INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY

HYDERABAD

Natural Language Inference

Final Report

COURSE CODE: INTRO TO NLP - CS7.401.S23

Course Instructor:
Prof. Manish Shrivastava

Team No: 12

Project Representatives:
Ardhendu Banerjee - 2022201005
Mayank Gupta - 2022201012
Aman Motwani - 2022201077

Academic year:
2022 - 2023

Contents

1 ABSTRACT

Natural Language Inference (NLI) is a vital task in Natural Language Processing (NLP) that involves determining the relationship between two pieces of text: the premise and the hypothesis. The premise provides the context or background for the hypothesis, which is a statement that may or may not be true based on the premise. The objective of NLI is to determine whether the hypothesis entailed, contradicted, or is neutral with respect to the premise.

NLI is a fundamental task in NLP and has a wide range of applications, including text classification, question answering, machine translation, and dialogue systems. Moreover, NLI can be leveraged to enhance the performance of other NLP tasks like sentiment analysis and information retrieval.

To start with the problem, we would be playing around with various approaches which come in handy in solving the problem. We would be exploring the datasets available to us, and try to formulate and understand what kinds of approaches or Machine Learning methodologies are best suited for this kind of problem-solving. Starting with very basic models, and would be going through the different research that had been done in this field, will also try out the latest model architectures in the field of NLP.

2 DATA OVERVIEW

2.1 SNLI Dataset

SNLI corpus is a benchmark dataset for natural language inference (NLI) tasks. It contains 570,152 English sentence pairs labeled with entailment, contradiction, or neutral categories, covering a wide range of topics and genres. The corpus was created to promote research and development in NLP and provide a standardized benchmark for evaluating NLI models. The dataset has been widely used in numerous studies, including deep learning models like neural networks and transformer-based models such as BERT. The SNLI corpus is publicly available for research purposes and has helped advance the field of NLP by providing a standardized dataset for evaluating and comparing NLI models.

2.2 MultiNLI Dataset

MultiNLI is a comprehensive dataset comprising more than 400,000 sentence pairs, each of which is categorized as either entailment, contradiction, or neutral. These sentences are sourced from various genres of written and spoken text, including government reports, fiction, and telephone conversations. The diverse range of genres is designed to offer a representative sample of the kinds of language that people encounter in their daily lives. The dataset is split into three subsets: matched, mismatched, and tested. The matched subset is intended for training and validation, while the mismatched subset is used to evaluate the generalization ability of Natural Language Inference (NLI) models. The test subset is used for final evaluation and is only released with labels for official evaluation. The sentences in MultiNLI are pre-processed and tokenized.

3 DATA PREPROCESSING

A preliminary EDA and data pre-processing have been conducted on the SNLI dataset, involving the following steps:

- The dataset was imported into dataframes, and the necessary columns were extracted from it.
- Entries with the label '-' were removed from all train, dev, and test datasets.
- Rows with NaN values were dropped from the data.
- Unwanted punctuation was eliminated from the dataset
- The gold labels were converted into numerical representation as follows : **(0 - *entailment*, 1 - *neutral*, 2 - *contradiction*)**

4 PROJECT IMPLEMENTATION

4.1 Logistic Regression

The primal objective of this problem is to classify a given pair of sentences (premise and hypothesis) into three classes (entailment, neutral, and contradiction). The root problem is a classification problem, so the initial set of experiments starts to form a Logistic Regression model approach.

To create feature vectors for the sentences, we have chosen to use TfidfVectorizer from the sklearn library. In this approach, we initially use the training data set, which includes both sentence1 and sentence2, to train the TfidfVectorizer to prepare the weights. Subsequently, we perform vectorization of each sentence separately, which produces a fixed-size vector for each sentence. This vector representation of the sentences carries the tfidf values for each of the words present in the sentences. As we know the tfidf generated higher values for those words which have a higher significance for the words which is more important or relevant. And for the less relevant terms which are very frequently occurring in all sentences to be approaching zero.

The vector shape of the vectorized set of all premise/hypothesis sentences is (549361, 33227), where each sentence has been transformed into a high-dimensional vector of length 33227. The first dimension 549361 indicates the total number of sentences in the training dataset. The second number 33227 represents the vocabulary size or the number of unique words identified by the TfidfVectorizer used to vectorize the sentences.

The data preparation process for feeding into a Logisitic Regressor involved concatenating the vectors of the two sentences in each entry. The resulting data was then used to train a Logisitic Regression Model from the sklearn library.

