Project report on:

Study of ML algorithms with encoding applied to Sequence Classification problem

As part of Natural Language Processing (NLP) subject Coursework

Prepared by:

Name - Sayonesh Chatterjee
University of Surrey

April 26, 2024

TABLE OF CONTENTS

1 INTRODUCTION

The study in this report is about analyzing different combinations of machine learning algorithms and encoding techniques applied to sequence classification problem. The machine learning algorithms in study scope included Random Forest, Artificial Neural Network, Convolutional Neural Network, and encoding techniques spacy and word2vec. In the end, the analysis section proposes the best combination outcome (considering highest accuracy) from the experiments carried out.

The approach for building the sequence classifier involves the following steps:

- Tokens are taken as inputs and fed to the spacy library to convert them into embeddings to be provided to machine learning algorithms.
- Subsequently, these algorithms provide predictions on the validation and test data.
- Finally, we measure the F1 score and analyze confusion matrix to evaluate the performance of the model.

2 DATASET ANALYSIS AND VISUALIZATION

Exploratory Data Analysis (EDA) is conducted on the dataset, which involved visualising and understanding the structure and patterns present within the dataset. Some of the visualizations carried out are:

1. The below word cloud shows the most common tokens in the training data. Higher frequency is indicated by the larger size of the words, e.g., ‘cell’ is the highest, followed by ‘using’. This distribution helps us with finding recurrent phrases and patterns within the dataset.

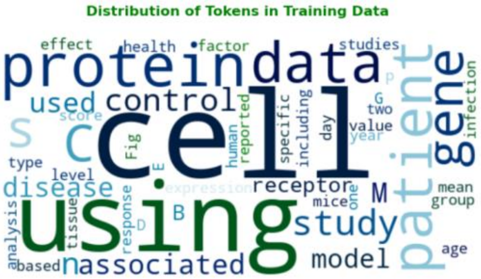


Fig 1: Distribution of Tokens in Training Data

2. The distribution of NER tags in the training data is displayed by the pie chart below. This was useful to learn which entities were most common in the collection, like it can be seen that 'B-O' tags comprise the portion of the area of the pie chart (82.4%), meaning that the dataset is highly imbalanced.

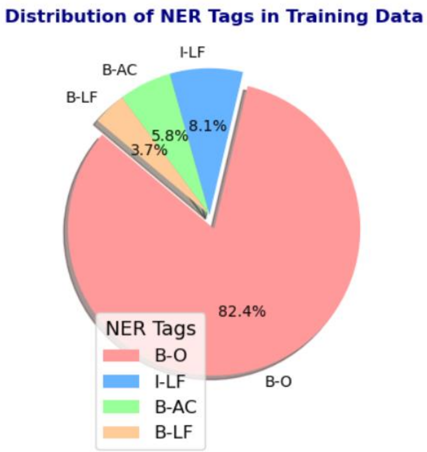


Fig 2: Distribution of NER Tags in Training Data

3. The number of tokens in each dataset is known by examining the below bar charts. Comprehension of the amount of text data that is available for testing, validation and training is also known, like it can be seen that the training data has the largest token counts (40000) followed by the validation and test data at 5000 each.

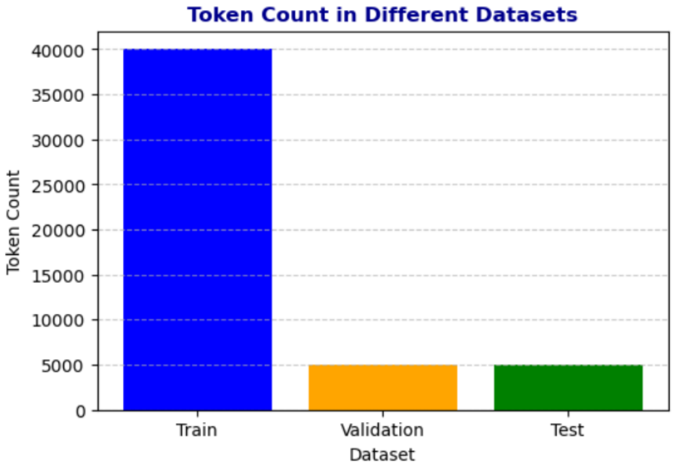


Fig 3: Count of token in different datasets

4. The variety of tokens found in each dataset can be obtained by analysing the below bar charts. It is important to know how many distinct tokens are available during training, validation and testing because it helps to evaluate how rich the dataset is, and the generalisation potential of the models trained using it. In the figure shown below, it can be

seen that validation has the highest number of unique tokens followed by training and testing.

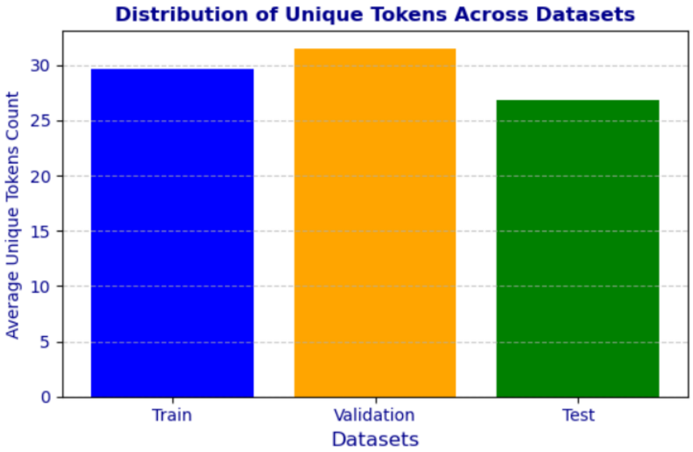


Fig 4: Unique token distribution across datasets

5. The distribution of various NER Tags throughout the datasets is obtained by the examination of the stacked bar chart. The distribution of entities and their frequencies within each dataset is required to assess a dataset's eligibility for NER tags. It can be seen below that the distribution of NER tags is unbalanced, like the 'B-O' tags are present in abundance in the train dataset compared to the validation and testing datasets and the same can be seen for other tags as well.

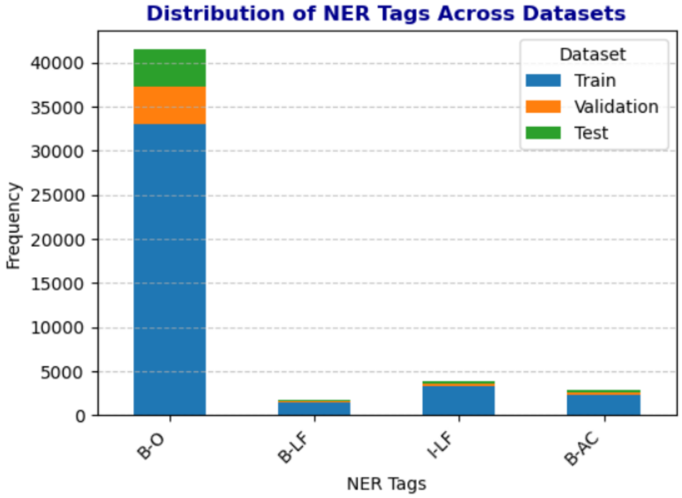


Fig 5: NER tag distribution across datasets

3 EXPERIMENTAL SETUPS

3.1 COMPARING TRADITIONAL ALGORITHMS

- First experiment - using spacy along with decision tree classifier.

