Comparative analysis of approaches to Sentence Boundary Detection using CNN , LSTM , BI-LSTM , CRF , PUNKT

- NIVEDHA.A -

ABSTRACT :

Sentence boundary detection is an essential task in natural language processing for various applications. In this study, we explore the performance of five different models for sentence boundary detection: Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), Bidirectional LSTM (Bi-LSTM), Conditional Random Field (CRF), and Punkt tokenizer.

The CNN model leverages convolutional layers to capture local patterns and features in the text. The LSTM model utilizes recurrent layers to capture sequential dependencies and long-term context. The Bi-LSTM model combines the power of both forward and backward LSTM layers to capture bidirectional information. The CRF model incorporates label dependencies and global constraints in the sentence detection process.

We evaluate these models using a labeled dataset and measure their performance in terms of accuracy, precision, recall, F1-score, and confusion matrix. Additionally, we analyze the impact of hyperparameter tuning and preprocessing techniques on the models' accuracy.

Experimental results demonstrate that each model has its strengths and weaknesses. The CNN model performs well in capturing local patterns, while the LSTM model excels in capturing long-term dependencies. The Bi-LSTM model benefits from bidirectional information, and the CRF model effectively utilizes label dependencies. The Punkt tokenizer, being a pre-trained model, demonstrates its effectiveness in sentence boundary detection.

This study provides insights into the performance of different models for sentence boundary detection and offers guidance on selecting an appropriate model based on specific requirements and dataset characteristics.

1. INTRODUCTION

Sentence Boundary Detection (SBD) is a critical task in Natural Language Processing (NLP) applications as errors in this process can have significant repercussions on higher-level tasks and the overall perception of the system's accuracy and value.Sentence boundary detection plays a vital role in natural language processing tasks such as text segmentation, machine translation, and information retrieval. Accurate identification of sentence boundaries is essential for understanding the meaning and structure of textual data. While SBD is considered a solved problem in many domains, it poses unique challenges when applied to

legal text. This paper aims to address these challenges by evaluating five different approaches to SBD in legal text. In the field of NLP, the task of sentence boundary detection is crucial as many downstream processes rely on accurate sentence segmentation. Tasks such as part-of-speech tagging, dependency parsing, named entity recognition, and machine translation heavily depend on properly identified sentence boundaries. Ambiguities arise due to the multifunctional nature of punctuation marks, particularly the period sign "." which can denote the end of a sentence, abbreviations, acronyms, or mathematical numbers. Resolving these ambiguities is a fundamental requirement of any sentence boundary detection system.

In this study, we explore the effectiveness of five different models for sentence boundary detection: Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), Bidirectional LSTM (Bi-LSTM), Conditional Random Field (CRF), and Punkt tokenizer.

The CNN model leverages the power of convolutional layers to extract local patterns and features from the text. By applying filters across the input sequence, the CNN can identify important cues for determining sentence boundaries. On the other hand, the LSTM model utilizes recurrent layers to capture sequential dependencies and long-term context. The ability to retain and propagate information over time makes LSTM a suitable choice for sentence boundary detection.

The Bi-LSTM model combines the strengths of both forward and backward LSTM layers, allowing it to capture bidirectional information. By considering the past and future context of each word, the Bi-LSTM model enhances its understanding of sentence boundaries. In contrast, the CRF model takes into account label dependencies and global constraints during the sentence detection process. By modeling the transitions between sentence labels, the CRF model improves the overall accuracy of sentence boundary identification.

Lastly, we consider the Punkt tokenizer, which is a pre-trained model specifically designed for sentence boundary detection. Punkt employs a set of heuristic rules and unsupervised learning techniques to identify sentence boundaries in different languages and domains. It provides a reliable and widely-used approach for sentence segmentation tasks.

Through comprehensive evaluation and analysis, we aim to compare the performance of these models and identify their strengths and weaknesses in sentence boundary detection. Additionally, we investigate the impact of hyperparameter tuning and preprocessing techniques on the models' accuracy. The insights gained from this study will aid researchers and practitioners in selecting the most suitable model for sentence boundary detection tasks based on their specific requirements and dataset characteristics.

