NLP TP4 REPORT

Natural Language Processing with Sequence Models

SEHILI CHAIMA

GROUP 01

2024

Named entity recognition (NER) is a natural language processing (NLP) method that extracts information from text. NER involves **detecting** and **categorizing** important information in text known as named entities. Named entities refer to the key subjects of a piece of text, such as names, locations, companies, events and products, as well as themes, topics, times, monetary values and percentages.

NER is also referred to as entity extraction, chunking and identification. It's used in many fields in artificial intelligence (AI), including machine learning (ML), deep learning and neural networks. NER is a key component of NLP systems, such as chatbots, sentiment analysis tools and search engines. It's used in healthcare, finance, human resources (HR), customer support, higher education and social media analysis.

Dataset:

CoNLL-2003 is a named entity recognition dataset released as a part of CoNLL-2003 shared task: **language-independent named entity recognition.**

There total have four NER types, **LOC, MISC, ORG, PER,** combine with the three location type, **B,I,O.** This task is a 9 classification task.

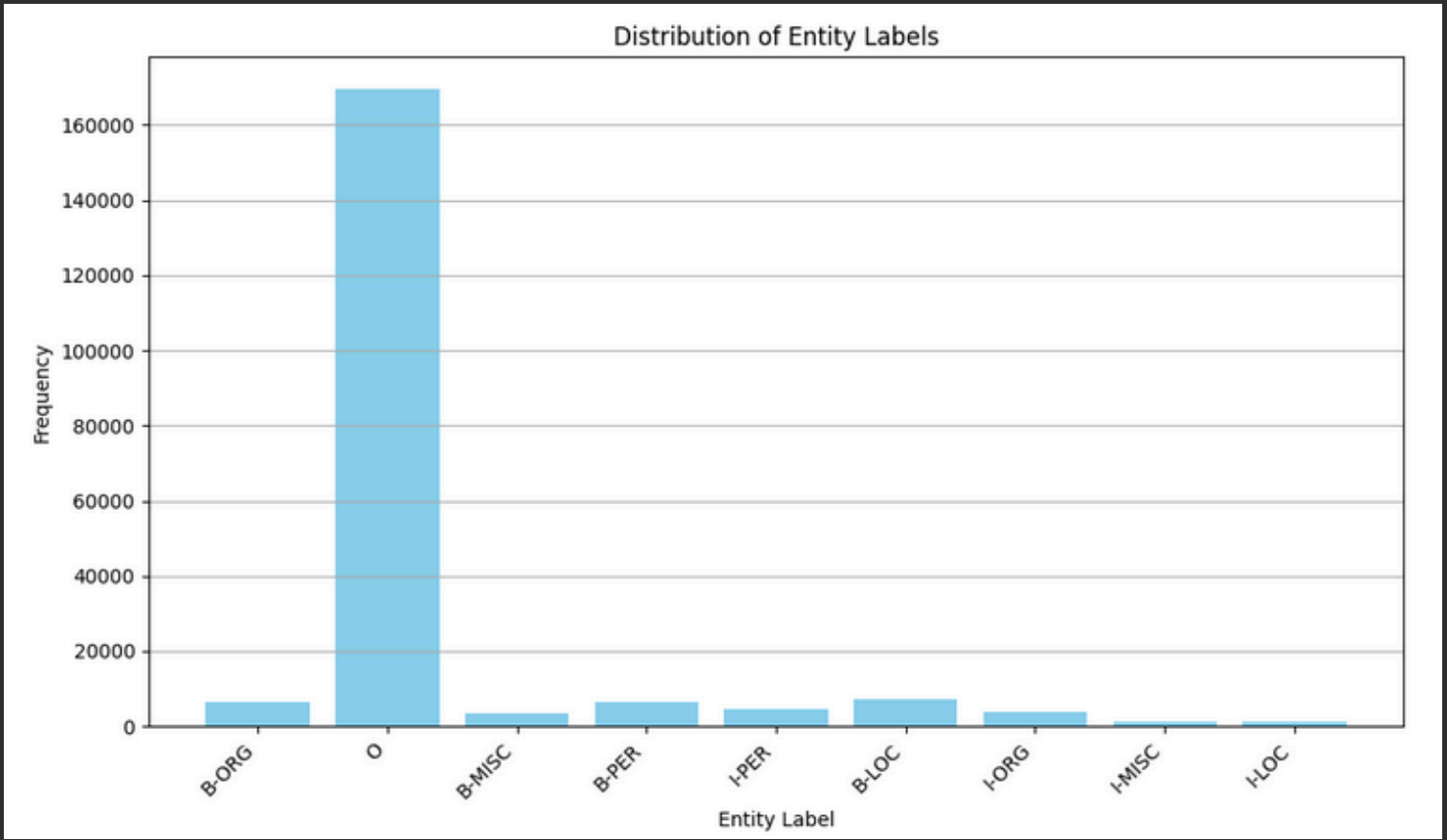
```
{'I-PER': 0,
 'I-ORG': 1,
 'O': 2,
 'B-ORG': 3,
 'B-MISC': 4,
 'I-LOC': 5,
 'I-MISC': 6,
 'B-LOC': 7,
 'B-PER': 8}
```

