Individual Experimentation for Natual Language Processing

Introduction

This coursework addresses a specific challenge within sequence classification: detecting and labelling abbreviations and long forms in biomedical texts, using the BIO (Begin, Inside, Outside) labelling format. The dataset was from PLOS journal articles and information extraction in the biomedical domain. Inside the dataset on hugging face there are 50,000 labelled tokens. Abbreviations (tagged as AC) and long forms (tagged as LF) are pervasive in scientific literature, making their identification essential for the accurate interpretation of content. This report will start with an understanding and visualization of the dataset, approach to experimentation, testing for the experimentation, observation of the outcomes, explanation of the outcomes and finally the evaluation of the experiments. All packages needed for experiments are written in the notebook file. Fasttext was removed from the experiment and no helper files were required for the current experiment setup.

1. Analysis of the Dataset

After loading the dataset, training Data: Consists of 1,072 entries. This is the largest subset and is used to train the model. Testing Data: Comprises 153 entries. It is used to assess the model's performance after it has been trained and tuned. Validation Data: Contains 126 entries. This subset can be used to tune the parameters of the model and prevent overfitting.

The training dataset will mainly used for training so further plotting was done for the dataset:

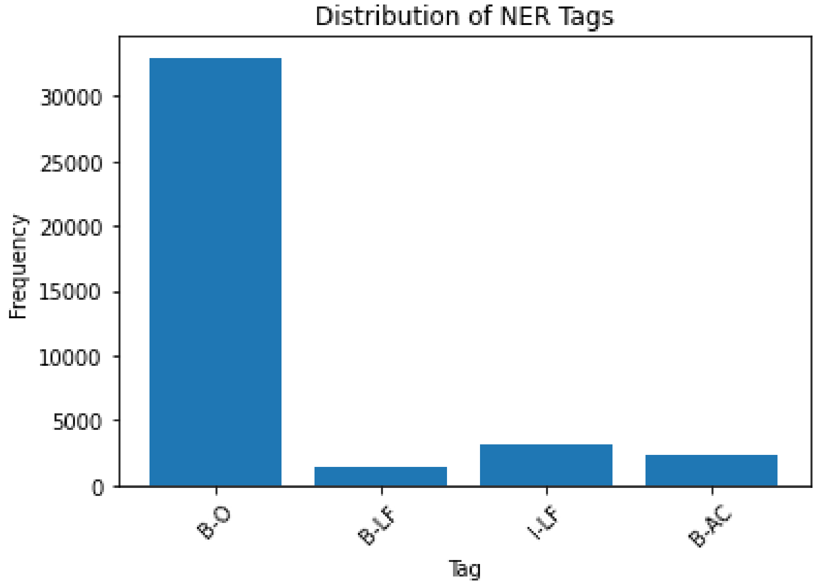


Image 1.1
The distribution of NER tags shows that the B-O tag category has the highest frequency; most tokens in the dataset are labelled as outside-of-name entities. This shows a significant imbalance in the dataset, which might lead to poor performance during the model training process.

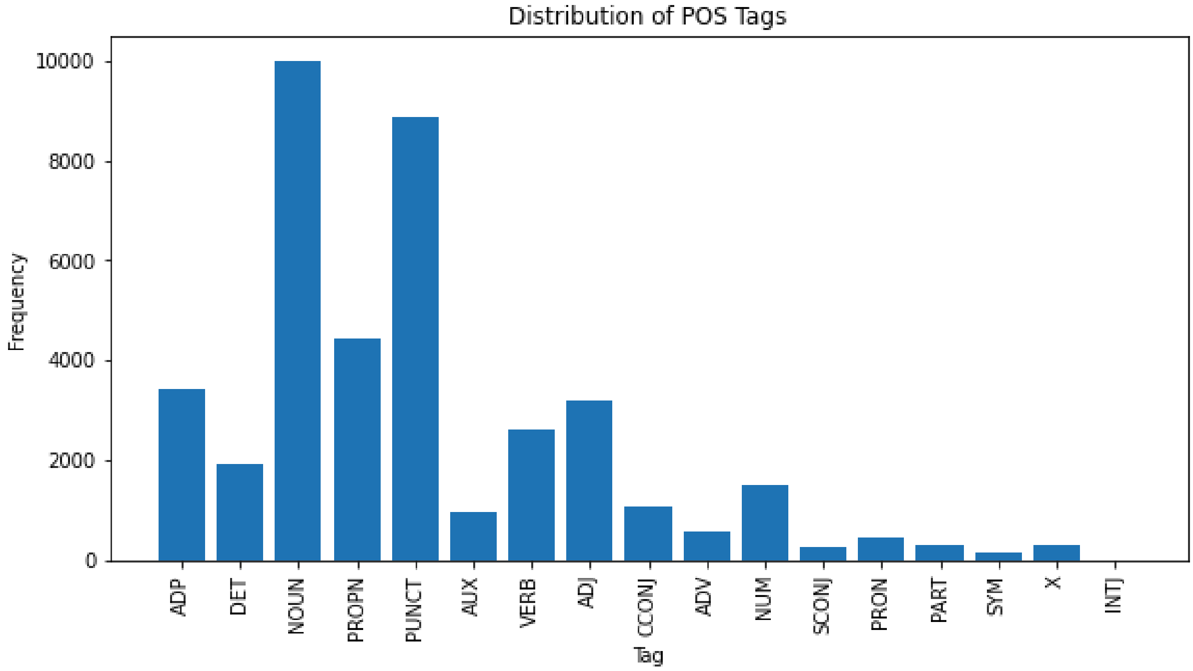


Image 1.2
The distribution of POS tags depicts the frequency of different POS (Part of Speech) tags, Each bar represents the count of a specific part of speech in the dataset, such as nouns,

verbs, adjectives, etc. It can be used as an option.

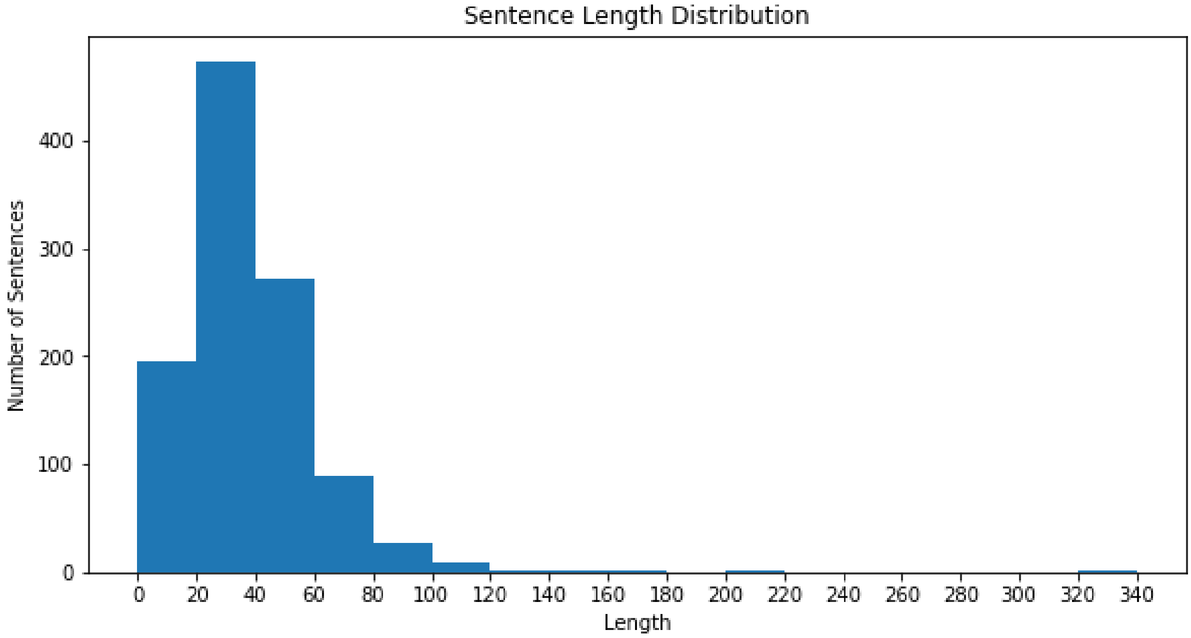
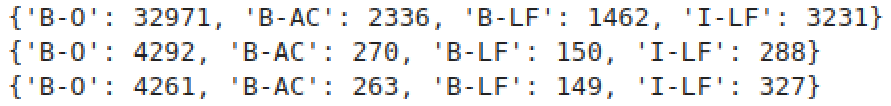


Image 1.3

The distribution of Sentence Length shows that the majority of sentences are short. With a mean length of 37 in the train dataset. The maximum sequence length was given, so if a model has a maximum length limit, it must be modified. Depending on the experiments, operations like padding, truncation, and removing outliers can be performed.



```
{'B-O': 32971, 'B-AC': 2336, 'B-LF': 1462, 'I-LF': 3231}
{'B-O': 4292, 'B-AC': 270, 'B-LF': 150, 'I-LF': 288}
{'B-O': 4261, 'B-AC': 263, 'B-LF': 149, 'I-LF': 327}
```

Image 1.4

Test and Validation datasets also show an imbalance in the distribution of NER Tags.

