In [1]:
```
# Assumptions as per post 519:
# 1) conlleval.py present in the directory
# 2) glove.6B.100d.txt present in the directory
```

In [2]:
```
!pip3 install torch torchvision torchaudio
!pip install numpy
!pip install -q datasets
!pip install scikit-learn
```

...

In [3]:
```
!pip freeze > requirements.txt
```

In [4]:
```
import itertools
from collections import Counter
import torch
import torch.nn as nn
from torch.utils.data import DataLoader, Dataset
from torch.optim import Adam, AdamW
import numpy as np
from conlleval import evaluate
import datasets
```

In [5]:
```
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
```

In [6]:
```
# Load the dataset
dataset = datasets.load_dataset("conll2003")
```

In [7]:

```
# Calculating the word frequency
word_frequency = Counter(itertools.chain(*dataset['train']['tokens']))
# Creating a dictionary with words having frequency greater than 2
word_frequency = {
    word: frequency
    for word, frequency in word_frequency.items()
    if frequency >= 3
}

# Adding the index and UNK and PAD to handle padding and unknown tokens:
word2idx = {
    word: index
    for index, word in enumerate(word_frequency.keys(), start=2)
}
word2idx['[PAD]'] = 0
word2idx['[UNK]'] = 1
```

In [8]:

```
# Iterating the dataset to replace unknown tokens with [UNK]
dataset = (
    dataset
    .map(lambda x: {
            'input_ids': [
                word2idx.get(word, word2idx['[UNK]'])
                for word in x['tokens']
            ]
        }
    )
)
```

```
Map:   0%|          | 0/14041 [00:00<?, ? examples/s]

Map:   0%|          | 0/3250 [00:00<?, ? examples/s]

Map:   0%|          | 0/3453 [00:00<?, ? examples/s]
```

In [9]:
```python
dataset
```

Out[9]:
```
DatasetDict({
    train: Dataset({
        features: ['id', 'tokens', 'pos_tags', 'chunk_tags', 'ner_tags', 'input_ids'],
        num_rows: 14041
    })
    validation: Dataset({
        features: ['id', 'tokens', 'pos_tags', 'chunk_tags', 'ner_tags', 'input_ids'],
        num_rows: 3250
    })
    test: Dataset({
        features: ['id', 'tokens', 'pos_tags', 'chunk_tags', 'ner_tags', 'input_ids'],
        num_rows: 3453
    })
})
```

In [10]:
```python
# Removing columns pos_tags & chunk_tags; Renaming column ner_tags to labels
for split in dataset.keys():
    dataset[split] = dataset[split].remove_columns(['pos_tags', 'chunk_tags'])
    dataset[split] = dataset[split].rename_column('ner_tags', 'labels')
```

In [11]:
```python
vocab_size = len(word2idx)
print(vocab_size)
```

```
8128
```

In [12]:
```python
# NER tag mapping
ner_tags = {'O': 0, 'B-PER': 1, 'I-PER': 2, 'B-ORG': 3, 'I-ORG': 4, 'B-LOC': 5, 'I-LOC': 6, 'B-MISC': 7,
'I-MISC': 8, '[PAD]': 9}
idx2tag = {idx: tag for tag, idx in ner_tags.items()}
```

Task 1

In [13]:

```
# Model Architecture:
class BiLSTM(nn.Module):
    def __init__(self, vocab_size, embedding_dim, lstm_hidden_dim, output_dim):
        dropout_val = 0.33
        super(BiLSTM, self).__init__()
        self.embedding = nn.Embedding(vocab_size, embedding_dim)
        self.bilstm = nn.LSTM(embedding_dim, lstm_hidden_dim, num_layers = 1, bidirectional=True,
batch_first=True)
        self.dropout = nn.Dropout(dropout_val)
        self.linear = nn.Linear(lstm_hidden_dim*2, output_dim)
        self.elu = nn.ELU()
        self.classifier = nn.Linear(output_dim, len(ner_tags)-1)

    def forward(self, x, labels=None):
        embed = self.embedding(x)
        lstm_out, _ = self.bilstm(embed)
        drop = self.dropout(lstm_out)
        linear = self.linear(drop)
        elu_out = self.elu(linear)
        logits = self.classifier(elu_out)
        loss = None
        if labels is not None:
            logits_flatten = logits.view(-1, logits.shape[-1])
            labels_flatten = labels.view(-1)
            loss = nn.functional.cross_entropy(logits_flatten, labels_flatten,ignore_index=9)

        return logits, loss
```

In [14]:

```
train_data = dataset['train']
validation_data = dataset['validation']
test_data = dataset['test']
```

In [15]:
```
def collate_fun(batch):

    input_ids = [torch.tensor(item['input_ids'], device=device) for item in batch]
    labels = [torch.tensor(item['labels'], device=device) for item in batch]
    lengths = [len(label) for label in labels]

    input_ids_padded = torch.nn.utils.rnn.pad_sequence(input_ids, batch_first=True, padding_value=0)
    labels_padded = torch.nn.utils.rnn.pad_sequence(labels, batch_first=True, padding_value=9)
    return {'input_ids': input_ids_padded, 'labels': labels_padded, 'lengths': lengths}

batch_size = 32
train_loader = DataLoader(train_data, batch_size= batch_size, collate_fn = collate_fun)
test_loader = DataLoader(test_data, batch_size= batch_size, collate_fn = collate_fun)
validation_loader = DataLoader(validation_data, batch_size= batch_size, collate_fn = collate_fun)
```

