NLP Assignment 2

Group Members:

- ➢ Sanya Madan (2021561)
- ➢ Prashasti Gupta (2021346)
- ➢ Shruti Gupta (2021353)
- ➢ Parisha Agrawal (2021270)

Part 1: Data Preparation

Part 1A (dataset 1)

- The NER_TRAIN_JUDGEMENT.json file is first split into train and val files using stratified split.
- Code first filters out items with empty 'annotations' lists and ensures there is a valid result in the annotations.
- Then extracts labels from the filtered data for stratified splitting based on the first annotation result in each item.
- Stratified splitting is performed using scikit-learn's train_test_split.
- The split data into training and testing subsets are written into separate JSON files, 'NER_TRAIN.json and 'NER_VAL.json', respectively with the same structure as the original files.
- After splitting, the assign_bio_labels function tokenizes the text based on space and initializes all labels as 'O' (indicating the token is outside of any entity).
- It iterates through annotations, extracts entity information, and finds the corresponding token indices for each entity in the text.
- For each token, it assigns a BIO label ('B' for the beginning of an entity, 'I' for inside) based on the entity's position in the text and label annotations which are added after the underscore.
- Each data entry is processed to assign these BIO labels, and the processed data is saved to new JSON files.

Original:


```
{
    "id": "a6149e6f168e43a38085c4a5b3ad5d29",
    "annotations": [
        {
            "result": [
                {
                    "value": {
                        "start": 4,
                        "end": 17,
                        "text": "N. M. Ghatate",
                        "labels": [
                            "OTHER_PERSON"
                        ]
                    },
                    "id": "GJLFUMKX",
                    "from_name": "label",
                    "to_name": "text",
                    "type": "labels"
                },
                {
                    "value": {
                        "start": 36,
                        "end": 47,
                        "text": "CA. 1658/80",
                        "labels": [
                            "CASE_NUMBER"
                        ]
                    },
                    "id": "13UP806E",
                    "from_name": "label",
                    "to_name": "text",
                    "type": "labels"
                }
            ]
        }
    ],
    "data": {
        "text": "Dr. N. M. Ghatate for Respondent in CA. 1658/80."
    },
    "meta": {
        "source": "civil_supremecourts judgement https://indiankanoon.org/doc/174675/"
    }
```


Pre-processed:


```
{
    "id": "a6149e6f168e43a38085c4a5b3ad5d29",
    "text": "Dr. N. M. Ghatate for Respondent in CA. 1658/80.",
    "labels": [
        "O",
        "B_OTHER_PERSON",
        "I_OTHER_PERSON",
        "I_OTHER_PERSON",
        "O",
        "O",
        "O",
        "B_CASE_NUMBER",
        "I_CASE_NUMBER"
    ]
},
```


Part 1B (dataset 2)

- The function BIO_labels takes raw words and aspects as inputs and generates BIO labels according to the indices and labels in the original file. It tokenizes the raw words using split by space, initializes all labels as 'O', then iterates through aspects and marks the corresponding tokens as 'B' (for the beginning of an aspect) or 'I' (for inside).
- Each entry in the dataset is processed by extracting words and aspects, indices and corresponding labels from the original file and generating BIO labels.

- Processed data is stored in a dictionary format, which is then converted to json files corresponding to each train, test and validation.

Original:

```
{
  "raw_words": "Great laptop that offers many great features !",
  "words": [
    "Great",
    "laptop",
    "that",
    "offers",
    "many",
    "great",
    "features",
    "!"
  ],
  "aspects": [
    {
      "index": 0,
      "from": 6,
      "to": 7,
      "polarity": "POS",
      "term": [
        "features"
      ]
    }
  ],
  "opinions": [
    {
      "index": 0,
      "from": 5,
      "to": 6,
      "term": [
        "great"
      ]
    }
  ]
},
```

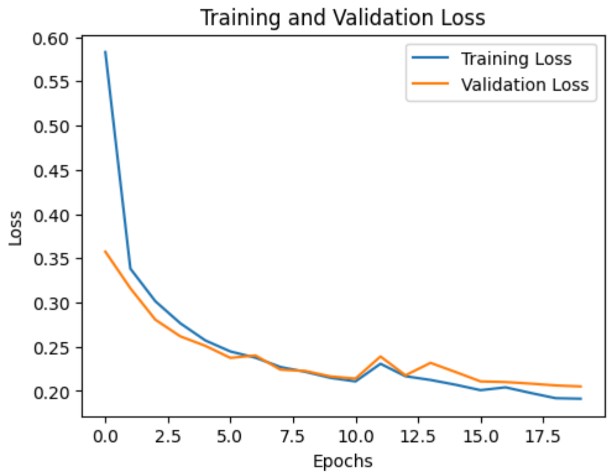
Pre-processed:

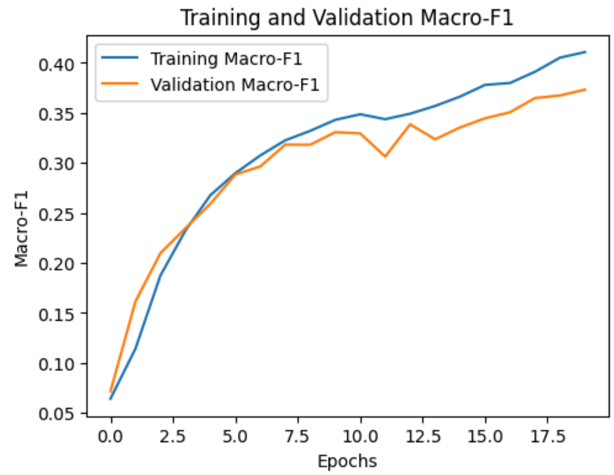
```
"4": {
    "text": "Great laptop that offers many great features !",
    "labels": [
        "O",
        "O",
        "O",
        "O",
        "O",
        "O",
        "B",
        "O"
    ]
},
```

Part 2: Baseline Models Implementation

Installation required: pip install `gensim`

- **RNN on dataset 1 using word2vec**

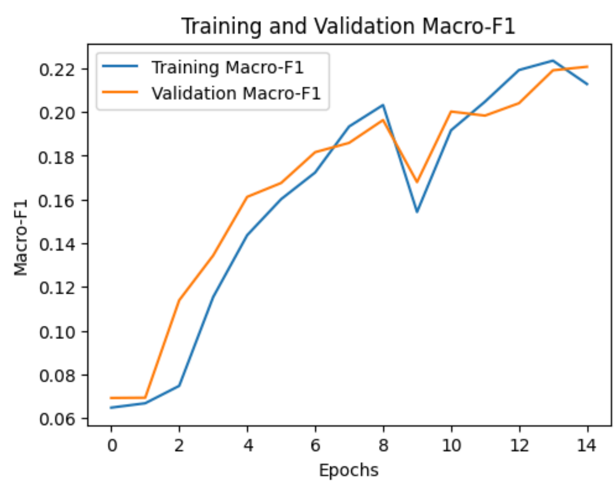




Epoch 20/20, Training Loss: 36.8973, Validation Loss: 7.1763, Training Macro-F1: 0.4108, Validation Macro-F1: 0.3763

Analysis: The model is effective while learning and generalizing. The loss consistently decreases for both training and validation datasets which indicates good generalization. The gap between training and testing curves are small indicating that the model efficiently learns the model and does not overfit. Similarly, the F1 scores increase for both training and validation datasets suggesting that the model learns and generalizes well on unseen data. After 20 epochs, validation loss is 7.1763 while the validation macro f1 is 0.3763.

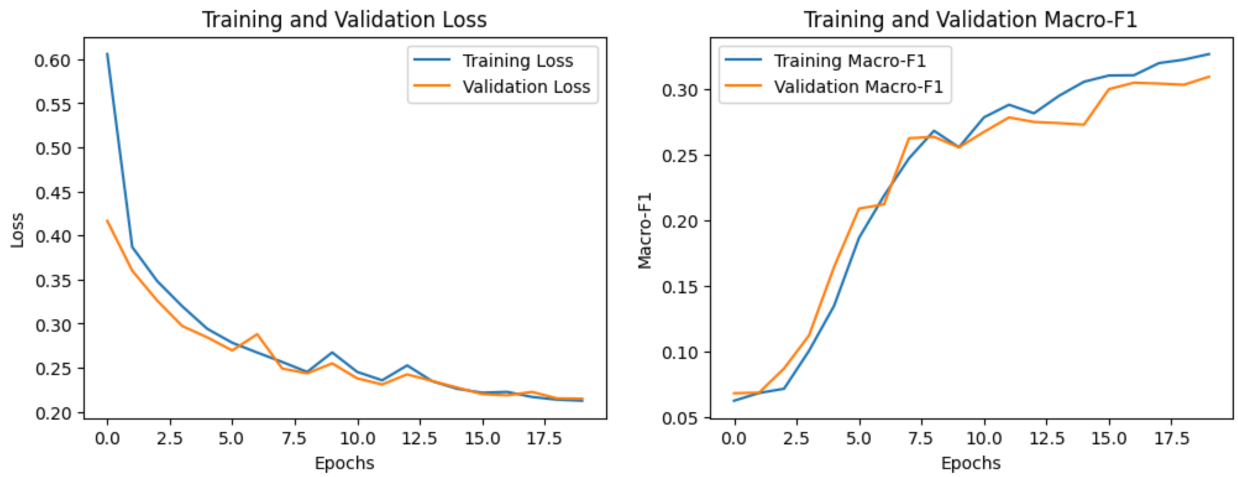
- **RNN on dataset 1 using GloVe**



Epoch 15/15, Training Loss: 48.9729, Validation Loss: 8.6550, Training Macro-F1: 0.2210, Validation Macro-F1: 0.2190

Analysis: The model is effective while learning and generalizing. The loss consistently decreases for both training and validation datasets while the F1 scores increase suggesting that model learns and generalizes well on unseen data. After 15 epochs, validation loss is 8.6550 while the validation macro f1 is 0.2190.