2. Approach to Experiments

Experiment 1: Comparing different data pre-processing techniques.
System 1: No preprocessing, Conditional Random Fields.
Vs.
System 2: Preprocess to lowercase, Conditional Random Fields.
Vs.
System 3: Preprocess to lowercase and remove stop words, Conditional Random Fields.

The success of a CRF model depends greatly on the quality and structure of the input data. Therefore, data preprocessing plays a role in enhancing its efficiency. The selection of methods is influenced by the data distribution. The dataset contains tokens with varying capitalization. Incorporates common stopwords found in academic papers. Implementing methods can simplify the CRF model's feature set, improving its capacity to identify relevant patterns for sequence classification.

Experiment 2: Comparing CRF to SVM
The same vectorization method and best-performing preprocess method, the DictVectorizer, is used to convert the list of feature dictionaries for each token into sparse matrices.

Choosing to go with an SVM following the utilization of a CRF could prove advantageous since SVMs are adept at handling high dimensional data, against overfitting and making quick predictions. This decision streamlines the model while potentially preserving or even enhancing its performance in scenarios where sequence dependencies play a role.

Experiment 3: Comparing different tokenizers for SVM
System 1: TF-IDF Vectorization without data-balancing
Vs.
System 2: Word2Vec with data-balancing
Vs.
System 3: TF-IDF Vectorization without data-balancing

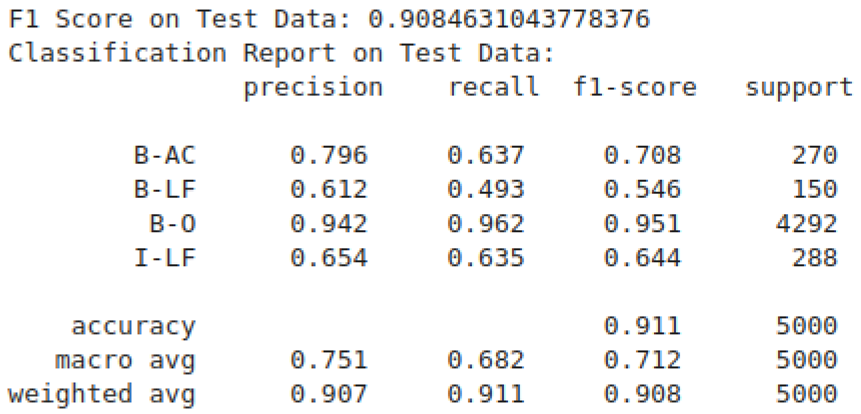
Compare the accuracy of different tokenizers; instead of handcrafted features, TF-IDF can provide a baseline using traditional token frequency measures, emphasizing feature representation without addressing class imbalance. It is also worth trying semantic embeddings like Word2Vec. Finally, see if data banking can improve performance.

Experiment 4: Hyperparameter optimisation is the best-performing system from above. One system will be CRF, and another SVM to try to improve performance further.

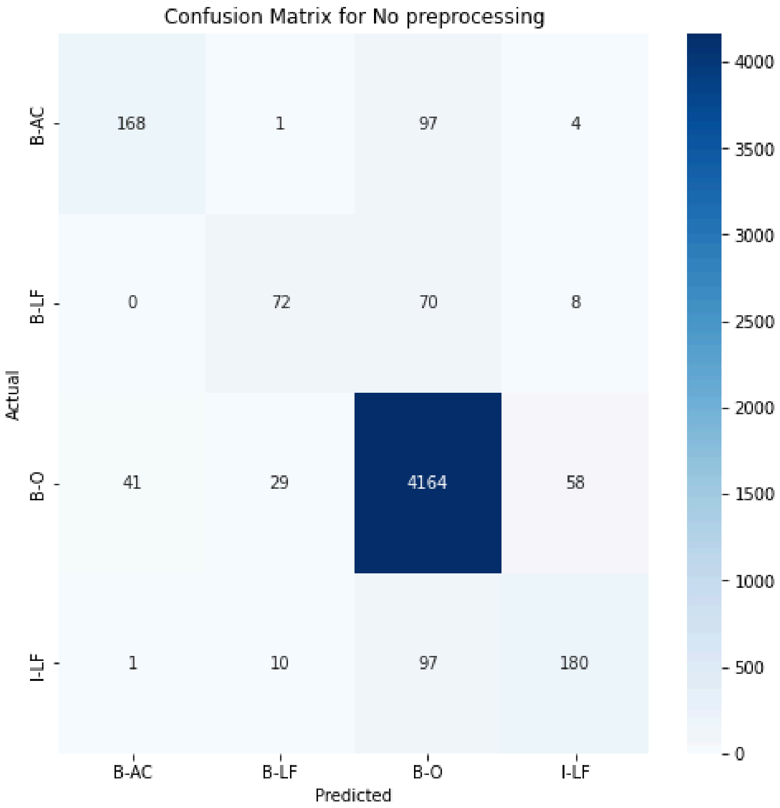
Testing for the experimentation

3.1:

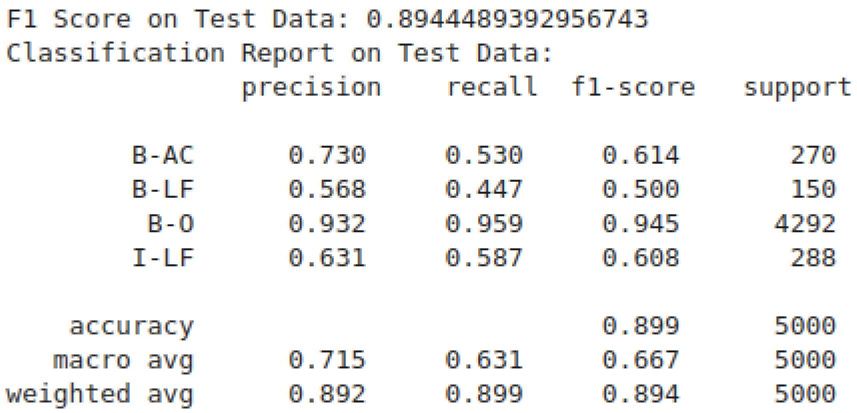
CRF with no data pre-processing:



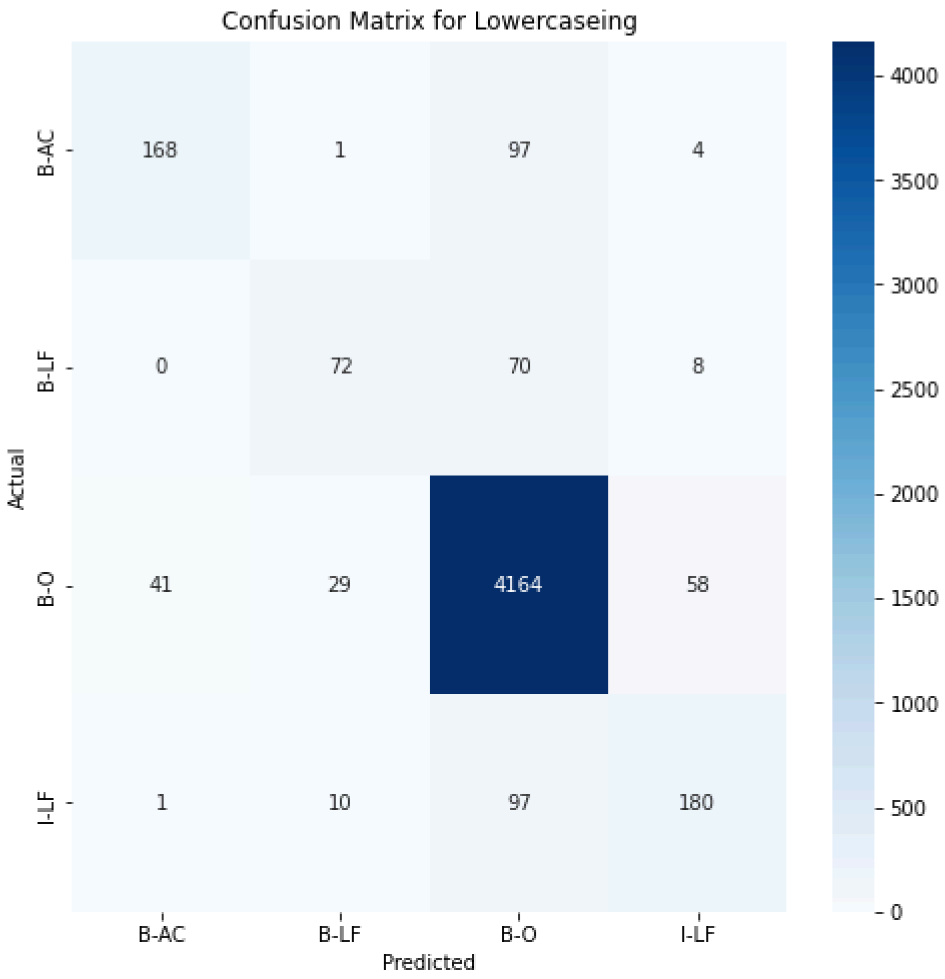
```
F1 Score on Test Data: 0.9084631043778376
Classification Report on Test Data:
               precision    recall  f1-score   support

        B-AC       0.796     0.637     0.708       270
        B-LF       0.612     0.493     0.546       150
         B-O       0.942     0.962     0.951      4292
        I-LF       0.654     0.635     0.644       288

    accuracy                           0.911      5000
   macro avg       0.751     0.682     0.712      5000
weighted avg       0.907     0.911     0.908      5000
```