4.1.1 Experiments & Results with SNLI Dataset

Figure 1: Train Accuracy with Logistic Regression on SNLI Dataset

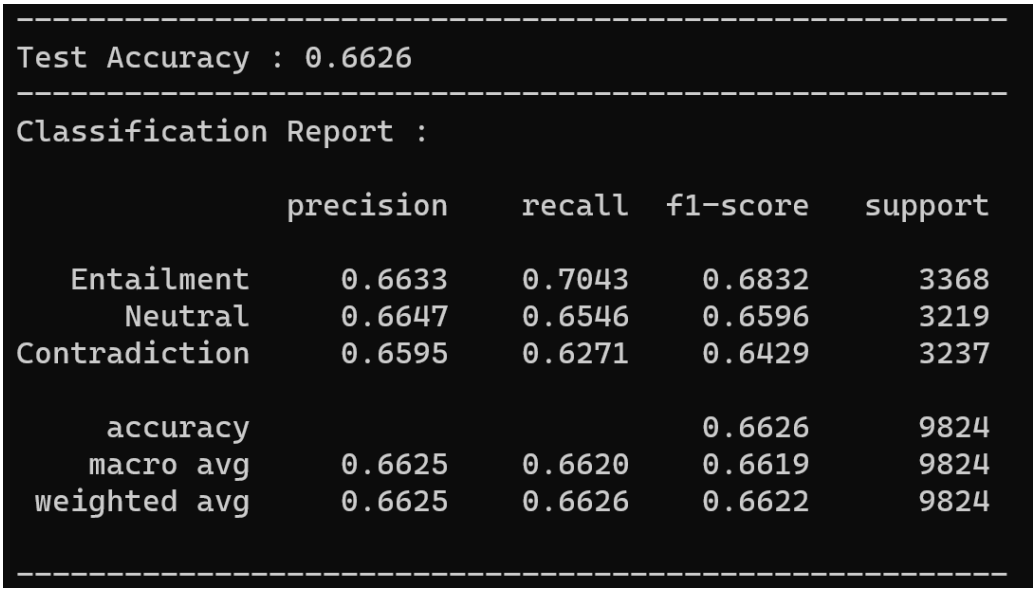


Figure 2: Test Accuracy with Logistic Regression on SNLI Dataset

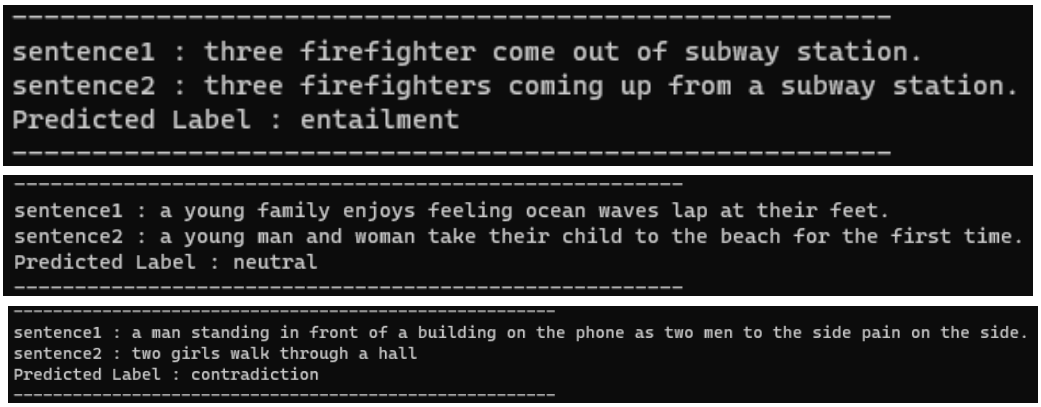


Figure 3: Inference on Input sentence with Logistic Regression on SNLI Dataset

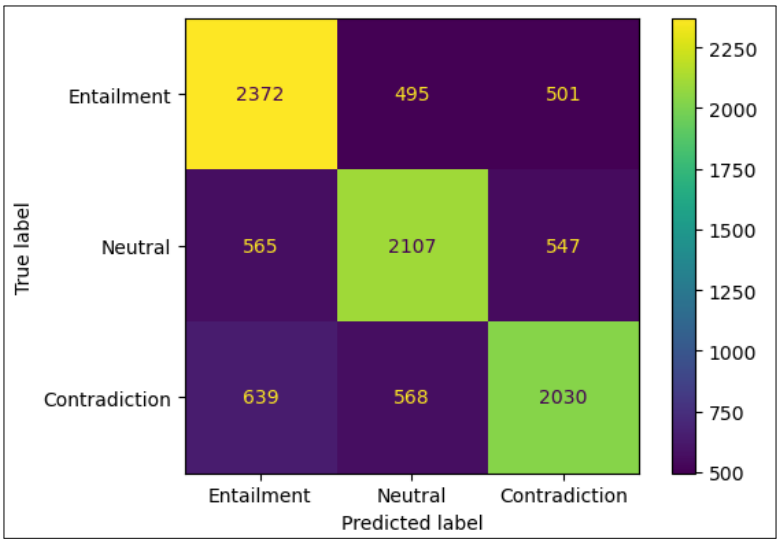


Figure 4: Confusion Matrix with Logistic Regression on SNLI Dataset

4.1.2 Experiments & Results with MultiNLI Dataset

```
-----------------------------------------------------
Train Accuracy : 0.6222
-----------------------------------------------------
```

Figure 5: Train Accuracy with Logistic Regression on MultiNLI Dataset

```
-----------------------------------------------------
Test Accuracy : 0.5867
-----------------------------------------------------
Classification Report :

              precision    recall  f1-score   support

   Entailment     0.5382    0.6420    0.5855      3279
      Neutral     0.5591    0.5064    0.5314      3371
Contradiction     0.6774    0.6136    0.6439      3333

     accuracy                         0.5867      9983
    macro avg     0.5916    0.5873    0.5870      9983
 weighted avg     0.5917    0.5867    0.5868      9983

-----------------------------------------------------
```

Figure 6: Test Accuracy with Logistic Regression on MultiNLI Dataset

```
----------------------------------------------------------
sentence1 : three firefighter come out of subway station.
sentence2 : three firefighters coming up from a subway station.
Predicted Label : entailment
----------------------------------------------------------
```

```
--------------------------------------------------
sentence1 : a young family enjoys feeling ocean waves lap at their feet.
sentence2 : a young man and woman take their child to the beach for the first time.
Predicted Label : neutral
--------------------------------------------------
```

```
----------------------------------------------------------
sentence1 : three firefighter come out of subway station.
sentence2 : three firefighters playing cards inside a fire station.
Predicted Label : contradiction
----------------------------------------------------------
```

Figure 7: Inference on Input sentence with Logistic Regression on MultiNLI Dataset