In [16]:
```
model = BiLSTM(vocab_size=vocab_size, embedding_dim=100, lstm_hidden_dim=256, output_dim=128).to(device)
optimizer = AdamW(model.parameters(), lr=0.001)
best_val_f1 = 0.0
```

```
In [17]: num_epochs = 20  # Number of epochs
for epoch in range(num_epochs):
    model.train()
    total_loss = 0
    total_f1 = 0
    for data in train_loader:
        input_ids, labels, lengths = data['input_ids'], data['labels'], data['lengths']
        # clear the gradients of all optimized variables
        optimizer.zero_grad()
        # forward pass: compute predicted outputs by passing inputs to the model
        logits, loss = model(input_ids, labels)

        if loss is not None:
            loss.backward()
            optimizer.step()
            total_loss += loss.item()

        predictions = torch.argmax(logits, dim=-1).view(-1)
        labels_flat = labels.view(-1)

    model.eval()
    valid_loss, valid_f1 = 0, 0
    all_val_predictions, all_val_labels = [], []

    with torch.no_grad():
        for data in validation_loader:

            input_ids, labels, lengths = data['input_ids'], data['labels'], data['lengths']
            # print(input_ids, '\n', '-----------------------------', labels)
            logits, loss = model(input_ids, labels)
            valid_loss += loss.item()

            val_predictions = torch.argmax(logits, dim=-1).tolist()

            for i, length in enumerate(lengths):
              seq_preds = val_predictions[i][:length]
              seq_labels = labels[i, :length].tolist()
              mapped_seq_preds = [idx2tag[p] for p in seq_preds]
              mapped_seq_labels = [idx2tag[l] for l in seq_labels]

              all_val_predictions.extend(mapped_seq_preds)
              all_val_labels.extend(mapped_seq_labels)
```

```
        flat_preds = list(itertools.chain(*all_val_predictions))
        flat_labels = list(itertools.chain(*all_val_labels))
        precision, recall, f1 = evaluate(all_val_labels, all_val_predictions)


        print(f"Epoch: {epoch}, Train Loss: {total_loss / len(train_loader)}")
        print(f"Epoch: {epoch}, Validation Loss: {valid_loss / len(validation_loader)}")

        # Saving best model based on best F1 score
        if f1 > best_val_f1:
            print(f'Validation F1 increased ({best_val_f1:.4f} --> {f1:.4f}). Saving model...')
            torch.save(model.state_dict(), 'model.pt')
            best_val_f1 = f1
```

```
processed 51362 tokens with 5942 phrases; found: 3177 phrases; correct: 1923.
accuracy:  34.77%; (non-O)
accuracy:  88.70%; precision:  60.53%; recall:  32.36%; FB1:  42.18
              LOC: precision:  72.06%; recall:  49.70%; FB1:  58.83  1267
             MISC: precision:  53.37%; recall:  10.30%; FB1:  17.27  178
              ORG: precision:  41.58%; recall:  30.20%; FB1:  34.99  974
              PER: precision:  67.28%; recall:  27.69%; FB1:  39.23  758
Epoch: 0, Train Loss: 0.6636327108991988
Epoch: 0, Validation Loss: 0.3902442083493167
Validation F1 increased (0.0000 --> 42.1757). Saving model...
processed 51362 tokens with 5942 phrases; found: 4331 phrases; correct: 3154.
accuracy:  56.18%; (non-O)
accuracy:  92.38%; precision:  72.82%; recall:  53.08%; FB1:  61.40
              LOC: precision:  83.06%; recall:  64.34%; FB1:  72.52  1423
             MISC: precision:  78.18%; recall:  49.35%; FB1:  60.51  582
              ORG: precision:  55.31%; recall:  48.92%; FB1:  51.92  1186
              PER: precision:  75.53%; recall:  46.74%; FB1:  57.75  1140
Epoch: 1, Train Loss: 0.3100220473538363
Epoch: 1, Validation Loss: 0.24765182019887017
Validation F1 increased (42.1757 --> 61.4037). Saving model...
processed 51362 tokens with 5942 phrases; found: 4863 phrases; correct: 3754.
accuracy:  66.30%; (non-O)
accuracy:  94.01%; precision:  77.20%; recall:  63.18%; FB1:  69.49
              LOC: precision:  86.17%; recall:  72.95%; FB1:  79.01  1555
             MISC: precision:  81.93%; recall:  63.45%; FB1:  71.52  714
              ORG: precision:  62.08%; recall:  59.58%; FB1:  60.81  1287
              PER: precision:  78.81%; recall:  55.92%; FB1:  65.42  1307
Epoch: 2, Train Loss: 0.1860341761454278
Epoch: 2, Validation Loss: 0.1905136772057078
Validation F1 increased (61.4037 --> 69.4863). Saving model...
processed 51362 tokens with 5942 phrases; found: 5122 phrases; correct: 4067.
accuracy:  71.37%; (non-O)
accuracy:  94.85%; precision:  79.40%; recall:  68.44%; FB1:  73.52
              LOC: precision:  87.19%; recall:  76.70%; FB1:  81.61  1616
             MISC: precision:  81.10%; recall:  68.87%; FB1:  74.49  783
              ORG: precision:  66.06%; recall:  64.43%; FB1:  65.23  1308
              PER: precision:  81.91%; recall:  62.92%; FB1:  71.17  1415
Epoch: 3, Train Loss: 0.12183876473940315
Epoch: 3, Validation Loss: 0.16661824801178904
Validation F1 increased (69.4863 --> 73.5177). Saving model...
processed 51362 tokens with 5942 phrases; found: 5302 phrases; correct: 4242.
accuracy:  74.44%; (non-O)
accuracy:  95.18%; precision:  80.01%; recall:  71.39%; FB1:  75.45
```

```
              LOC: precision:  87.99%; recall:  79.37%; FB1:  83.46  1657
             MISC: precision:  83.94%; recall:  70.28%; FB1:  76.51  772
              ORG: precision:  64.15%; recall:  67.26%; FB1:  65.67  1406
              PER: precision:  84.12%; recall:  66.99%; FB1:  74.58  1467
Epoch: 4, Train Loss: 0.08455812154207294
Epoch: 4, Validation Loss: 0.16090877685814584
Validation F1 increased (73.5177 --> 75.4536). Saving model...
processed 51362 tokens with 5942 phrases; found: 5417 phrases; correct: 4383.
accuracy:  76.65%; (non-O)
accuracy:  95.51%; precision:  80.91%; recall:  73.76%; FB1:  77.17
              LOC: precision:  90.48%; recall:  81.22%; FB1:  85.60  1649
             MISC: precision:  81.27%; recall:  72.02%; FB1:  76.37  817
              ORG: precision:  67.38%; recall:  68.23%; FB1:  67.80  1358
              PER: precision:  82.36%; recall:  71.23%; FB1:  76.39  1593
Epoch: 5, Train Loss: 0.05989609160797314
Epoch: 5, Validation Loss: 0.15836506466582126
Validation F1 increased (75.4536 --> 77.1723). Saving model...
processed 51362 tokens with 5942 phrases; found: 5513 phrases; correct: 4379.
accuracy:  76.86%; (non-O)
accuracy:  95.45%; precision:  79.43%; recall:  73.70%; FB1:  76.46
              LOC: precision:  88.29%; recall:  82.47%; FB1:  85.28  1716
             MISC: precision:  79.31%; recall:  71.91%; FB1:  75.43  836
              ORG: precision:  65.82%; recall:  67.64%; FB1:  66.72  1378
              PER: precision:  81.74%; recall:  70.25%; FB1:  75.56  1583
Epoch: 6, Train Loss: 0.04306901675829444
Epoch: 6, Validation Loss: 0.16944996220049421
processed 51362 tokens with 5942 phrases; found: 5440 phrases; correct: 4387.
accuracy:  76.59%; (non-O)
accuracy:  95.43%; precision:  80.64%; recall:  73.83%; FB1:  77.09
              LOC: precision:  89.98%; recall:  81.65%; FB1:  85.62  1667
             MISC: precision:  76.45%; recall:  72.89%; FB1:  74.63  879
              ORG: precision:  68.05%; recall:  68.46%; FB1:  68.25  1349
              PER: precision:  83.95%; recall:  70.41%; FB1:  76.59  1545