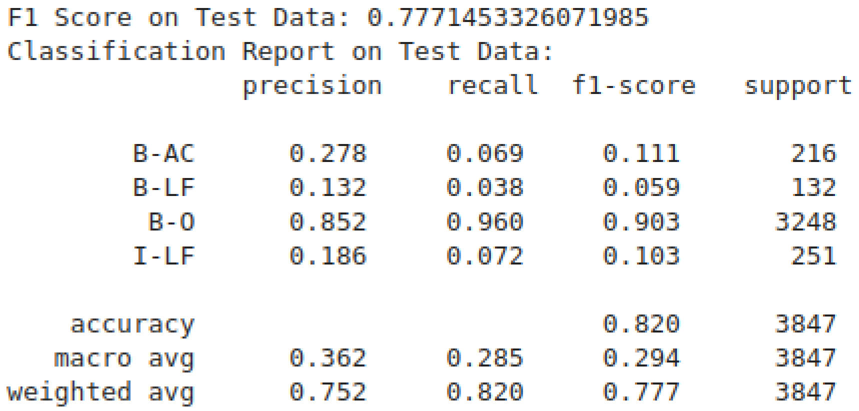
CRF with change to lowercase:



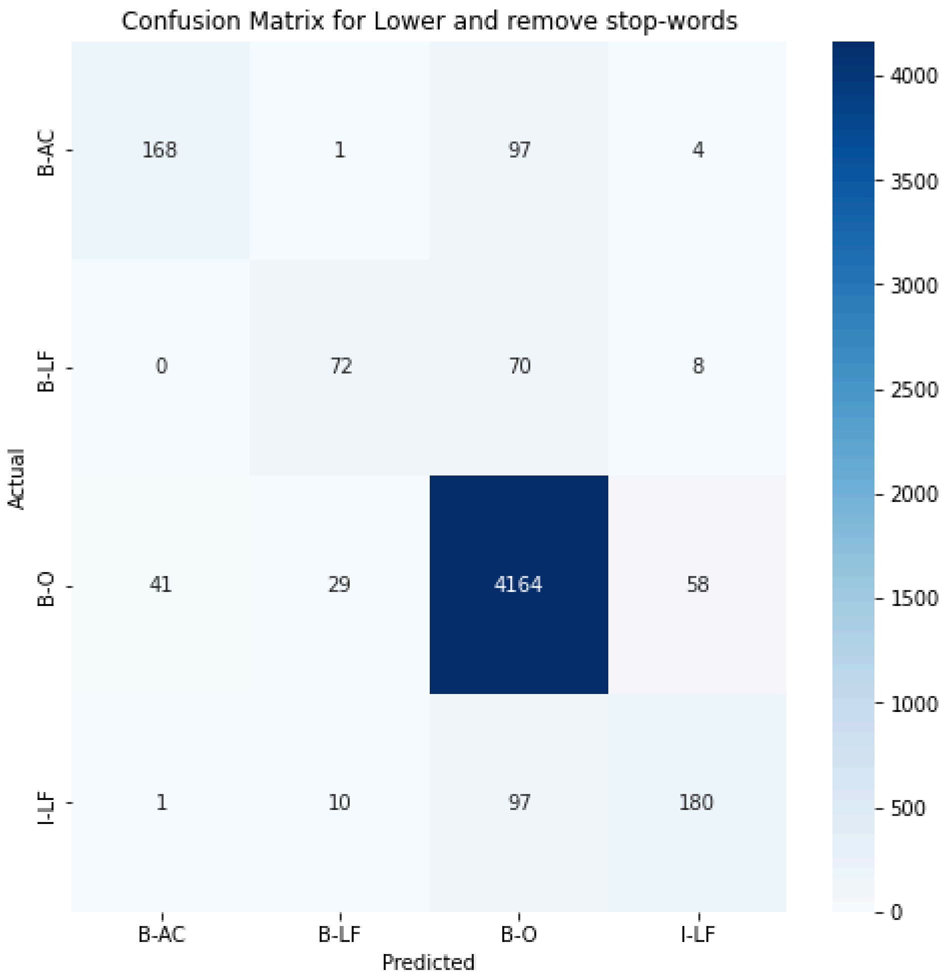
```
F1 Score on Test Data: 0.8944489392956743
Classification Report on Test Data:
               precision    recall  f1-score   support

        B-AC      0.730     0.530     0.614       270
        B-LF      0.568     0.447     0.500       150
         B-O      0.932     0.959     0.945      4292
        I-LF      0.631     0.587     0.608       288

    accuracy                          0.899      5000
   macro avg      0.715     0.631     0.667      5000
weighted avg      0.892     0.899     0.894      5000
```



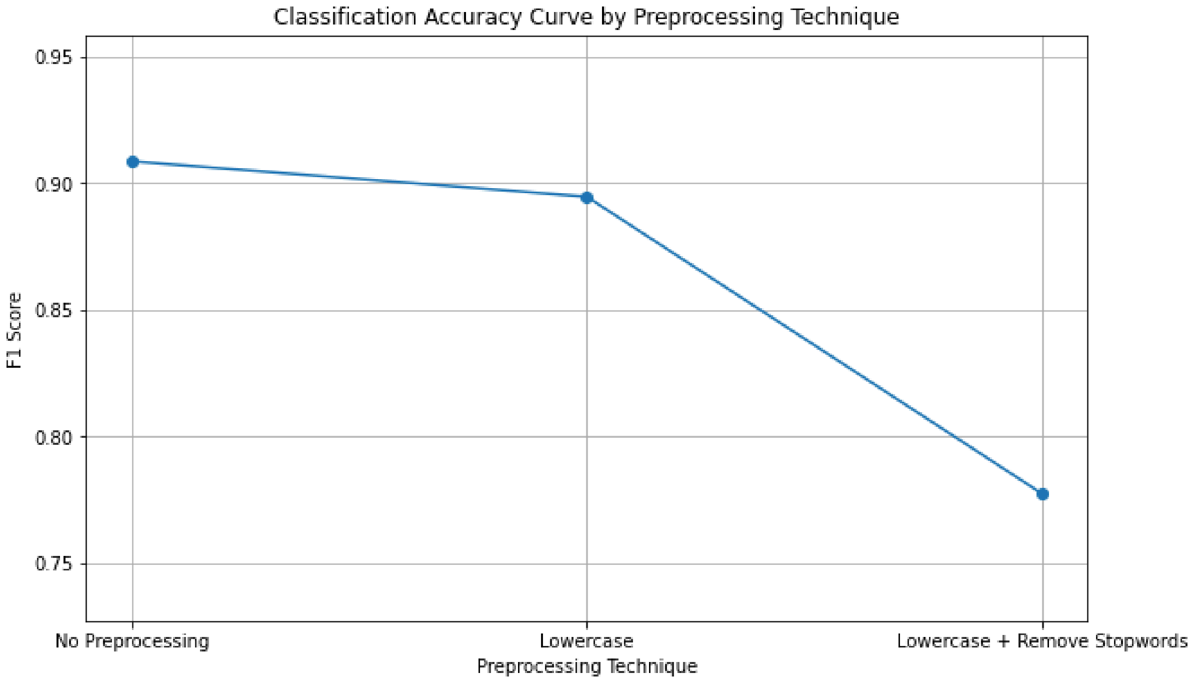
CRF with lowercasing and remove stop words:



```
F1 Score on Test Data: 0.7771453326071985
Classification Report on Test Data:
               precision    recall  f1-score   support

        B-AC      0.278     0.069     0.111       216
        B-LF      0.132     0.038     0.059       132
         B-O      0.852     0.960     0.903      3248
        I-LF      0.186     0.072     0.103       251

    accuracy                          0.820      3847
   macro avg      0.362     0.285     0.294      3847
weighted avg      0.752     0.820     0.777      3847
```

