

## 模型架構：

```

SecBERT_BiLSTM_CNN(
  (bert): BertModel([
    (embeddings): BertEmbeddings(
      (word_embeddings): Embedding(52000, 768, padding_idx=0)
      (position_embeddings): Embedding(514, 768)
      (token_type_embeddings): Embedding(1, 768)
      (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
      (dropout): Dropout(p=0.1, inplace=False)
    )
    (encoder): BertEncoder(
      (layer): ModuleList(
        (0-5): 6 x BertLayer(
          (attention): BertAttention(
            (self): BertSdpaSelfAttention(
              (query): Linear(in_features=768, out_features=768, bias=True)
              (key): Linear(in_features=768, out_features=768, bias=True)
              (value): Linear(in_features=768, out_features=768, bias=True)
              (dropout): Dropout(p=0.1, inplace=False)
            )
            (output): BertSelfOutput(
              (dense): Linear(in_features=768, out_features=768, bias=True)
              (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
              (dropout): Dropout(p=0.1, inplace=False)
            )
          )
          (intermediate): BertIntermediate(
            (dense): Linear(in_features=768, out_features=3072, bias=True)
            (intermediate_act_fn): GELUActivation()
          )
          (output): BertOutput(

```

```

      (pooler): BertPooler(
        (dense): Linear(in_features=768, out_features=768, bias=True)
        (activation): Tanh()
      )
    )
  )
  (lstm): LSTM(768, 256, batch_first=True, bidirectional=True)
  (cnn): Conv1d(512, 128, kernel_size=(3,), stride=(1,), padding=(1,))
  (dropout): Dropout(p=0.1, inplace=False)
  (classifier): Linear(in_features=128, out_features=29, bias=True)
)

```

## 1. 使用 SecBERT，作為 backbone

## 嵌入層 (Embedding):

- word\_embeddings: 詞嵌入層，單詞表大小為 52000，維度為 768。
- position\_embeddings: 位置嵌入層，最大序列長度為 514。
- token\_type\_embeddings: 分段嵌入，僅使用一個 segment。
- LayerNorm + Dropout: 提升穩定性與防止過擬合

編碼器 (Encoder):

含有 6 層 Transformer 編碼器 (BertLayer)。

自注意力 (Self-Attention)

前饋神經網路 (Feed-Forward Network, FFN)

LayerNorm 與 Dropout 構成殘差連接 (Residual Connections)

## 2. 雙向 LSTM 層 (BiLstm)

輸入維度：768 (與 BERT 輸出一致)

隱藏層維度：256

雙向輸出：所以最終輸出維度為 512 (256 forward + 256 backward)

功能：學習序列中上下文關係，加強 BERT 表徵的序列建模能力。

## 3. 一維卷積層 CNN

將 BiLSTM 輸出的 shape [batch\_size, seq\_len, 512] 轉換為 [batch\_size, 512, seq\_len] 傳入 CNN。