Epoch: 7, Train Loss: 0.03227576900841451
Epoch: 7, Validation Loss: 0.18479545962698324
processed 51362 tokens with 5942 phrases; found: 5858 phrases; correct: 4471.
accuracy:  78.14%; (non-O)
accuracy:  95.15%; precision:  76.32%; recall:  75.24%; FB1:  75.78
              LOC: precision:  85.12%; recall:  83.45%; FB1:  84.28  1801
             MISC: precision:  70.13%; recall:  73.86%; FB1:  71.95  971
              ORG: precision:  64.05%; recall:  70.54%; FB1:  67.14  1477
              PER: precision:  81.48%; recall:  71.17%; FB1:  75.98  1609
Epoch: 8, Train Loss: 0.026288396433106643
```

```
Epoch: 8, Validation Loss: 0.18995527111110277
processed 51362 tokens with 5942 phrases; found: 5640 phrases; correct: 4465.
accuracy:  78.10%; (non-O)
accuracy:  95.48%; precision:  79.17%; recall:  75.14%; FB1:  77.10
              LOC: precision:  90.58%; recall:  82.14%; FB1:  86.15  1666
             MISC: precision:  75.98%; recall:  73.43%; FB1:  74.68  891
              ORG: precision:  66.48%; recall:  69.50%; FB1:  67.95  1402
              PER: precision:  80.13%; recall:  73.13%; FB1:  76.47  1681
Epoch: 9, Train Loss: 0.022471409347392786
Epoch: 9, Validation Loss: 0.20204291901527363
processed 51362 tokens with 5942 phrases; found: 5735 phrases; correct: 4506.
accuracy:  78.94%; (non-O)
accuracy:  95.46%; precision:  78.57%; recall:  75.83%; FB1:  77.18
              LOC: precision:  87.10%; recall:  83.78%; FB1:  85.41  1767
             MISC: precision:  77.94%; recall:  73.97%; FB1:  75.90  875
              ORG: precision:  65.73%; recall:  69.95%; FB1:  67.77  1427
              PER: precision:  80.85%; recall:  73.13%; FB1:  76.80  1666
Epoch: 10, Train Loss: 0.018384253433334073
Epoch: 10, Validation Loss: 0.212188676871615
Validation F1 increased (77.1723 --> 77.1774). Saving model...
processed 51362 tokens with 5942 phrases; found: 5761 phrases; correct: 4546.
accuracy:  79.17%; (non-O)
accuracy:  95.53%; precision:  78.91%; recall:  76.51%; FB1:  77.69
              LOC: precision:  87.16%; recall:  83.89%; FB1:  85.49  1768
             MISC: precision:  75.38%; recall:  74.40%; FB1:  74.89  910
              ORG: precision:  68.48%; recall:  69.50%; FB1:  68.99  1361
              PER: precision:  80.55%; recall:  75.30%; FB1:  77.83  1722
Epoch: 11, Train Loss: 0.014689100629115945
Epoch: 11, Validation Loss: 0.21830930830932915
Validation F1 increased (77.1774 --> 77.6895). Saving model...
processed 51362 tokens with 5942 phrases; found: 5637 phrases; correct: 4551.
accuracy:  79.29%; (non-O)
accuracy:  95.76%; precision:  80.73%; recall:  76.59%; FB1:  78.61
              LOC: precision:  90.45%; recall:  82.47%; FB1:  86.28  1675
             MISC: precision:  78.35%; recall:  75.38%; FB1:  76.84  887
              ORG: precision:  72.94%; recall:  69.35%; FB1:  71.10  1275
              PER: precision:  78.39%; recall:  76.60%; FB1:  77.48  1800
Epoch: 12, Train Loss: 0.011818865107701793
Epoch: 12, Validation Loss: 0.23496087240519223
Validation F1 increased (77.6895 --> 78.6078). Saving model...
processed 51362 tokens with 5942 phrases; found: 5656 phrases; correct: 4523.
accuracy:  78.60%; (non-O)
accuracy:  95.65%; precision:  79.97%; recall:  76.12%; FB1:  78.00
```

```
              LOC: precision:  86.90%; recall:  84.10%; FB1:  85.48  1778
             MISC: precision:  79.91%; recall:  73.75%; FB1:  76.71  851
              ORG: precision:  70.02%; recall:  69.50%; FB1:  69.76  1331
              PER: precision:  80.54%; recall:  74.16%; FB1:  77.22  1696
Epoch: 13, Train Loss: 0.012203457638822074
Epoch: 13, Validation Loss: 0.2450696681744197
processed 51362 tokens with 5942 phrases; found: 5642 phrases; correct: 4537.
accuracy:  78.68%; (non-O)
accuracy:  95.77%; precision:  80.41%; recall:  76.35%; FB1:  78.33
              LOC: precision:  86.56%; recall:  84.16%; FB1:  85.34  1786
             MISC: precision:  81.09%; recall:  73.97%; FB1:  77.37  841
              ORG: precision:  76.02%; recall:  66.89%; FB1:  71.16  1180
              PER: precision:  76.95%; recall:  76.66%; FB1:  76.80  1835
Epoch: 14, Train Loss: 0.011745665162984685
Epoch: 14, Validation Loss: 0.2504376052004541
processed 51362 tokens with 5942 phrases; found: 5655 phrases; correct: 4519.
accuracy:  78.93%; (non-O)
accuracy:  95.65%; precision:  79.91%; recall:  76.05%; FB1:  77.93
              LOC: precision:  87.79%; recall:  83.72%; FB1:  85.71  1752
             MISC: precision:  81.58%; recall:  73.97%; FB1:  77.59  836
              ORG: precision:  71.22%; recall:  69.95%; FB1:  70.58  1317
              PER: precision:  77.77%; recall:  73.89%; FB1:  75.78  1750
Epoch: 15, Train Loss: 0.01050128613831782
Epoch: 15, Validation Loss: 0.25423536887483705
processed 51362 tokens with 5942 phrases; found: 5799 phrases; correct: 4610.
accuracy:  79.97%; (non-O)
accuracy:  95.71%; precision:  79.50%; recall:  77.58%; FB1:  78.53
              LOC: precision:  87.12%; recall:  85.03%; FB1:  86.06  1793
             MISC: precision:  84.77%; recall:  72.45%; FB1:  78.13  788
              ORG: precision:  65.93%; recall:  71.59%; FB1:  68.64  1456
              PER: precision:  80.59%; recall:  77.09%; FB1:  78.80  1762
Epoch: 16, Train Loss: 0.009613973399300988
Epoch: 16, Validation Loss: 0.24981880438637888
processed 51362 tokens with 5942 phrases; found: 5584 phrases; correct: 4507.
accuracy:  78.53%; (non-O)
accuracy:  95.79%; precision:  80.71%; recall:  75.85%; FB1:  78.21
              LOC: precision:  88.81%; recall:  82.53%; FB1:  85.55  1707
             MISC: precision:  85.52%; recall:  74.30%; FB1:  79.51  801
              ORG: precision:  72.38%; recall:  68.98%; FB1:  70.64  1278
              PER: precision:  76.81%; recall:  74.97%; FB1:  75.88  1798
Epoch: 17, Train Loss: 0.00844681326858555
Epoch: 17, Validation Loss: 0.27087047008152176
processed 51362 tokens with 5942 phrases; found: 5636 phrases; correct: 4557.
```

```
accuracy:  79.36%; (non-O)
accuracy:  95.83%; precision:  80.86%; recall:  76.69%; FB1:  78.72
              LOC: precision:  88.59%; recall:  83.23%; FB1:  85.83  1726
             MISC: precision:  86.92%; recall:  72.78%; FB1:  79.22  772
              ORG: precision:  73.00%; recall:  69.95%; FB1:  71.44  1285
              PER: precision:  76.58%; recall:  77.04%; FB1:  76.81  1853
Epoch: 18, Train Loss: 0.007829904260111138
Epoch: 18, Validation Loss: 0.28675997486991534
Validation F1 increased (78.6078 --> 78.7183). Saving model...
processed 51362 tokens with 5942 phrases; found: 5431 phrases; correct: 4481.
accuracy:  78.07%; (non-O)
accuracy:  95.85%; precision:  82.51%; recall:  75.41%; FB1:  78.80
              LOC: precision:  90.90%; recall:  82.14%; FB1:  86.30  1660
             MISC: precision:  83.31%; recall:  74.73%; FB1:  78.79  827
              ORG: precision:  74.74%; recall:  69.50%; FB1:  72.02  1247
              PER: precision:  79.61%; recall:  73.34%; FB1:  76.35  1697
Epoch: 19, Train Loss: 0.007166095389616431
Epoch: 19, Validation Loss: 0.2979401738826134
Validation F1 increased (78.7183 --> 78.8007). Saving model...
```