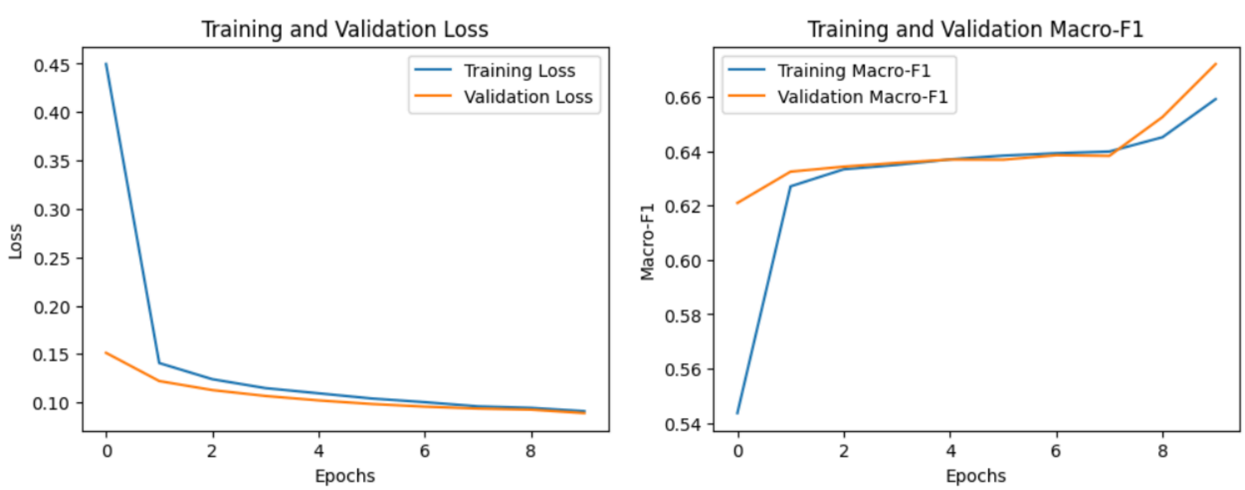
- **RNN on dataset 1 using fasttext**



Epoch 20/20, Training Loss: 41.0159, Validation Loss: 7.5194, Training Macro-F1: 0.3262, Validation Macro-F1: 0.3090

Analysis: The model demonstrates effectiveness in both learning and generalization, as indicated by a consistent decrease in loss for both training and validation datasets. This consistency suggests that the model is adept at learning and doesn't suffer from overfitting. Moreover, the increasing F1 scores for both datasets imply that the model learns and generalizes effectively to unseen data. The loss decreases sharply after 2 epochs. After 20 epochs, validation loss is 7.5194 while the validation macro f1 is 0.3090.

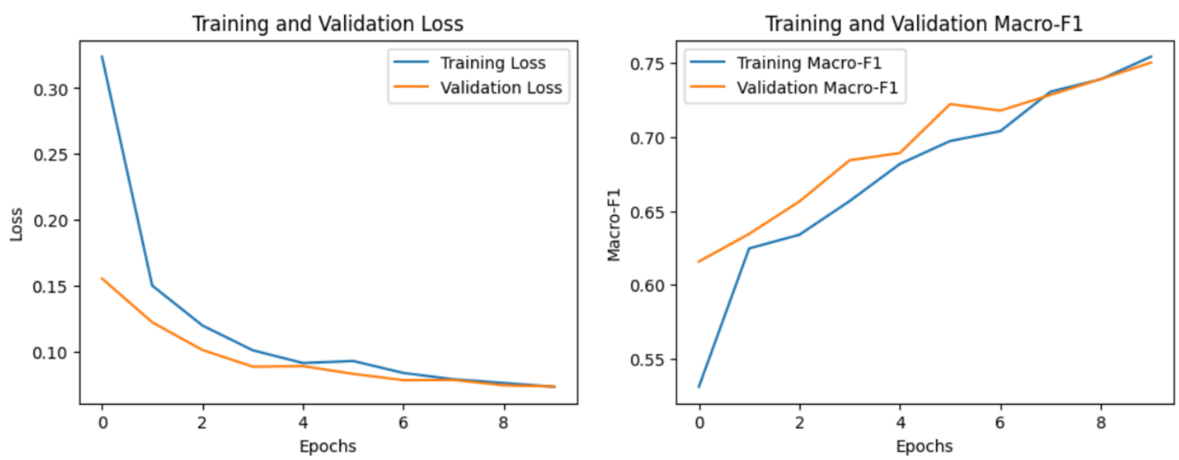
- **RNN on dataset 2 using word2vec**



Epoch 10/10, Training Loss: 2.6986, Validation Loss: 0.6376, Training Macro-F1: 0.6536, Validation Macro-F1: 0.6720

Analysis:
- Both training and validation losses decrease as the number of epochs increases, which indicates that the model is learning and improving over time.
- The validation loss is consistently lower than the training loss, suggesting that the model generalizes well to unseen data. The rate of decrease in both losses slows down after 6 epochs, indicating that the model is converging.
- Both training and validation Macro-F1 scores increase as the number of epochs increases, showing that the model is getting better at classifying the data.
- The validation Macro-F1 score is consistently lower than the training Macro-F1 score, which is expected as the model is tuned to the training data.
- Overall, the model performs reasonably well, achieving a validation Macro-F1 score of around 0.67 after 10 epochs.

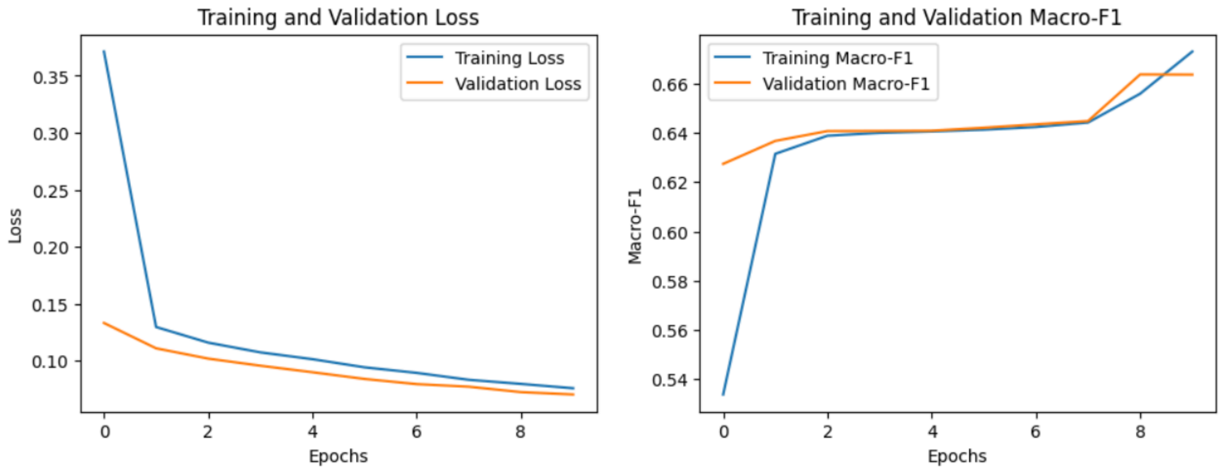
- **<u>RNN on dataset 2 using GloVe</u>**



Epoch 10/10, Training Loss: 2.1220, Validation Loss: 0.5140, Training Macro-F1: 0.7545, Validation Macro-F1: 0.7505

Analysis: The model demonstrates effective learning and generalization, with both training and validation loss decreasing consistently across epochs. The validation loss decreases more slowly, indicating good generalization. The training and validation Macro-F1 scores increase steadily, reaching around 0.75 after 10 epochs. This suggests that the model learns effectively and generalizes well to unseen data. After 10 epochs, validation loss is 0.5140 while the validation macro f1 is 0.7505.