  The steps are as follows:

  1. Pre-processing Data:
     a) Tokenization: It refers to dividing text into discrete tokens (words or sub words).
     b) Choice of Tokenization: The tokenizer used here is spacy's tokenizer for language models, which have already been trained, like "en_core_web_sm".
     c) Using Pre-Trained Language Models: Pre-trained language models such as "en_core_web_sm", has been used here for tokenizing a text, making use of spacy's in-built tokenizer.
  2. Text Encoding/Transformation into Numerical Vectors:
     a) Word Embeddings: Words are converted into numerical vectors by text encoding.
     b) Spacy Word Embeddings: Used because that has already been trained with the “en_core_web_sm” model and hence can capture syntactic and semantic information well.
  3. Applying NLP Algorithms/Techniques:
     a) Decision Tree Classifier: Initial performance is evaluated by employing it as a baseline model.
     b) Justification of choice: Simplicity and interpretability is offered, alongside being useful for understanding feature importance.

- Second experiment - spacy along with Support Vector Machine (SVM) classifier.

  The steps are as follows:

  - Pre-processing Data:
    a) Tokenization: It refers to dividing text into discrete tokens (words or sub words).
    b) Choice of Tokenization: The tokenizer used here is spacy's tokenizer for language models, which have already been trained, like "en_core_web_sm".
    c) Using Pre-Trained Language Models: Pre-trained language models such as "en_core_web_sm", has been used here for tokenizing a text, making use of spacy's in-built tokenizer.
  - Text Encoding/Transformation into Numerical Vectors:
    a) Word Embeddings: Words are converted into numerical vectors by text encoding.

b) Spacy Word Embeddings: Used because that has already been trained with the "en_core_web_sm" model and hence can capture syntactic and semantic information well.

- Applying NLP Algorithms/Techniques:

  a) Support Vector Machine (SVM): This classifier was used because of its ability to handle high dimensional data well and also to capture intricate relationships between features.

3.2 COMPARING FEATURES/VECTORIZATION METHODS

➢ First experiment - Word2Vec along with Random Forest classifier.
The steps are as follows:

- Pre-processing data:

  a) Tokenization: The Word2Vec model is trained using tokens from the training dataset.

- Text Encoding/Transformation into Numerical Vectors:

  a) Word2Vec Embeddings: Word2Vec embeddings trained on the training dataset's tokenized data is utilized.

  b) Used because: Semantic links between words in a continuous vector space is recorded and rich representations are produced with the help of Word2Vec embeddings. It also helps to capture contextual data.

- Applying NLP Algorithms/Techniques:

  a) Random Forest Classifier: Used because of its versatility in handling high dimensional data and also ability to capture intricate correlations between attributes.

  b) Used because: An adaptable ensemble learning method that works well in case of classification problems. Also, it is resilient against overfitting and it can handle high-dimensional feature spaces.

➢ Second experiment - spacy along with Random Forest classifier.
The steps are as follows:

- Pre-processing data:

  a) Tokenization: The text is divided into discrete tokens (words or sub words), by applying spacy's pre-trained language model.

- Text Encoding/Transformation into Numerical Vectors:

  a) Word Embeddings: Pre-trained word embeddings of the spacy "en_core_web_sm" model is used.

  b) Used because: Syntactic and semantic information in large volumes of representation of words can be obtained from the text.

- Applying NLP Algorithms/Techniques:

  a) Random Forest Classifier: Chosen because of its ability and efficiency in managing high dimensional data.

b) Used because: Ability to manage high dimensional feature fields. Also, works well in case of classification, due to its ability to find intricate interactions between features and its resiliency to overfitting.

3.3 COMPARING DEEP LEARNING ALGORITHMS

➢ First experiment - spacy is used along with Convolutional Neural Networks (CNNs).
The steps are as follows:

- Pre-processing data:
  a) Tokenization: The data is tokenized into discrete tokens by applying spacy's pre-trained language model.
- Text encoding/Transformation into Numerical Vectors:
  a) Word Embeddings: Using spacy's pre-trained word embeddings, tokens are represented as dense numerical vectors.
  b) Used because: By capturing semantic information, word embeddings yield greater representation of words in the text. Also, they work well related to NER activities.
- Applying NLP Algorithms/Techniques:
  a) Convolutional Neural Network(CNN): CNN was selected as a classifier because of its ability to identify both local and global patterns in sequential data.
  b) Used because: Good at collecting hierarchical patterns in sequential data and hence, they are suited for NER tasks where word context is important.

➢ Second one - spacy is used along with Artificial Neural Networks (ANNs).
The steps are as follows:

- Pre-processing data:
  a) Tokenization: The data is tokenized into discrete tokens by applying spacy's pre-trained language model.
- Text encoding/Transformation into Numerical Vectors:
  a) Word Embeddings: Using spacy's pre-trained word embeddings, tokens are represented as dense numerical vectors.
  b) Used because: By capturing semantic information, word embeddings give rich representation of words. Also, they work well for gathering background data.
- Applying NLP Algorithms/Techniques:
  a) Artificial Neural Networks (ANN): Chosen because of capacity to discover intricate patterns in data.
  b) Used because: Flexible enough to describe intricate interactions between input and output data.

3.4 COMPARING DIFFERENT LOSS FUNCTIONS AND OPTIMIZERS

- <u>First experiment - model is built using spacy and ANN.</u>
  Steps same as Section 3.3. Then, various combinations of loss functions and optimizers are incorporated. The steps remain the same as the one mentioned above ("spacy is used along with Artificial Neural Networks(ANNs)"). The only additional last step is:
    - Choices of loss functions and optimizers: Loss functions used are sparse categorical cross entropy and cross entropy.
    - Along with the above, various combinations of optimizers such as Adam, RMSProp and SGD are used to study model training and performance.
    - By taking combinations of loss functions and optimizers, the combination which maximises the F1 score on both validation and test datasets is selected.

- <u>Second experiment - model is built using spacy and CNN.</u>
  Steps same as Section 3.3. Then, various combinations of loss functions and optimizers are incorporated. The steps remain the same as the one mentioned above ("spacy is used along with Convolutional Neural Networks (CNNs)"). The only additional last step is:
    - Choices of loss functions and optimizers: Loss functions used are sparse categorical cross entropy and cross entropy.
    - Along with the above, various combinations of optimizers such as Adam, RMSProp and SGD are used to study model training and performance.
    - By taking combinations of loss functions and optimizers, the combination which maximises the F1 score on both validation and test datasets is selected.

4 TESTING AND ERROR ANALYSIS

4.1 COMPARING TRADITIONAL ALGORITHMS

- Regarding the first experiment (Section 3.1), the accuracy testing of the decision tree model with spacy is done.
    - F1-score Evaluation:

a) An F1 score of 0.858 is obtained for the validation dataset and it indicates that the model performs well even on unseen data, after the training.
b) An F1 score of 0.843 is obtained for the test dataset, which indicates a robust performance. One reason for this could be that the test dataset has the least number of unique token count (which we get from Exploratory Data Analysis) amongst all the three datasets.

- Error Analysis:

a) From the confusion matrix shown below, we can see that:
Point 1 – 'B-O' is the most accurately predicted label, followed by 'B-AC'. The level of accuracy in predicting 'B-LF' and 'I-LF' labels is the same.
Point 2 – It can also be seen that few labels have been misclassified like 7 labels are predicted as 'B-LF', they are actually 'B-AC' and 13 labels are predicted as 'I-LF', but they are actually 'B-AC'.
Point 3 – This is mainly because the dataset is highly imbalanced, consisting of the largest portion of 'B-O', followed by 'I-LF', 'B-AC' and 'B-LF'.