In [18]:

```python
# Model architechture:
print(model)
```

```
BiLSTM(
  (embedding): Embedding(8128, 100)
  (bilstm): LSTM(100, 256, batch_first=True, bidirectional=True)
  (dropout): Dropout(p=0.33, inplace=False)
  (linear): Linear(in_features=512, out_features=128, bias=True)
  (elu): ELU(alpha=1.0)
  (classifier): Linear(in_features=128, out_features=9, bias=True)
)
```

Hyperparameters:

1. Number Of Epochs: 20
2. Optimizer: AdamW
3. Learning rate: 0.001
4. Best Model saved based on F1 score
5. Dropout: 0.33
6. Vocab size: 8128

Results on Validation data

In [19]:
```
# Model on validation set
model.load_state_dict(torch.load('model.pt'))
model.eval()
preds = []
label_list = []
with torch.no_grad():
    for data in validation_loader:
        input_ids, labels, lengths = data['input_ids'], data['labels'], data['lengths']
        logits, loss = model(input_ids, labels)
        predictions = torch.argmax(logits, dim=2)

        for pred, label, length in zip(predictions.tolist(), labels.tolist(), lengths):
            decoded_label = [idx2tag[l] for l in label]
            label_list.extend([decoded_label[:length]])
            trimmed_pred = pred[:length]
            decoded_pred = [idx2tag[p] for p in trimmed_pred]
            preds.extend([decoded_pred])

flat_preds = list(itertools.chain(*preds))
flat_labels = list(itertools.chain(*label_list))
precision, recall, f1 = evaluate(flat_labels, flat_preds)

print(f"Precision: {precision}")
print(f"Recall: {recall}")
print(f"F1 Score: {f1}")
```

```
processed 51362 tokens with 5942 phrases; found: 5431 phrases; correct: 4481.
accuracy:  78.07%; (non-O)
accuracy:  95.85%; precision:  82.51%; recall:  75.41%; FB1:  78.80
              LOC: precision:  90.90%; recall:  82.14%; FB1:  86.30  1660
             MISC: precision:  83.31%; recall:  74.73%; FB1:  78.79  827
              ORG: precision:  74.74%; recall:  69.50%; FB1:  72.02  1247
              PER: precision:  79.61%; recall:  73.34%; FB1:  76.35  1697
Precision: 82.50782544651078
Recall: 75.41231908448334
F1 Score: 78.80066824936252
```

In [20]:
```
# What are the precision, recall, and F1 score on the validation data?.
# precision:  82.51%; recall:  75.41%; FB1:  78.80
```

Results on test data

In [21]:
```
# Model on test set
model.load_state_dict(torch.load('model.pt'))
model.eval()
preds = []
label_list = []
with torch.no_grad():
    for data in test_loader:
        input_ids, labels, lengths = data['input_ids'], data['labels'], data['lengths']
        logits, loss = model(input_ids, labels)
        predictions = torch.argmax(logits, dim=2)

        for pred, label, length in zip(predictions.tolist(), labels.tolist(), lengths):
            decoded_label = [idx2tag[l] for l in label]
            label_list.extend([decoded_label[:length]])
            trimmed_pred = pred[:length]
            decoded_pred = [idx2tag[p] for p in trimmed_pred]
            preds.extend([decoded_pred])

flat_preds = list(itertools.chain(*preds))
flat_labels = list(itertools.chain(*label_list))
precision, recall, f1 = evaluate(flat_labels, flat_preds)

print(f"Precision: {precision}")
print(f"Recall: {recall}")
print(f"F1 Score: {f1}")
```

```
processed 46435 tokens with 5648 phrases; found: 4981 phrases; correct: 3709.
accuracy:  70.38%; (non-O)
accuracy:  93.86%; precision:  74.46%; recall:  65.67%; FB1:  69.79
              LOC: precision:  85.47%; recall:  74.04%; FB1:  79.34  1445
             MISC: precision:  65.77%; recall:  62.68%; FB1:  64.19  669
              ORG: precision:  70.23%; recall:  59.66%; FB1:  64.52  1411
              PER: precision:  71.63%; recall:  64.50%; FB1:  67.88  1456
Precision: 74.4629592451315
Recall: 65.66926345609065
F1 Score: 69.79019663185625
```

In [22]:
```
# What are the precision, recall, and F1 score on the test data?.
# precision:  74.46%; recall:  65.67%; FB1:  69.79
```

Task 2

In [23]:
```
vocab, embeddings = [], []
with open('glove.6B.100d.txt', 'rt', encoding='utf-8') as fi:
    full_content = fi.read().strip().split('\n')

for line in full_content:
    parts = line.split(' ')
    word = parts[0]
    embedding = [float(val) for val in parts[1:]]
    vocab.append(word)
    embeddings.append(embedding)

vocab = ['[PAD]', '[UNK]'] + vocab
pad_emb_npa = np.zeros((1, 100))  # embedding for '<pad>' token
unk_emb_npa = np.mean(embeddings, axis=0, keepdims=True)  # embedding for '<unk>' token

# Insert embeddings for pad and unk tokens at the top of embs_npa.
embs_npa = np.vstack((pad_emb_npa, unk_emb_npa, embeddings))

vocab_npa = np.array(vocab)
embs_npa = np.array(embs_npa)

print(len(embs_npa))
print(len(vocab_npa))
```

```
400002
400002
```

In [24]:
```
vocab_size = len(vocab_npa)
```

In [25]:
```
vocab_npa
```

Out[25]:
```
array(['[PAD]', '[UNK]', 'the', ..., 'rolonda', 'zsombor', 'sandberger'],
      dtype='<U68')
```

In [26]:
```python
dataset_glove = datasets.load_dataset("conll2003")

word_frequency = Counter()

word2idx_glove = {
    word: index
    for index, word in enumerate(vocab_npa)
}

# Iterating the dataset to replace unknown tokens with [UNK]
dataset_glove = (
    dataset_glove
    .map(lambda x: {
            'input_ids': [
                word2idx_glove.get(word.lower(), word2idx_glove['[UNK]'])
                for word in x['tokens']
            ]
        }
    )
)

# Removing columns pos_tags & chunk_tags; Renaming column ner_tags to labels
for split in dataset_glove.keys():
    dataset_glove[split] = dataset_glove[split].remove_columns(['pos_tags', 'chunk_tags'])
    dataset_glove[split] = dataset_glove[split].rename_column('ner_tags', 'labels')

train_data = dataset_glove['train']
validation_data = dataset_glove['validation']
test_data = dataset_glove['test']
```

In [27]:

```
# Model Architecture
class BiLSTMGlove(nn.Module):
    def __init__(self, vocab_size, embedding_dim, lstm_hidden_dim, output_dim):
        super(BiLSTMGlove, self).__init__()
        droupout_val = 0.33

        self.embedding = nn.Embedding.from_pretrained(torch.from_numpy(embs_npa).float(), freeze=True)
        self.upper_embedding = nn.Embedding(2,10)
        self.lower_embedding = nn.Embedding(2,10)
        self.title_embedding = nn.Embedding(2,10)

        self.bilstm = nn.LSTM(embedding_dim+30, lstm_hidden_dim, num_layers = 1, bidirectional=True,
batch_first=True)
        self.dropout = nn.Dropout(droupout_val)
        self.linear = nn.Linear(lstm_hidden_dim*2, output_dim)
        self.elu = nn.ELU()
        self.classifier = nn.Linear(output_dim, len(ner_tags)-1)

    def forward(self, x, is_upper, lower_case, title_case, labels=None):
        embed = self.embedding(x)
        upper = self.upper_embedding(is_upper)
        lower = self.lower_embedding(lower_case)
        title = self.lower_embedding(title_case)
        features = torch.cat((embed, upper, lower, title), dim=-1)
        lstm_out, _ = self.bilstm(features)
        drop = self.dropout(lstm_out)
        linear = self.linear(drop)
        elu_out = self.elu(linear)
        logits = self.classifier(elu_out)

        loss = None
        if labels is not None:
            logits_flatten = logits.view(-1, logits.shape[-1])
            labels_flatten = labels.view(-1)
            loss = nn.functional.cross_entropy(logits_flatten, labels_flatten,ignore_index=9)

        return logits, loss
```