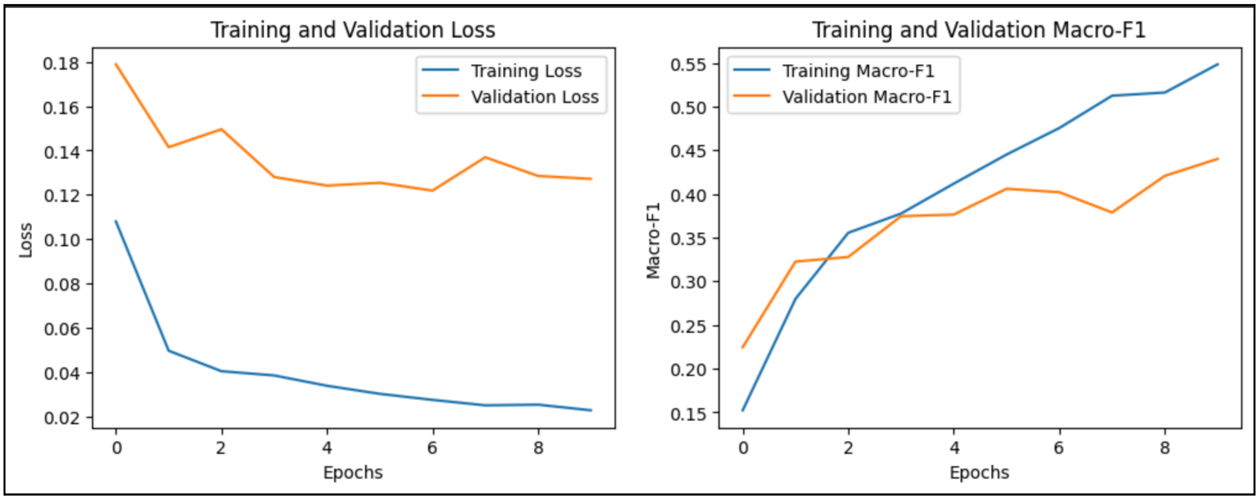
- **<u>RNN on dataset 2 using fasttext</u>**



Epoch 10/10, Training Loss: 2.2017, Validation Loss: 0.4932, Training Macro-F1: 0.6730, Validation Macro-F1: 0.6636

Analysis: The model demonstrates effective learning and generalization, with both training and validation loss decreasing consistently across epochs. The validation loss decreases more slowly, indicating good generalization. This suggests that the model learns effectively and generalizes well to unseen data. The loss decreases and the f1 increases sharply after 1 epoch. After 10 epochs, validation loss is 0.4932 while the validation macro f1 is 0.6636.

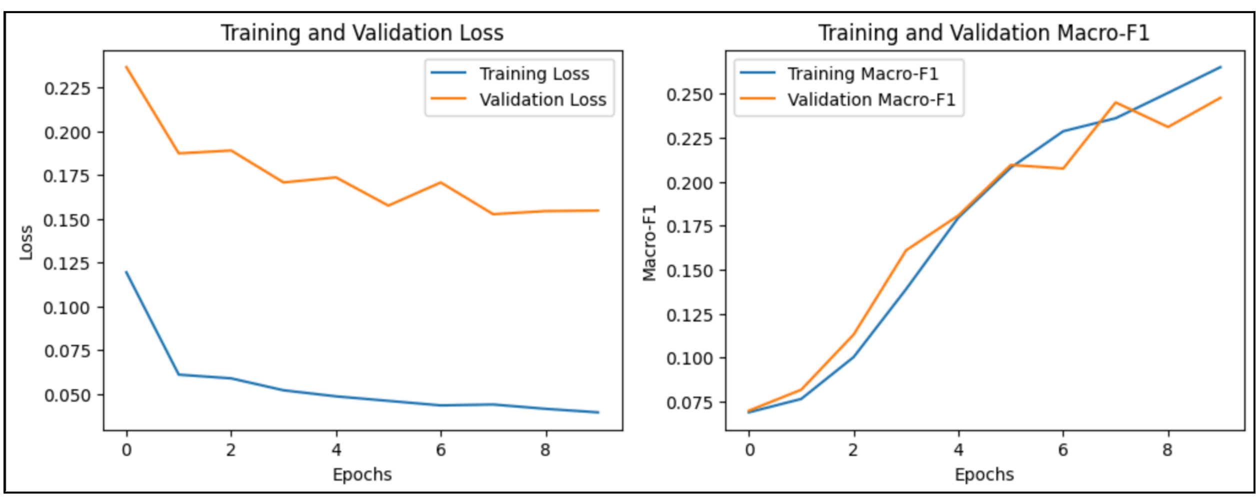
- **LSTM on dataset 1 using word2vec**



Epoch 10/10, Training Loss: 4.3810, Validation Loss: 0.4520, Training Macro-F1: 0.5485, Validation Macro-F1: 0.4402

Analysis: There is a large gap between the loss curves of the training and the validation datasets but a very narrow gap in their macro f1 scores. This indicates that the model is overfitting the data to some extent however it is still able to learn the underlying patterns in the data well, which may be memorizing some details of the training set. After 10 epochs, validation loss is 0.4932 while the validation macro f1 is 0.6636.

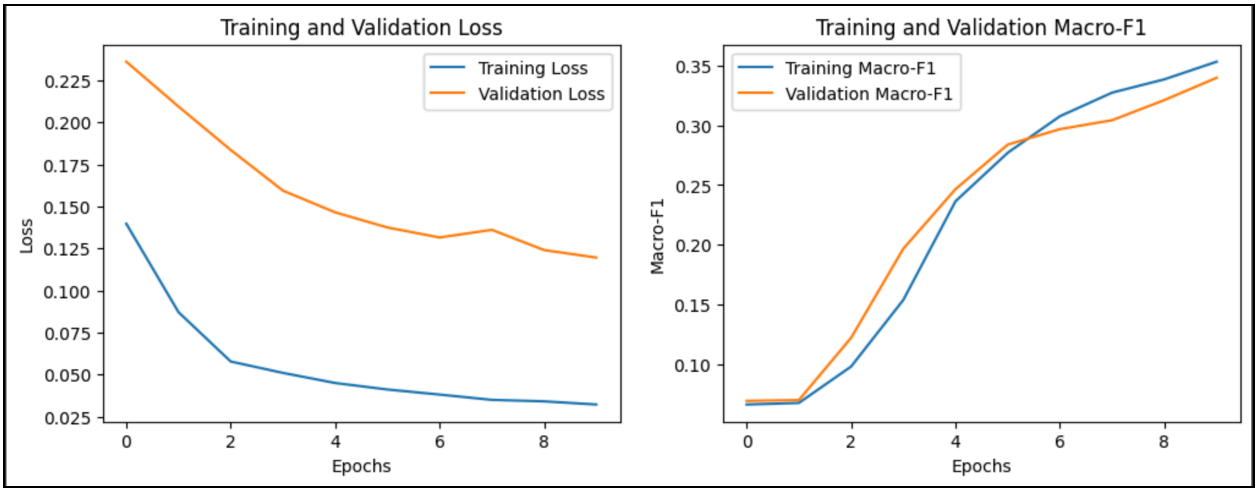
- **LSTM on dataset 1 using GloVe**



Epoch 8/10, Training Loss: 13.8146, Validation Loss: 6.9458, Training Macro-F1: 0.1389, Validation Macro-F1: 0.1347

Analysis: There is a large gap between the loss curves of the training and the validation datasets but a very narrow gap in their macro f1 scores. This indicates that the model is overfitting the data to some extent however it is still able to learn the underlying patterns in the data well, which may be memorizing some details of the training set. The F1 values are significantly low suggesting that model along with the word embedding does not work well on the given dataset. After 10 epochs, validation loss is 6.9458 while the validation macro f1 is 0.1347.

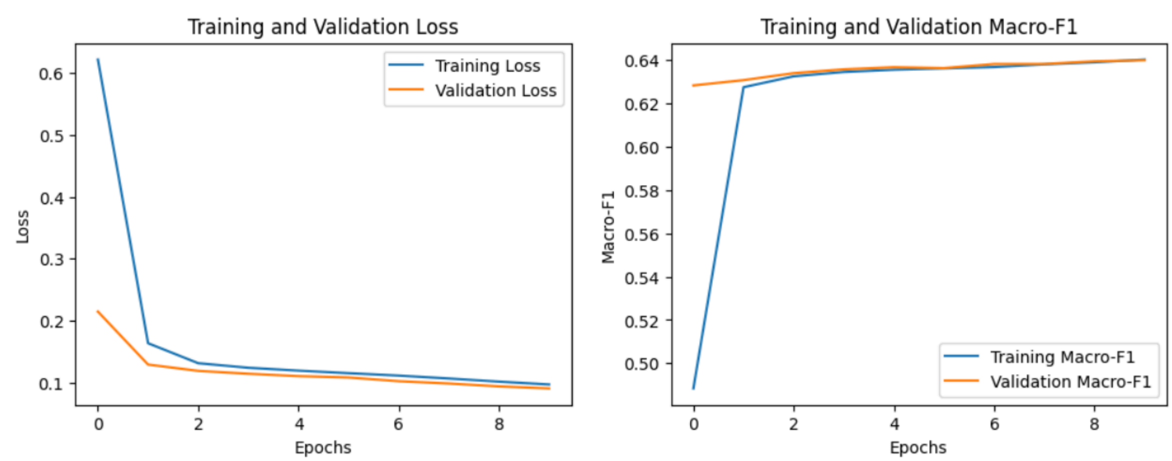
- **LSTM on dataset 1 using fasttext**



Epoch 10/10, Training Loss: 6.2220, Validation Loss: 4.1870, Training Macro-F1: 0.3533, Validation Macro-F1: 0.3399

Analysis: There is a large gap between the loss curves of the training and the validation datasets but a very narrow gap in their macro f1 scores. This indicates that the model is overfitting the data to some extent however it is still able to learn the underlying patterns in the data well, which may be memorizing some details of the training set. After 10 epochs, validation loss is 4.1870 while the validation macro f1 is 0.3399.