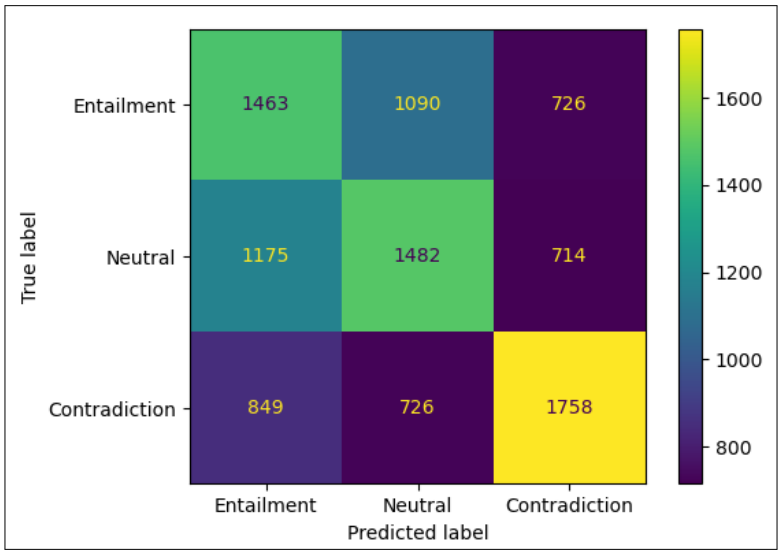


Figure 8: Confusion Matrix with Logistic Regression on MultiNLI Dataset

4.1.3 Observation

Logistic Regression is a machine learning model that heavily relies on the input features provided to it. In the case of text data, vectorization using Tfidf may not be sufficient to capture the contextual meaning of sentences, as it only considers the frequency of individual words.

From this experiment, we have found that a very simple model such as a Logistic Regression has been of some significance as we were able to achieve 66% accuracy in the SNLI dataset.

Our next approaches would be focused on better context-capturing techniques, to be able to do a better classification.

4.2 LSTM-based Neural Network with character level embeddings

LSTM, or Long Short-Term Memory, is a type of neural network architecture that excels in processing sequential data, such as text. Unlike other traditional neural networks that process inputs independently, LSTM models process each input in a sequence one by one and store the output of each input as a hidden state to learn patterns and dependencies within the sequence.

LSTM models excel at capturing the complex connections and dependencies between words in a sequence. They possess the unique capability to retain information from previous inputs, enabling them to comprehend the context and significance of each word in the sequence. This exceptional feature of LSTM models makes them a valuable tool for enhancing the representation of data, surpassing what is achievable through TF-IDF.

To create an LSTM-based neural network model, we utilized certain available layers :

1. **Embedding Layer :** We utilized the embedding layer in our LSTM-based neural network model to convert words or characters into vectors, which enables the model to better represent sentence information. In our experiment, we pre-calculated the weights of the embedding layer before training the model, which means that the layer was not trained during the training process.

2. **Dropout Layer :** we incorporated a dropout layer to selectively eliminate some information during each interaction while training, thus allowing the essential information to be retained better by the model. We utilized a dropout layer with a dropout rate of 0.1 in our experiment.

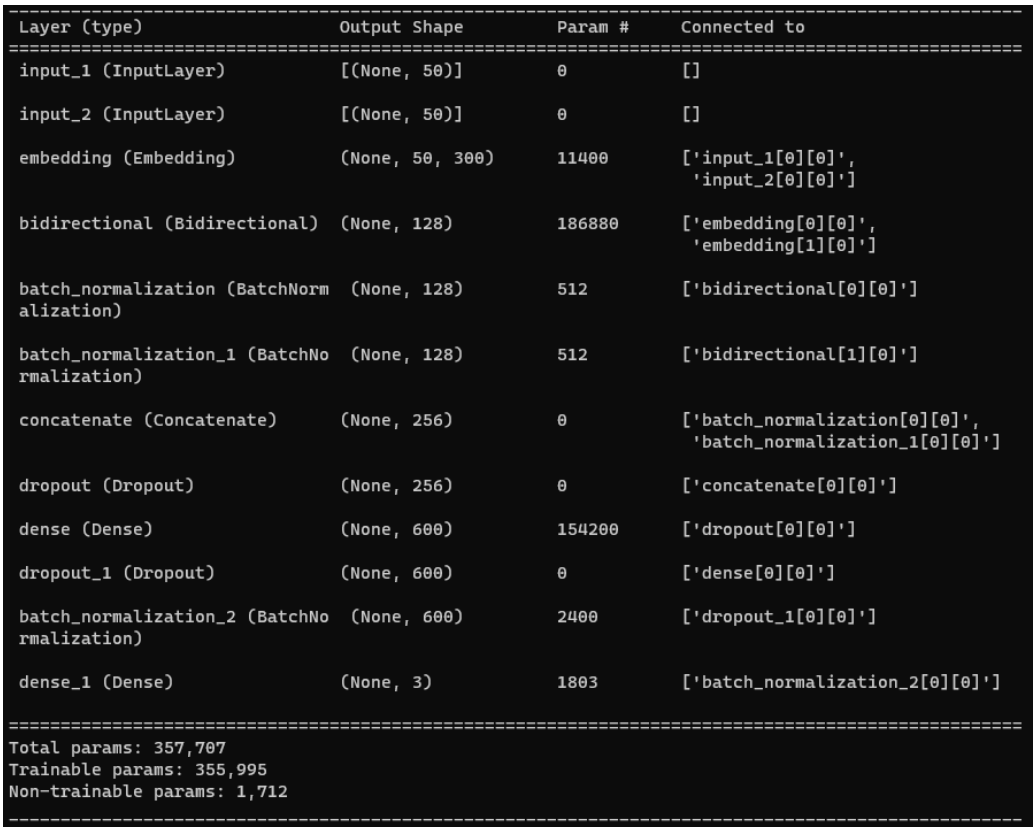
3. **LSTM Layer :** The LSTM layer in our LSTM-based neural network model is designed with a forget gate and an input gate, which enables the model to retain some of the previous information while learning new information. To enhance the learning process further, we utilized a variant of the LSTM layer called BiLSTM, which reads the data from both ends and improves the contextual understanding of the sentence. We chose to use BiLSTM in our experiment as the contextual information of the sentence is not only valid in a forward direction but also in reverse. By processing the data in reverse as well, we are able to preserve the same contextual information.