In [28]:
```
def collate_fun_glove(batch):

    input_ids = [torch.tensor(item['input_ids'], device=device) for item in batch]
    labels = [torch.tensor(item['labels'], device=device) for item in batch]
    lengths = [len(label) for label in labels]

    # Additional features
    upper_case = [torch.tensor([1 if word.isupper() else 0 for word in item['tokens']],
dtype=torch.long,device=device) for item in batch]
    lower_case = [torch.tensor([1 if word.islower() else 0 for word in item['tokens']],
dtype=torch.long,device=device) for item in batch]
    title_case = [torch.tensor([1 if word.istitle() else 0 for word in item['tokens']],
dtype=torch.long,device=device) for item in batch]

    upper_case_padded = torch.nn.utils.rnn.pad_sequence(upper_case, batch_first=True, padding_value=0)
    lower_case_padded = torch.nn.utils.rnn.pad_sequence(lower_case, batch_first=True, padding_value=0)
    title_case_padded = torch.nn.utils.rnn.pad_sequence(title_case, batch_first=True, padding_value=0)

    input_ids_padded = torch.nn.utils.rnn.pad_sequence(input_ids, batch_first=True, padding_value=0)
    labels_padded = torch.nn.utils.rnn.pad_sequence(labels, batch_first=True, padding_value=9)

    return {'input_ids': input_ids_padded, 'labels': labels_padded, 'lengths': lengths, 'upper_case':
upper_case_padded, 'lower_case': lower_case_padded,
            'title_case': title_case_padded}

batch_size = 32
train_loader = DataLoader(train_data, batch_size= batch_size, collate_fn = collate_fun_glove)
test_loader = DataLoader(test_data, batch_size= batch_size, collate_fn = collate_fun_glove)
validation_loader = DataLoader(validation_data, batch_size= batch_size, collate_fn = collate_fun_glove)
```

In [29]:
```
model_glove = BiLSTMGlove(vocab_size=vocab_size, embedding_dim=100, lstm_hidden_dim=256,
output_dim=128).to(device)
optimizer = AdamW(model_glove.parameters(),lr=0.001)
best_val_f1 = 0.0
```