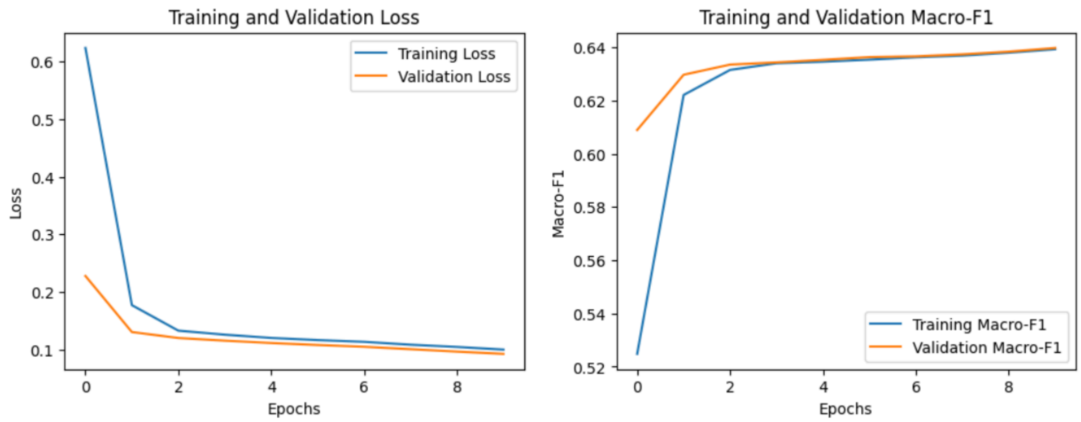
- **LSTM on dataset 2 using word2vec**



Epoch 10/10, Training Loss: 2.8155, Validation Loss: 0.6430, Training Macro-F1: 0.6399, Validation Macro-F1: 0.6393

Analysis: The model demonstrates effective learning and generalization, with both training and validation loss decreasing consistently across epochs. The loss decreases sharply initially and then slowly after 1 epoch. Similar increase pattern can be observed in the macro f1 scores. After 10 epochs, validation loss is 2.8155 while the validation macro f1 is 0.6393.

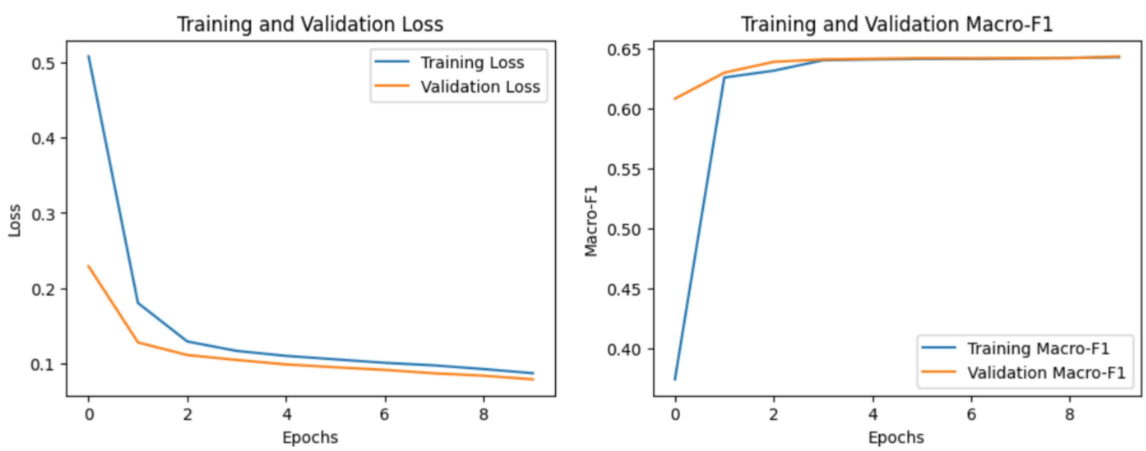
- **LSTM on dataset 2 using GloVe**



Epoch 10/10, Training Loss: 2.0712, Validation Loss: 0.4699, Training Macro-F1: 0.6755, Validation Macro-F1: 0.6550

Analysis: The model demonstrates effective learning and generalization, with both training and validation loss decreasing consistently across epochs. The loss decreases sharply initially and then slowly after 1 epoch. Similar increase patterns can be observed in the macro f1 scores. After 10 epochs, validation loss is 0.4699 while the validation macro f1 is 0.6550.

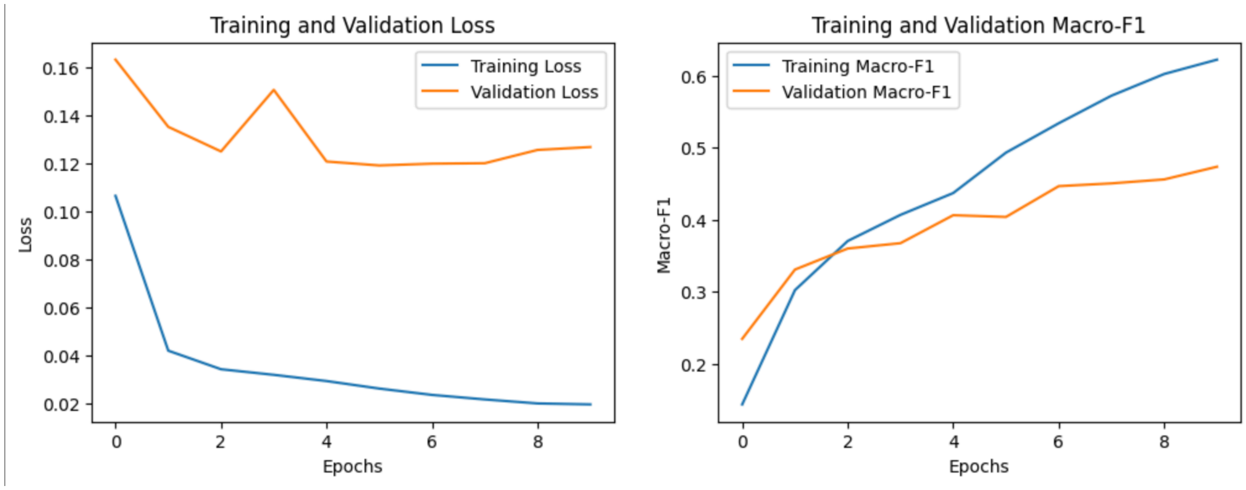
- **LSTM on dataset 2 using fasttext**



Epoch 10/10, Training Loss: 2.6696, Validation Loss: 0.5812, Training Macro-F1: 0.6423, Validation Macro-F1: 0.6428

Analysis: The model exhibits proficient learning and generalization, evidenced by a steady decrease in both training and validation loss throughout epochs. Initially, the loss diminishes sharply before settling into a more gradual decline. Similarly, the training and validation Macro-F1 scores experience a rapid ascent early on, reaching approximately 0.62 after the first epoch, followed by a steady increase thereafter. The minimal disparity between the training and validation F1 scores implies the model effectively learns and generalizes to new data.

