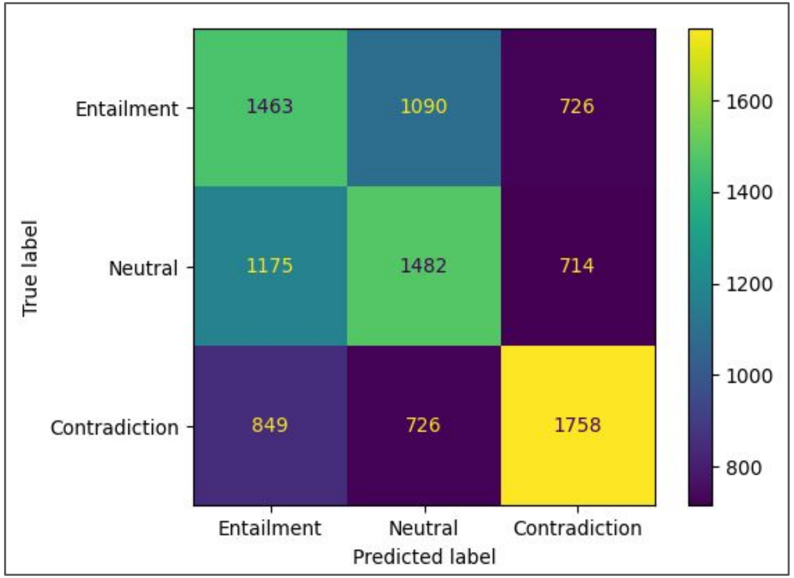
Models	SNLI Dataset	MultiNLI Dataset
Logistic Regression	66.26	47.11
BiLSTM (Char embedding)	70.62	54.94
BiLSTM (Word embedding)	74.07	58.67
Transformer - BERT	82.10	90.16

Confusion Matrix for Logistic Regression

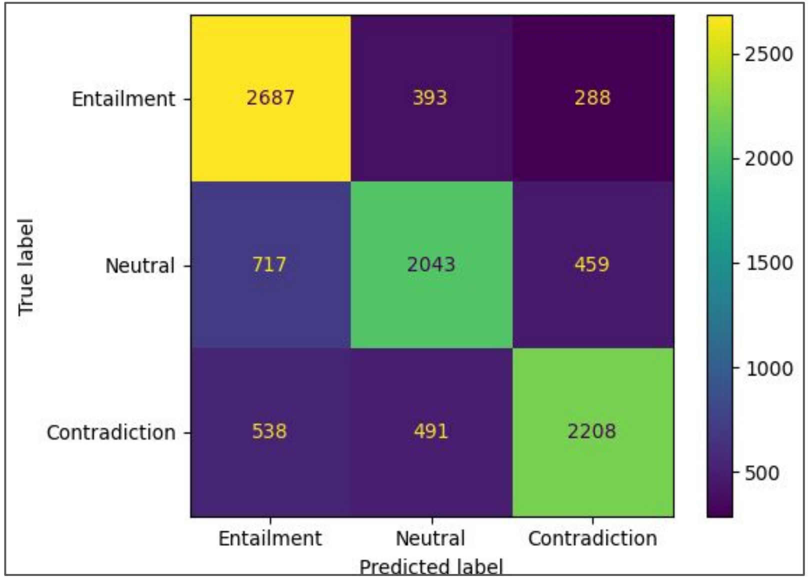


SNLI Dataset

MultiNLI Dataset

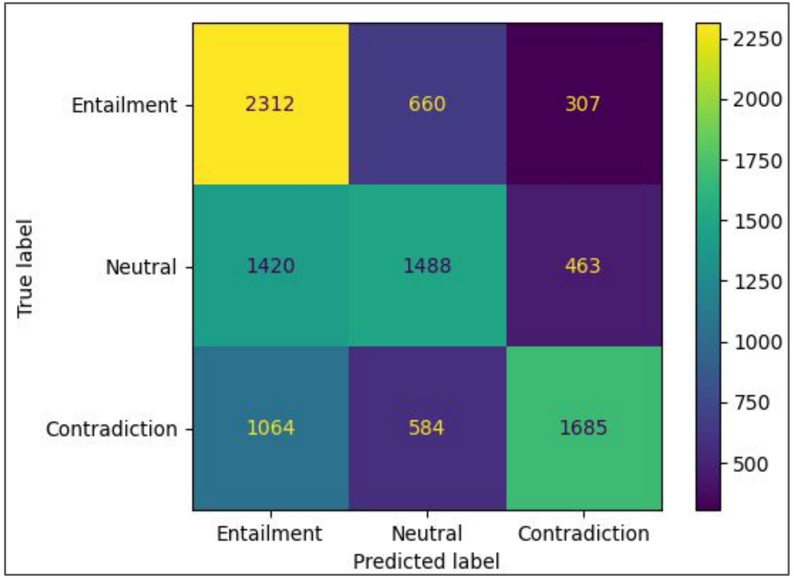


Confusion Matrix for LSTM (Char Level Embeddings)