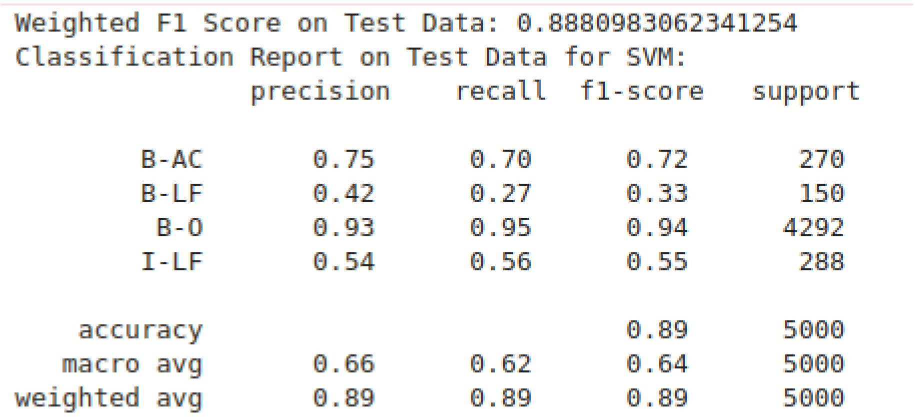


Classification Accuracy Curve by Preprocessing Techniques:

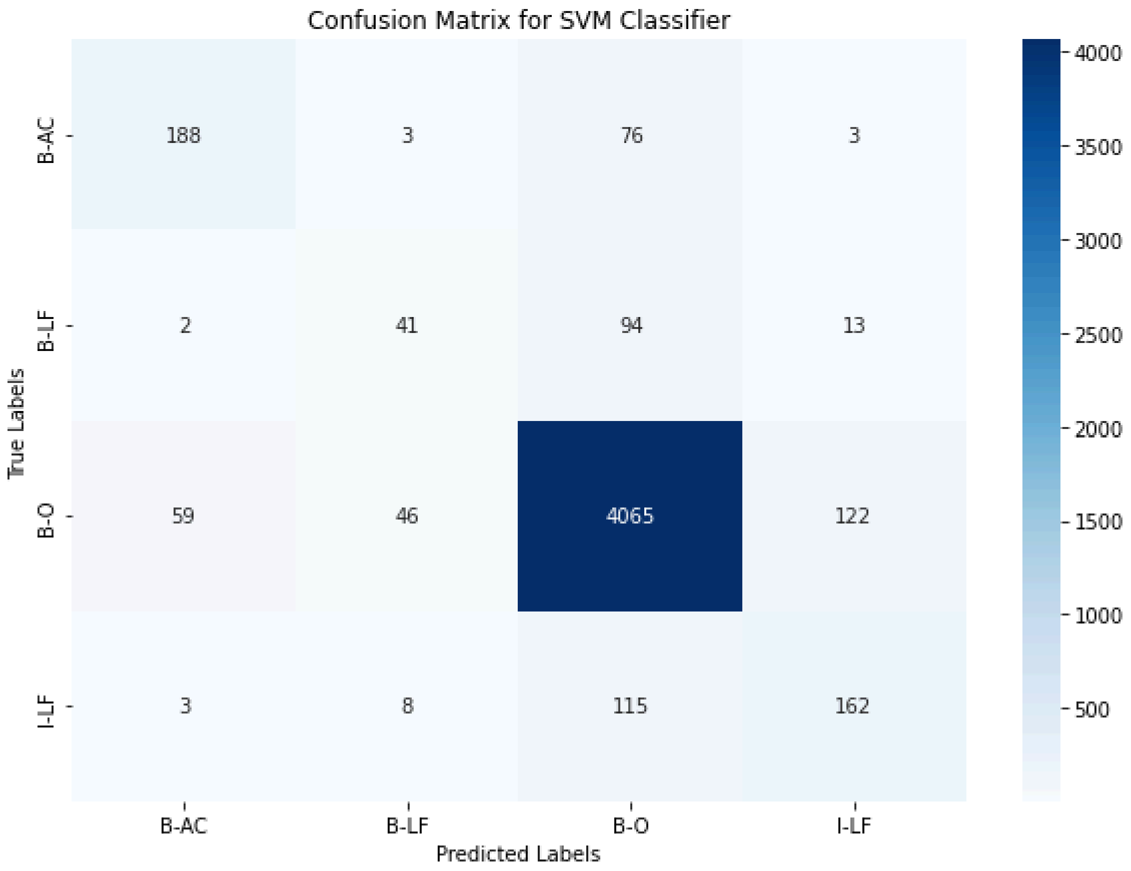


3.2:

Compare CRF with SVM:



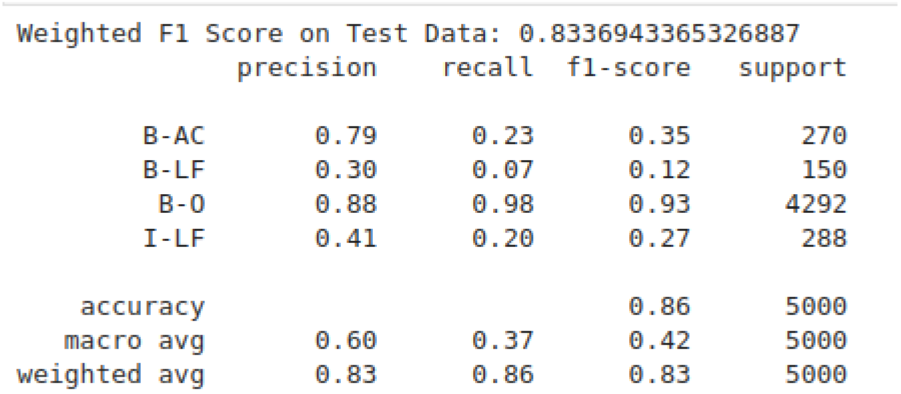
```
Weighted F1 Score on Test Data: 0.8880983062341254
Classification Report on Test Data for SVM:
               precision    recall  f1-score   support

        B-AC       0.75      0.70      0.72       270
        B-LF       0.42      0.27      0.33       150
         B-O       0.93      0.95      0.94      4292
        I-LF       0.54      0.56      0.55       288

    accuracy                           0.89      5000
   macro avg       0.66      0.62      0.64      5000
weighted avg       0.89      0.89      0.89      5000
```



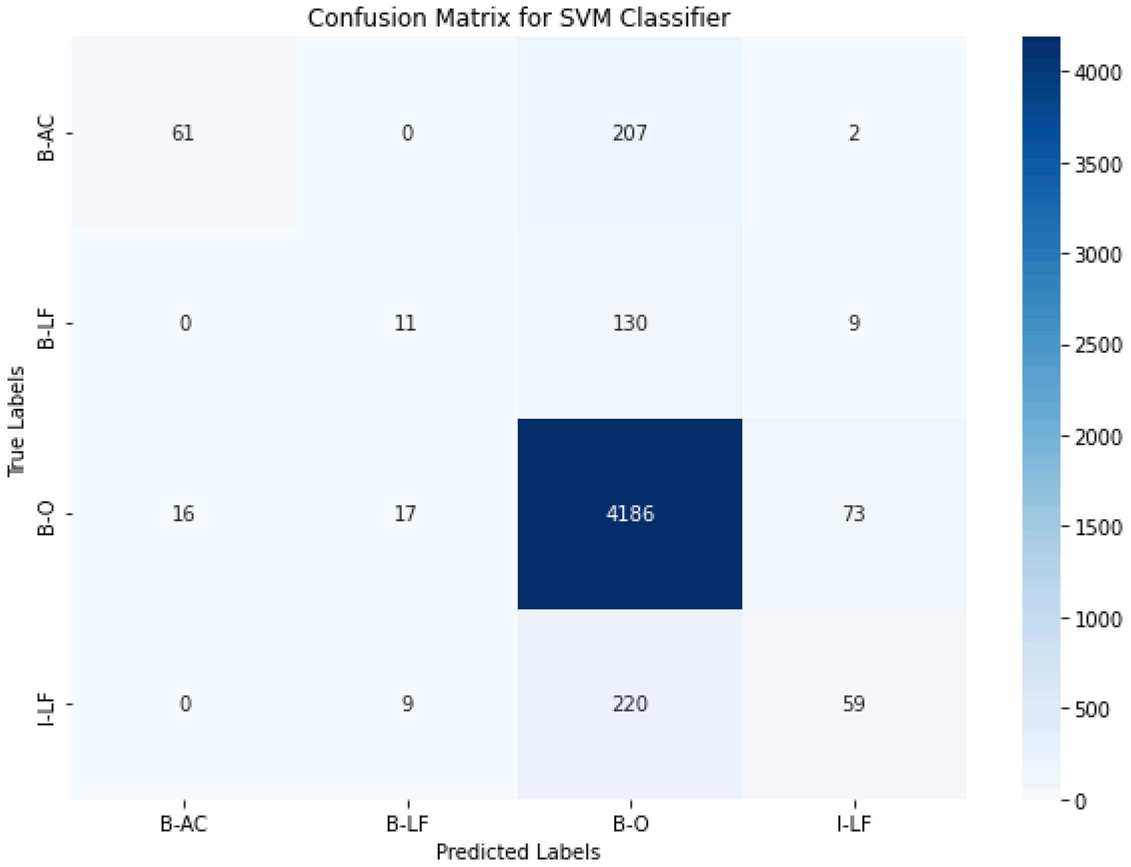
	B-AC	B-LF	B-O	I-LF
B-AC	188	3	76	3
B-LF	2	41	94	13
B-O	59	46	4065	122
I-LF	3	8	115	162