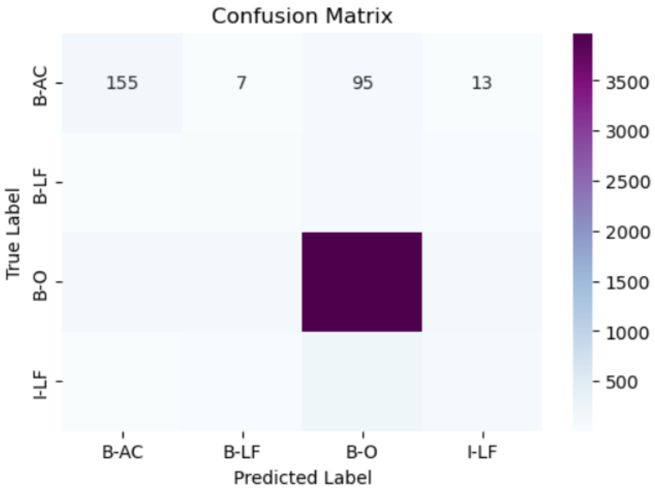


Fig 6: Confusion Matrix

- Receiver Operator Characteristic (ROC) Curve:

a) From the ROC Curve below, it can be seen that:
Point 1- Class 'B-AC' has an area of 0.78, indicating of its ability to perform well in correctly identifying instances of class 'B-AC', while reducing false-positives.
Point 2 – Class 'B-LF' exhibits an area under the curve(AUC) of 0.69, indicating that there is scope for improvement in reducing false-positive rates.
Point 3 – The classes 'B-O' and 'I-LF' show AUC values of 0.73 and 0.70 respectively, which indicates that the model can distinguish these classes from negatives with moderate accuracy.

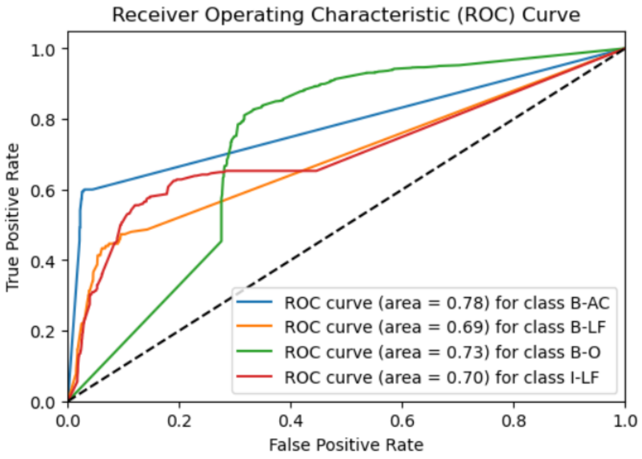


Fig 7: ROC Curve

- Regarding second experiment (Section 3.1), the accuracy testing of the SVM model with spacy is done.
  - F1-score Evaluation:
    a) An F1 score of 0.836 obtained for the validation and an F1 score of 0.839 obtained for the test dataset means that the model is performing reasonably well.

  - Error Analysis:
    a) From the confusion matrix shown below, we can see that:
    Point 1 – 'B-O' is the most accurately predicted label.
    Point 2 – It can also be seen that few labels have been misclassified like 105 labels are predicted as 'B-O', they are actually 'B-AC'.
    Point 3 – This is mainly because the dataset is highly imbalanced, consisting of the largest portion of 'B-O' and also because the SVM algorithm is not that effective in handling complex interactions.

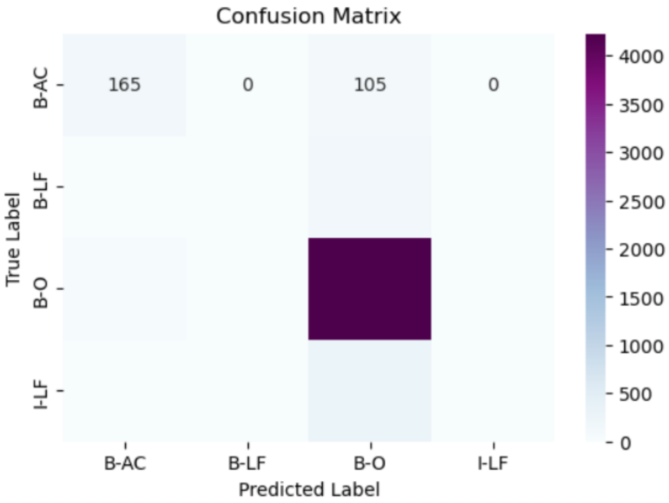


Fig 8: Confusion Matrix

- Receiver Operator Characteristic (ROC) Curve:
  a) From the ROC Curve below, it can be seen that:
  Point 1 – Class ‘B-AC’, with an area of 0.96 indicates that the model has a very high distinguishing power between positive instances of ‘B-AC’ and negative instances.
  Point 2 – For class ‘B-LF’, area of 0.82 suggests good discriminatory power but not as high as ‘B-AC’.
  Point 3 – For class ‘B-O’, area of 0.74 suggests moderate performance to distinguish between positive and negative instances.
  Point 4 – AUC for class ‘I-LF’ (0.68) is not as great compared to the other classes, due to imbalanced data.

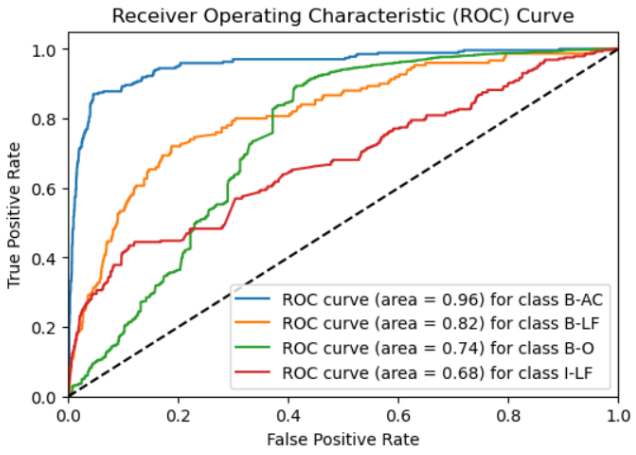


Fig 9: ROC Curve

4.2 COMPARING FEATURES/VECTORIZATION METHODS

- Regarding the first experiment (Section 3.2), the accuracy testing of Word2Vec with Random Forest is done.
  - F1-score Evaluation:
    a) The validation data has an F1 score of 0.769 and the test data has an F1 score of 0.773, indicating an average performance in classifying named entities.
  - Error Analysis:
    a) From the confusion matrix shown below, we can see that:

Point 1 – ‘B-O’ is the most accurately predicted label.
Point 2 – It can also be seen that few labels have been misclassified like 200 labels are predicted as ‘B-O’, they are actually ‘B-AC’.
Point 3 – This is mainly because the dataset is highly imbalanced, consisting of the largest portion of ‘B-O’ and also because of the fact that Word2Vec is not as efficient as spacy in handling linguistic complexities like special characters.