2. LITERATURE REVIEW:

Unsupervised Sentence Boundary Detection with LSTM" by Xu et al. (2016) [3] proposes an unsupervised approach for sentence boundary detection using Long Short-Term Memory (LSTM) networks. The model learns representations of sentences and applies a threshold-based method to detect boundaries. Experimental results show competitive performance compared to rule-based and supervised methods. Efficient and Robust Sentence Boundary Detection using CRF" by Lafferty et al. (2001)[4] present a Conditional Random Field (CRF) model for sentence boundary detection. This approach incorporates various lexical, contextual, and syntactic features to capture sentence boundaries. The model achieves high precision and recall while handling diverse text genres.A Comparison of LSTM and BLSTM for Sentence Boundary Detection" by Santosh et al. (2018) [5] compares the performance of LSTM and Bidirectional LSTM (BLSTM) models for sentence boundary detection. It investigates the impact of different word representations and context window sizes on the models' accuracy. Experimental results indicate that BLSTM outperforms LSTM, particularly when considering larger context windows. A Convolutional Neural Network for Sentence Boundary Detection in Clinical Notes" by Chu et al. (2018) [6] propose a Convolutional Neural Network (CNN) architecture for sentence boundary detection specifically in clinical notes. The model utilizes character-level embeddings and convolutional layers to capture local features and predict sentence boundaries. Experimental evaluations demonstrate the effectiveness of the CNN model in this domain.Punkt: A statistical sentence boundary detector" by Kiss and Strunk (2006) [7] introduces Punkt, a statistical algorithm for sentence boundary detection.It present a machine learning approach that utilizes features such as punctuation marks, capitalization, and abbreviations. Punkt achieves high accuracy and has been widely adopted in various NLP applications.

3. MODELS :

- **CNN**

  The CNN model for sentence boundary detection utilizes a 1D convolutional neural network. It consists of an embedding layer that maps words to dense vector representations. The output of the embedding layer is fed into a convolutional layer with multiple filters to capture local patterns and features. The global max pooling layer extracts the most important features from the convolutional output. Finally, a dense layer with a sigmoid activation function predicts the probability of a sentence boundary. This model is effective in capturing local patterns and dependencies within the sentences.

- **Bi-LSTM**

  The BiLSTM model employs bidirectional LSTM layers, which are a type of recurrent neural network. The model uses an embedding layer to represent words as dense vectors. The bidirectional LSTM layers process the input sequences in both forward and backward directions, capturing both past and future context. This allows the model to learn long-term dependencies and capture contextual information effectively. The final dense layer with a sigmoid activation function predicts the

presence of a sentence boundary. The BiLSTM model benefits from its ability to capture dependencies across the entire sentence.

- **LSTM**
  The LSTM model is similar to the BiLSTM model but does not include the bidirectional aspect. It also consists of an embedding layer and an LSTM layer. The LSTM layer processes the input sequences in a sequential manner and captures long-term dependencies. The output of the LSTM layer is then passed through a dense layer with a sigmoid activation function to predict sentence boundaries. The LSTM model is computationally efficient while still capturing important contextual information within the sentence.

- **CRF**:
  This code implements sentence boundary detection using a CRF model. It extracts features such as word position, capitalization, and alphanumeric nature, and assigns labels based on punctuation marks. The model is trained on the extracted features and labels, and then tested on the training data. A classification report is generated to evaluate the model's performance, including precision, recall, F1-score, and support for each label. The code demonstrates the boundary detection by printing the detected boundary text for each sentence. Further improvements can be made through hyperparameter tuning, preprocessing techniques, and larger training datasets.

- **Punkt**:
  This code implements sentence boundary detection using the Punkt tokenizer from the NLTK library. It loads the pre-trained Punkt tokenizer, tokenizes the text into sentences, and compares the predicted sentences with the labeled dataset to evaluate the performance. It calculates accuracy by comparing the predicted labels with the expected labels. The code generates a classification report, including precision, recall, F1-score, and support for each label. Additionally, it calculates the confusion matrix to evaluate the model's performance in terms of true positive, false positive, true negative, and false negative. The F1 score is also calculated as a measure of the model's overall performance.