4. **Linear Layer :** The linear layer within a neural network is responsible for learning the weights of different segments that will contribute to the final output of the model. These weights are trained by the model to contribute to the final values associated with the three classes (entailed, contradicted, or neutral) in our experiment.

5. **BatchNorm1d :** BatchNorm1d is a type of layer that is commonly used in deep learning models to normalize the input data. Its purpose is to minimize the impact of changing input values during the training process, which can make it difficult for the model to learn new weights. By calculating the mean and variance of the input data and applying normalization, BatchNorm1d helps prevent covariate shifts, leading to improved model performance.

4.2.1 Modeling of Layers

1. The sentences were vectorized at the character level by identifying unique characters and converting them to integers using a dictionary.
2. Weights for these integers were obtained from a pre-trained word vectorization model (Glove) and used in the Embedding layer of the LSTM-based Neural Network Model.
3. The integer-based sentence data from the premise and hypothesis were fed separately to the embedding layers.
4. The output from the embedding layers was then passed through a BiLSTM layer with ReLU activation functions to eliminate negative weights.
5. The data from the BiLSTM layers are passed through a BatchNormalization layer, which returns the normalized set of values.
6. The outputs from both BatchNormalization were concatenated and fed to a series of Linear, Dropout, and Normalization layers sequentially.
7. Finally a Linear Layer transforming the data into a dimension of the number of labels and softmax function was applied to the 3 dimensions corresponding to the 3 labels to obtain the final output.



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 (BatchNormalization)	(None, 128)	512	['bidirectional[0][0]']
batch_normalization_1 (BatchNormalization)	(None, 128)	512	['bidirectional[1][0]']
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 (BatchNormalization)	(None, 600)	2400	['dropout_1[0][0]']
dense_1 (Dense)	(None, 3)	1803	['batch_normalization_2[0][0]']

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

Figure 9: BiLSTM Model with Character level Embeddings

4.2.2 Experiments & Results with SNLI Dataset

```
--------------------------------------------------------
Test Accuracy : 0.7062
--------------------------------------------------------
Classification Report :

              precision    recall  f1-score   support

   Entailment     0.6816    0.7978    0.7352      3368
      Neutral     0.6980    0.6347    0.6648      3219
Contradiction     0.7472    0.6821    0.7132      3237

     accuracy                         0.7062      9824
    macro avg     0.7089    0.7049    0.7044      9824
 weighted avg     0.7086    0.7062    0.7049      9824

--------------------------------------------------------
```

Figure 10: Test Accuracy with BiLSTM(Char Level) on SNLI Dataset

```
--------------------------------------------------------
sentence1 : Three girls blow out the candles of a cake made of Peeps.
sentence2 : There are three girls and a cake.
Predicted Label : entailment
--------------------------------------------------------
--------------------------------------------------
sentence1 : A guy riding a motorcycle near junk cars
sentence2 : The man is test driving a motorcycle to decide whether or not he will buy it.
Predicted Label : neutral
--------------------------------------------------
--------------------------------------------------------
sentence1 : A woman is painting a mural of a woman's face.
sentence2 : There is a man tying his shoes.
Predicted Label : contradiction
--------------------------------------------------------
```

Figure 11: Inference on Input sentence with BiLSTM(Char Level on SNLI Dataset

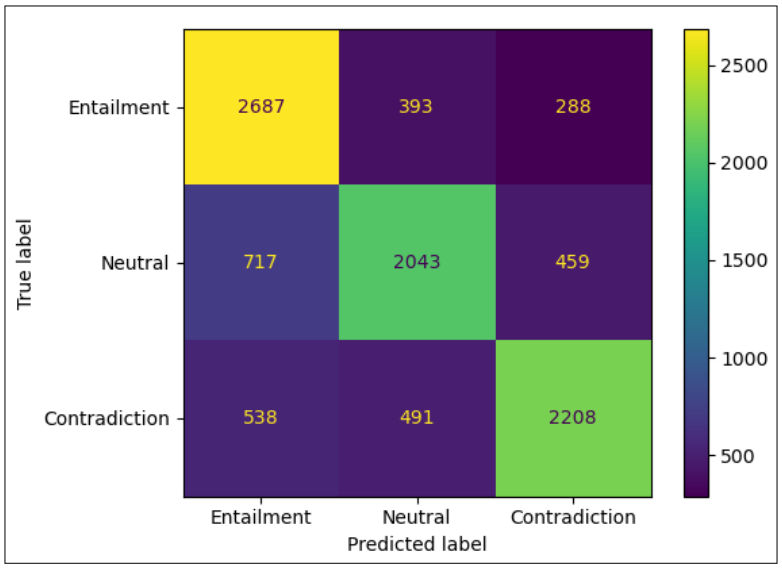


Figure 12: Confusion Matrix with BiLSTM(Char Level) on SNLI Dataset

4.2.3 Experiments & Results with MultiNLI Dataset

```
----------------------------------------------------------
Test Accuracy : 0.5494
----------------------------------------------------------
Classification Report :

              precision    recall  f1-score   support

   Entailment     0.4821    0.7051    0.5726      3279
      Neutral     0.5447    0.4414    0.4876      3371
Contradiction     0.6864    0.5056    0.5822      3333

     accuracy                         0.5494      9983
    macro avg     0.5710    0.5507    0.5475      9983
 weighted avg     0.5714    0.5494    0.5471      9983

----------------------------------------------------------
```