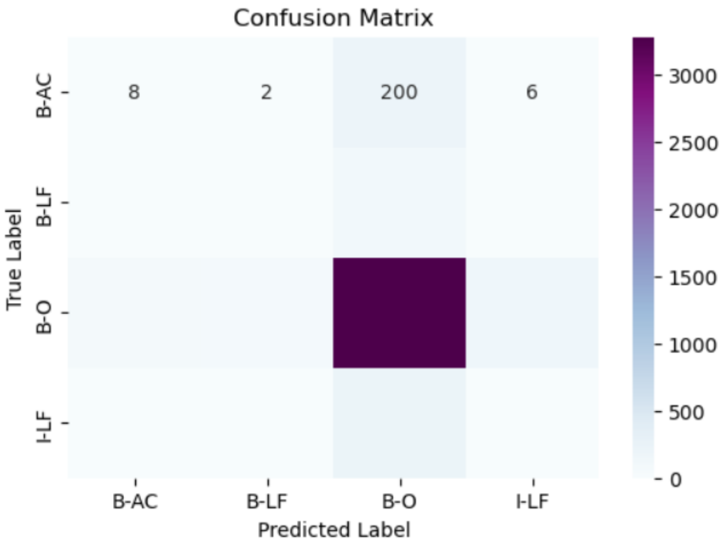


Fig 10: Confusion Matrix

- Receiver Operator Characteristic (ROC) Curve:

a) From the ROC Curve below, it can be seen that:
Point 1 – Class ‘B-AC’, with an area of 0.49 indicates that the model’s ability to classify instances of ‘B-AC’ is not very strong.
Point 2 – For class ‘B-LF’, area of 0.50 suggests that the classifier’s ability to distinguish instances of class ‘B-LF’ is equivalent to random guessing.
Point 3 – For class ‘B-O’, area of 0.47 suggests that the classifier’s ability to classify instances of ‘B-O’ is not effective.
Point 4 – AUC for class ‘I-LF’ (0.47) suggests that the classifier’s ability to classify instances of ‘I-LF’ is not strong, due to imbalanced data.

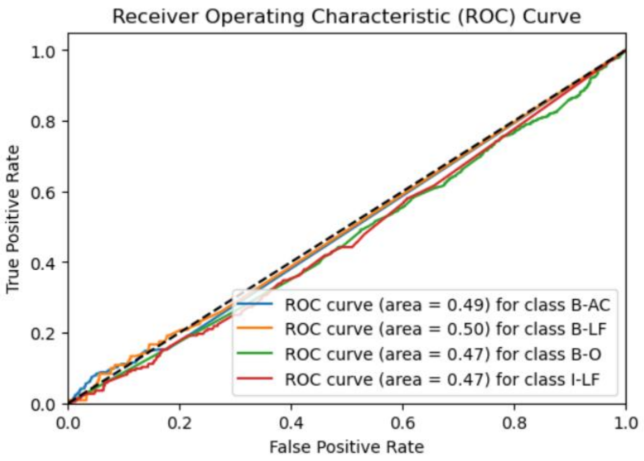


Fig 11: ROC Curve

- Regarding second experiment (Section 3.2), the accuracy testing of spacy with Random Forest is done.
  - F1-score Evaluation:
    a) The validation data has an F1 score of 0.875 and the test data has an F1 score of 0.84, indicating that the model is performing well even on unseen data.

  - Error Analysis:
    a) From the confusion matrix shown below, we can see that:
    Point 1 – 'B-O' is the most accurately predicted label, followed by 'B-AC'.
    Point 2 – It can also be seen that few labels have been misclassified like 127 labels are predicted as 'B-O', they are actually 'B-AC'.
    Point 3 – This is mainly because the dataset is highly imbalanced, consisting of the largest portion of 'B-O'.

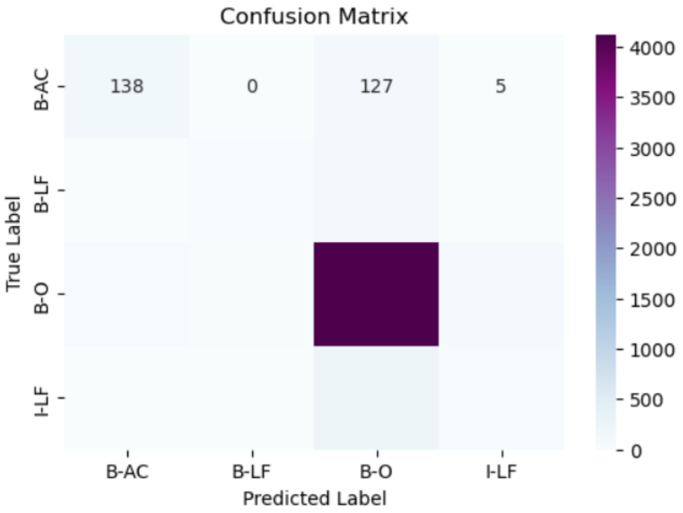


Fig 12: Confusion Matrix

- Receiver Operator Characteristic (ROC) Curve:
  a) From the ROC Curve below, it can be seen that:
  Point 1 – Class ‘B-AC’, with an area of 0.97 indicates that the model has a high true positive rate and a low false positive rate, making it highly reliable.
  Point 2 – For class ‘B-LF’, area of 0.85 suggests that the classifier portrays a relatively high true positive rate and a moderate false positive rate.
  Point 3 – For class ‘B-O’, area of 0.87 suggests that the model has a high true positive rate and a moderate false positive rate.
  Point 4 – AUC for class ‘I-LF’ (0.81) suggests that the model shows a moderate true-positive rate and a relatively high false-positive rate.

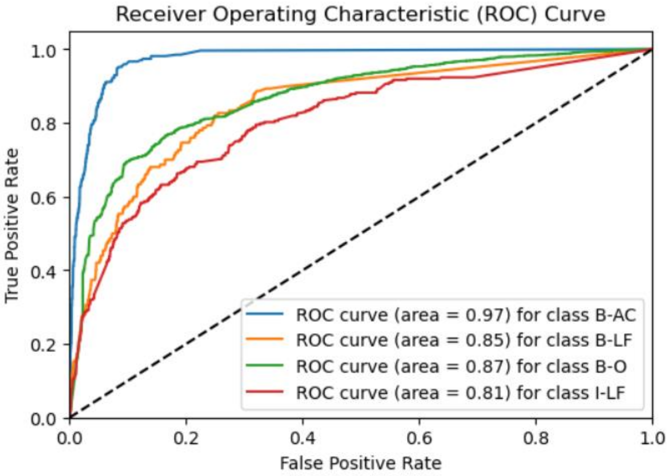


Fig 13: ROC Curve

4.3 COMPARING DEEP LEARNING ALGORITHMS

- Regarding first experiment (Section 3.3), the accuracy testing of spacy with CNN is done.
  - F1-score Evaluation:
    a) The validation data has an F1 score of 0.845 and the test data has an F1 score of 0.846, indicating that the model does well on unseen data.

  - Error Analysis:
    a) From the confusion matrix shown below, we can see that:
    Point 1 – ‘B-O’ is the most accurately predicted label, followed by ‘B-AC’ and ‘I-LF’.
    Point 2 – It can also be seen that few labels have been misclassified like 115

labels are predicted as ‘B-O’, they are actually ‘B-LF’.

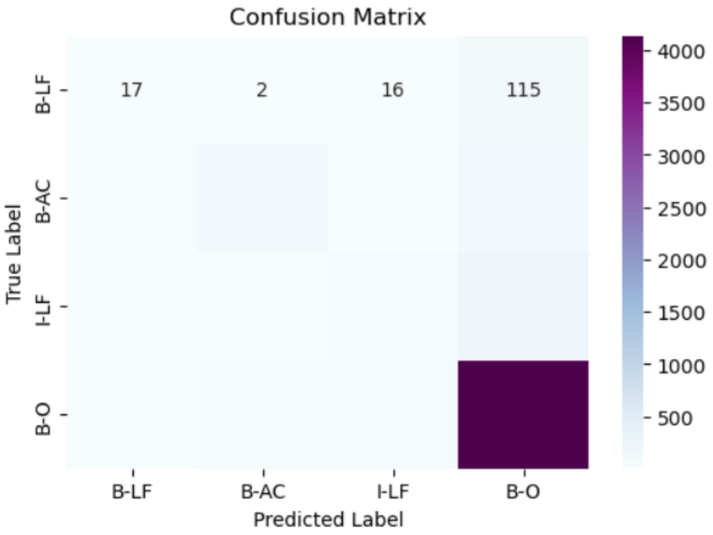


Fig 14: Confusion Matrix

- Receiver Operator Characteristic (ROC) Curve:

a) From the ROC Curve below, it can be seen that:

Point 1 – Class ‘B-AC’, with an area of 0.94 indicates that the model has a high true positive rate and a low false positive rate.