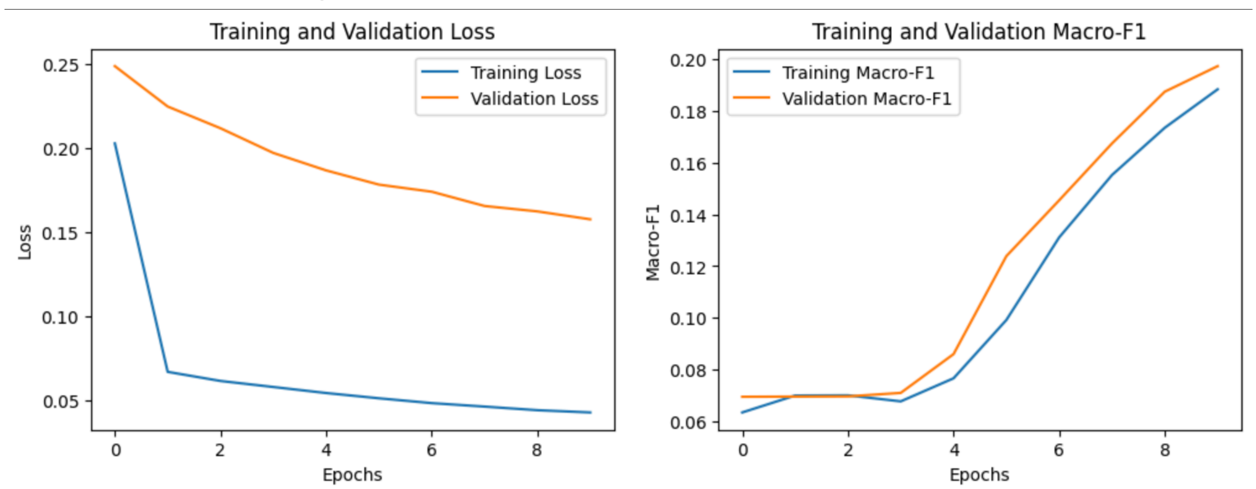
GRU on dataset 1 using word2vec



Epoch 10/10, Training Loss: 3.8216, Validation Loss: 4.4412, Training Macro-F1: 0.6223, Validation Macro-F1: 0.4735

Analysis: There is a large gap between the loss curves of the training and the validation datasets but a very narrow gap in their macro f1 scores. This indicates that the model is overfitting the data to some extent however it is still able to learn the underlying patterns in the data well, which may be memorizing some details of the training set. After 10 epochs, validation loss is 4.1870 while the validation macro f1 is 0.3399.

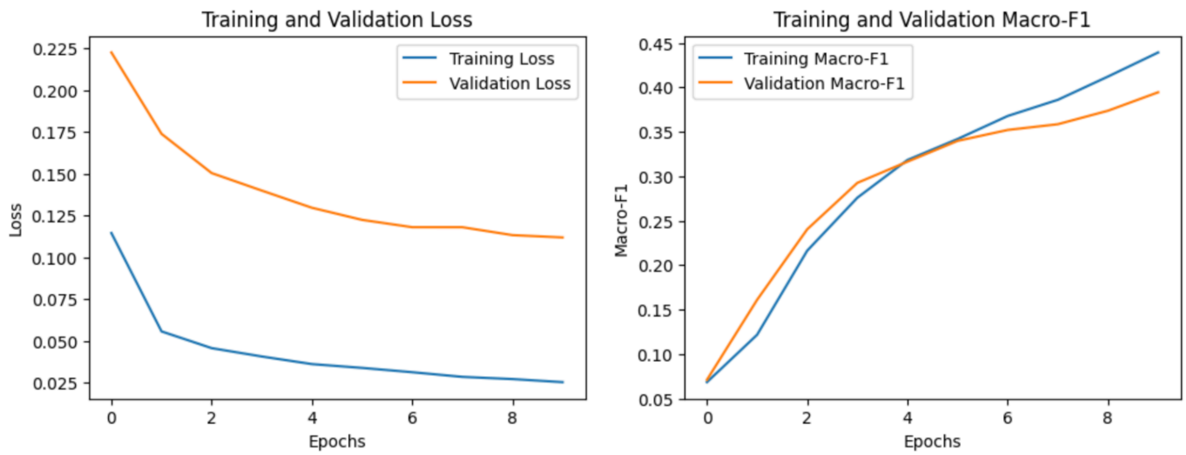
- **GRU on dataset 1 using GloVe**



Epoch 15/15, Training Loss: 5.6434, Validation Loss: 5.3262, Training Macro-F1: 0.4108, Validation Macro-F1: 0.3241

Analysis: Both training and validation loss decrease consistently across epochs. The training loss decreases rapidly and then becomes steady indicating effective learning. The validation loss decreases more slowly, indicating good generalization to unseen data. The training and validation Macro-F1 scores increase steadily, reaching around 0.20 after 10 epochs. This model is not as good due to less F1, the accuracy is good but the precision or recall might be less.

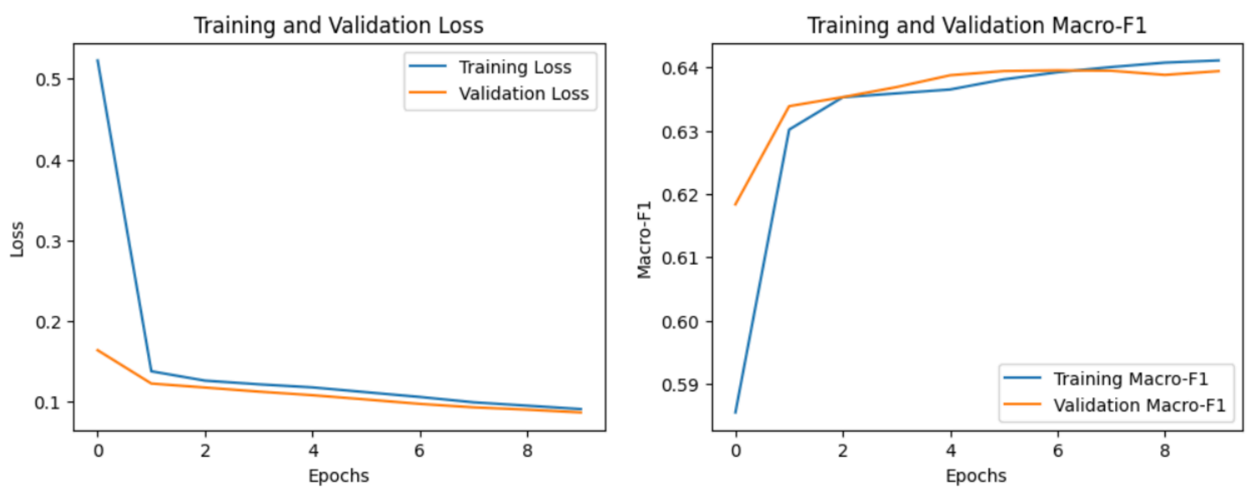
- **GRU on dataset 1 using fasttext**



Epoch 10/10, Training Loss: 4.8831, Validation Loss: 3.9140, Training Macro-F1: 0.4392, Validation Macro-F1: 0.3944

Analysis: The training and validation data show big differences in their loss curves, but their macro f1 scores are very close. This suggests that the model is learning the data's patterns well, even though it might be too focused on specific details from the training set. After 10 rounds of training, the validation loss is 0.125, and the validation macro f1 score is 0.38, indicating that the model performs well on new, unseen data.

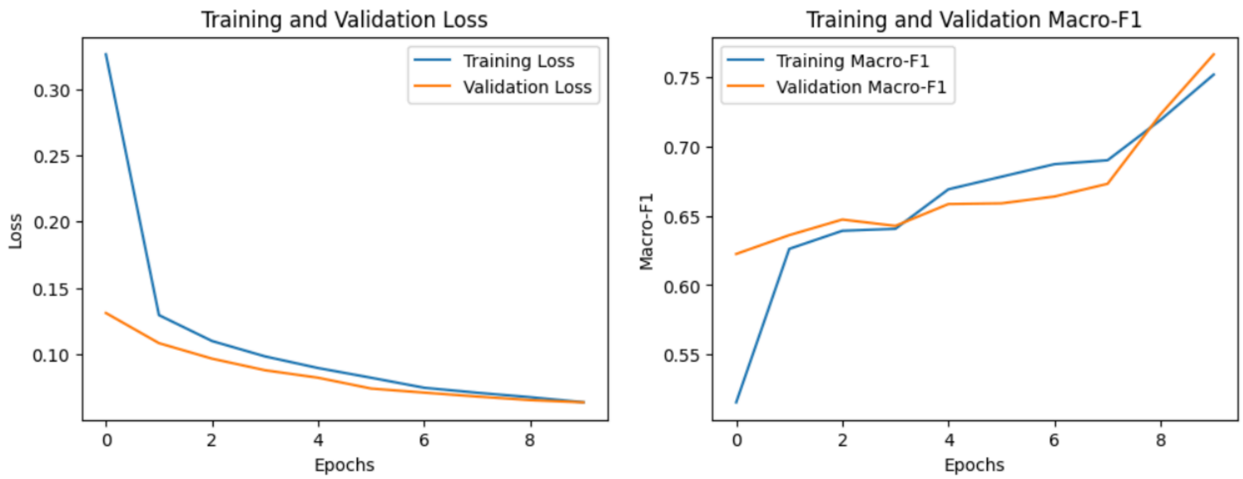
- **GRU on dataset 2 using word2vec**



Epoch 10/10, Training Loss: 2.6387, Validation Loss: 0.6078, Training Macro-F1: 0.6411, Validation Macro-F1: 0.6394

Analysis: The training loss decreases sharply indicating effective learning, while the validation loss decreases slowly, showing it also generalizes well. The F1 score also increases sharply till 0.64 in 10 epochs. The model demonstrates effective learning and generalization.