3.3:

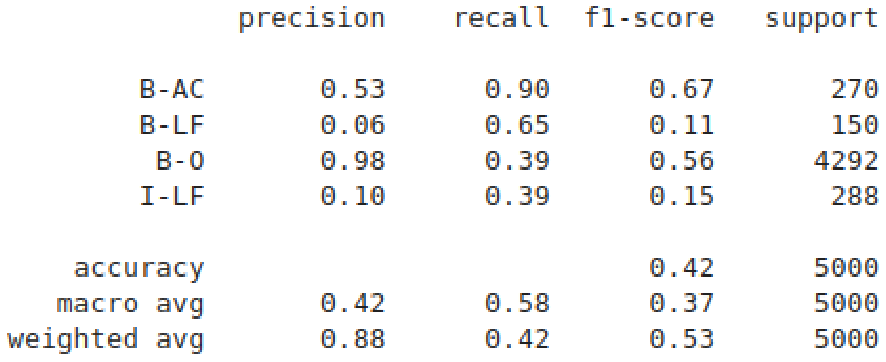
SVM using TfidfVectorizer:



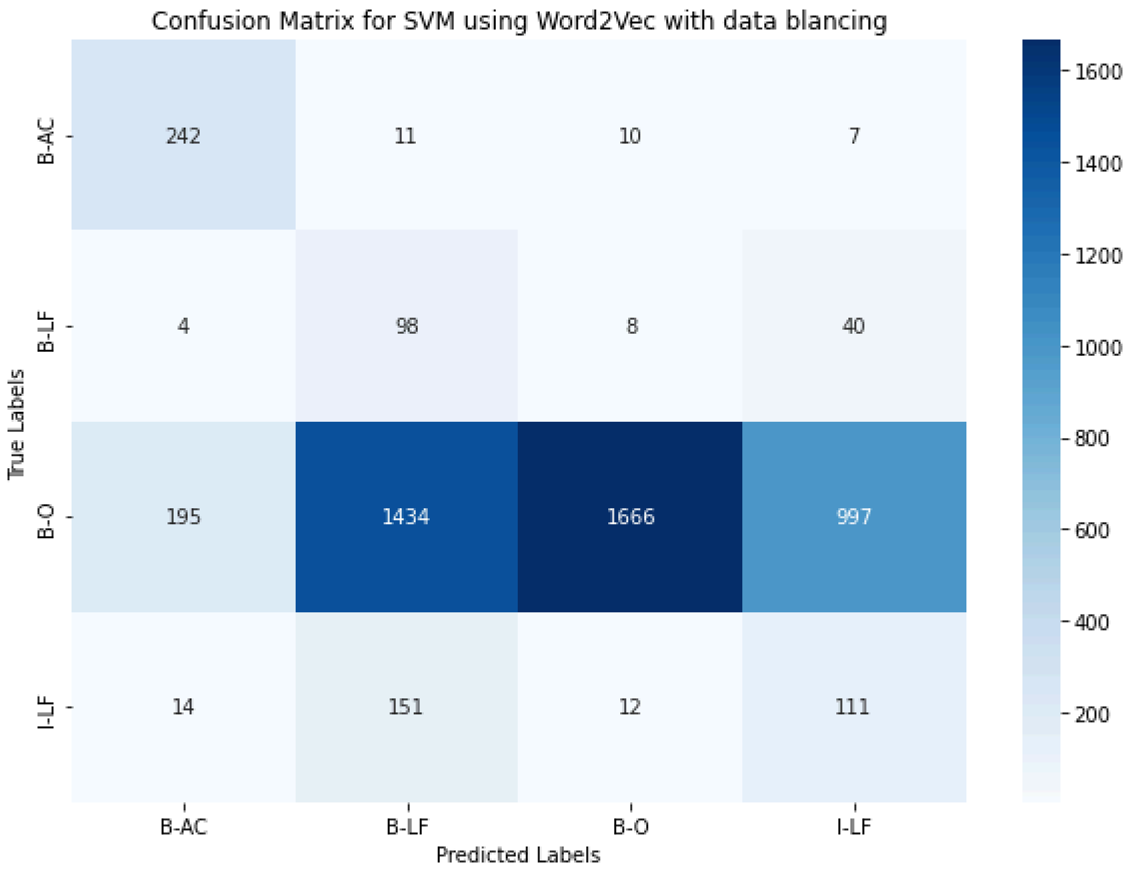
```
Weighted F1 Score on Test Data: 0.8336943365326887
              precision    recall  f1-score   support

        B-AC       0.79      0.23      0.35       270
        B-LF       0.30      0.07      0.12       150
         B-O       0.88      0.98      0.93      4292
        I-LF       0.41      0.20      0.27       288

    accuracy                           0.86      5000
   macro avg       0.60      0.37      0.42      5000
weighted avg       0.83      0.86      0.83      5000
```



SVM using Word2Vec with data balancing:

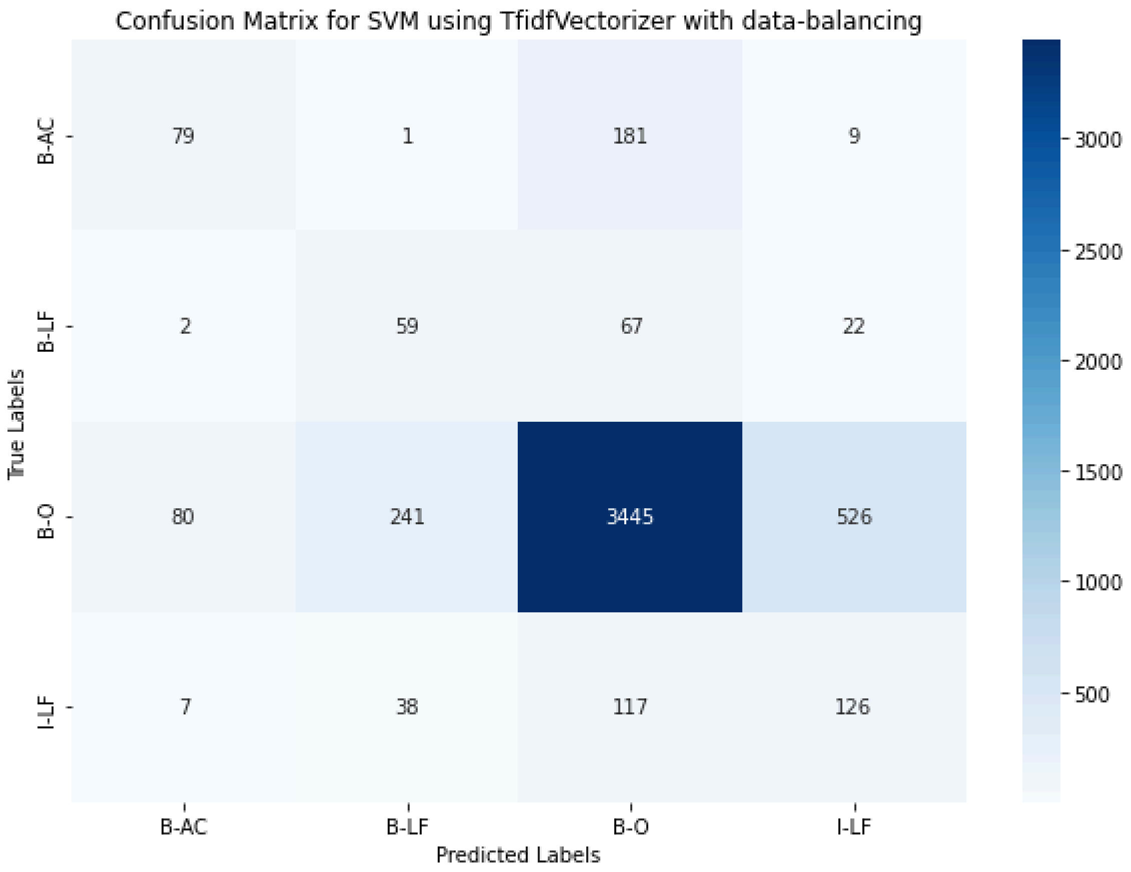


```
              precision    recall  f1-score   support

        B-AC       0.53      0.90      0.67       270
        B-LF       0.06      0.65      0.11       150
         B-O       0.98      0.39      0.56      4292
        I-LF       0.10      0.39      0.15       288

    accuracy                           0.42      5000
   macro avg       0.42      0.58      0.37      5000
weighted avg       0.88      0.42      0.53      5000
```