Figure 13: Test Accuracy with BiLSTM(Char Level) on MultiNLI Dataset

```
----------------------------------------------------------
sentence1 : but maybe i'll try it one day
sentence2 : Perhaps in the future I will try it.
Predicted Label : entailment
----------------------------------------------------------
```

```
----------------------------------------------------------
sentence1 : thank you it's nice to have someone understand that
sentence2 : I have had a hard time explaining this to other people.
Predicted Label : neutral
----------------------------------------------------------
```

```
----------------------------------------------------------
sentence1 : There are other differences as well.
sentence2 : No differences are known.
Predicted Label : contradiction
----------------------------------------------------------
```

Figure 14: Inference on Input sentence with BiLSTM(Char Level) on MultiNLI Dataset

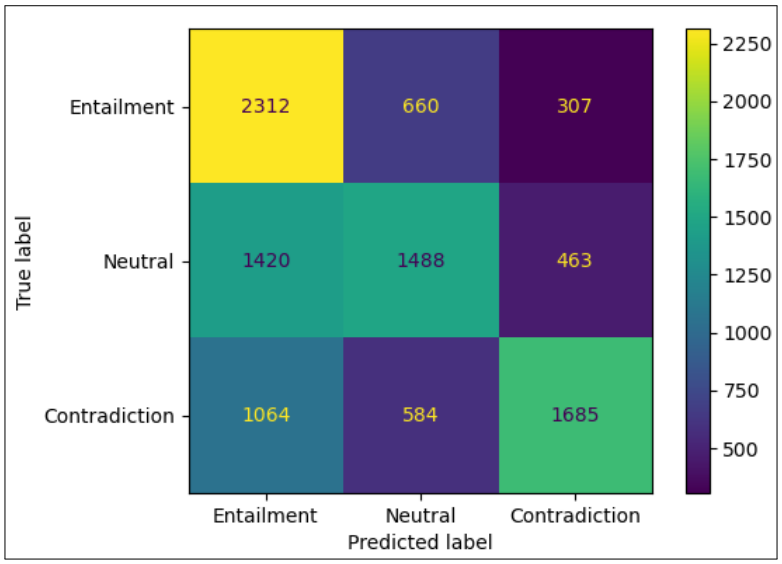


Figure 15: Confusion Matrix with BiLSTM(Char Level) on MultiNLI Dataset

4.2.4 Observation

From the results, we can conclude that the use of LSTM with a combination of a character-level embeddings has a significant improvement in the performance of the model, where it has been successful in capturing some of the contexts of the sentences. With over 300K trainable parameters in the model, the model has proved to be better than the simple Logistic Regression model.

4.3 LSTM-based Neural Network with word-level embeddings

In the previous model, the language model was based on character-level embeddings. We found that some of the contexts were surely preserved in the model, as there was a significant increase in the performance of the model. Now to improve the preservation of the context, we would be building out embeddings over words instead of characters. As done prior we would be using an embedding layer and a BiLSTM layer trying to learn the context of the sentences in a Bidirectional manner.

4.3.1 Modeling of Layers

1. The sentences are initially converted into indexes based on a word index dictionary.
2. Following this the premise and the hypothesis are fed to the model in the form of an integer array.
3. The first layer of the model consists of an Embedding layer, which is trainable, and this tries to get embeddings of the words from the input sentences.
4. The output of the embedding layer for both the premise ad the hypothesis is then traversed through the Bi-directional LSTM layer individually.
5. The results from the LSTM layer for both the premise and the hypothesis are then concatenated.
6. The concatenated results are then passed through a series of 3 Linear layers serially which reduces the dimension of the data gradually to the size of the number of labels i.e. 3.

```
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()
)
```

Figure 16: BiLSTM Model with Word level Embeddings

4.3.2 Experiments & Results with SNLI Dataset

```
--------------------------------------------------------
Test Accuracy : 0.7062
--------------------------------------------------------
Classification Report :

              precision    recall  f1-score   support

   Entailment     0.6816    0.7978    0.7352      3368
      Neutral     0.6980    0.6347    0.6648      3219
Contradiction     0.7472    0.6821    0.7132      3237

     accuracy                         0.7062      9824
    macro avg     0.7089    0.7049    0.7044      9824
 weighted avg     0.7086    0.7062    0.7049      9824

--------------------------------------------------------
```

Figure 17: Test Accuracy with BiLSTM(Word Level) on SNLI Dataset

```
--------------------------------------------------
sentence1 : A statue at a museum that no seems to be looking at.
sentence2 : There is a statue that not many people seem to be interested in.
Predicted Label : entailment
--------------------------------------------------
--------------------------------------------------
sentence1 : A man playing an electric guitar on stage.
sentence2 : A man is performing for cash.
Predicted Label : neutral
--------------------------------------------------
--------------------------------------------------
sentence1 : A man reads the paper in a bar with green lighting.
sentence2 : The man is climbing a mountain.
Predicted Label : contradiction
--------------------------------------------------
```