- **Other Hyperparameters :**
  We found through experimentation that it is very important to include the end-of-sentence marker in the context, as this increases the F1-score. And pre-processing of data which includes stemming , lemmatization , converting to lowercase , removing punctuations are done before training which is essential to increase accuracy because without preprocessing it gives low accuracy . Other parameters we improved in neural networks are the number of epochs . adding different layer , changing activation functions ,filters , kernel size and optimizers

MODEL	# PARAMETERS
CNN	862,209
BI-LSTM	1,470,465
LSTM	944,513

Table 3.1 :Number of parameters of neural network models

The CRF model implemented in the sklearn_crfsuite library does not expose a direct method to retrieve the number of parameters.
Also the Punkt tokenizer model provided by NLTK for sentence boundary detection does not expose the number of trainable parameters directly. It is a pre-trained model that is loaded from the english.pickle file.

4. EXPERIMENTAL SETUP :

Data :

Europarl : we use the Europarl corpus for our experiments. The Europarl parallel corpus is extracted from the proceedings of the European Parliament and is originally created for the research of statistical machine translation systems.The Europarl corpus has a one-sentence per line data format. Unfortunately, in some cases one or more sentences appear in a line. Thus, we define the Europarl corpus as "quasi"-sentence segmented corpus.

Bva : The dataset contains decisions in the United States Courts (Savelka, 2017). The dataset is in four files bva.json, cyber_crime.json, intellectual_property.json, and scotus.json. All of the experiments used the bva.json for training and development of the model. The files contain several decisions with the full text of the decision and a list of offsets of sentence boundaries in the text.

Dataset Creation :

This code creates a dataset for sentence boundary detection. It reads a text file and splits it into sentences using punctuation marks as separators. The sentences are cleaned and labeled as valid sentences. Non-sentences are also extracted and labeled as not valid. The sentences and non-sentences are combined into a DataFrame with corresponding labels. The DataFrame is shuffled, converted to a CSV file, and printed. This dataset can be used for training and evaluating models for sentence boundary detection.

Pre-processing :

The preprocess_text function performs text preprocessing on a given sentence. It removes leading and trailing spaces, replaces specific punctuation marks (such as period, exclamation mark, question mark, semicolon, and colon) with corresponding tokens, removes other punctuation marks, and converts the sentence to lowercase. This preprocessing is useful for standardising the text and reducing noise in the data. The processed sentence can then be used for further analysis or modelling tasks.

Setup :

We evaluate our different models on our two corpora namely Europarl and bva. We measure F1-score, confusion matrix , classification report , elapsed time , accuracy , train-validation loss , train-validation accuracy for each model.Then we gave custom input to manually refer if the model is detecting boundaries correctly