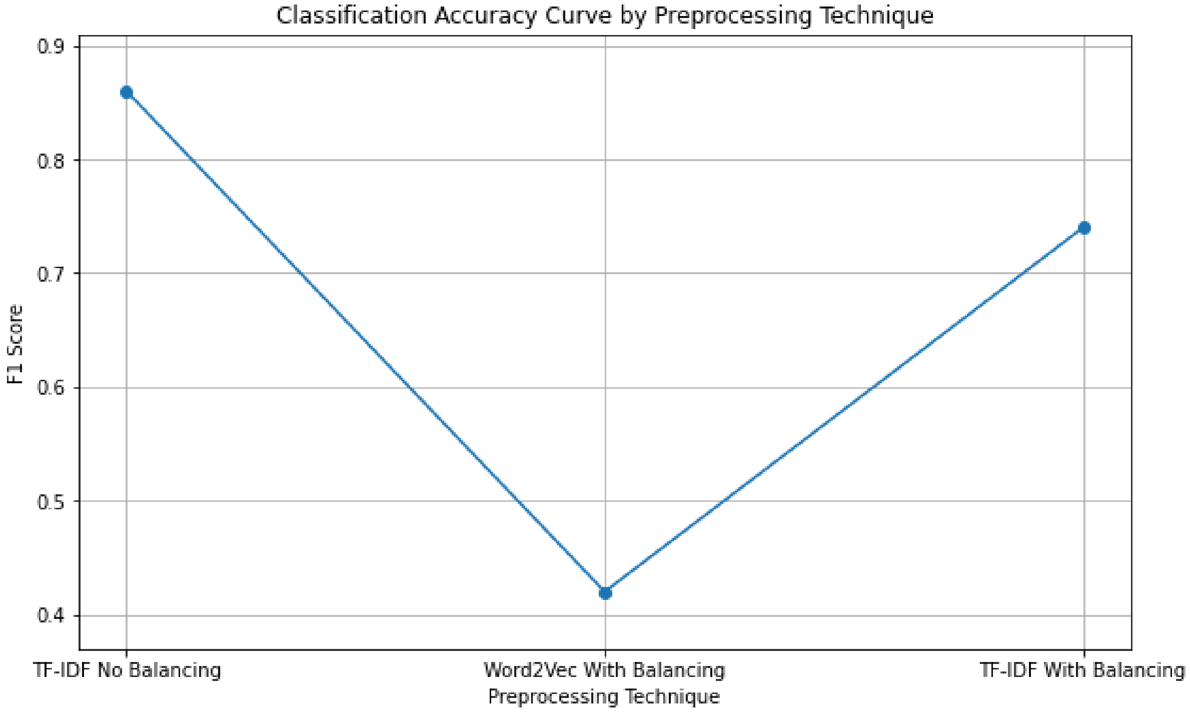


SVM using TfidfVectorizer with data-balancing:

```
              precision    recall  f1-score   support

        B-AC       0.47      0.29      0.36       270
        B-LF       0.17      0.39      0.24       150
         B-O       0.90      0.80      0.85      4292
        I-LF       0.18      0.44      0.26       288

    accuracy                           0.74      5000
   macro avg       0.43      0.48      0.43      5000
weighted avg       0.82      0.74      0.77      5000
```

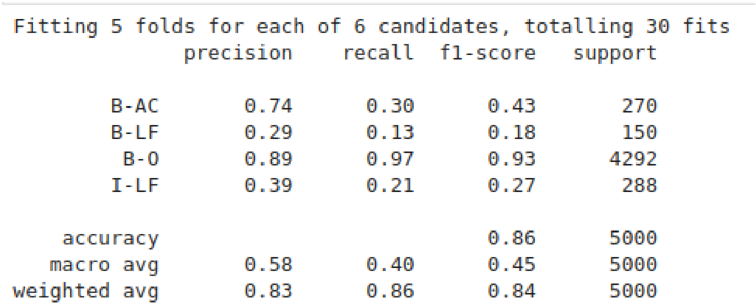


Classification Accuracy Curve by three different tokenizer for SVM

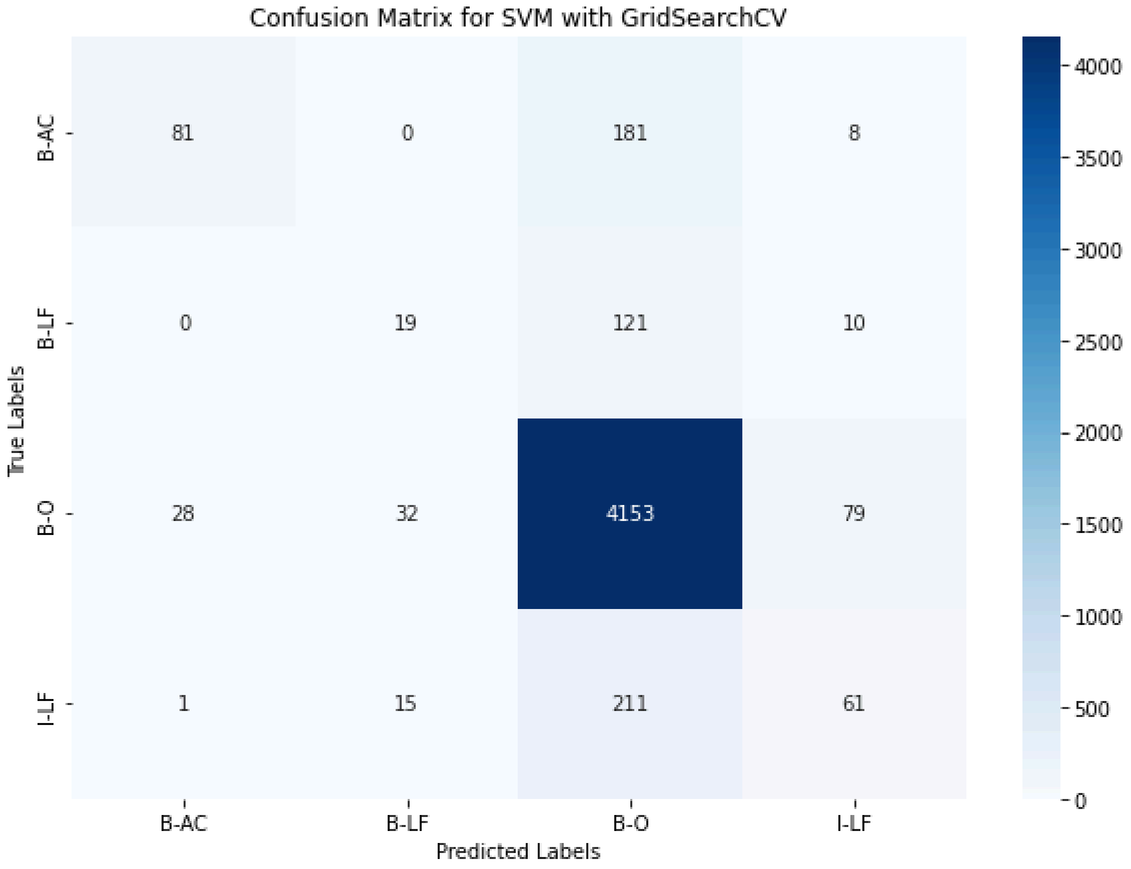


3.4:

SVM with GridSearchCV:

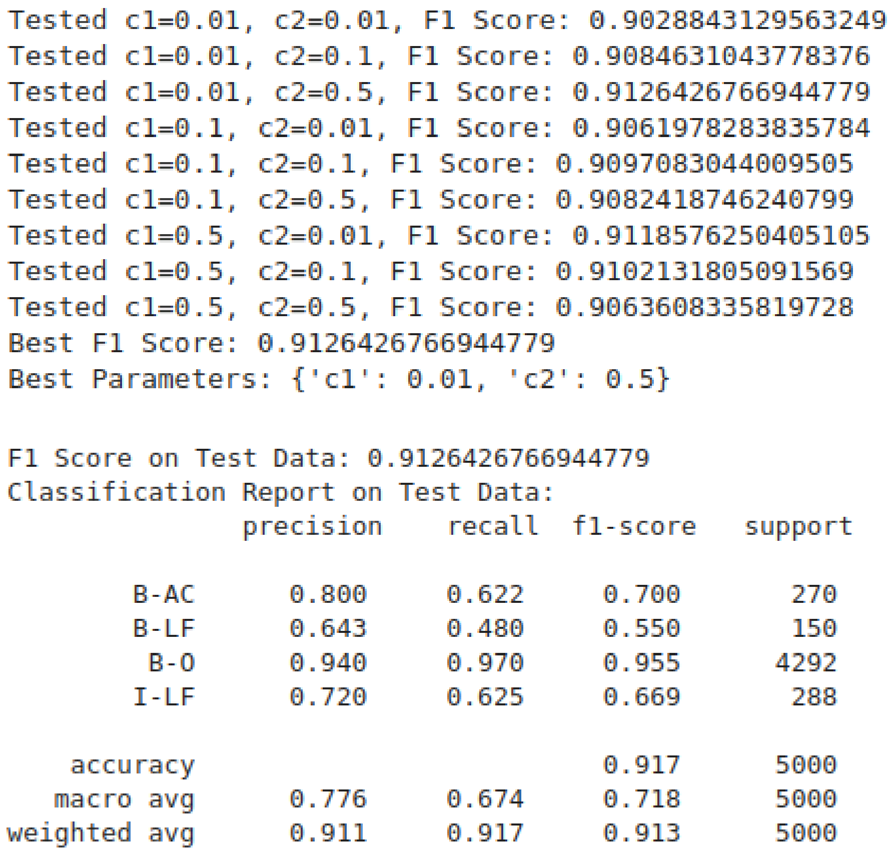


```
Fitting 5 folds for each of 6 candidates, totalling 30 fits
              precision    recall  f1-score   support

        B-AC       0.74      0.30      0.43       270
        B-LF       0.29      0.13      0.18       150
         B-O       0.89      0.97      0.93      4292
        I-LF       0.39      0.21      0.27       288

    accuracy                           0.86      5000
   macro avg       0.58      0.40      0.45      5000
weighted avg       0.83      0.86      0.84      5000
```