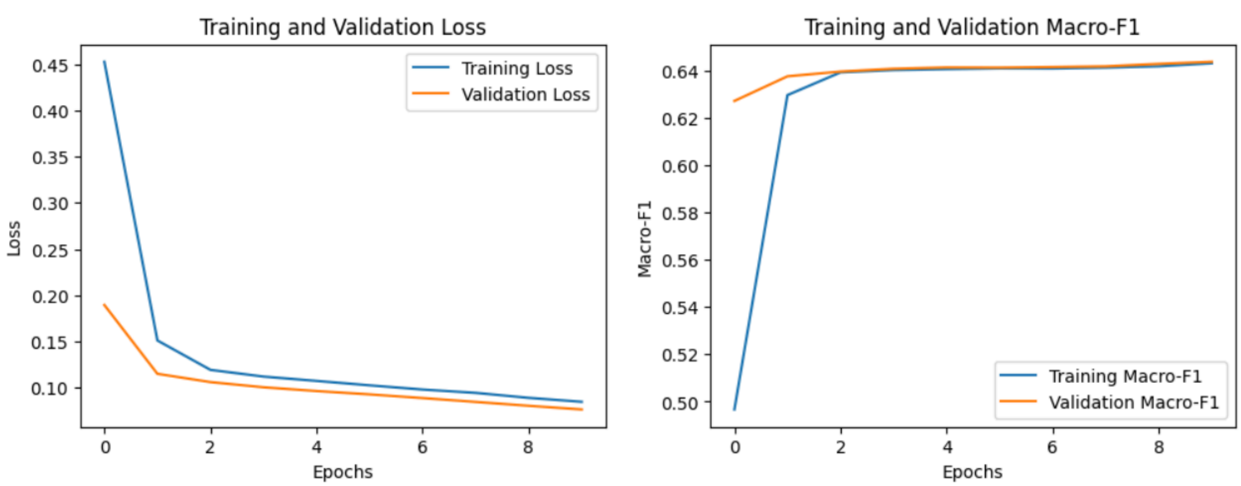
- **GRU on dataset 2 using GloVe**



Epoch 10/10, Training Loss: 1.8423, Validation Loss: 0.4441, Training Macro-F1: 0.7520, Validation Macro-F1: 0.7666

Analysis: The model exhibits efficient learning and generalization, as evidenced by a consistent decrease in both training and validation loss throughout epochs. Along with that, the validation loss shows a slower decline, implying strong generalization capabilities. Finally, the training and validation Macro-F1 scores steadily rise, peaking at approximately 0.7520 and 0.7666 respectively after 10 epochs. These results suggest the model effectively learns from the data and can generalize effectively to unseen data, indicating its robustness and reliability for real-world applications.

- **GRU on dataset 2 using fasttext**



Epoch 10/10, Training Loss: 2.4529, Validation Loss: 0.5340, Training Macro-F1: 0.6432, Validation Macro-F1: 0.6438

Analysis: The model demonstrates effective learning and generalization, with both training and validation loss decreasing consistently across epochs. The validation loss decreases more slowly, indicating good generalization. The training and validation Macro-F1 scores increase steadily, reaching around 0.64 after 10 epochs, though it converges after around 4 epochs and training further on the same may cause overfitting. As the gap between training and validation F1 score is small, it suggests that the model learns effectively and generalizes well to unseen data.

Note: Due to the huge size of the pt files, we have used two separate notebooks (one for dataset 1 and the other for dataset 2) to load the .pt files for each model in part 2 with each embedding.

<u>**Part 3: BiLSTM-CRF Model Implementation**</u>

- **BiLSTM-CRF on dataset 1 using word2vec**

Epoch 20/20, Training Loss: 10.4564, Validation Loss: 7.0715, Training Macro-F1: 0.8457, Validation Macro-F1: 0.5073

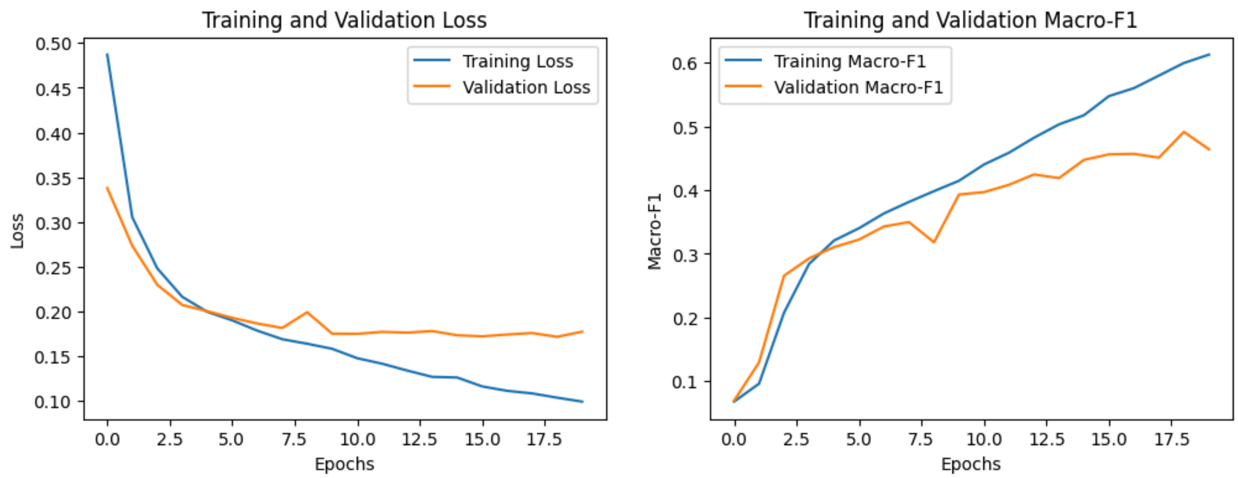
Analysis:
The loss plot shows the training and validation losses decreasing steadily over epochs, indicating the model is learning and improving its performance. The F1 plot also shows that training and validation macro-F1-scores are increasing over epochs, indicating that the model is becoming more accurate. We also have to prevent overfitting and stop as model loss becomes converging which we can see here that as loss stabilizes, we are not training for more epochs. After 10 epochs, validation loss is 7.0715 while the validation macro f1 is 0.5073.

- **BiLSTM-CRF on dataset 1 using GloVe**

Epoch 20/20, Training Loss: 10.5041, Validation Loss: 9.3608, Training Macro-F1: 0.7785, Validation Macro-F1: 0.4063

Analysis: The loss plot illustrates a consistent decline in both training and validation losses across epochs, suggesting the model is actively learning and enhancing its performance. Similarly, the F1 plot depicts a progressive increase in both training and validation macro-F1 scores, indicating the model's improving accuracy over time. To prevent overfitting, it's imperative to halt training as the model's loss stabilizes. In this instance, after 10 epochs, the validation loss stands at 9.3608, with a corresponding validation macro F1 score of 0.4063.

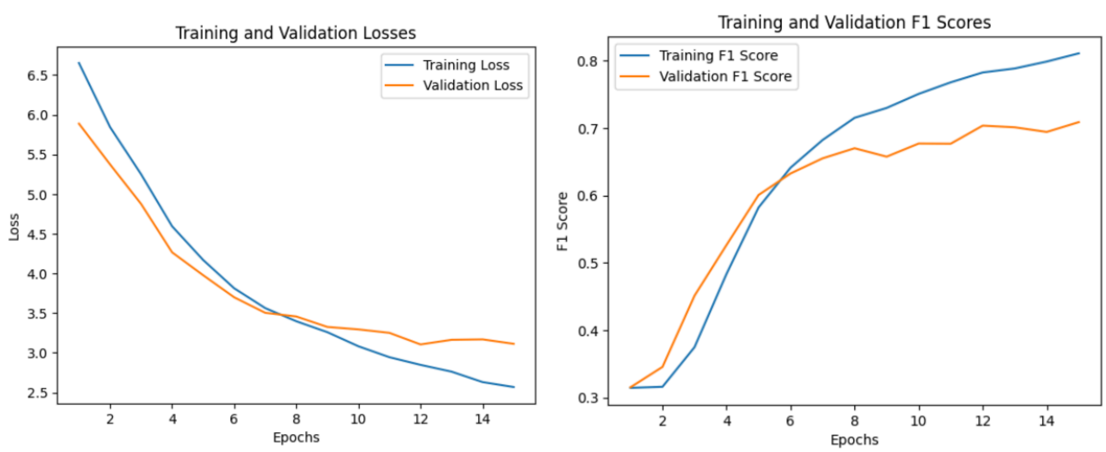
- **BiLSTM-CRF on dataset 1 using fasttext**



Epoch 20/20, Training Loss: 19.9627, Validation Loss: 5.8441, Training Macro-F1: 0.5998, Validation Macro-F1: 0.4915

Analysis: Both validation and training loss decrease steadily. Similarly, their f1 scores increase steadily. The gap between them is narrow indicating the model accurately understands the dataset, finds the underlying patterns and generalizes it. In this case. After 20 epochs, validation loss is 5.8441 while the validation macro f1 is 0.4915.