Figure 18: Inference on Input sentence with BiLSTM(Word Level) on SNLI Dataset

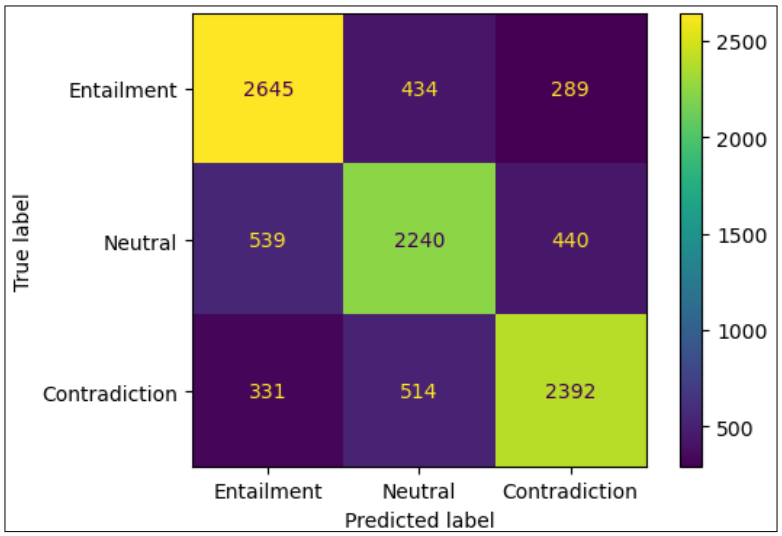


Figure 19: Confusion Matrix with BiLSTM(Word Level) on SNLI Dataset

4.3.3 Experiments & Results with MultiNLI Dataset

```
----------------------------------------------------
Test Accuracy : 0.5494
----------------------------------------------------
Classification Report :

              precision    recall  f1-score   support

   Entailment     0.4821    0.7051    0.5726      3279
      Neutral     0.5447    0.4414    0.4876      3371
Contradiction     0.6864    0.5056    0.5822      3333

     accuracy                         0.5494      9983
    macro avg     0.5710    0.5507    0.5475      9983
 weighted avg     0.5714    0.5494    0.5471      9983

----------------------------------------------------
```

Figure 20: Test Accuracy with BiLSTM(Word Level) on MultiNLI Dataset

```
----------------------------------------------------------
sentence1 : Dove Cottage was their first home.
sentence2 : Dove Cottage was the first place that they lived.
Predicted Label : entailment
----------------------------------------------------------
----------------------------------------------------
sentence1 : But a simple solution might be at hand.
sentence2 : There may be a simple solution available that would benefit all involved parties.
Predicted Label : neutral
----------------------------------------------------
----------------------------------------------------------
sentence1 : There are also risks to going it alone.
sentence2 : Going it alone is risk-free.
Predicted Label : contradiction
----------------------------------------------------------
```

Figure 21: Inference on Input sentence with BiLSTM(Word Level) on MultiNLI Dataset

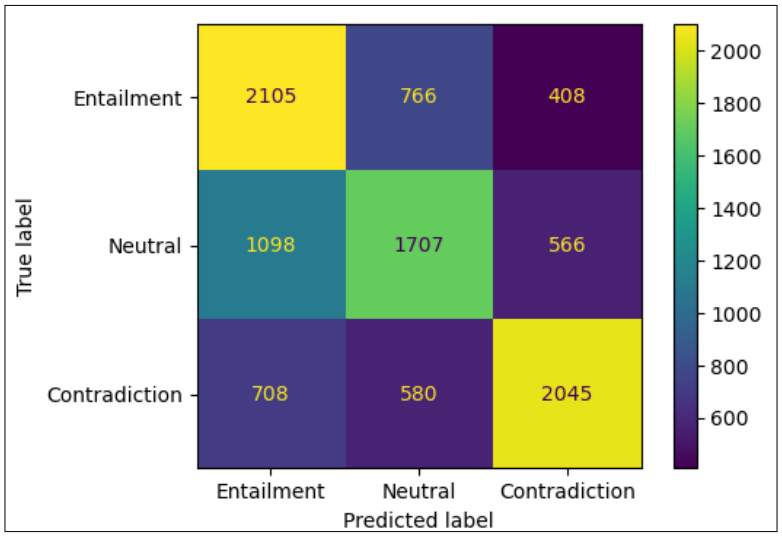


Figure 22: Confusion Matrix with BiLSTM(Word Level) on MultiNLI Dataset

4.4 Transformer based Model with Pre-trained Transformer(BERT)

A transformer is a type of neural network architecture that was introduced in the paper "Attention Is All You Need" by Vaswani et al. (2017). Transformers are designed to process sequential data, such as text, and are particularly effective for tasks such as language modeling, machine translation, and text classification.

Transformers utilize "self-attention", a mechanism that enables the network to attend selectively to various segments of the input sequence while processing it. This mechanism allows the transformer to capture dependencies that span long distances within the input sequence, making it an efficient tool for tasks that require contextual comprehension of words and phrases.

In real-world applications, transformers are frequently used as the foundation of pre-trained language models, which can subsequently be fine-tuned to tackle specific NLP tasks such as sentiment analysis or question answering. The use of pre-trained transformers has resulted in significant performance improvements across various NLP tasks and has opened up new possibilities for applications that were previously thought to be challenging.

BERT is a transformers model pre-trained on a large corpus of English data in a self-supervised fashion. This means it was pre-trained on the raw texts only, with no humans labeling them in any way (which is why it can use lots of publicly available data) with an automatic process to generate inputs and labels from those texts.

4.4.1 Data Pre-Processing

- The idea is to use the BERT model to return us a vector with the sentences as input, which we will be using with Linear layers to transform it to the dimension of the number of layers. For this purpose of this, we are using a pre-trained tokenizer for BERT to tokenize both sentences and then convert the words in the sentences into integers.
- Next we will need to fine-tune the BERT model with the type of data/sentences that we will be testing out the model on. The model will need an attention mask and a token sequence along with the sentences.
- As the model will be trained in batches, all the sentences will be padded to the same length based on the max length of the batch. The attention mask for the padded part of the sentences is given as '0', and for the actual part of the data the mask is kept as '1'. This will provide support to the model to focus on the actual content of the sentences and have no attention to the padded data.
- The token sequence is differentiated as the separation of the premise and the hypothesis. The token sequence for the premise part is kept to be '0' and the rest of the sentence including the padded part has the token sequence be '1'. This will help the model differences between the two parts of the sentence.