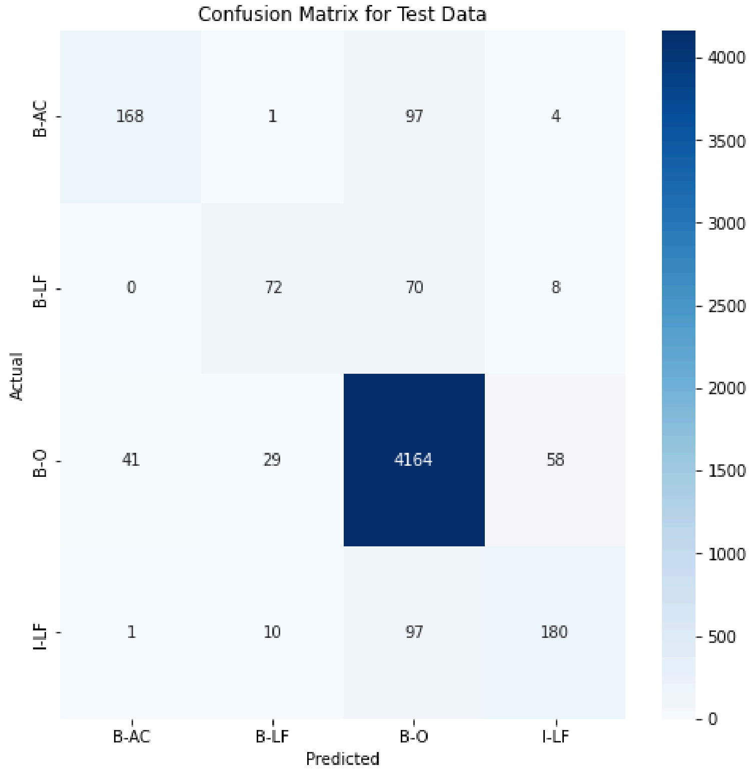


Confusion Matrix for SVM with GridSearchCV

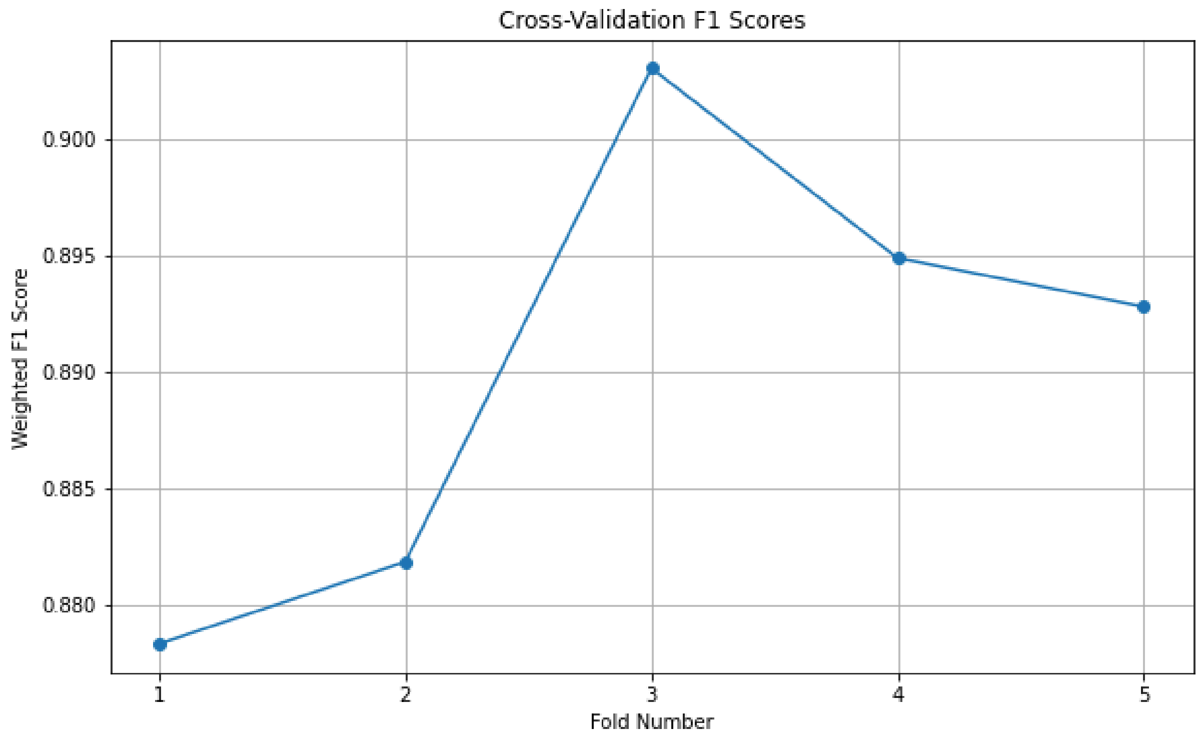
Using grid search to find the best regularization parameters for CRF:

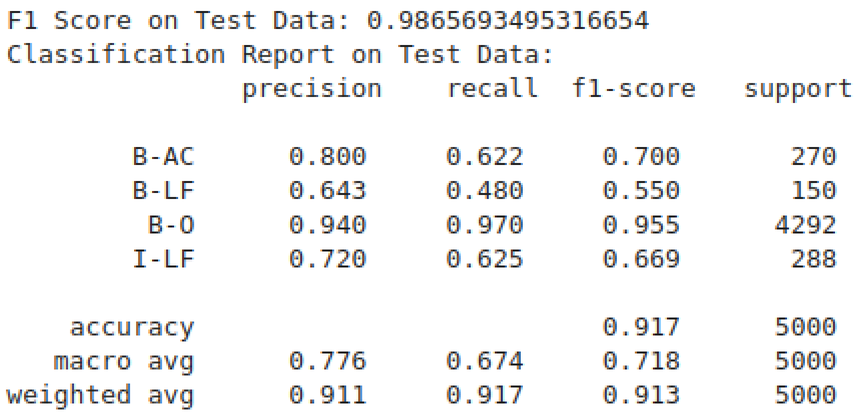


```
Tested c1=0.01, c2=0.01, F1 Score: 0.9028843129563249
Tested c1=0.01, c2=0.1, F1 Score: 0.9084631043778376
Tested c1=0.01, c2=0.5, F1 Score: 0.9126426766944779
Tested c1=0.1, c2=0.01, F1 Score: 0.9061978283835784
Tested c1=0.1, c2=0.1, F1 Score: 0.9097083044009505
Tested c1=0.1, c2=0.5, F1 Score: 0.9082418746240799
Tested c1=0.5, c2=0.01, F1 Score: 0.9118576250405105
Tested c1=0.5, c2=0.1, F1 Score: 0.9102131805091569
Tested c1=0.5, c2=0.5, F1 Score: 0.9063608335819728
Best F1 Score: 0.9126426766944779
Best Parameters: {'c1': 0.01, 'c2': 0.5}

F1 Score on Test Data: 0.9126426766944779
Classification Report on Test Data:
              precision    recall  f1-score   support

        B-AC      0.800     0.622     0.700       270
        B-LF      0.643     0.480     0.550       150
         B-O      0.940     0.970     0.955      4292
        I-LF      0.720     0.625     0.669       288

    accuracy                          0.917      5000
   macro avg      0.776     0.674     0.718      5000
weighted avg      0.911     0.917     0.913      5000
```



5-fold Crocss-vaildation best parameters for CRF:





```
F1 Score on Test Data: 0.9865693495316654
Classification Report on Test Data:
              precision    recall  f1-score   support

        B-AC      0.800     0.622     0.700       270
        B-LF      0.643     0.480     0.550       150
         B-O      0.940     0.970     0.955      4292
        I-LF      0.720     0.625     0.669       288

    accuracy                          0.917      5000
   macro avg      0.776     0.674     0.718      5000
weighted avg      0.911     0.917     0.913      5000
```

4. Observation of the outcomes

Experiment 1:

System 1: No Preprocessing
Best Performance. Maintaining the text's full context and syntax, including capitalisation, has been beneficial. This information can be particularly valuable in literature, where specific terms and entities are often capitalised.

System 2: Lowercase Preprocessing
Slight Decrease in Performance: The step of converting to lowercase may have caused the loss of crucial capitalisation features specific to entities like 'B-AC' (abbreviation) and 'B-LF' (long form), making it harder to distinguish between them.