Point 2 – For class ‘B-LF’, area of 0.85 suggests that the model has a slightly greater false positive rate compared to ‘B-AC’.

Point 3 – For class ‘B-O’, area of 0.87 suggests that the model performs well, similar to class ‘B-LF’.

Point 4 – AUC for class ‘I-LF’ (0.85) suggests that the model depicts a good discrimination between positive instances and negative instances.

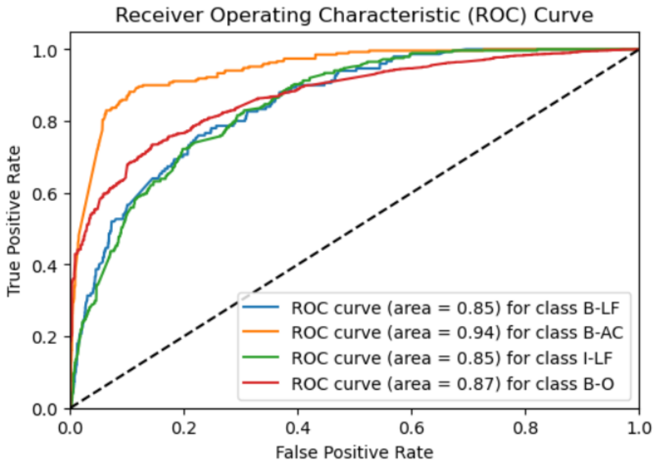


Fig 15: ROC Curve

- Regarding second experiment (Section 3.3), the accuracy testing of spacy with ANN is done.
  - F1-score Evaluation:
    a) The validation data has an F1 score of 0.846 and the test data has an F1 score of 0.842, indicating reliable and accurate predictions.

  - Error Analysis:
    a) From the confusion matrix shown below, we can see that:
    Point 1 – 'B-O' is the most accurately predicted label, followed by 'B-AC'.
    Point 2 – It can also be seen that few labels have been misclassified like 128 labels are predicted as 'B-O', they are actually 'B-LF'.

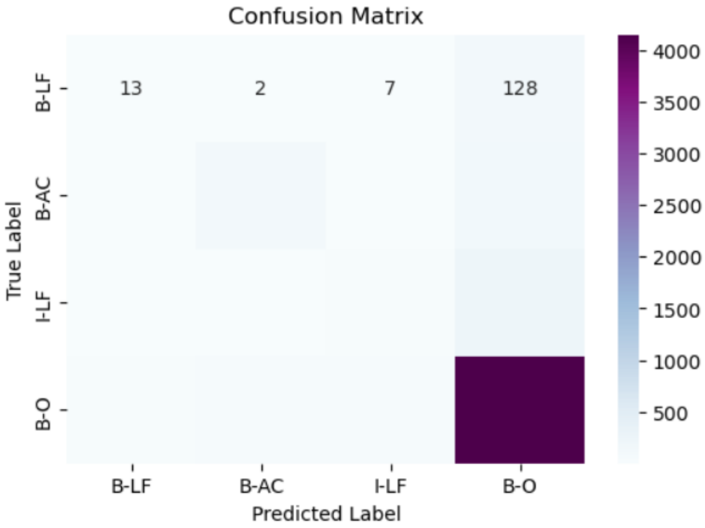


Fig 16: Confusion Matrix

  - Receiver Operator Characteristic (ROC) Curve:
    a) From the ROC Curve below, it can be seen that:
    Point 1 – Class 'B-AC', with an area of 0.95 indicates that the model has performed excellently to distinguish between true positives and false positives.
    Point 2 – For class 'B-LF', area of 0.86 suggests that the model has a great ability to differentiate between true and false positives.
    Point 3 – For class 'B-O', area of 0.87 suggests a good discriminatory ability of the model to distinguish between positive and negative instances.
    Point 4 – AUC for class 'I-LF' (0.85) suggests that the model depicts a decent ability to separate true positives from false positives.

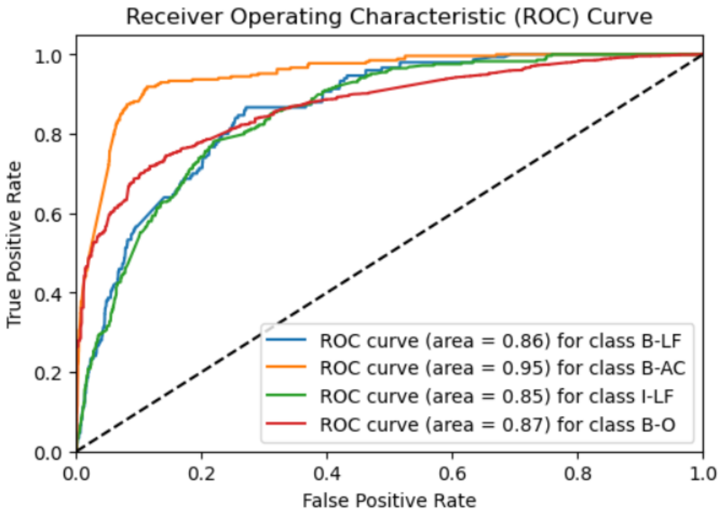


Fig 17: ROC Curve

4.4 COMPARING DIFFERENT LOSS FUNCTIONS AND OPTIMIZERS

- ➢ Regarding the first experiment (Section 3.4). The accuracy testing of spacy with ANN is done.
  1. First, Adam optimizer and sparse categorical cross entropy loss is used.
     - o F1-score Evaluation:
       a) The validation data has an F1 score of 0.837 and the test data has an F1 score of 0.838, indicating that the model performs consistently across both the datasets.
     - o Error Analysis:
       a) From the confusion matrix shown below, we can see that:
       Point 1 – 'B-O' is the most accurately predicted label, followed by 'B-AC'.
       Point 2 – It can also be seen that few labels have been misclassified like 140 labels are predicted as 'B-O', they are actually 'B-LF'.

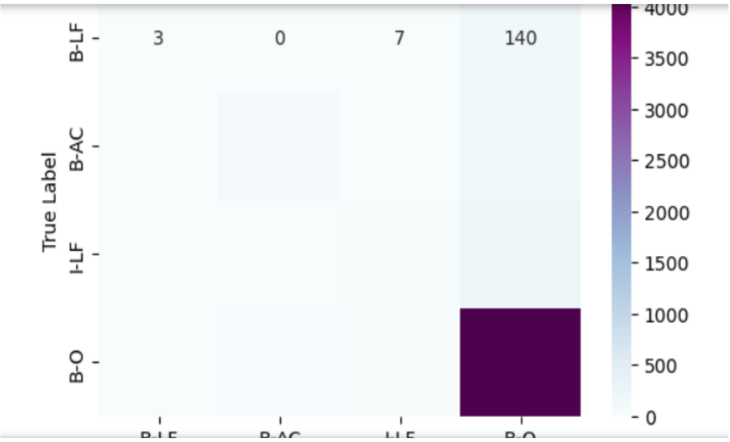


Fig 18: Confusion Matrix

2. Second, Adam optimizer and categorical cross entropy loss is used.
   - F1-score Evaluation:
     a) The validation data has an F1 score of 0.82 and the test data has an F1 score of 0.822, indicating that the model performs consistently across both the datasets.