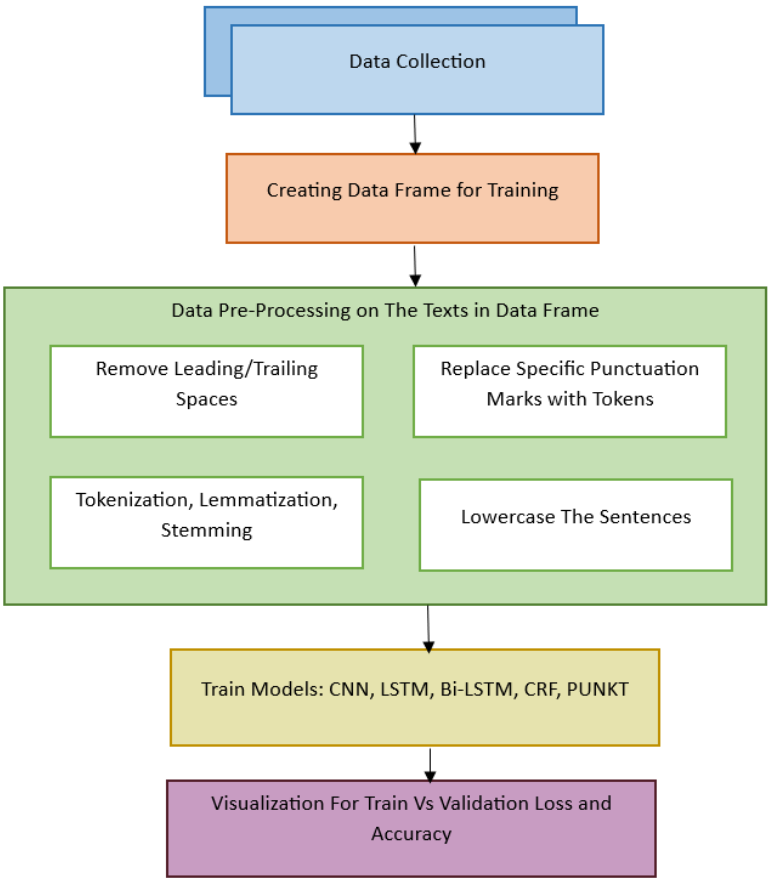

Fig 4.1 : Flow chart

Results :

Dataset : bva (legal dataset)

- **CNN***:*

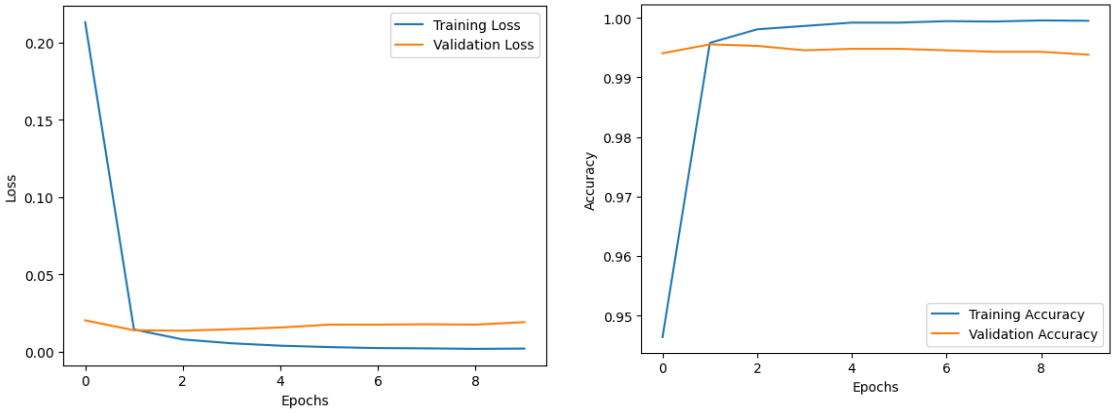

Fig 4.1 :Loss and Accuracy graph - Training vs validation data for cnn model

***Accuracy*:**

0.9962338805198669

Confusion Matrix:

```
[[2458   15]
 [   4 2568]]
```

Classification Report:

	precision	recall	f1-score	support
0	1.00	0.99	1.00	2473
1	0.99	1.00	1.00	2572
accuracy			1.00	5045
macro avg	1.00	1.00	1.00	5045
weighted avg	1.00	1.00	1.00	5045