In [30]:

```
num_epochs = 30  # Number of epochs
for epoch in range(num_epochs):
    model_glove.train()
    total_loss = 0
    total_f1 = 0
    for data in train_loader:
        input_ids, labels, lengths, upper_case, lower_case, title_case = data['input_ids'], data['labels'],
data['lengths'], data['upper_case'], data['lower_case'], data['title_case']
        optimizer.zero_grad()
        # forward pass: compute predicted outputs by passing inputs to the model
        logits, loss = model_glove(input_ids, upper_case, lower_case, title_case, labels)

        if loss is not None:
            loss.backward()
            # perform a single optimization step (parameter update)
            optimizer.step()
            total_loss += loss.item()

        predictions = torch.argmax(logits, dim=-1).view(-1)
        labels_flat = labels.view(-1)

    model_glove.eval()
    valid_loss, valid_f1 = 0, 0
    all_val_predictions, all_val_labels = [], []

    with torch.no_grad():
        for data in validation_loader:
            input_ids, labels, lengths, upper_case, lower_case, title_case = data['input_ids'],
data['labels'], data['lengths'], data['upper_case'], data['lower_case'], data['title_case']
            logits, loss = model_glove(input_ids,  upper_case,  lower_case, title_case, labels)
            valid_loss += loss.item()

            val_predictions = torch.argmax(logits, dim=-1).tolist()

            for i, length in enumerate(lengths):
                seq_preds = val_predictions[i][:length]
                seq_labels = labels[i, :length].tolist()
                mapped_seq_preds = [idx2tag[p] for p in seq_preds]
                mapped_seq_labels = [idx2tag[l] for l in seq_labels]

                all_val_predictions.extend(mapped_seq_preds)
                all_val_labels.extend(mapped_seq_labels)
```

```python
        flat_preds = list(itertools.chain(*all_val_predictions))
        flat_labels = list(itertools.chain(*all_val_labels))
        precision, recall, f1 = evaluate(all_val_labels, all_val_predictions)


        print(f"Epoch: {epoch}, Train Loss: {total_loss / len(train_loader)}")
        print(f"Epoch: {epoch}, Validation Loss: {valid_loss / len(validation_loader)}")

        if f1 > best_val_f1:
            print(f'Validation F1 increased ({best_val_f1:.4f} --> {f1:.4f}). Saving model...')
            torch.save(model_glove.state_dict(), 'model_glove.pt')
            best_val_f1 = f1
```

```
processed 51362 tokens with 5942 phrases; found: 6050 phrases; correct: 4851.
accuracy:  82.66%; (non-O)
accuracy:  96.85%; precision:  80.18%; recall:  81.64%; FB1:  80.90
              LOC: precision:  84.45%; recall:  86.34%; FB1:  85.38  1878
             MISC: precision:  69.18%; recall:  71.58%; FB1:  70.36  954
              ORG: precision:  67.93%; recall:  68.38%; FB1:  68.15  1350
              PER: precision:  90.36%; recall:  91.64%; FB1:  91.00  1868
Epoch: 0, Train Loss: 0.2715232573366875
Epoch: 0, Validation Loss: 0.10504391064922161
Validation F1 increased (0.0000 --> 80.9039). Saving model...
processed 51362 tokens with 5942 phrases; found: 6041 phrases; correct: 5154.
accuracy:  87.25%; (non-O)
accuracy:  97.67%; precision:  85.32%; recall:  86.74%; FB1:  86.02
              LOC: precision:  88.01%; recall:  93.47%; FB1:  90.65  1951
             MISC: precision:  73.92%; recall:  79.93%; FB1:  76.81  997
              ORG: precision:  78.57%; recall:  74.65%; FB1:  76.56  1274
              PER: precision:  93.40%; recall:  92.24%; FB1:  92.82  1819
Epoch: 1, Train Loss: 0.10237841387167955
Epoch: 1, Validation Loss: 0.0793897450203076
Validation F1 increased (80.9039 --> 86.0219). Saving model...
processed 51362 tokens with 5942 phrases; found: 6019 phrases; correct: 5289.
accuracy:  89.52%; (non-O)
accuracy:  98.07%; precision:  87.87%; recall:  89.01%; FB1:  88.44
              LOC: precision:  91.47%; recall:  94.56%; FB1:  92.99  1899
             MISC: precision:  78.32%; recall:  81.89%; FB1:  80.06  964
              ORG: precision:  81.30%; recall:  79.42%; FB1:  80.35  1310
              PER: precision:  93.82%; recall:  94.03%; FB1:  93.93  1846
Epoch: 2, Train Loss: 0.07497241848375402
Epoch: 2, Validation Loss: 0.06275374148732654
Validation F1 increased (86.0219 --> 88.4374). Saving model...
processed 51362 tokens with 5942 phrases; found: 5998 phrases; correct: 5346.
accuracy:  90.51%; (non-O)
accuracy:  98.24%; precision:  89.13%; recall:  89.97%; FB1:  89.55
              LOC: precision:  92.51%; recall:  94.77%; FB1:  93.63  1882
             MISC: precision:  78.78%; recall:  84.16%; FB1:  81.38  985
              ORG: precision:  84.91%; recall:  80.98%; FB1:  82.90  1279
              PER: precision:  94.11%; recall:  94.63%; FB1:  94.37  1852
Epoch: 3, Train Loss: 0.05710048237067683
Epoch: 3, Validation Loss: 0.057212442708630844
Validation F1 increased (88.4374 --> 89.5477). Saving model...
processed 51362 tokens with 5942 phrases; found: 5986 phrases; correct: 5379.
accuracy:  91.22%; (non-O)
accuracy:  98.36%; precision:  89.86%; recall:  90.53%; FB1:  90.19
```

```
              LOC: precision:  94.23%; recall:  94.28%; FB1:  94.26  1838
             MISC: precision:  80.39%; recall:  84.92%; FB1:  82.59  974
              ORG: precision:  84.06%; recall:  83.37%; FB1:  83.71  1330
              PER: precision:  94.69%; recall:  94.79%; FB1:  94.74  1844
Epoch: 4, Train Loss: 0.0450371005955568
Epoch: 4, Validation Loss: 0.05253829852867342
Validation F1 increased (89.5477 --> 90.1911). Saving model...
processed 51362 tokens with 5942 phrases; found: 6022 phrases; correct: 5426.
accuracy:  91.70%; (non-O)
accuracy:  98.41%; precision:  90.10%; recall:  91.32%; FB1:  90.71
              LOC: precision:  94.28%; recall:  95.05%; FB1:  94.66  1852
             MISC: precision:  79.76%; recall:  85.90%; FB1:  82.72  993
              ORG: precision:  85.34%; recall:  84.19%; FB1:  84.76  1323
              PER: precision:  94.88%; recall:  95.49%; FB1:  95.18  1854
Epoch: 5, Train Loss: 0.03613554203625188
Epoch: 5, Validation Loss: 0.05176268622973094
Validation F1 increased (90.1911 --> 90.7054). Saving model...
processed 51362 tokens with 5942 phrases; found: 5998 phrases; correct: 5446.
accuracy:  92.12%; (non-O)
accuracy:  98.49%; precision:  90.80%; recall:  91.65%; FB1:  91.22
              LOC: precision:  94.33%; recall:  95.10%; FB1:  94.71  1852
             MISC: precision:  82.51%; recall:  85.47%; FB1:  83.96  955
              ORG: precision:  86.22%; recall:  85.38%; FB1:  85.80  1328
              PER: precision:  94.79%; recall:  95.87%; FB1:  95.33  1863
Epoch: 6, Train Loss: 0.02866752836070313
Epoch: 6, Validation Loss: 0.05244049765086874
Validation F1 increased (90.7054 --> 91.2228). Saving model...
processed 51362 tokens with 5942 phrases; found: 5985 phrases; correct: 5451.
accuracy:  92.37%; (non-O)
accuracy:  98.53%; precision:  91.08%; recall:  91.74%; FB1:  91.41
              LOC: precision:  94.70%; recall:  95.26%; FB1:  94.98  1848
             MISC: precision:  81.22%; recall:  85.36%; FB1:  83.24  969
              ORG: precision:  87.42%; recall:  86.06%; FB1:  86.73  1320
              PER: precision:  95.24%; recall:  95.55%; FB1:  95.39  1848
Epoch: 7, Train Loss: 0.022723367547240865
Epoch: 7, Validation Loss: 0.05320340288502242
Validation F1 increased (91.2228 --> 91.4061). Saving model...
processed 51362 tokens with 5942 phrases; found: 6043 phrases; correct: 5473.
accuracy:  92.37%; (non-O)
accuracy:  98.50%; precision:  90.57%; recall:  92.11%; FB1:  91.33
              LOC: precision:  93.81%; recall:  95.75%; FB1:  94.77  1875
             MISC: precision:  79.62%; recall:  86.88%; FB1:  83.09  1006
              ORG: precision:  87.01%; recall:  85.91%; FB1:  86.45  1324
```

```
              PER: precision:  95.81%; recall:  95.60%; FB1:  95.71  1838
Epoch: 8, Train Loss: 0.018257219020112615
Epoch: 8, Validation Loss: 0.057465917216819325
processed 51362 tokens with 5942 phrases; found: 5972 phrases; correct: 5428.
accuracy:  91.61%; (non-O)
accuracy:  98.42%; precision:  90.89%; recall:  91.35%; FB1:  91.12
              LOC: precision:  94.73%; recall:  94.83%; FB1:  94.78  1839
             MISC: precision:  79.84%; recall:  85.47%; FB1:  82.56  987
              ORG: precision:  87.48%; recall:  85.46%; FB1:  86.46  1310
              PER: precision:  95.42%; recall:  95.11%; FB1:  95.27  1836
Epoch: 9, Train Loss: 0.014468162008682514
Epoch: 9, Validation Loss: 0.06627412902682908
processed 51362 tokens with 5942 phrases; found: 6046 phrases; correct: 5454.
accuracy:  91.78%; (non-O)
accuracy:  98.43%; precision:  90.21%; recall:  91.79%; FB1:  90.99
              LOC: precision:  92.87%; recall:  96.46%; FB1:  94.63  1908
             MISC: precision:  79.38%; recall:  86.01%; FB1:  82.56  999
              ORG: precision:  87.62%; recall:  84.41%; FB1:  85.99  1292
              PER: precision:  95.13%; recall:  95.39%; FB1:  95.26  1847
Epoch: 10, Train Loss: 0.012428276478133332
Epoch: 10, Validation Loss: 0.07182222778490041
processed 51362 tokens with 5942 phrases; found: 6055 phrases; correct: 5466.
accuracy:  92.39%; (non-O)
accuracy:  98.46%; precision:  90.27%; recall:  91.99%; FB1:  91.12
              LOC: precision:  93.03%; recall:  95.97%; FB1:  94.48  1895
             MISC: precision:  81.75%; recall:  86.01%; FB1:  83.83  970
              ORG: precision:  86.09%; recall:  86.28%; FB1:  86.18  1344
              PER: precision:  94.96%; recall:  95.17%; FB1:  95.07  1846
Epoch: 11, Train Loss: 0.010326147472391238
Epoch: 11, Validation Loss: 0.06595770871047303
processed 51362 tokens with 5942 phrases; found: 6060 phrases; correct: 5489.
accuracy:  92.74%; (non-O)
accuracy:  98.53%; precision:  90.58%; recall:  92.38%; FB1:  91.47
              LOC: precision:  92.31%; recall:  96.73%; FB1:  94.47  1925
             MISC: precision:  83.42%; recall:  86.23%; FB1:  84.80  953
              ORG: precision:  87.52%; recall:  85.76%; FB1:  86.63  1314
              PER: precision:  94.59%; recall:  95.93%; FB1:  95.26  1868