   - Error Analysis:
     a) From the confusion matrix shown below, we can see that:
     Point 1 – 'B-O' is the most accurately predicted label.
     Point 2 – It can also be seen that few labels have been misclassified like 137 labels are predicted as 'B-O', they are actually 'B-LF'.

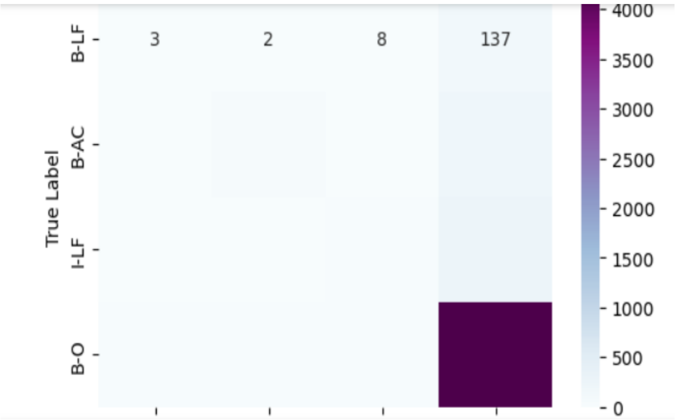


Fig 19: Confusion Matrix

3. Third, RMSprop optimizer and sparse categorical cross entropy loss is used.
   - F1-score Evaluation:
     a) The validation data has an F1 score of 0.838 and the test data has an F1 score of 0.841, indicating that the model performs consistently across both the datasets.

- Error Analysis:

  a) From the confusion matrix shown below, we can see that:
  Point 1 – 'B-O' is the most accurately predicted label, followed by 'B-AC'.
  Point 2 – It can also be seen that few labels have been misclassified like 132 labels are predicted as 'B-O', they are actually 'B-LF'.

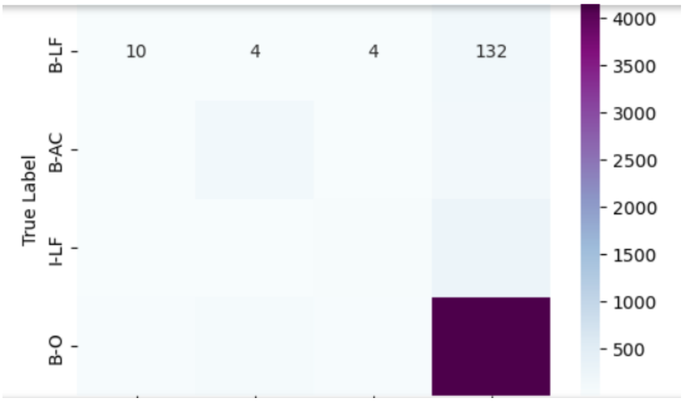


Fig 20: Confusion Matrix

4. Fourth, RMSprop optimizer and categorical cross entropy loss is used.
   - F1-score Evaluation:

     a) The validation data has an F1 score of 0.839 and the test data has an F1 score of 0.835, indicating that the model generalizes well to unseen data.

   - Error Analysis:

     a) From the confusion matrix shown below, we can see that:
     Point 1 – 'B-O' is the most accurately predicted label, followed by 'B-AC'.
     Point 2 – It can also be seen that few labels have been misclassified like 130 labels are predicted as 'B-O', they are actually 'B-LF'.

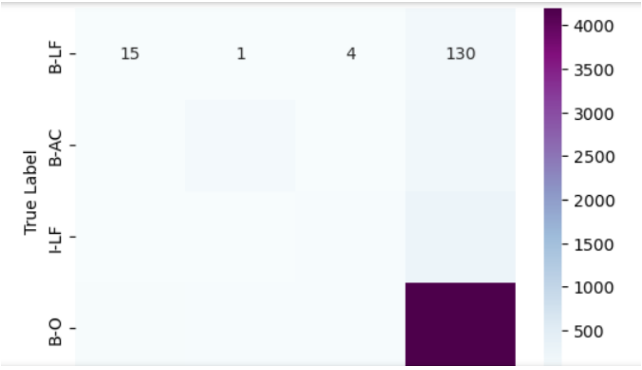


Fig 21: Confusion Matrix

5. Fifth, Stochastic Gradient Descent (SGD) optimizer and sparse categorical cross entropy loss is used.
   - F1-score Evaluation:

a) The validation data has an F1 score of 0.839 and the test data has an F1 score of 0.833, indicating that the model performs consistently across both the datasets.

- Error Analysis:

  a) From the confusion matrix shown below, we can see that:
  Point 1 – ‘B-O’ is the most accurately predicted label.
  Point 2 – It can also be seen that few labels have been misclassified like 128 labels are predicted as ‘B-O’, they are actually ‘B-LF’.

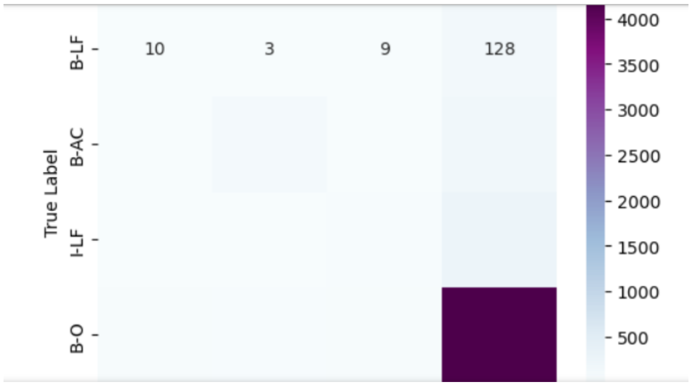


Fig 22: Confusion Matrix

6. Lastly, Stochastic Gradient Descent (SGD) optimizer and categorical cross entropy loss is used.

- F1-score Evaluation:

  a) The validation data has an F1 score of 0.839 and the test data has an F1 score of 0.833, indicating that the model performs consistently across both the datasets.

- Error Analysis:

  a) From the confusion matrix shown below, we can see that:
  Point 1 – ‘B-O’ is the most accurately predicted label.
  Point 2 – It can also be seen that few labels have been misclassified like 127 labels are predicted as ‘B-O’, they are actually ‘B-LF’.

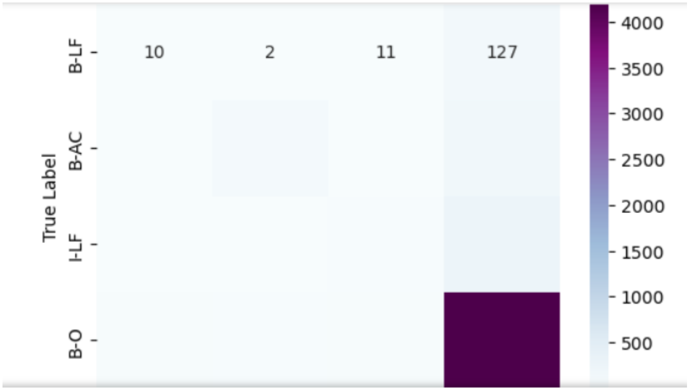


Fig 23: Confusion Matrix

- Regarding second experiment (Section 3.4). The accuracy testing of spacy with CNN is done.
  1. First, Adam optimizer and sparse categorical cross entropy loss is used.
     - F1-score Evaluation:
       a) The validation data has an F1 score of 0.839 and the test data has an F1 score of 0.843, indicating a slight improvement in the test data.
     - Error Analysis:
       a) From the confusion matrix shown below, we can see that:
       Point 1 – 'B-O' is the most accurately predicted label, followed by 'B-AC'.
       Point 2 – It can also be seen that few labels have been misclassified like 119 labels are predicted as 'B-O', they are actually 'B-LF'.