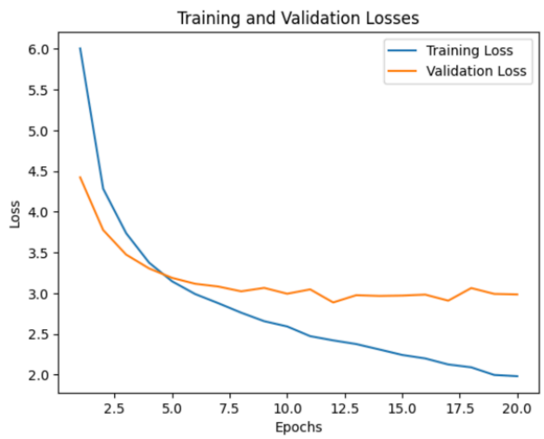
- **BiLSTM-CRF on dataset 2 using word2vec**

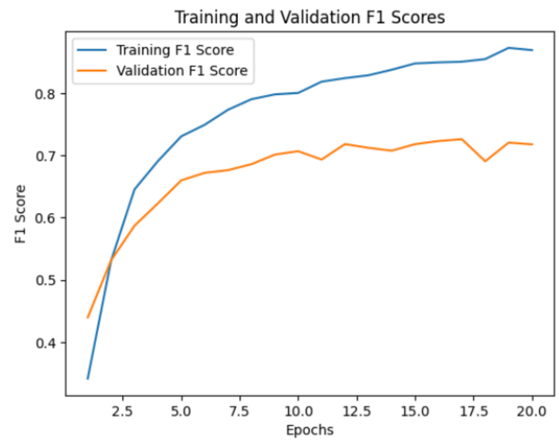


Epoch 15/15, Training Loss: 2.5679, Validation Loss: 3.1126, Training F1: 0.8111, Validation F1: 0.7090

Analysis: Both validation and training loss decrease steadily and have a narrow gap between them. Similarly, their f1 scores increase steadily. This indicates model accurately understands the dataset, finds the underlying patterns and generalizes it. In this case. After 15 epochs, the validation loss stands at 3.1126, with a corresponding validation macro F1 score of 0.7090.

- **BiLSTM-CRF on dataset 2 using GloVe**

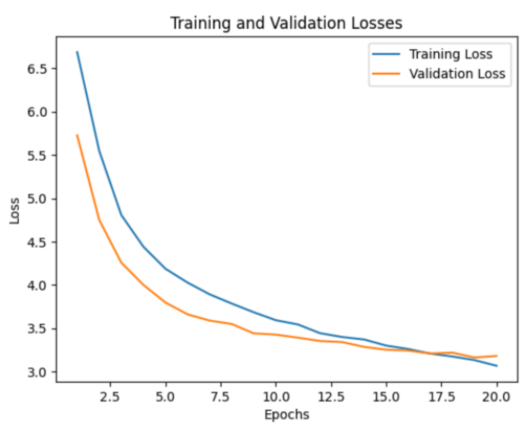


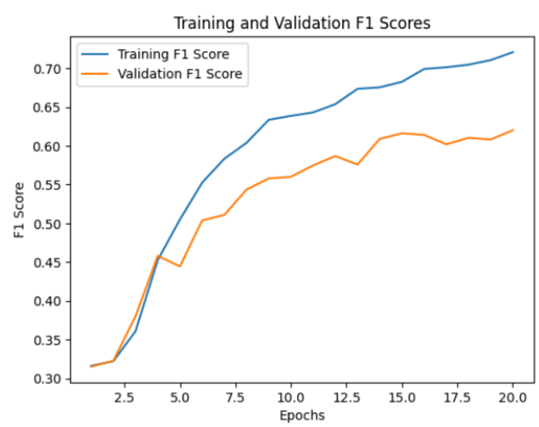


Epoch 20/20, Training Loss: 1.9829, Validation Loss: 2.9860, Training F1: 0.8690, Validation F1: 0.7177

Analysis: The training loss decreases more steeply than the validation loss. Similar pattern is observed in the increase of their macro f1 scores. The decrease indicates that model is able to generalize the data well while the increase is a measure of its performance. The F1 plot also shows that both training and validation macro-F1-scores are increasing over epochs, indicating the model is becoming more accurate. After 20 epochs, validation loss is 2.9860 while the validation macro f1 is 0.7177.

- **BiLSTM-CRF on dataset 2 using fasttext**





Epoch 20/20, Training Loss: 3.0675, Validation Loss: 3.1803, Training F1: 0.7207, Validation F1: 0.6200

Analysis: The loss plot demonstrates a steady decline in both training and validation losses throughout epochs, underscoring the model's ongoing refinement and learning journey. Likewise, the F1 plot depicts a gradual rise in both training and validation macro-F1 scores, indicating improved accuracy as training progresses. It's crucial to mitigate overfitting by halting training when the model's loss stabilizes, as evidenced in this case where additional epochs may offer diminishing returns. Following 20 epochs, the validation loss stands at 3.0675, accompanied by a validation macro F1 score of 0.6200.

<u>Tables for all Accuracies and F1</u>

Dataset 1:

Model_No	**Embedding_used**	**Accuracy**	**Macro_F1**
1 (RNN)	word2vec	0.7319755624269435	0.15168516856642836
1 (RNN)	GloVe	0.6011504776382387	0.1548107474017858
1 (RNN)	fasttext	0.6932808190387351	0.21895290263053943
2 (LSTM)	word2vec	0.8053343467670887	0.20253776600570392
2 (LSTM)	GloVe	0.8087086205208819	0.3592128770312555
2 (LSTM)	fasttext	0.8104303062758037	0.3430796054165031
3 (GRU)	word2vec	0.8065493156443078	0.42832298598121293
3 (GRU)	GloVe	0.8091347233498593	0.3656067483078713
3 (GRU)	fasttext	0.8069063207172347	0.3847831567967505
4 (BiLSTM-CRF)	word2vec	0.9566702586206897	0.4625821460029647
4 (BiLSTM-CRF)	GloVe	0.9494827586206896	0.3455169452643622
4 (BiLSTM-CRF)	fasttext	0.9597198275862069	0.4277152762183393

BiLSTM-CRF is the best model in word2vec and fasttext embedding

```
T1 Model 4 Word2Vec Test Accuracy: 0.9566702586206897  Test F1 Score: 0.4625821460029647
T1 Model 4 GloVe Test Accuracy: 0.9494827586206896  Test F1 Score: 0.3455169452643622
T1 Model 4 Fasttext Test Accuracy: 0.9597198275862069  Test F1 Score: 0.4277152762183393
```

```
T1 Model 1 Word2Vec Test Accuracy: 0.7319755624269435
T1 Model 1 GloVe Test Accuracy: 0.6011504776382387
T1 Model 1 Fasttext Test Accuracy: 0.6932808190387351
T1 Model 2 Word2Vec Test Accuracy: 0.8053343467670887
T1 Model 2 GloVe Test Accuracy: 0.8087086205208819
T1 Model 2 Fasttext Test Accuracy: 0.8104303062758037
T1 Model 3 Word2Vec Test Accuracy: 0.8065493156443078
T1 Model 3 GloVe Test Accuracy: 0.8091347233498593
T1 Model 3 Fasttext Test Accuracy: 0.8069063207172347
```

```
T1 Model 1 Word2Vec Test F1 Score: 0.15168516856642836
T1 Model 1 GloVe Test F1 Score: 0.1548107474017858
T1 Model 1 Fasttext Test F1 Score: 0.21895290263053943
T1 Model 2 Word2Vec Test F1 Score: 0.20253776600570392
T1 Model 2 GloVe Test F1 Score: 0.3592128770312555
T1 Model 2 Fasttext Test F1 Score: 0.3430796054165031
T1 Model 3 Word2Vec Test F1 Score: 0.42832298598121293
T1 Model 3 GloVe Test F1 Score: 0.3656067483078713
T1 Model 3 Fasttext Test F1 Score: 0.3847831567967505
```

Dataset 2:

Model_No	**Embedding_used**	**Accuracy**	**Macro_F1**
1 (RNN)	word2vec	0.9636293369975953	0.6418273258905921
1 (RNN)	GloVe	0.9705427688079697	0.7345606849465055
1 (RNN)	fasttext	0.9688680865681897	0.642376971497277
2 (LSTM)	word2vec	0.9644881484026108	0.6326217148585592
2 (LSTM)	GloVe	0.9692545517004466	0.643781609162235
2 (LSTM)	fasttext	0.968610443146685	0.6368809473533991
3 (GRU)	word2vec	0.965218138096874	0.6331449380684212
3 (GRU)	GloVe	0.9709721745104775	0.7176116417133045
3 (GRU)	fasttext	0.9688680865681897	0.6371674868854306
4 (BiLSTM-CRF)	word2vec	0.9007928833881261	0.6461923148655623
4 (BiLSTM-CRF)	GloVe	0.8975053181202862	0.6575802407713208
4 (BiLSTM-CRF)	fasttext	0.8822278089344421	0.5624527222363326