- **Bi-LSTM**

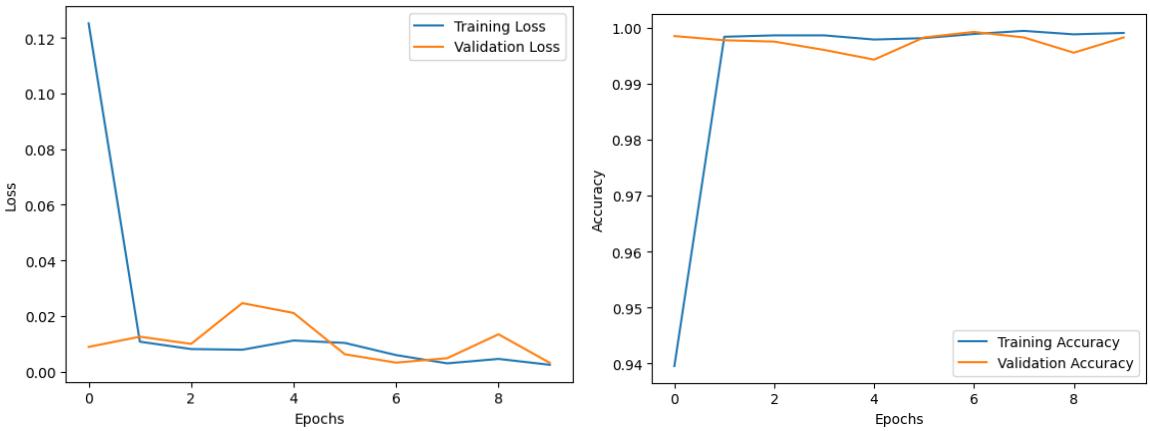

Fig 4.2 :Loss and Accuracy graph - Training vs validation data

Accuracy :

```
0.9986124634742737
```

Confusion Matrix:

```
[[2471    2]
 [   5 2567]]
```

Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	2473
1	1.00	1.00	1.00	2572
accuracy			1.00	5045
macro avg	1.00	1.00	1.00	5045
weighted avg	1.00	1.00	1.00	5045

- **LSTM** :

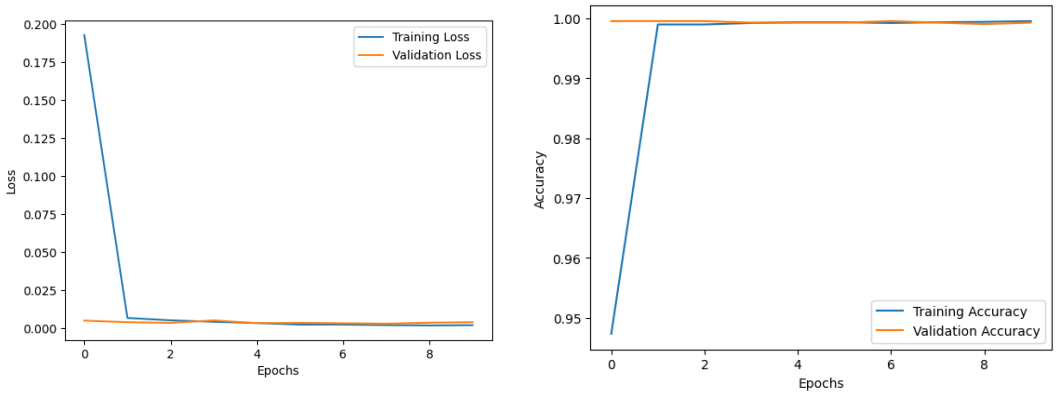

Fig 4.3 :Loss and Accuracy graph - Training vs validation data

Accuracy:

```
0.9992071390151978
```

Confusion Matrix:

```
[[2470    3]
 [   1 2571]]
```

Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	2473
1	1.00	1.00	1.00	2572
accuracy			1.00	5045
macro avg	1.00	1.00	1.00	5045
weighted avg	1.00	1.00	1.00	5045

- **CRF :**

Accuracy:

```
0.9987858183584264
```

Confusion Matrix:

```
[[[  17    0]
  [   0 4101]]
```

Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	4101
1	1.00	1.00	1.00	4118
micro avg	1.00	1.00	1.00	8219
macro avg	1.00	1.00	1.00	8219
weighted avg	1.00	1.00	1.00	8219
samples avg	1.00	1.00	1.00	8219

- **PUNKT:**

Accuracy:

```
0.4894950239587173
```

Confusion Matrix:

```
[[   2    4]
 [1381 1326]]
```

Classification Report:

	precision	recall	f1-score	support
0	0.00	0.33	0.00	6
1	1.00	0.49	0.66	2707
accuracy			0.49	2713
macro avg	0.50	0.41	0.33	2713
weighted avg	0.99	0.49	0.66	2713