- **O:** word doesn’t correspond to any entity.
- **B-PER/I-PER:** word corresponds to the beginning of / is inside a person entity.
- **B-ORG/I-ORG:** word corresponds to the beginning of / is inside an organization entity.
- **B-LOC/I-LOC:** word corresponds to the beginning of / is inside a location entity.
- **B-MISC/I-MISC:** word corresponds to the beginning of / is inside a miscellaneous entity.

Train Samples:

	train_sentences	train_labels
0	[eu, rejects, german, call, to, boycott, briti...	[B-ORG, O, B-MISC, O, O, O, B-MISC, O, O]
1	[peter, blackburn]	[B-PER, I-PER]
2	[brussels, 1996-08-22]	[B-LOC, O]
3	[the, european, commission, said, on, thursday...	[O, B-ORG, I-ORG, O, O, O, O, O, O, B-MISC, O,...
4	[germany, 's, representative, to, the, europea...	[B-LOC, O, O, O, O, B-ORG, I-ORG, O, O, O, B-P...

Distribution of entity labels:



Tokenization and padding:

The tokenization process assigns a unique index to each unique word and each unique label in the dataset.

```
word2idx = {word: idx + 1 for idx, word in
            enumerate(set(word for sentence in train_sentences
                          for word in sentence))}
tag2idx = {tag: idx for idx, tag in enumerate(set(tag for labels in train_labels
                                                  for tag in labels))}
max_len = max(len(sentence) for sentence in train_sentences)
```

```
X_train = [[word2idx.get(word.lower(), 0) for word in sentence]
           for sentence in train_sentences]
X_train = pad_sequences(maxlen=max_len, sequences=X_train, padding="post", value=0)
y_train = [[tag2idx[tag] for tag in labels] for labels in train_labels]
y_train = pad_sequences(maxlen=max_len, sequences=y_train,
                        padding="post", value=tag2idx["O"])
y_train = [to_categorical(label, num_classes=len(tag2idx)) for label in y_train]
```

Tokenization and Padding:

- It tokenizes the text data in train_sentences, valid_sentences, and test_sentences using the word2idx dictionary, which maps words to integer indices. Unknown words are assigned index 0.
- The tokenized sequences are then padded to ensure uniform length.

Encoding Labels:

- It encodes the entity labels in train_labels, valid_labels, and test_labels into numerical format using the tag2idx dictionary, which maps labels to integer indices.
- The encoded sequences are then padded similarly to the text sequences, ensuring they match the corresponding text data in length.

One-Hot Encoding:

- Finally, it performs one-hot encoding on the encoded label sequences.

```
X_train shape: (14041, 113)
y_train shape: (14041, 113, 9)
X_valid shape: (3250, 113)
y_valid shape: (3250, 113, 9)
X_test shape: (3453, 113)
y_test shape: (3453, 113, 9)
```

Training with Biderctional LSTM:

Model Architecture:

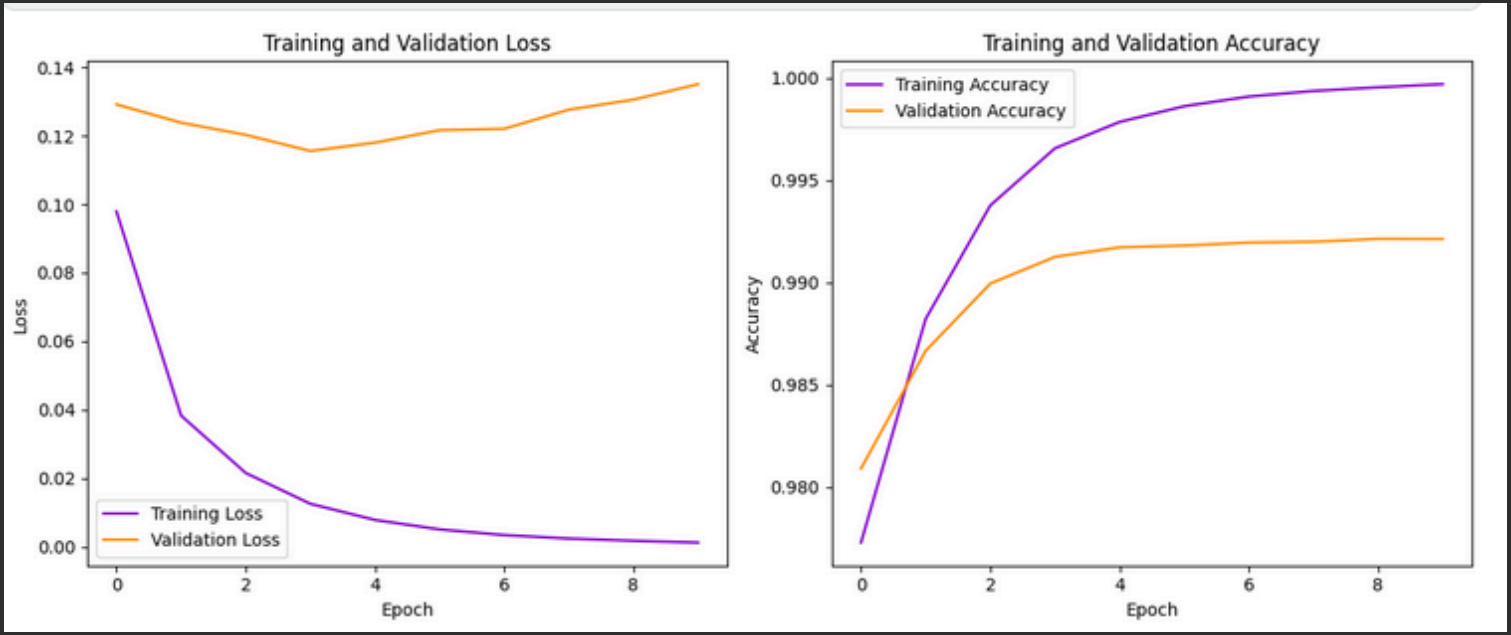
Layer (type)	Output Shape	Param #
embedding_2 (Embedding)	(None, 113, 50)	1,050,500
bidirectional_2 (Bidirectional)	(None, 113, 200)	120,800
dense_2 (Dense)	(None, 113, 9)	1,809

Total params: 1,173,109 (4.48 MB)

Trainable params: 1,173,109 (4.48 MB)

Non-trainable params: 0 (0.00 B)

Learning Curves:



Results:

The model achieved an overall accuracy of **99%,** indicating that it performed very well on the test data.

```
Test Loss: 0.18002460896968842
Test Accuracy: 0.9897642135620117
```

- **B-LOC:**The precision, recall, and F1-score for this class are 0.85, 0.81, and 0.83, respectively. The model performed well in identifying the beginning of location entities.
- **B-MISC:** The precision, recall, and F1-score for this class are 0.80, 0.65, and 0.72, respectively. The model's performance is decent but lower compared to other classes.
- **B-ORG:** The precision, recall, and F1-score for this class are 0.84, 0.57, and 0.68, respectively. The model had higher precision but lower recall for organization entities.
- **B-PER :** The precision, recall, and F1-score for this class are 0.93, 0.44, and 0.60, respectively. The model performed well in identifying the beginning of person entities, especially in terms of precision.

	precision	recall	f1-score	support
B-LOC	0.85	0.81	0.83	1667
B-MISC	0.80	0.65	0.72	702
B-ORG	0.84	0.57	0.68	1660
B-PER	0.93	0.44	0.60	1616
I-LOC	0.79	0.65	0.71	257
I-MISC	0.59	0.42	0.49	216
I-ORG	0.74	0.49	0.59	835
I-PER	0.94	0.23	0.38	1155
O	0.99	1.00	1.00	382081
accuracy			0.99	390189
macro avg	0.83	0.59	0.67	390189
weighted avg	0.99	0.99	0.99	390189

- **I-LOC, I-MISC, I-ORG,I-PER**

These classes represent entities that occur inside the named entities. The model's performance is reasonable but lower compared to the beginning of entity classes.

- **O:** The precision, recall, and F1-score for this class are exceptionallyhigh, indicating that the model correctly identifies non-entity tokens.