System 3: Lowercase and Stop Words Removal

Significant Decrease in Performance: Eliminating stop words has diminished the clues CRF models rely on for sequence prediction. In texts, stop words can play a role in terminologies, and their removal might disrupt the necessary flow and structure for accurate entity detection.

Observations:

Imbalance in Classes: The dominance of the 'B-O' class indicates overfitting on this class while showing performance on less common classes like 'B-AC' and 'B-LF'.
Loss of Distinctive Features: Both converting to lowercase and removing stop words could result in losing features for the CRF model to capture the intricacies of biomedical terminology. When preprocessing data, it is crucial to avoid overgeneralising and missing out on the subtleties of patterns.

Improvements idea:

1. Strategic Data Preparation: Leverage domain expertise to maintain terms in their form.
2. Feature Enhancement: Integrate part of speech tagging or named entity attributes for feature sets.
3. Model Fine Tuning: Experiment with a range of hyperparameters and regularisation methods to prevent overfitting to the class.
4. Balanced Data Representation: Implement sampling techniques to ensure the representation of minority classes during training.

Experiment 2:

CRF - Better Performance: Capitalising on the full context and syntax of the text led to superior performance. CRF models utilise token sequences effectively, which can be crucial for entity recognition tasks within scientific literature where terms are often case-sensitive.

SVM - Decrease in Performance: Although the SVM model scored a weighted F1 Score of 0.888, it showed a decline in effectively distinguishing between the 'B-AC', 'B-LF', and 'I-LF' classes compared to the CRF. The SVM's classification report and the confusion matrix indicated misclassifications, especially among minority classes.

Observations from SVM:

The SVM's tendency to misclassify minority classes, as shown in the confusion matrix, suggests a model bias toward the overrepresented 'B-O' class.

SVM's lower performance in 'B-LF' and 'I-LF' could stem from the inadequacy of the feature set generated via DictVectorizer to capture the sequential dependencies and contextual nuances that CRF models inherently leverage.

Improvements idea:

1. Enhanced Feature Engineering: Develop richer feature representations that can capture the individual token attributes and contextual information around them. Advanced NLP techniques, such as embeddings or contextual features, could be beneficial.
2. Class Weight Adjustment: Implement class weight balancing within the SVM training process to improve the model's sensitivity to minority classes, addressing the class imbalance issue highlighted by the confusion matrix.
3. Data Augmentation: Increasing the representation of underrepresented classes in the training data could help improve the SVM's ability to recognise and classify these entities more accurately.

Experiment 3:

System 1: TF-IDF Vectorization without Data Balancing
Highest weighted F1 Score in the current experiment. The precision and recall for 'B-O' are high, strong performance on the dominant class.

System 2: Word2Vec with Data Balancing
Notable drop in weighted F1 Score (0.53). Despite data balancing, there is a significant decrease in precision for 'B-O' while recall for less frequent classes like 'B-LF' and 'I-LF' improved. This could indicate that while data balancing helps recognise minority classes, it may also lead to misclassifications of the majority class due to the model becoming less biased towards the 'B-O' class.

System 3: TF-IDF Vectorization with Data Balancing

Decreased weighted F1 Score (0.77) compared to System 1. Due to data balancing, this system shows improvements in recall for 'B-LF' and 'I-LF', yet there is a trade-off with a drop in precision, especially for the 'B-O' class.

Observations:

The poor performance of Word2Vec could be due to the loss of syntactic information that TF-IDF preserves. Word2Vec focuses on semantic meaning and may average discriminative features when used to create sentence embeddings. It also leads to a longer running time. The SVM may not effectively separate classes when features are purely semantic and not syntactically informative.

Methods like fasttext and glove were also implemented for this experiment. As it shows a 0 index for minority classes and increases the running time, it shows it is not suitable for the current experiment setup. It is removed eventually.

While data balancing helps improve the recognition of minority classes, data balancing appears to harm the classifier's ability to predict the majority class correctly.

Improvements idea:

1. Refined Vectorisation: a combination of TF-IDF and Word2Vec features or other embeddings that capture semantic and syntactic features to better represent the text data.
2. Class-Specific Models: Develop separate models for the minority classes or a hierarchical classification system that identifies the main class and then specialises in distinguishing sub-classes.
3. Post-Processing Heuristics: Introduce rules or heuristics in post-processing to refine SVM predictions, especially for borderline cases between classes.
4. Advanced Balancing Techniques: Apply more advanced balancing techniques that ensure minority classes are upsampled without degrading the SVM's ability to classify the majority class accurately.

Experiment 4:

SVM - After GridSearchCV, SVM showed a slight improvement but still lagged behind the CRF. The precision and recall for the 'B-O' class remained strong, but for minority classes like 'B-AC' and 'I-LF', the performance gains were minimal.
After hyperparameter tuning, the SVM seems unable to overcome the limitations in handling sequential data as effectively as the CRF, designed explicitly for sequence prediction tasks.

CRF - with parameters optimised through cross-validation outperformed the same CRF with Grid Search best parameters on the test set, showing an F1 Score of 0.98. Seems very high and might indicate data leakage or an overfitting issue. The model had prior access to test data and validation data during training, which would invalidate the test results.

5-fold cross-validation CRF- using only the training and validation data for cross-validation, the test set is left untouched for a final, unbiased evaluation. This could overfitting and prevent data leakage. Gives a more reasonable average F1 Score of 0.89.

Improvements idea:

1. Data Handling: More strict separation of training, validation, and test data to prevent data leakage. Re-evaluate the cross-validation procedure to confirm that test data was not used in any way during the model selection or parameter tuning processes.
2. Robustness Checks: Perform additional checks and validations to ensure the reported results are robust and reproducible. This may involve additional cross-validation rounds, using different data splits, or even hold-out validation sets.

5. Evaluation

Four experiments conducted provided valuable insights into the performance of various approaches for biomedical entity recognition. Each experiment has different aspects of model performance, highlighting strengths, weaknesses, and areas for improvement.

Experiment 1 explored the impact of preprocessing techniques on model performance. While System 1, without preprocessing, exhibited the best performance by preserving the text's full context and syntax, Systems 2 and 3 decreased performance due to lowercase conversion and stop words removal, respectively. The observations emphasised the importance of strategic data preparation, feature enhancement, and balanced data representation for improving model accuracy.

Experiment 2 compared the performance of CRF and SVM models, with CRF demonstrating superior performance in handling case-sensitive text data and capturing sequential dependencies effectively. The observations from SVM highlighted the need for enhanced feature engineering, class weight adjustment, and data augmentation techniques to address limitations in capturing subtle contextual details and mitigating class imbalance issues.

Experiment 3 focused on different tokenisation approaches; TF-IDF showed the highest F1 score, and Word2Vec exhibited notable performance degradation, indicating the importance of balancing semantic and syntactic features for accurate entity recognition. The findings challenge data balancing and the need for refined vectorisation techniques and advanced balancing strategies to achieve optimal performance.

Experiment 4 delved into hyperparameter optimisation for SVM and CRF models, with CRF outperforming SVM in handling sequential data effectively. However, concerns regarding potential data leakage and overfitting were raised, so another 5-fold cross-validation with separated train and validation datasets was added. The exploration of different model evaluation methodologies provided a view into unbiased model evaluation and addressed concerns of potential data leakage.

In conclusion, while striving for the most accurate model, it is essential to consider practical constraints such as computational cost and time limitations. The main reason some experiments were removed during the process, and in most cases, only train and test datasets were used. Machine learning experimentation and refinement require careful consideration of various factors beyond performance metrics, including interpretability, computational complexity, and resource efficiency. The findings from these experiments lay the foundation for future research and model development in biomedical entity recognition, emphasising the importance of continuous improvement and validation.

References:

Mehmetlaudatekman (2020) Text classification: SVM explained, Kaggle. Available at: https://www.kaggle.com/code/mehmetlaudatekman/text-classification-svm-explained (Accessed: 26 March 2024).

Bedi, G. (2020) Simple guide to text classification(nlp) using SVM and Naive Bayes with python, Medium. Available at: https://medium.com/@bedigunjit/simple-guide-to-text-classification-nlp-using-svm-and-naive-bayes-with-python-421db3a72d34 (Accessed: 30 March 2024).

Getoor, L. and Taskar, B. (2019) Introduction to statistical relational learning. Cambridge, MA: MIT Press.

Sutton, C., McCallum, A. (2012). An Introduction to Conditional Random Fields for Relational Learning. In: Introduction to Statistical Relational Learning. MIT Press.

Wallace, B.C. (2012). Machine Learning for Conditional Random Fields. Tutorial Presentation at the Association for Computational Linguistics (ACL).

Wang, Y., Afzal, N., Fu, S., Wang, X., & Liu, H. (2018). A Comparison of Word Embeddings for the Biomedical Natural Language Processing. In Proceedings of the International Conference on Bioinformatics and Biomedical Engineering.

Wei, C. H., Peng, Y., Leaman, R., Davis, A. P., Mattingly, C. J., Li, J., & Wiegers, T. C. (2015). Overview of the BioCreative V chemical disease relation (CDR) task.