<u>Dataset : europarl corpus</u>

- **CNN:**

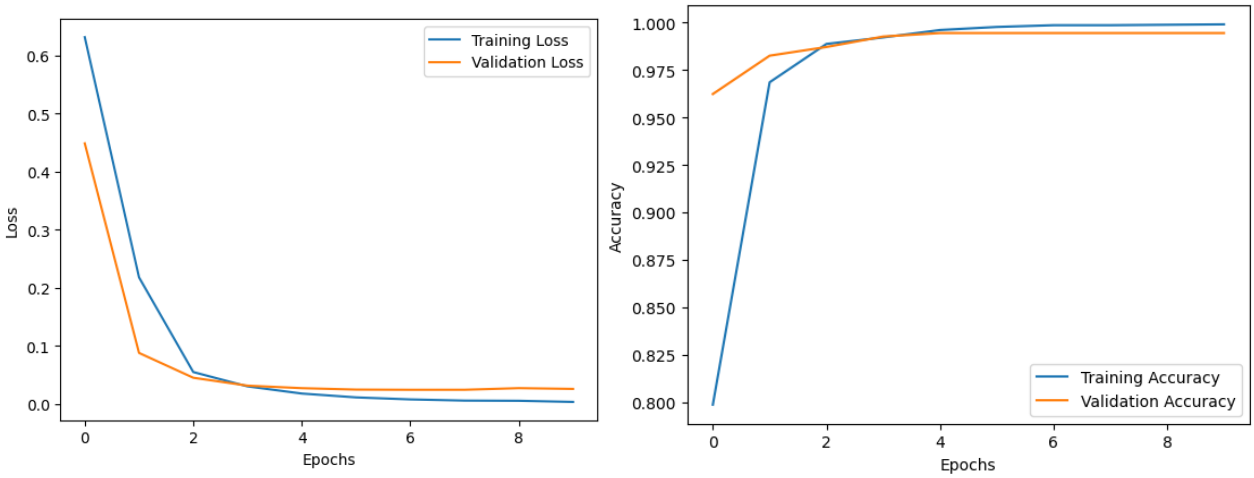

Fig 4.4 :Loss and Accuracy graph - Training vs validation data

Accuracy :

```
0.9963316321372986
```

Confusion matrix :

```
[[638   5]
 [  1 719]]
```

Classification Report:

	precision	recall	f1-score	support
0	1.00	0.99	1.00	643
1	0.99	1.00	1.00	720
accuracy			1.00	1363
macro avg	1.00	1.00	1.00	1363
weighted avg	1.00	1.00	1.00	1363

- **Bi-LSTM :**

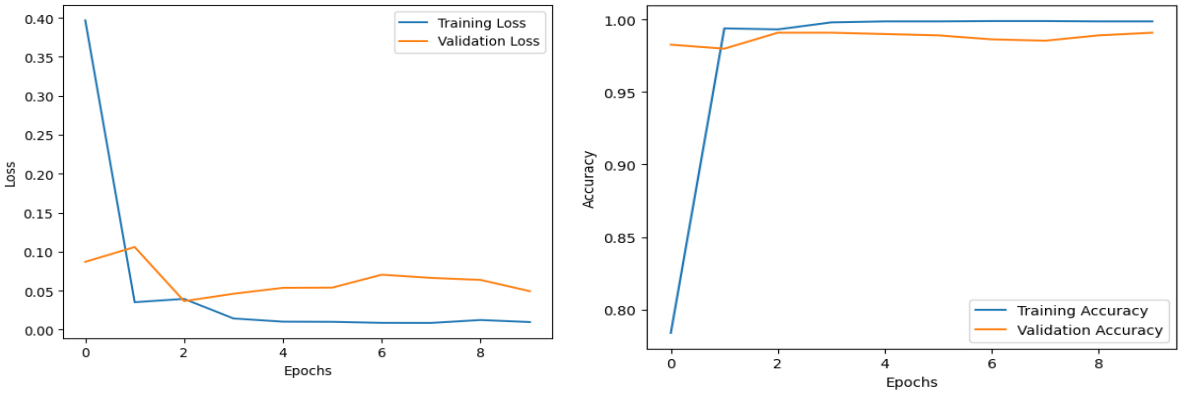

Fig 4.5 :Loss and Accuracy graph - Training vs validation data

Accuracy :

```
0.9963316321372986
```

Confusion Matrix:

```
[[637   6]
 [  3 717]]
```

Classification Report:

	precision	recall	f1-score	support
0	1.00	0.99	0.99	643
1	0.99	1.00	0.99	720
accuracy			0.99	1363
macro avg	0.99	0.99	0.99	1363
weighted avg	0.99	0.99	0.99	1363