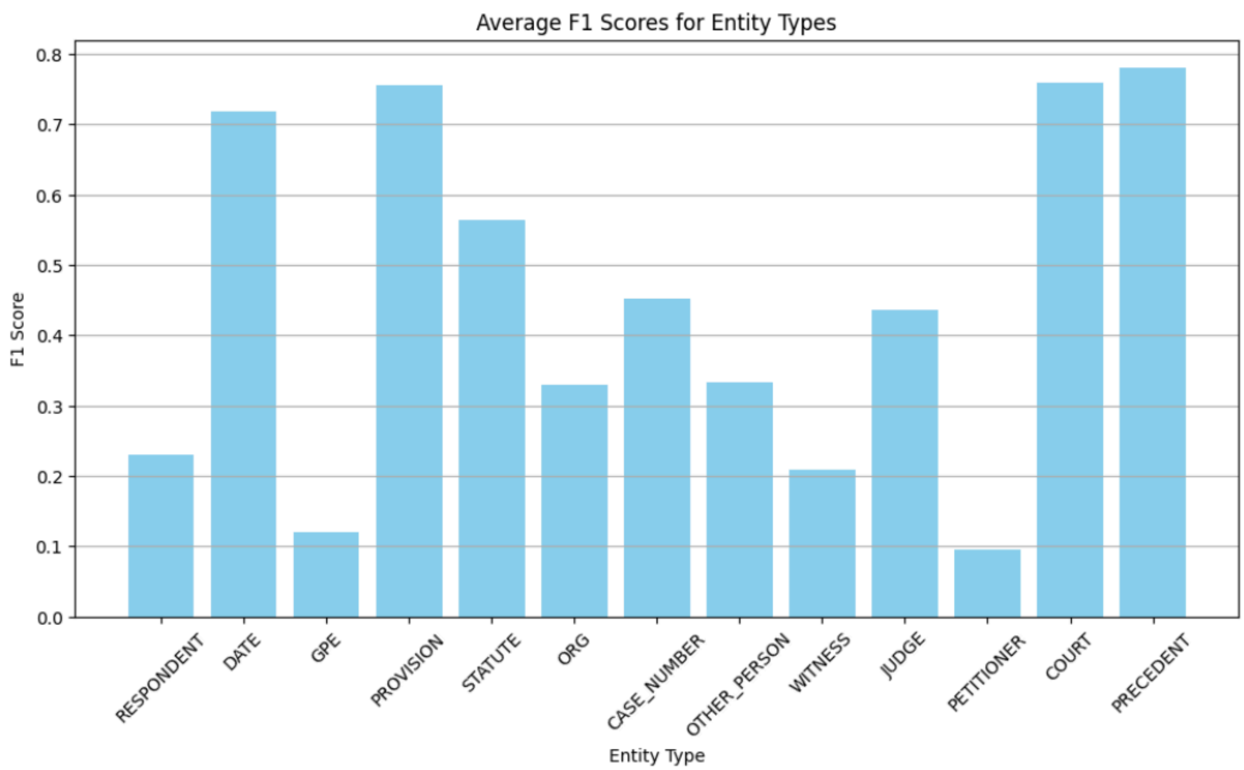
BiLSTM-CRF is the best model in word2vec embedding

```
T2 Model 1 Word2Vec Test Accuracy: 0.9636293369975953  Test F1 Score: 0.6418273258905921
T2 Model 1 GloVe Test Accuracy: 0.9705427688079697  Test F1 Score: 0.7345606849465055
T2 Model 1 Fasttext Test Accuracy: 0.9688680865681897  Test F1 Score: 0.642376971497277
T2 Model 2 Word2Vec Test Accuracy: 0.9644881484026108  Test F1 Score: 0.6326217148585592
T2 Model 2 GloVe Test Accuracy: 0.9692545517004466  Test F1 Score: 0.643781609162235
T2 Model 2 Fasttext Test Accuracy: 0.968610443146685  Test F1 Score: 0.6368809473533991
T2 Model 3 Word2Vec Test Accuracy: 0.965218138096874  Test F1 Score: 0.6331449380684212
T2 Model 3 GloVe Test Accuracy: 0.9709721745104775  Test F1 Score: 0.7176116417133045
T2 Model 3 Fasttext Test Accuracy: 0.9688680865681897  Test F1 Score: 0.6371674868854306
```

```
T2 Model 4 Word2Vec Test Accuracy: 0.9007928833881261  Test F1 Score: 0.6461923148655623
T2 Model 4 GloVe Test Accuracy: 0.8975053181202862  Test F1 Score: 0.6575802407713208
T2 Model 4 Fasttext Test Accuracy: 0.8822278089344421  Test F1 Score: 0.5624527222363326
```

Plot of the label-wise F1 scores (13 labels) on test data of Dataset_1 using best model

On Dataset 1 for BILSTM-CRF using word2vec embedding: (plot_model4_word2vec.ipynb)

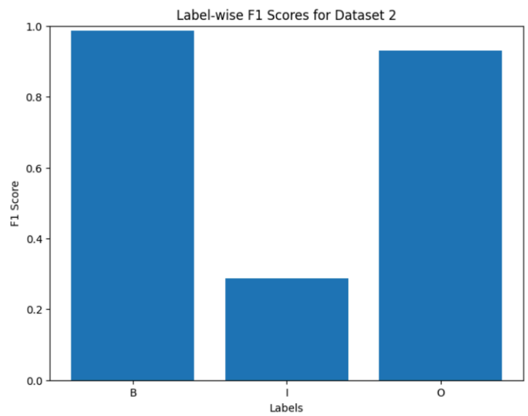


F1 scores:

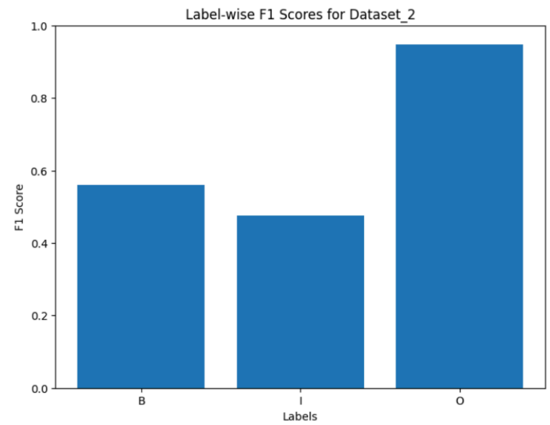
```
{'RESPONDENT': 0.23026315789473684,
 'DATE': 0.7181919572926979,
 'GPE': 0.1197352149559904,
 'PROVISION': 0.75561531161025,
 'STATUTE': 0.5635831494098456,
 'ORG': 0.32927349667508266,
 'CASE_NUMBER': 0.452765243868804,
 'OTHER_PERSON': 0.33266150608211753,
 'WITNESS': 0.20840787119856888,
 'JUDGE': 0.43529411764705883,
 'PETITIONER': 0.09610983981693363,
 'COURT': 0.7596037455749687,
 'PRECEDENT': 0.7806451612903226}
```

<u>For dataset 2:</u>

(RNN GloVe)



(BILSTM GloVe)



Contribution:

Member	Question done
Sanya Madan (2021561)	Part 1A, 1B (Data Preparation) Task 1 BILSTM-CRF
Prashasti Gupta (2021346)	Task 1,2 LSTM Task 2 RNN, GRU
Shruti Gupta (2021353)	Part 1A, 1B (Data Preparation) Task 1 BILSTM-CRF
Parisha Agrawal (2021270)	Task 2 RNN, GRU Task 2 BILSTM-CRF Model Load, accuracy and plots

All members contributed equally. The report was made by everyone.

Drive Link with all files and saved models:
https://drive.google.com/drive/folders/120woa9MdqYhbBur0Ym7Y_FDy79ZT6zPp?usp=sharing

Note: As .pt files take too much size and thus time and memory in zipping and uploading to the classroom in 1 zip, we have uploaded the .pt files in the above drive link which we have given viewer access to all IIITD account. All the other files and codes are in the submitted zip.

In Zip file:

- [x] ~~JSON files for Part 1A and 1B~~
- [x] ~~12 models for dataset 1 (4 models * 3 embedding * 2 dataset) and code as well~~
- [x] ~~12 models for dataset 2 (4 models * 3 embedding * 2 dataset) and code as well~~
- [x] ~~For Part 2: final output file, load models, extract named entities (t1) or aspect terms (t2) from test data and report accuracy and F1 using nine print statements for each dataset.~~
- [x] ~~For Part 3: In the final output file, load models, extract the named entities (t1) or aspect terms (t2) from test data, and report accuracy and F1 scores using three print statements for each dataset.~~
- [x] ~~Ipynb notebooks for:~~

> Create separate .ipynb files for each part. The file name should follow the format: “A2_<Part number>.ipynb”
> Create a single .ipynb file for generating the final outputs that are required for submittables. It should be named as “A2_<Group No>_infer.ipynb”. Clearly indicate which cell corresponds to the output of which task/subtask. Outputs will be checked from this inference file only by TAs.