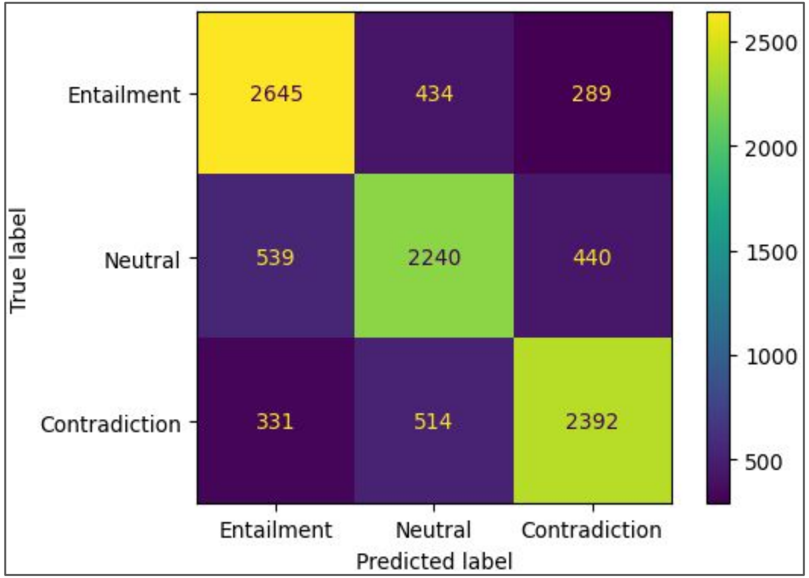


SNLI Dataset

MultiNLI Dataset

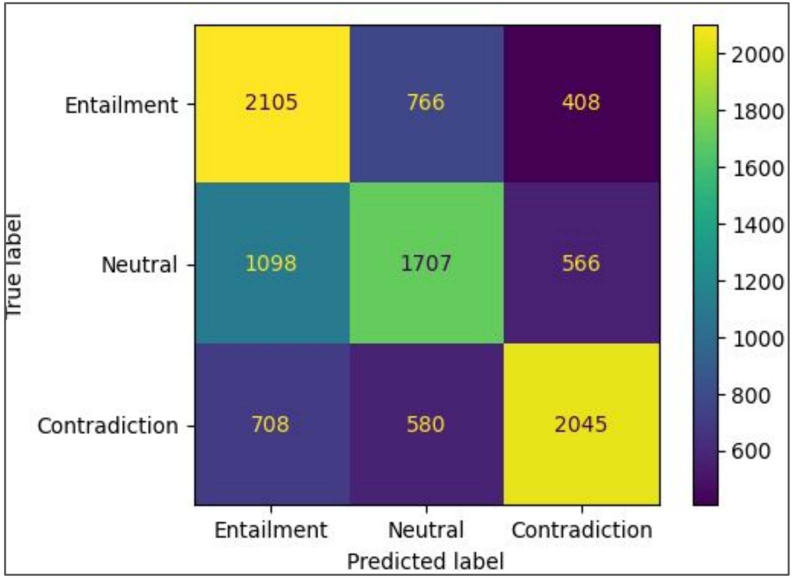


Confusion Matrix for LSTM (Word Level Embeddings)