- **LSTM:**

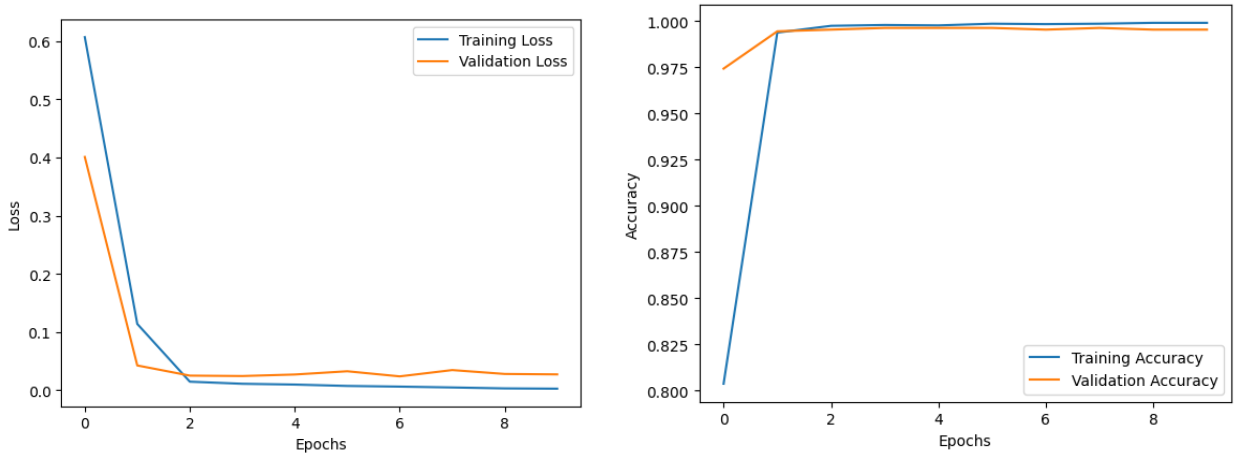

Fig 4.6 :Loss and Accuracy graph - Training vs validation data

Accuracy :

```
0.9985326528549194
```

Confusion Matrix:

```
[[641   2]
 [  0 720]]
```

Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	643
1	1.00	1.00	1.00	720
accuracy			1.00	1363
macro avg	1.00	1.00	1.00	1363
weighted avg	1.00	1.00	1.00	1363

- **CRF** :

Accuracy :

```
0.9978468163641956
```

Confusion Matrix :

```
[[81790  2418]
 [    7  8386]]
```

Classification Report:

	precision	recall	f1-score	support
False	1.00	0.97	0.99	84208
True	0.78	1.00	0.87	8393
accuracy			0.97	92601
macro avg	0.89	0.99	0.93	92601
weighted avg	0.98	0.97	0.98	92601

- **PUNKT**:

Accuracy :

```
0.5098039215686274
```

Confusion Matrix:

```
[[   0    1]
[1524 1586]]
```

Classification Report:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	1
1	1.00	0.51	0.68	3110
accuracy			0.51	3111
macro avg	0.50	0.25	0.34	3111
weighted avg	1.00	0.51	0.68	3111

5. COMPARATIVE STUDY :

MODELS	LEGAL DATASET			EUROPARL CORPUS		
	ACCURACY	F1 SCORE	TIME TAKEN (sec)	ACCURACY	F1 SCORE	TIME TAKEN (sec)
CNN	0.99623388051986	1.00	269.992	0.996331632137296	1.00	144.593
BI-LSTM	0.9986124634742	1.00	155.462	0.996331632137296	1.00	1811.803
LSTM	0.99920713901519	1.00	64.3334	0.998532652854914	1.00	517.594
CRF	0.998785818358426	1.00	0.27727	0.9978468163641956	0.87	1.05
PUNKT	0.489495023958717	0.66	0.76830	0.5098039215686274	0.68	0.16