## 4. Dropout 層:

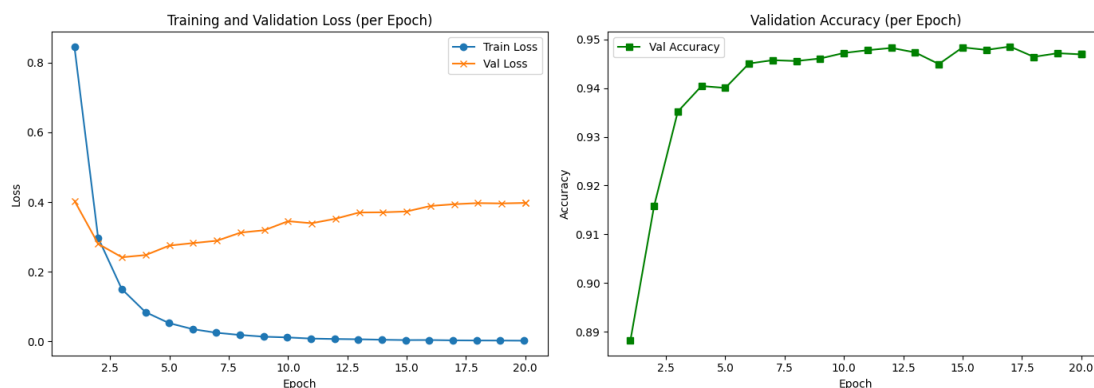
使用  $p=0.1$ ，隨機丟棄部分特徵以減少過擬合。

## 5. 分類層 (classifier)

全連接層，輸入維度為 CNN 輸出的 128，輸出為 29 個類別 (對應 NER 的 tag 數量)。

每個 token 對應一個標籤，使用 CrossEntropyLoss 計算損失

## 訓練結果:



在訓練損失的部分可以看到上圖，從第五個 epoch 開始損失的下降就趨於平緩，至於驗證集的損失，從第五個 epoch 左右開始緩慢上升，總結來說模型訓練到後面能提升的表現較有限。

再來到驗證集的精確度部分，從第一個 epoch 到第四個 epoch 有大幅度的精確度提升，但從第六個 epoch 開始驗證集的精確度提升就開始趨於平緩。總結來說因為是拿預訓練的模型去做微調，所以第一個 epoch 開始就能有 0.89 的精確度，訓練到後面也能提升到快 0.95 的精確度。

## 測試集結果:

	precision	recall	f1-score	support
0	0.99	0.98	0.98	14093
In	0.00	0.00	0.00	0
It	0.00	0.00	0.00	0
B-HackOrg	0.81	0.87	0.84	368
I-HackOrg	0.78	0.83	0.80	133
B-OffAct	0.90	0.83	0.86	150
I-OffAct	0.90	0.81	0.86	91
B-SamFile	0.93	0.91	0.92	248
I-SamFile	0.94	0.93	0.93	95
B-SecTeam	0.95	0.95	0.95	152
I-SecTeam	0.78	0.93	0.85	56
B-Time	0.91	0.93	0.92	167
I-Time	0.94	0.90	0.92	101
B-Org	0.81	0.79	0.80	137
I-Org	0.77	0.84	0.80	101
B-Purp	0.85	1.00	0.92	115
I-Purp	0.83	1.00	0.91	145
B-Exp	0.98	1.00	0.99	132
I-Exp	0.98	1.00	0.99	49
B-Idus	0.94	0.96	0.95	127
I-Idus	0.84	1.00	0.92	38
B-Area	0.88	0.89	0.89	216
I-Area	0.78	0.88	0.83	52
B-Way	0.91	1.00	0.95	98
I-Way	0.91	1.00	0.95	88
B-Features	0.95	1.00	0.97	116
I-Features	0.96	1.00	0.98	151
B-Tool	0.82	0.80	0.81	312
I-Tool	0.79	0.77	0.78	153
accuracy			0.96	17684
macro avg	0.82	0.86	0.84	17684
weighted avg	0.96	0.96	0.96	17684

從測試結果來看，模型 NER 任務中達到 整體準確率 96%，顯示模型具備穩定且高效的辨識能力。

常見標籤如 B-Exp、I-Exp、B-Features、I-Features 等的 precision、recall、f1-score 幾乎都接近 1.0，表示模型對這些實體有非常高的辨識能力。

B-Purp、I-Purp、B-Way、I-Way 等類別也都達到或接近滿分 recall，表示模型在這些關鍵實體上的召回能力非常好，不容易漏判。

另外像 B-Tool、I-Tool 雖然 precision 和 recall 稍低，但依然保持在 0.78~0.82 的範圍，屬於可接受的水準。

少數類別如 B-Org 和 I-Org 的 f1-score 落在 0.80 左右，可能因為樣本數較少或標註不一致造成模型表現略微下降。

類別 In 和 It，應該在讀取測試集，因為測試集格式的關係，標籤錯的類別，但實際應該是沒有這個類別，所以測試集中未出現這兩個標籤為正常的表現。

從 macro average（平均考慮各類別）來看：

Precision: 0.82

Recall: 0.86

F1-score: 0.84

代表模型在不同標籤間的表現差異相對穩定，沒有嚴重的偏向單一類別。

整體來說，本模型在處理 DNRTI 資料集上表現良好，對大部分類別都具備良好的辨識與泛化能力，已能有效支援 NER 任務。