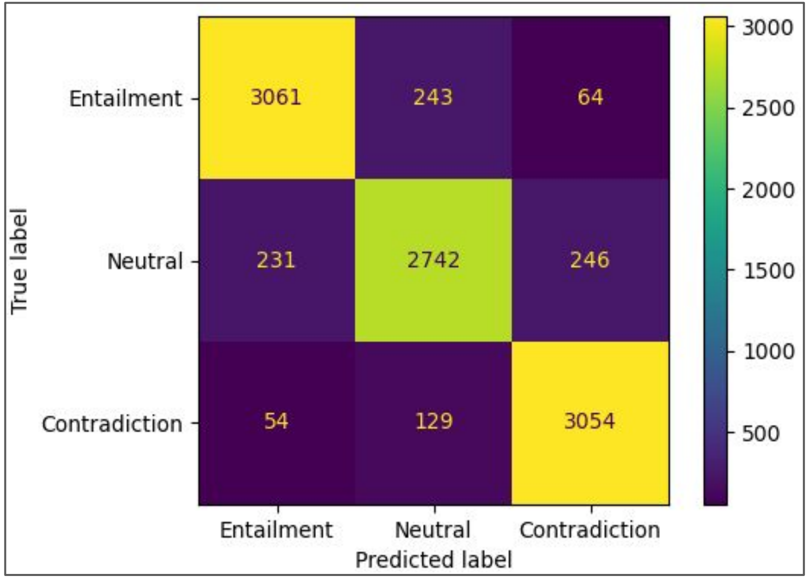


SNLI Dataset

MultiNLI Dataset

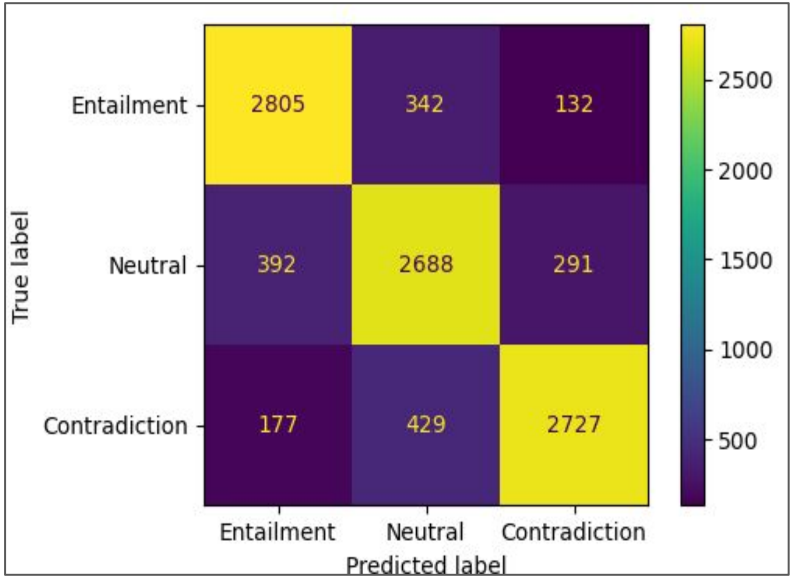


Confusion Matrix for Transformer Based BERT Model

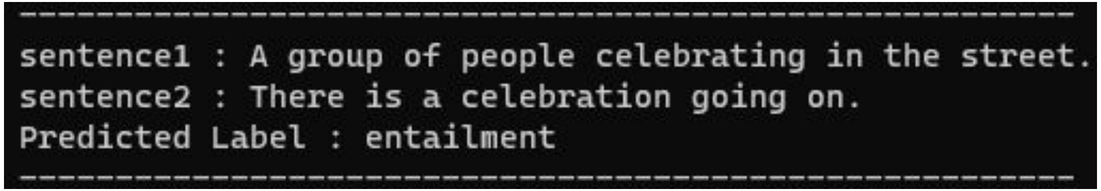


SNLI Dataset

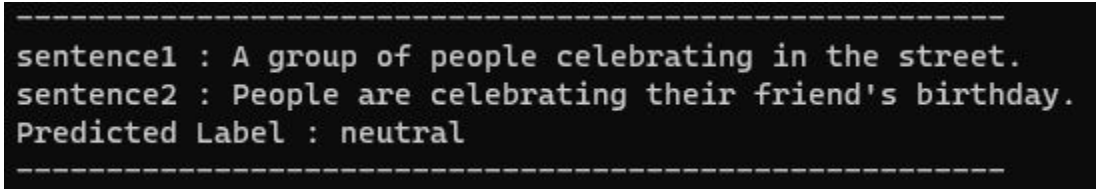
MultiNLI Dataset



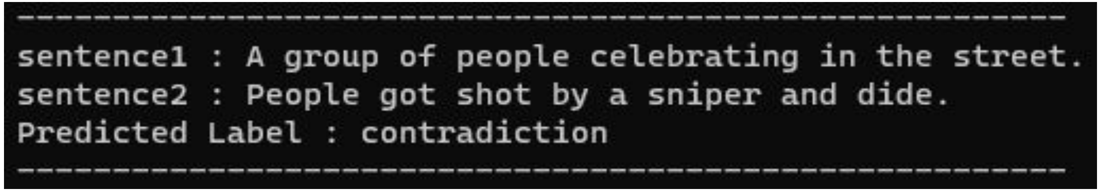
Inference Results on BERT Model (SNLI Dataset)



```
------------------------------------------------------------------
sentence1 : A group of people celebrating in the street.
sentence2 : There is a celebration going on.
Predicted Label : entailment
------------------------------------------------------------------
```