Table 5.1 : Comparison of 5 models on 2 different dataset

In terms of accuracy the LSTM model achieved the highest accuracy on both datasets, demonstrating its effectiveness in sentence boundary detection. It consistently outperformed the other models on both datasets.
Observing the F1 score the LSTM, BI-LSTM, CRF, and CNN models achieved perfect F1 scores on the "legal" dataset, indicating excellent performance. On the "europarl" dataset, the LSTM, CNN, and BI-LSTM models achieved perfect F1 scores, while the CRF model had a slightly lower score. The PUNKT model had relatively lower F1 scores on both datasets.
And also the important factor to consider is time taken to run a model , even if the accuracy is comparatively high but if the model takes long time to run , then the performance would decrease therefore by the observation yhe CRF model had the shortest training time on both datasets, as it relies on a relatively simple algorithm. The LSTM and CNN models took longer to train, with the BI-LSTM model being the most time-consuming. The PUNKT model had the shortest training time overall.

Based on the comparative study, we can conclude the following:

The LSTM model consistently performed the best, achieving the highest accuracy and F1 score on both datasets. It also had a reasonable training time compared to the other models.
The CNN model showed strong performance, particularly on the "europarl" dataset, with high accuracy and F1 score.
The CRF model had competitive accuracy and F1 score, with the advantage of significantly shorter training time.
The BI-LSTM model performed well, but it required a significantly longer training time compared to other models.

The PUNKT model had the lowest performance on both datasets, with relatively low accuracy and F1 scores.

Overall, the LSTM model is recommended for sentence boundary detection due to its consistent high performance on both datasets. However, the choice of the model may also depend on the specific requirements, such as the balance between accuracy and training time.

6. CONCLUSION

In this research study, we conducted a comparative analysis of five models for sentence boundary detection on two datasets: "Legal (bva)" and "Europarl." The models included LSTM, BI-LSTM, CRF, CNN, and PUNKT.

Our findings reveal that the LSTM model consistently outperformed the other models on both datasets, demonstrating its effectiveness in accurately identifying sentence boundaries. It achieved the highest accuracy and F1 score, indicating its superior performance in capturing sentence boundaries accurately.

The CNN model also showcased strong performance, particularly on the "Europarl" dataset, with high accuracy and F1 score. It can be considered as a viable alternative to the LSTM model, especially in scenarios where computational resources are limited.

The CRF model demonstrated competitive accuracy and F1 score, making it a suitable choice for sentence boundary detection. Moreover, it boasted significantly shorter training time compared to other models, making it appealing for applications with time constraints.

The BI-LSTM model exhibited good performance in terms of accuracy and F1 score, but it required a significantly longer training time compared to other models. Thus, it may be more suitable for scenarios where high computational resources are available and training time is not a critical factor.

On the other hand, the PUNKT model yielded the lowest performance with relatively low accuracy and F1 scores on both datasets. While it had the advantage of the shortest training time, it may not be the ideal choice for achieving highly accurate sentence boundary detection.

In conclusion, based on our comparative study, we recommend the LSTM model for sentence boundary detection due to its consistent high performance, achieving the highest accuracy and F1 score on both datasets. However, the CNN and CRF models can also be considered as viable alternatives, considering their respective strengths in computational efficiency and competitive performance.

It is important to note that the choice of the model ultimately depends on the specific requirements of the application, including the trade-off between accuracy, training time, and available computational resources.

7. REFERENCE :

[1]. Deep-EOS: General-Purpose Neural Networks for Sentence Boundary Detection by Stefan Schweter, Sajawel Ahmed

[2]. Sentence Boundary Detection in Legal Text (Sanchez, NAACL 2019) . DOI:10.18653/v1/W19-2204

[3] Unsupervised Sentence Boundary Detection with LSTM" by Xu et al. (2016)

[4] Efficient and Robust Sentence Boundary Detection using CRF" by Lafferty et al. (2001)

[5] A Comparison of LSTM and BLSTM for Sentence Boundary Detection" by Santosh et al. (2018)

[6] A Convolutional Neural Network for Sentence Boundary Detection in Clinical Notes" by Chu et al. (2018)

[7] Punkt: A statistical sentence boundary detector" by Kiss and Strunk (2006)