Epoch: 12, Train Loss: 0.00924966951802193
Epoch: 12, Validation Loss: 0.07136137768559944
Validation F1 increased (91.4061 --> 91.4681). Saving model...
processed 51362 tokens with 5942 phrases; found: 6005 phrases; correct: 5481.
accuracy:  92.74%; (non-O)
accuracy:  98.57%; precision:  91.27%; recall:  92.24%; FB1:  91.76
```

```
              LOC: precision:  94.71%; recall:  95.48%; FB1:  95.09  1852
             MISC: precision:  83.48%; recall:  84.38%; FB1:  83.93  932
              ORG: precision:  87.34%; recall:  89.04%; FB1:  88.18  1367
              PER: precision:  94.66%; recall:  95.28%; FB1:  94.97  1854
Epoch: 13, Train Loss: 0.0074038172105146734
Epoch: 13, Validation Loss: 0.06954462036934413
Validation F1 increased (91.4681 --> 91.7553). Saving model...
processed 51362 tokens with 5942 phrases; found: 6072 phrases; correct: 5521.
accuracy:  93.49%; (non-O)
accuracy:  98.63%; precision:  90.93%; recall:  92.91%; FB1:  91.91
              LOC: precision:  94.69%; recall:  96.19%; FB1:  95.44  1866
             MISC: precision:  84.18%; recall:  86.01%; FB1:  85.09  942
              ORG: precision:  84.72%; recall:  89.34%; FB1:  86.97  1414
              PER: precision:  95.30%; recall:  95.71%; FB1:  95.50  1850
Epoch: 14, Train Loss: 0.006931918180019451
Epoch: 14, Validation Loss: 0.06828830824260386
Validation F1 increased (91.7553 --> 91.9094). Saving model...
processed 51362 tokens with 5942 phrases; found: 6034 phrases; correct: 5489.
accuracy:  92.85%; (non-O)
accuracy:  98.61%; precision:  90.97%; recall:  92.38%; FB1:  91.67
              LOC: precision:  95.73%; recall:  95.10%; FB1:  95.41  1825
             MISC: precision:  82.03%; recall:  85.14%; FB1:  83.56  957
              ORG: precision:  85.43%; recall:  90.08%; FB1:  87.70  1414
              PER: precision:  95.16%; recall:  94.95%; FB1:  95.05  1838
Epoch: 15, Train Loss: 0.006898024848981494
Epoch: 15, Validation Loss: 0.07125043617563778
processed 51362 tokens with 5942 phrases; found: 6074 phrases; correct: 5497.
accuracy:  93.13%; (non-O)
accuracy:  98.53%; precision:  90.50%; recall:  92.51%; FB1:  91.49
              LOC: precision:  96.14%; recall:  94.83%; FB1:  95.48  1812
             MISC: precision:  79.86%; recall:  86.01%; FB1:  82.82  993
              ORG: precision:  84.56%; recall:  88.59%; FB1:  86.53  1405
              PER: precision:  95.17%; recall:  96.31%; FB1:  95.74  1864
Epoch: 16, Train Loss: 0.007662423328580554
Epoch: 16, Validation Loss: 0.0708807601856154
processed 51362 tokens with 5942 phrases; found: 6046 phrases; correct: 5515.
accuracy:  93.39%; (non-O)
accuracy:  98.64%; precision:  91.22%; recall:  92.81%; FB1:  92.01
              LOC: precision:  95.42%; recall:  95.37%; FB1:  95.40  1836
             MISC: precision:  83.88%; recall:  85.79%; FB1:  84.83  943
              ORG: precision:  85.74%; recall:  89.71%; FB1:  87.68  1403
              PER: precision:  94.90%; recall:  96.04%; FB1:  95.47  1864
Epoch: 17, Train Loss: 0.00511772343176702
```

```
Epoch: 17, Validation Loss: 0.06980832576995427
Validation F1 increased (91.9094 --> 92.0087). Saving model...
processed 51362 tokens with 5942 phrases; found: 6065 phrases; correct: 5500.
accuracy:  93.01%; (non-O)
accuracy:  98.55%; precision:  90.68%; recall:  92.56%; FB1:  91.61
              LOC: precision:  93.54%; recall:  96.90%; FB1:  95.19  1903
             MISC: precision:  81.58%; recall:  86.44%; FB1:  83.94  977
              ORG: precision:  87.86%; recall:  86.35%; FB1:  87.10  1318
              PER: precision:  94.54%; recall:  95.82%; FB1:  95.17  1867
Epoch: 18, Train Loss: 0.005656444585575619
Epoch: 18, Validation Loss: 0.08005778727617469
processed 51362 tokens with 5942 phrases; found: 6077 phrases; correct: 5546.
accuracy:  93.77%; (non-O)
accuracy:  98.70%; precision:  91.26%; recall:  93.34%; FB1:  92.29
              LOC: precision:  94.15%; recall:  96.35%; FB1:  95.24  1880
             MISC: precision:  84.65%; recall:  86.12%; FB1:  85.38  938
              ORG: precision:  86.91%; recall:  90.08%; FB1:  88.47  1390
              PER: precision:  94.92%; recall:  96.31%; FB1:  95.61  1869
Epoch: 19, Train Loss: 0.004142424101014657
Epoch: 19, Validation Loss: 0.07158923786378003
Validation F1 increased (92.0087 --> 92.2872). Saving model...
processed 51362 tokens with 5942 phrases; found: 6071 phrases; correct: 5528.
accuracy:  93.58%; (non-O)
accuracy:  98.66%; precision:  91.06%; recall:  93.03%; FB1:  92.03
              LOC: precision:  94.84%; recall:  96.03%; FB1:  95.43  1860
             MISC: precision:  85.43%; recall:  85.25%; FB1:  85.34  920
              ORG: precision:  85.37%; recall:  89.63%; FB1:  87.45  1408
              PER: precision:  94.32%; recall:  96.42%; FB1:  95.36  1883
Epoch: 20, Train Loss: 0.0035400247416768355
Epoch: 20, Validation Loss: 0.07703571870804557
processed 51362 tokens with 5942 phrases; found: 6089 phrases; correct: 5525.
accuracy:  93.56%; (non-O)
accuracy:  98.64%; precision:  90.74%; recall:  92.98%; FB1:  91.85
              LOC: precision:  95.24%; recall:  94.77%; FB1:  95.01  1828
             MISC: precision:  83.83%; recall:  86.01%; FB1:  84.90  946
              ORG: precision:  84.55%; recall:  90.98%; FB1:  87.64  1443
              PER: precision:  94.60%; recall:  96.15%; FB1:  95.37  1872
Epoch: 21, Train Loss: 0.003488394026173085
Epoch: 21, Validation Loss: 0.0817668135392054
processed 51362 tokens with 5942 phrases; found: 6051 phrases; correct: 5527.
accuracy:  93.54%; (non-O)
accuracy:  98.70%; precision:  91.34%; recall:  93.02%; FB1:  92.17
              LOC: precision:  95.02%; recall:  95.54%; FB1:  95.28  1847
```

```
             MISC: precision:  83.57%; recall:  87.74%; FB1:  85.61  968
              ORG: precision:  87.34%; recall:  89.04%; FB1:  88.18  1367
              PER: precision:  94.65%; recall:  96.04%; FB1:  95.34  1869
Epoch: 22, Train Loss: 0.0033994918974745185
Epoch: 22, Validation Loss: 0.08185144328134873
processed 51362 tokens with 5942 phrases; found: 6092 phrases; correct: 5512.
accuracy:  93.29%; (non-O)
accuracy:  98.55%; precision:  90.48%; recall:  92.76%; FB1:  91.61
              LOC: precision:  95.08%; recall:  94.77%; FB1:  94.93  1831
             MISC: precision:  80.16%; recall:  87.64%; FB1:  83.73  1008
              ORG: precision:  85.53%; recall:  89.49%; FB1:  87.46  1403
              PER: precision:  95.30%; recall:  95.71%; FB1:  95.50  1850
Epoch: 23, Train Loss: 0.0033876888932041797
Epoch: 23, Validation Loss: 0.08728449949725953
processed 51362 tokens with 5942 phrases; found: 6038 phrases; correct: 5509.
accuracy:  93.28%; (non-O)
accuracy:  98.65%; precision:  91.24%; recall:  92.71%; FB1:  91.97
              LOC: precision:  94.61%; recall:  95.59%; FB1:  95.10  1856
             MISC: precision:  83.40%; recall:  84.49%; FB1:  83.94  934
              ORG: precision:  86.38%; recall:  90.83%; FB1:  88.55  1410
              PER: precision:  95.54%; recall:  95.33%; FB1:  95.43  1838
Epoch: 24, Train Loss: 0.0032230220378077516
Epoch: 24, Validation Loss: 0.08627318399033744
processed 51362 tokens with 5942 phrases; found: 6053 phrases; correct: 5554.
accuracy:  93.82%; (non-O)
accuracy:  98.73%; precision:  91.76%; recall:  93.47%; FB1:  92.61
              LOC: precision:  95.44%; recall:  95.70%; FB1:  95.57  1842
             MISC: precision:  85.04%; recall:  85.68%; FB1:  85.36  929
              ORG: precision:  86.46%; recall:  91.42%; FB1:  88.87  1418
              PER: precision:  95.49%; recall:  96.63%; FB1:  96.06  1864
Epoch: 25, Train Loss: 0.0038534758859634544
Epoch: 25, Validation Loss: 0.08241541978256378
Validation F1 increased (92.2872 --> 92.6053). Saving model...
processed 51362 tokens with 5942 phrases; found: 6012 phrases; correct: 5524.
accuracy:  93.35%; (non-O)
accuracy:  98.71%; precision:  91.88%; recall:  92.97%; FB1:  92.42
              LOC: precision:  95.38%; recall:  95.43%; FB1:  95.40  1838
             MISC: precision:  84.62%; recall:  85.90%; FB1:  85.25  936
              ORG: precision:  87.87%; recall:  90.75%; FB1:  89.29  1385
              PER: precision:  95.09%; recall:  95.66%; FB1:  95.37  1853
Epoch: 26, Train Loss: 0.002702540738231117
Epoch: 26, Validation Loss: 0.08476743231450364
processed 51362 tokens with 5942 phrases; found: 6003 phrases; correct: 5537.
```

```
accuracy:  93.53%; (non-O)
accuracy:  98.73%; precision:  92.24%; recall:  93.18%; FB1:  92.71
              LOC: precision:  94.52%; recall:  96.62%; FB1:  95.56  1878
             MISC: precision:  86.99%; recall:  84.82%; FB1:  85.89  899
              ORG: precision:  88.06%; recall:  90.16%; FB1:  89.09  1373
              PER: precision:  95.57%; recall:  96.15%; FB1:  95.86  1853
Epoch: 27, Train Loss: 0.0023310799705559284
Epoch: 27, Validation Loss: 0.08612175240561767
Validation F1 increased (92.6053 --> 92.7082). Saving model...
processed 51362 tokens with 5942 phrases; found: 6014 phrases; correct: 5537.
accuracy:  93.63%; (non-O)
accuracy:  98.75%; precision:  92.07%; recall:  93.18%; FB1:  92.62
              LOC: precision:  95.26%; recall:  96.35%; FB1:  95.81  1858
             MISC: precision:  86.27%; recall:  84.49%; FB1:  85.37  903
              ORG: precision:  86.99%; recall:  91.28%; FB1:  89.08  1407
              PER: precision:  95.56%; recall:  95.77%; FB1:  95.66  1846
Epoch: 28, Train Loss: 0.0022321702571345818
Epoch: 28, Validation Loss: 0.08569300297338531
processed 51362 tokens with 5942 phrases; found: 6023 phrases; correct: 5550.
accuracy:  93.72%; (non-O)
accuracy:  98.77%; precision:  92.15%; recall:  93.40%; FB1:  92.77
              LOC: precision:  95.46%; recall:  96.08%; FB1:  95.77  1849
             MISC: precision:  85.25%; recall:  85.25%; FB1:  85.25  922
              ORG: precision:  88.17%; recall:  90.60%; FB1:  89.37  1378
              PER: precision:  95.20%; recall:  96.85%; FB1:  96.02  1874
Epoch: 29, Train Loss: 0.0019563890452288387
Epoch: 29, Validation Loss: 0.08553200845252461
Validation F1 increased (92.7082 --> 92.7706). Saving model...
```