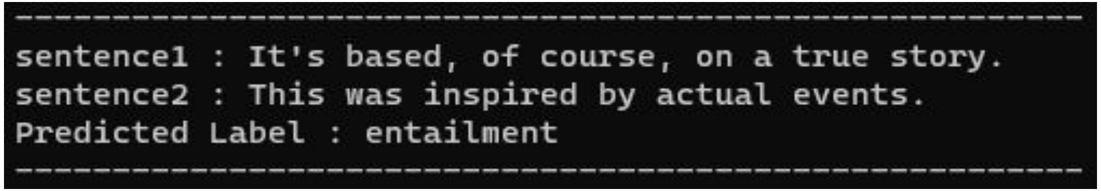


```
-------------------------------------------------------------
sentence1 : A group of people celebrating in the street.
sentence2 : People are celebrating their friend's birthday.
Predicted Label : neutral
-------------------------------------------------------------
```

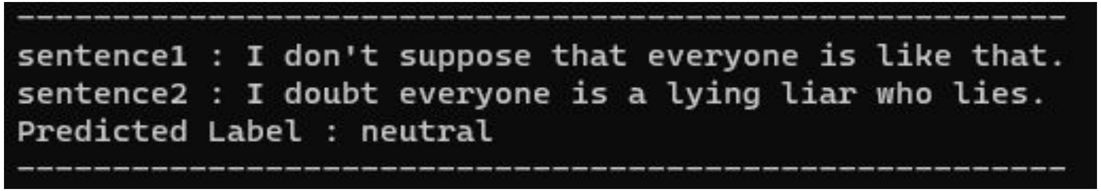


```
-----------------------------------------------------------------
sentence1 : A group of people celebrating in the street.
sentence2 : People got shot by a sniper and dide.
Predicted Label : contradiction
-----------------------------------------------------------------
```

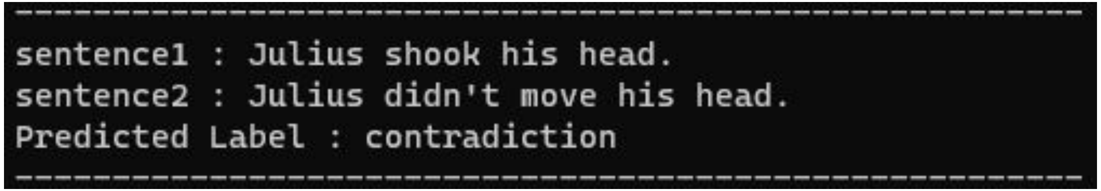
Inference Results on BERT Model (MultiNLI Dataset)



```
-------------------------------------------------------
sentence1 : It's based, of course, on a true story.
sentence2 : This was inspired by actual events.
Predicted Label : entailment
-------------------------------------------------------
```



```
------------------------------------------------------
sentence1 : I don't suppose that everyone is like that.
sentence2 : I doubt everyone is a lying liar who lies.
Predicted Label : neutral
------------------------------------------------------
```



```
-------------------------------------------------------
sentence1 : Julius shook his head.
sentence2 : Julius didn't move his head.
Predicted Label : contradiction
-------------------------------------------------------
```

Observations and Conclusion

- **Logistic Regression :** It depends on input features, which Tfidf may not fully capture for text data. However, it can still achieve significant accuracy, as demonstrated by a 66% accuracy rate in the SNLI dataset.
- **LSTM (Char Level Embedding) :** Using LSTM with character-level embeddings has significantly improved model performance by capturing sentence context. With 300K+ trainable parameters, it outperforms simple logistic regression.

Observations and Conclusion

- **LSTM (Word Level Embedding) :** By employing word embedding, we notice an improvement of 4% in accuracy for both corpora when compared to character embedding. This is due to a slightly better preservation of context.
- **Transformer Bases BERT Model :** It outperforms previous models as it capture the most amount of context. Self-attention in the BERT transformer is the reason for this significant increase in performance.

Thank You