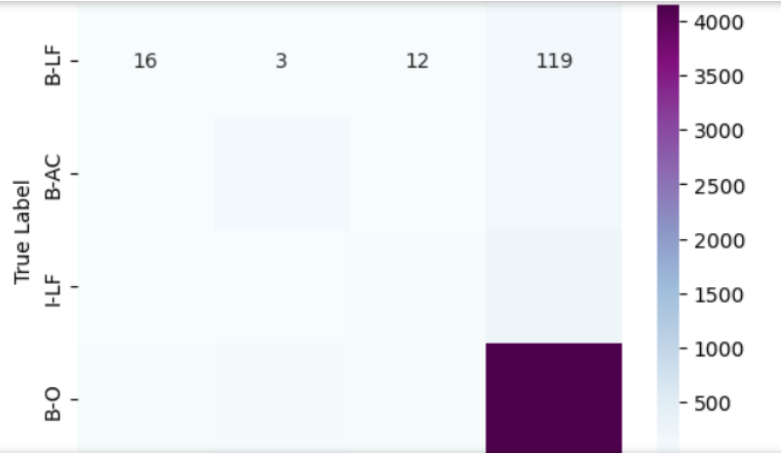


Fig 24: Confusion Matrix

  2. Second, Adam optimizer and categorical cross entropy loss is used.
     - F1-score Evaluation:
       a) The validation data has an F1 score of 0.844 and the test data has an F1 score of 0.843, indicating that the model strikes a good balance between precision and recall.
     - Error Analysis:
       a) From the confusion matrix shown below, we can see that:
       Point 1 – 'B-O' is the most accurately predicted label, followed by 'B-AC'.
       Point 2 – It can also be seen that few labels have been misclassified like 123 labels are predicted as 'B-O', they are actually 'B-LF'.

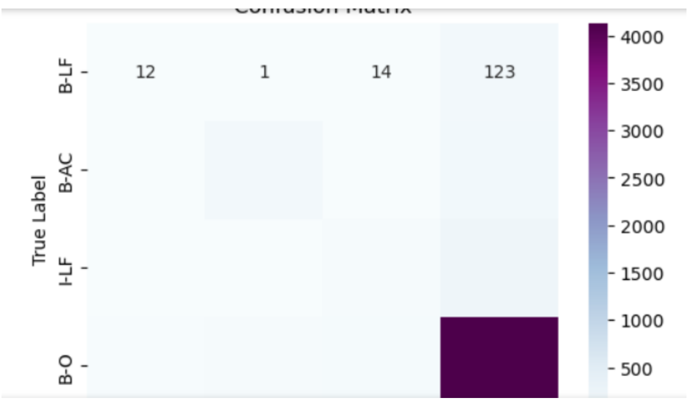
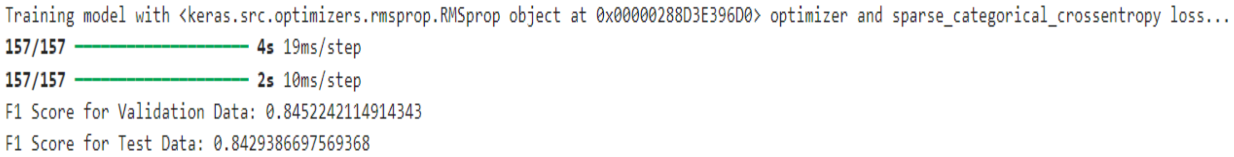


Fig 24: Confusion Matrix

3. Third, RMSprop optimizer and sparse categorical cross entropy loss is used.
   - F1-score Evaluation:
     a) The validation data has an F1 score of 0.845 and the test data has an F1 score of 0.843, indicating that the model does not suffer from overfitting or underfitting.

   - Error Analysis:
     a) From the confusion matrix shown below, we can see that:
        Point 1 – ‘B-O’ is the most accurately predicted label.
        Point 2 – It can also be seen that few labels have been misclassified like 109 labels are predicted as ‘B-O’, they are actually ‘B-LF’.



```
Training model with <keras.src.optimizers.rmsprop.RMSprop object at 0x00000288D3E396D0> optimizer and sparse_categorical_crossentropy loss...
157/157 ━━━━━━━━━━━━━━━━━━━━ 4s 19ms/step
157/157 ━━━━━━━━━━━━━━━━━━━━ 2s 10ms/step
F1 Score for Validation Data: 0.8452242114914343
F1 Score for Test Data: 0.8429386697569368
```

Fig 25: F1 Scores for Validation and Test Data

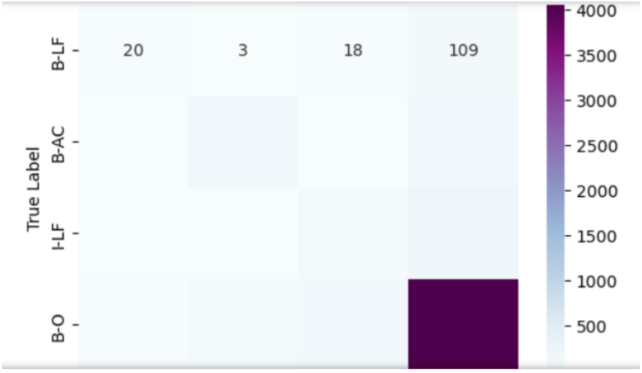


Fig 26: Confusion Matrix

4. Fourth, RMSprop optimizer and categorical cross entropy loss is used.
   - F1-score Evaluation:

a) The validation data has an F1 score of 0.849 and the test data has an F1 score of 0.845, indicating that the model generalizes well to unseen data and is robust.

- Error Analysis:

a) From the confusion matrix shown below, we can see that:
Point 1 – 'B-O' is the most accurately predicted label.
Point 2 – It can also be seen that few labels have been misclassified like 106 labels are predicted as 'B-O', they are actually 'B-LF'.

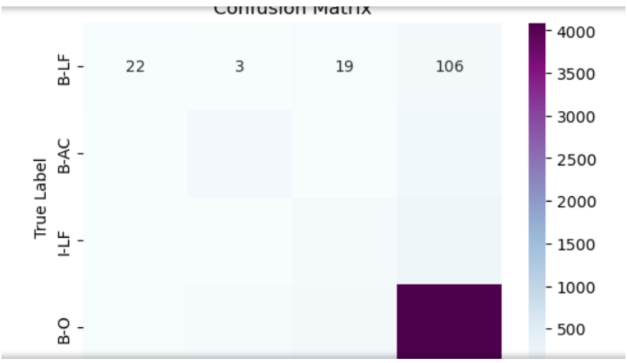


Fig 27: Confusion Matrix

5. Stochastic Gradient Descent (SGD) optimizer and sparse categorical cross entropy loss is used.

- F1-score Evaluation:

a) The validation data has an F1 score of 0.846 and the test data has an F1 score of 0.845, indicating that the model performs well on both the datasets.

- Error Analysis:

a) From the confusion matrix shown below, we can see that:
Point 1 – 'B-O' is the most accurately predicted label.
Point 2 – It can also be seen that few labels have been misclassified like 116 labels are predicted as 'B-O', they are actually 'B-LF'.