Followed by the data pre-processing the BERT model is fed 3 things, the data of the padded sentences in integer format, the attention mask, and the token sequence. The model returns a vector of [hidden dim] length.

This vector is then used by a Linear Layer to change the dimension to the number of classes i.e. 3. This gives us the prediction of the actual class of the given input of the premise and hypothesis input.

4.4.2 Experiments & Results with SNLI Dataset

```
--------------------------------------------------------
Test Accuracy : 0.9016
--------------------------------------------------------
Classification Report :

              precision    recall  f1-score   support

   Entailment     0.9148    0.9088    0.9118      3368
      Neutral     0.8805    0.8518    0.8659      3219
Contradiction     0.9078    0.9435    0.9253      3237

     accuracy                         0.9016      9824
    macro avg     0.9011    0.9014    0.9010      9824
 weighted avg     0.9013    0.9016    0.9012      9824

--------------------------------------------------------
```

Figure 23: Test Accuracy with BERT Model on 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
--------------------------------------------------------
```

Figure 24: Inference on Input sentence with BERT Model on SNLI Dataset

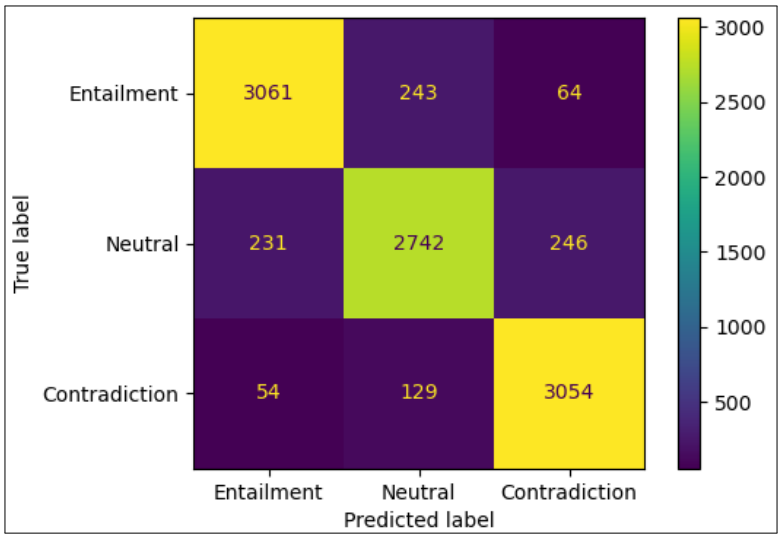


Figure 25: Confusion Matrix with BERT Model on SNLI Dataset

4.4.3 Experiments & Results with MultiNLI Dataset

```
------------------------------------------------------
Test Accuracy : 0.8234
------------------------------------------------------
Classification Report :

              precision    recall  f1-score   support

   Entailment     0.8314    0.8554    0.8432      3279
      Neutral     0.7771    0.7974    0.7871      3371
Contradiction     0.8657    0.8182    0.8413      3333

     accuracy                         0.8234      9983
    macro avg     0.8247    0.8237    0.8239      9983
 weighted avg     0.8245    0.8234    0.8236      9983

------------------------------------------------------
```

Figure 26: Test Accuracy with BERT Model on 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
--------------------------------------------------
```