Adding Time Distributed Dense Layer and increase number of units:

Model: "sequential_3"

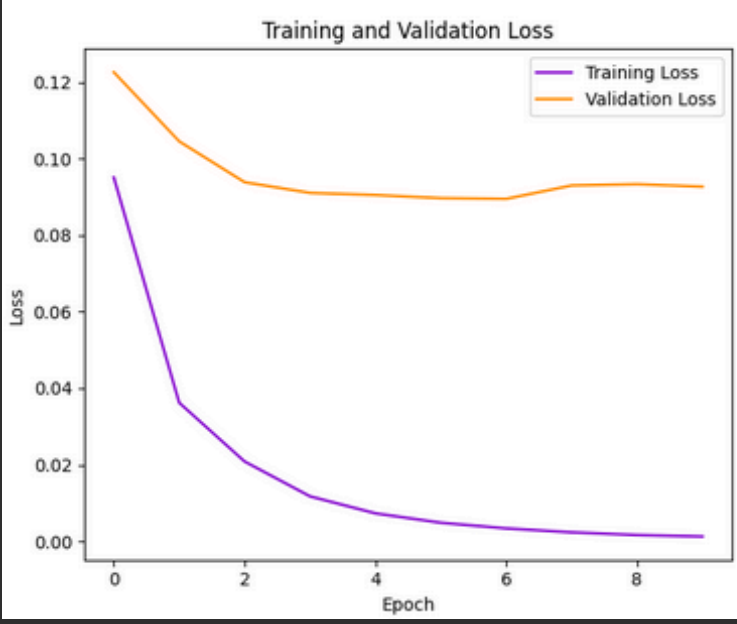
Layer (type)	Output Shape	Param #
embedding_3 (Embedding)	(None, 113, 50)	1,050,500
bidirectional_3 (Bidirectional)	(None, 113, 256)	183,296
time_distributed (TimeDistributed)	(None, 113, 9)	2,313

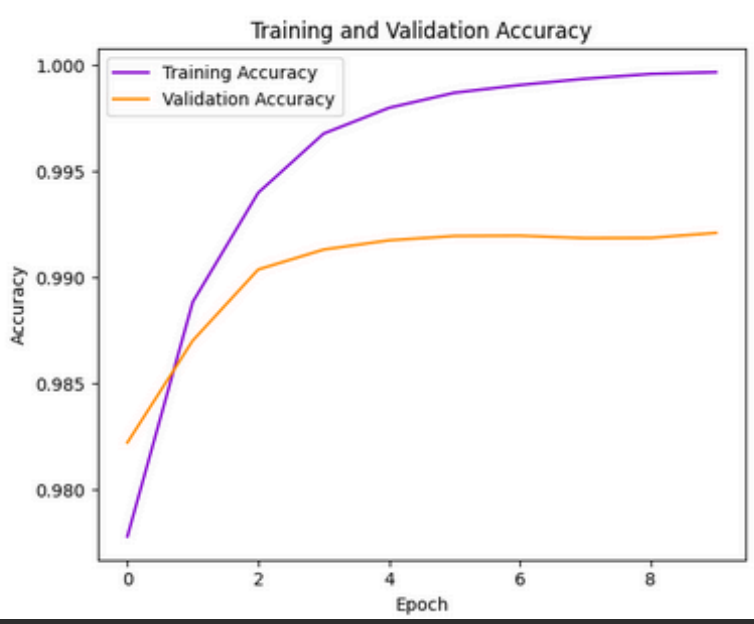
Total params: 1,236,109 (4.72 MB)

Trainable params: 1,236,109 (4.72 MB)

Non-trainable params: 0 (0.00 B)

Learning Curves:





Results:

The updated classification report shows some improvements in precision, recall, and F1-score for certain classes compared to the previous report. However, there are still areas where the model's performance can be enhanced.

```
Test Loss: 0.12103500962257385
Test Accuracy: 0.9898975491523743
```

	precision	recall	f1-score	support
B-LOC	0.88	0.81	0.84	1667
B-MISC	0.77	0.67	0.71	702
B-ORG	0.85	0.57	0.69	1660
B-PER	0.91	0.47	0.62	1616
I-LOC	0.81	0.65	0.72	257
I-MISC	0.64	0.34	0.45	216
I-ORG	0.78	0.46	0.58	835
I-PER	0.92	0.24	0.38	1155
O	0.99	1.00	1.00	382081
accuracy			0.99	390189
macro avg	0.84	0.58	0.66	390189
weighted avg	0.99	0.99	0.99	390189

Conclusion:

The use of Bidirectional LSTM layers allows the model to consider both past and future context when predicting entity labels, leading to better performance.
However, the effectiveness of the model depends on the quality and representativeness of the training data, as well as the hyperparameters chosen during training.