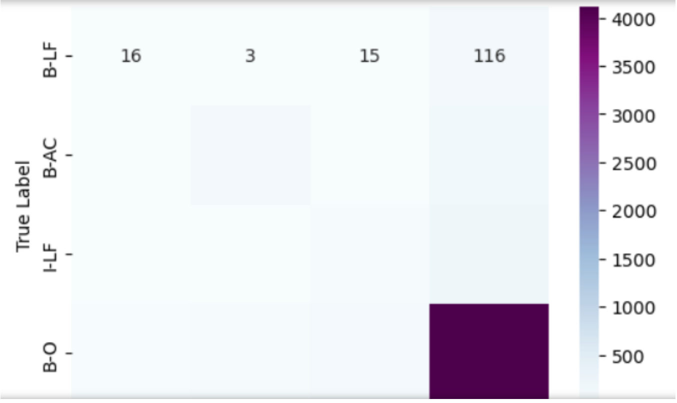


Fig 28: Confusion Matrix

6. Lastly, Stochastic Gradient Descent (SGD) optimizer and categorical cross entropy loss is used.
   - F1-score Evaluation:
     a) The validation data has an F1 score of 0.847 and the test data has an F1 score of 0.845, indicating that the model performs well on new unseen instances.

   - Error Analysis:
     a) From the confusion matrix shown below, we can see that:
        Point 1 – 'B-O' is the most accurately predicted label.
        Point 2 – It can also be seen that few labels have been misclassified like 117 labels are predicted as 'B-O', they are actually 'B-LF'.

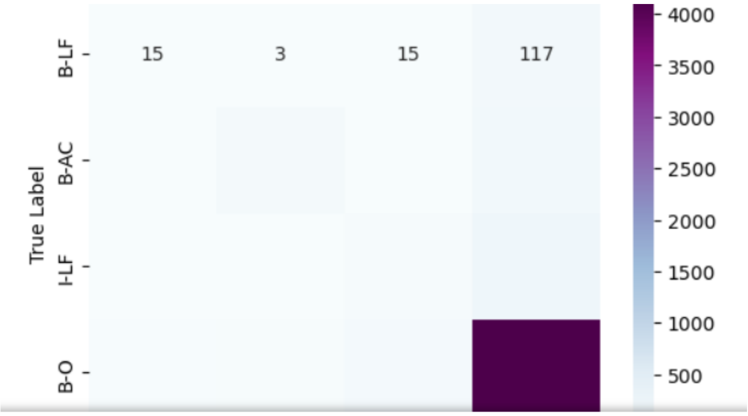


Fig 29: Confusion Matrix

5 RESULT

- From Section 3.1, the best result was obtained on testing the experiment which involved spacy and SVM, because SVM algorithm is less prone to overfitting compared to other algorithms.

- From Section 3.2, the best result was obtained on testing the experiment which involved spacy and Random Forest, because spacy is known for speed and efficiency, in comparison with Word2Vec.

- From Section 3.3, the best result was obtained on testing the experiment which involved spacy and CNN, because CNNs preserve the spatial structure of the input data with the help of convolutional layers.

- From Section 3.4, the best result was obtained on testing the experiment which involved spacy and CNN alongside RMSprop optimizer and categorical cross entropy loss. This is because RMSprop adjusts the learning rate, based on the average of recent gradients and categorical cross entropy loss helps to accurately capture relationships between predicted and true values, whilst CNNs help to preserve the spatial structure of the input data.

6 ANALYSIS & CONCLUSION

The result of experimentation is depicted below. “F1” score has been considered as the primary evaluation metric.

S. No	Experiment	Combination	F1 validation	F1 test	ROC Curve Values			
					Class B-AC	Class B-LF	Class B-O	Class I-LF
1	Comparing traditional algorithms	Using Spacy and Decision tree	0.86	0.84	0.78	0.69	0.73	0.7
		Using Spacy and SVM	0.83	0.84	0.96	0.82	0.74	0.68
2	Comparing Features/Vectorization Methods	Using Word2Vec and RF	0.76	0.77	0.49	0.5	0.47	0.47
		Using Spacy and RF	0.87	0.84	0.97	0.85	0.87	0.81
3	Comparing Deep Learning Algorithms	Using Spacy and CNN	0.84	0.84	0.94	0.85	0.87	0.85
		Using Spacy and ANN	0.84	0.84	0.95	0.86	0.87	0.85

4	Comparing Different Loss Functions and Optimizers	Using Spacy and ANN (Adam and sparse categorical cross entropy)	0.83	0.84	NA
		Using Spacy and CNN (Adam and sparse categorical cross entropy)	0.84	0.85	NA
5	Comparing Different Loss Functions and Optimizers	Using Spacy and ANN (Adam and categorical cross entropy)	0.82	0.82	NA
		Using Spacy and CNN (Adam and categorical cross entropy)	0.84	0.84	NA
6	Comparing Different Loss Functions and Optimizers	Using Spacy and ANN (RMSprop and sparse categorical cross entropy)	0.84	0.84	NA
		Using Spacy and CNN (RMSprop and sparse categorical cross entropy)	0.84	0.84	NA
7	Comparing Different Loss Functions and Optimizers	Using Spacy and ANN (RMSprop and categorical cross entropy)	0.84	0.83	NA
		Using Spacy and CNN (RMSprop and categorical cross entropy)	0.85	0.84	NA
8	Comparing Different Loss Functions and Optimizers	Using Spacy and ANN (SGD and sparse categorical cross entropy)	0.84	0.83	NA
		Using Spacy and CNN (SGD and sparse categorical cross entropy)	0.84	0.84	NA
9	Comparing Different Loss Functions and Optimizers	Using Spacy and ANN (SGD and categorical cross entropy)	0.84	0.83	NA
		Using Spacy and CNN (SGD and categorical cross entropy)	0.85	0.84	NA

- As can be seen from the above table, the models built fulfilled their purpose of sequence classification. This is because:
  - Their F1 scores are around 85%, on both the validation and test datasets.
  - The Area Under the ROC Curve (AUC) is close to 1 for almost all the experiments.

- A “good enough” F1 or accuracy score is determined by the specific domain and error tolerance. From the above table, 0.85 can be considered as a good enough F1 score. In certain applications, high accuracy is of prime importance e.g., financial fraud detection, whereas in some other applications like this, lesser accuracies are acceptable because it comes coupled with lesser compute power.

- From the above table, the Word2Vec encoder used with Random Forest yields a very low F1 score of 0.77. This is because of the inability of Word2Vec embeddings to capture the semantic ambiguity present in the dataset. Hence, to further increase the accuracy of the model, further investigation can be done using feature representation and hyperparameter optimization. Also, we can experiment with other encoding methods like Glove, fasttext and tfidf.

- Referring to the above table, the CNN model which used spacy embeddings (Adam and sparse categorical cross entropy), had an accuracy of 85%. However, the training was very slow. The model could be made efficient without compromising quality, by leveraging hardware accelerators and optimizing inference pipelines.

- From the above table, the ANN model with spacy embeddings (Adam and categorical cross entropy), took much lesser time for training (three minutes), than the CNN one (mentioned in the above point), and it gave 82% accuracy. Thus, the most accurate solution in this case would be the CNN model with spacy embeddings, whereas the most effective solution would be the ANN model with spacy embeddings.

7 REFERENCES

[1] Dataset - https://huggingface.co/datasets/surrey-nlp/PLOD-CW
[2] https://www.researchgate.net/publication/308007227_Exploratory_Data_Analysis
[3] https://www.analyticsvidhya.com/blog/2022/07/step-by-step-exploratory-data-analysis-eda-using-python/
[4] Lab – 06 NLP
[5] Jupyter Notebook with experimentation code (with comments)