Figure 27: Inference on Input sentence with BERT Model on MultiNLI Dataset

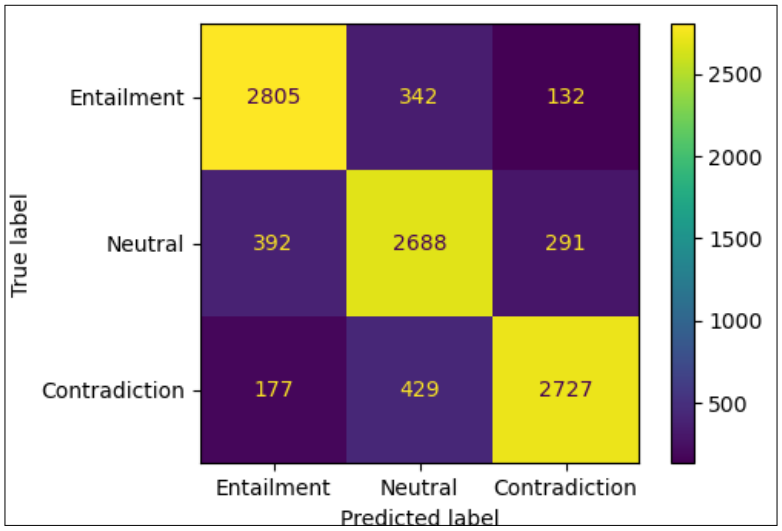


Figure 28: Confusion Matrix with BERT Model on MultiNLI Dataset

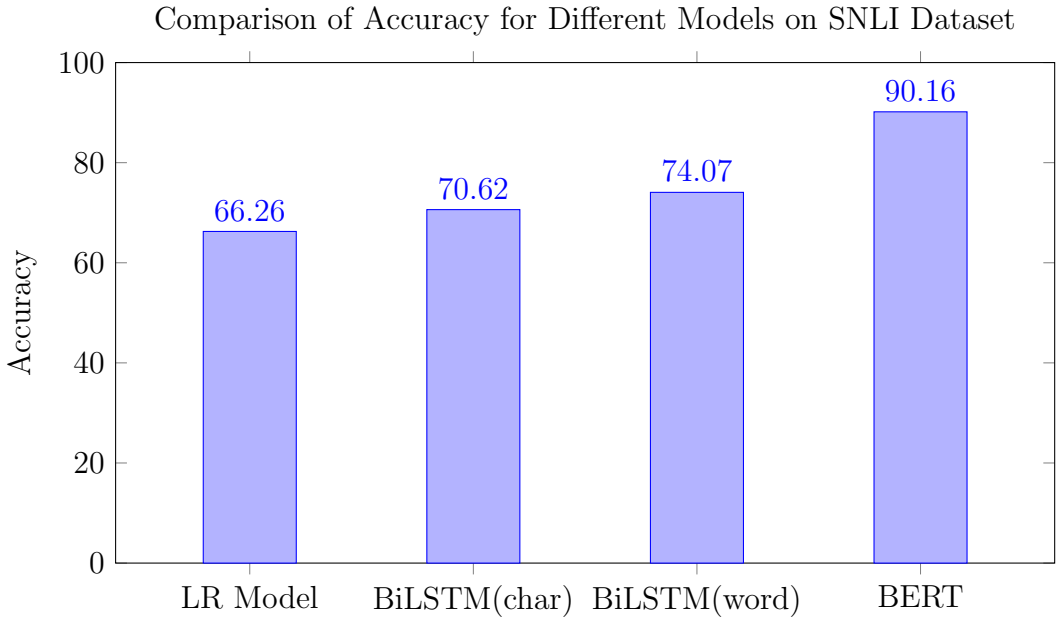
4.4.4 Observation

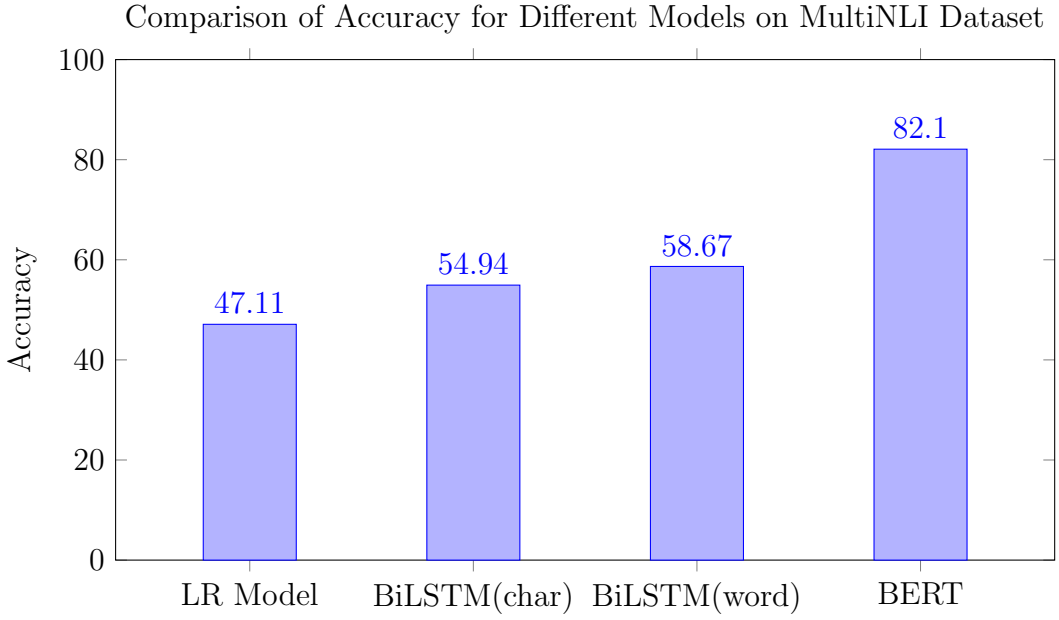
The BERT model which is already proven to capture the most amount of context in terms of distributional semantics here proves to be performing better than any of the previous models that were trying to capture the context of the sentences with the help of BiLSTM layers both with character level embeddings and the word level embeddings. This significant increase in the performance of the model is due to the self-attention property of the BERT transformer.

5 COMPARISON OF RESULTS

Table 1: Accuracy for different Models on different Datasets

Models	Accuracy (SNLI)	Accuracy (MultiNLI)
Logistic Regression Model	66.26	47.11
BiLSTM (character embedding)	70.62	54.94
BiLSTM (word embedding)	74.07	58.67
Transformer based BERT Model	90.16	82.10





6 CONCLUSION

I started this Natural Language Inference task with the basic Logistic Regression model and then proceeded with LSTMs and transformers. We came to an understanding that, although Logistic Regression works well as a classifier model with various kinds of data, it struggles to hold its performance when it comes to language data where the requirement is to learn to understand the context of the sentences.

Again going with LSTMs and BiLSTMs has proven to be having some efficiency where the context-preserving nature of the LSTM architectures has shown results where it was able to preserve the context of the sentences by going through the sentences sequentially both forward and backward.

Finally when it comes to understanding the context in languages the transformer's architecture has been the best research done in this field, wherein its architecture allows it to self-understand the important parts of the sentences and the self-attention provides the right amount of support for their operation. And the use of the transformers architecture along with the pre-trained BERT model has so far proved to be the most effective model to be successfully performing the NLI task.

References

[1] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*, 2015.

[2] Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, and Hui Jiang. Enhanced LSTM for natural language inference. *arXiv preprint arXiv:1609.06038*, 2016.

[3] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*, 2017.

[4] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *CoRR abs/1802.05365*, 2018.

[5] Shuohang Wang and Jing Jiang. Learning natural language inference with LSTM. *arXiv preprint arXiv:1512.08849*, 2015.