In [31]:
```python
# Model Architecture:
print(model_glove)
```

```
BiLSTMGlove(
  (embedding): Embedding(400002, 100)
  (upper_embedding): Embedding(2, 10)
  (lower_embedding): Embedding(2, 10)
  (title_embedding): Embedding(2, 10)
  (bilstm): LSTM(130, 256, batch_first=True, bidirectional=True)
  (dropout): Dropout(p=0.33, inplace=False)
  (linear): Linear(in_features=512, out_features=128, bias=True)
  (elu): ELU(alpha=1.0)
  (classifier): Linear(in_features=128, out_features=9, bias=True)
)
```

Hyperparameters

1. Number Of Epochs: 30
2. Optimizer: AdamW
3. Learning rate: 0.001
4. Best Model saved based on F1 score
5. Dropout: 0.33
6. Vocab size: 23625
7. Additional features used: isUpper, isTitle, isLower

Model with Validation data

In [32]:
```python
# Model on validation set
model_glove.load_state_dict(torch.load('model_glove.pt'))
model_glove.eval()
preds = []
label_list = []
with torch.no_grad():
    for data in validation_loader:
        input_ids, labels, lengths, upper_case, lower_case, title_case = data['input_ids'], data['labels'],
data['lengths'], data['upper_case'], data['lower_case'], data['title_case']
        logits, loss = model_glove(input_ids,  upper_case,  lower_case, title_case, labels)
        predictions = torch.argmax(logits, dim=2)

        for pred, label, length in zip(predictions.tolist(), labels.tolist(), lengths):
            decoded_label = [idx2tag[l] for l in label]
            label_list.extend([decoded_label[:length]])
            trimmed_pred = pred[:length]
            decoded_pred = [idx2tag[p] for p in trimmed_pred]
            preds.extend([decoded_pred])

flat_preds = list(itertools.chain(*preds))
flat_labels = list(itertools.chain(*label_list))
precision, recall, f1 = evaluate(flat_labels, flat_preds)

print(f"Precision: {precision}")
print(f"Recall: {recall}")
print(f"F1 Score: {f1}")
```

```
processed 51362 tokens with 5942 phrases; found: 6023 phrases; correct: 5550.
accuracy:  93.72%; (non-O)
accuracy:  98.77%; precision:  92.15%; recall:  93.40%; FB1:  92.77
              LOC: precision:  95.46%; recall:  96.08%; FB1:  95.77  1849
             MISC: precision:  85.25%; recall:  85.25%; FB1:  85.25  922
              ORG: precision:  88.17%; recall:  90.60%; FB1:  89.37  1378
              PER: precision:  95.20%; recall:  96.85%; FB1:  96.02  1874
Precision: 92.14677071226963
Recall: 93.40289464826658
F1 Score: 92.77058086084413
```

In [37]:
```
# What are the precision, recall, and F1 score on the validation data?.
# 98.77%; precision:  92.15%; recall:  93.40%; FB1:  92.77
```

Model with Test Data

In [34]:
```python
# Model on test set
model_glove.load_state_dict(torch.load('model_glove.pt',map_location=torch.device(device)))
model_glove.eval()
preds = []
label_list = []
with torch.no_grad():
    for data in test_loader:
        input_ids, labels, lengths, upper_case, lower_case, title_case = data['input_ids'], data['labels'],
data['lengths'], data['upper_case'], data['lower_case'], data['title_case']
        logits, loss = model_glove(input_ids,  upper_case,  lower_case, title_case, labels)
        predictions = torch.argmax(logits, dim=2)

        for pred, label, length in zip(predictions.tolist(), labels.tolist(), lengths):
            decoded_label = [idx2tag[l] for l in label]
            label_list.extend([decoded_label[:length]])
            trimmed_pred = pred[:length]
            decoded_pred = [idx2tag[p] for p in trimmed_pred]
            preds.extend([decoded_pred])

flat_preds = list(itertools.chain(*preds))
flat_labels = list(itertools.chain(*label_list))
precision, recall, f1 = evaluate(flat_labels, flat_preds)

print(f"Precision: {precision}")
print(f"Recall: {recall}")
print(f"F1 Score: {f1}")
```

```
processed 46435 tokens with 5648 phrases; found: 5785 phrases; correct: 5045.
accuracy:  90.62%; (non-O)
accuracy:  97.70%; precision:  87.21%; recall:  89.32%; FB1:  88.25
              LOC: precision:  89.53%; recall:  92.75%; FB1:  91.11  1728
             MISC: precision:  72.11%; recall:  77.35%; FB1:  74.64  753
              ORG: precision:  84.44%; recall:  86.57%; FB1:  85.49  1703
              PER: precision:  94.75%; recall:  93.82%; FB1:  94.28  1601
Precision: 87.20829732065687
Recall: 89.32365439093485
F1 Score: 88.25330184553486
```

In [36]:
```
# What are the precision, recall, and F1 score on the test data?.
# precision:  87.21%; recall:  89.32%; FB1:  88.25
```

BiLSTM with Glove Embeddings outperforms the model without. Can you provide a rationale for this?

Glove Embedding is a pretrained word embedding. It is trained on large datasets and captures the semantic and syntactic meaning of the word. Learning word embeddings from scratch is challenging. While learning from scratch on our dataset we will face issue of sparsity on the training data. The vocabulary of our training dataset might not be rich enough. This is overcome by using Glove embeddings. Due to these reasons BiLSTM with Glove Embeddings outperforms the model without Glove Embeddings.