Name: 陳以瑄 StudentID: 109705001

Part 0

■ Briefly explain the method you implemented and give an example

我做了以下四件事情:

1. 移除換行

   我使用 regex 將
移除

   舉例：

   原文：classic use word.<br / ><br / >It called OZ nickname given

   處理後：classic use word. It called OZ nickname given

2. 移除標點符號

   我使用 string.translate(str.maketrans('', '', string.punctuation))
   舉例：

   原文：classic use word. It called OZ nickname given

   處理後：classic use word It called OZ nickname given

3. 統一成小寫

   我使用 string.lower()
   舉例：

   原文：classic use word It called OZ nickname given

   處理後：classic use word it called oz nickname given

4. lemmatization
   我使用 WordNetLemmatizer()來還原字型
   舉例：

   原文：classic use word it called oz nickname given

   處理後：classic use word it call oz nickname give

Part 1

■ Briefly explain the concept of perplexity in report and discuss how it will be influenced.

Perplexity(困惑度)是用來衡量語言模型預測測試資料的好壞程度。它的概念是當語言模型訓練好後，它計算測試資料裡那些正常的語句的機率要越高越好，因為這表示它的預測能力較佳。而困惑度越低就表示它預測的越好。

當資料量比較大或比較多元時，或是模型比較複雜時，模型的困惑度會較低，因為它可以捕捉更多潛在 pattern。

■ Screenshot the outputs and tell your observations about the differences in the perplexity caused by the preprocessing methods

1. without preprocess

   ```
   Perplexity of ngram: 116.26046015880357
   ```

2. with remove stop words

   ```
   Perplexity of ngram: 195.43245350997685
   ```

3. with my method

   ```
   Perplexity of ngram: 293.70311020218367
   ```

隨著我預處理的步驟越多，我的模型困惑度越大。可能是因為在預處理時我移除了某些重要特徵，導致模型的預測能力變差。

Part 2

■ Briefly explain the two pre-training steps in BERT.

1. masking input

   隨機決定一些文字將它蓋掉成非原語言的字或是換成其他原語言的文字，希望 BERT 可以填出這個空缺。

2. next sentence prediction

將兩個句子用[SEP]分開，希望 BERT 能區分此兩句子是否相接。

■ Briefly explain four different BERT application scenarios

1. 輸入一個 sequence，輸出一個類別

   例如 sentiment analysis，區分輸入的句子是正面或是負面。

2. 輸入一個 sequence，輸入相同長度的 sequence

   例如詞性標註，對於輸入句子的每個詞輸出他們的詞性。

3. 輸入兩個 sequence，輸出一個類別

   例如 natural language inference (NLI)，給定前提與假設兩個句子，推出兩者的關係是衝突、因果或是無關。

4. 問答系統

   給 BERT 一篇文章，並且詢問答案就在文章中的問題，希望 BERT 回答出答案是文章中的第幾個字到第幾個字。

■ Discuss the difference between BERT and distilBERT?

DistilBERT 是簡化版的 BERT，它的參數只有 BERT 的四成，層數只有 BERT 的一半。因為具有較小的模型所以 DistilBERT 的速度較快，但同時又保有好的準確度。

■ Screenshot the required test F1-score.

1. without preprocess

   ```
   Epoch: 0, F1 score: 0.936, Precision: 0.936, Recall: 0.936, Loss: 0.0
   ```

2. with remove stop words

   ```
   Epoch: 0, F1 score: 0.9328, Precision: 0.9328, Recall: 0.9328, Loss: 0.0001
   ```

■ Explain the relation of the Transformer and BERT and the core of the Transformer.

Transformer 的核心是使用 self-attention 做 encoder-decoder，使它在訓練時他考慮輸入的整個 sequence。而 BERT 是由 Transformer 的 Encoder 組成。

Part 3

■ Briefly explain the difference between vanilla RNN and LSTM.

LSTM 是在 vanilla RNN 上加了 input gate, forget gate 跟 output gate，使它可以控制哪些舊資訊要忘掉，哪些新資訊要被記下以及多少資訊要被輸出，這有助於它在長期記憶的表現。

■ Please explain the meaning of each dimension of the input and output for each layer in the model.

我採用的架構如下：
nn.Embedding(num_embeddings = vocab_size, embedding_dim = 250)
nn.GRU(input_size = 250, hidden_dim = 500, num_layers = 5)
nn.Linear(in_features = 500, out_features = 2)
Embedding 層的輸入大小是資料集中有多少種詞彙(不重複)，輸出維度為 250。
再來的 GRU 共有五層，第一層的輸入為度是 250，輸出為 500；接下來的四層輸入輸出維度都是 500。
最後的 Linear 層輸入為 500，而輸出為 2 即 label 的種類。

■ Screenshot the required test F1-score.

1. without preprocess

```
Epoch: 0, F1 score: 0.3333, Precision: 0.75, Recall: 0.5, Loss: 0.0006
Epoch: 1, F1 score: 0.3333, Precision: 0.75, Recall: 0.5, Loss: 0.0005
Epoch: 2, F1 score: 0.497, Precision: 0.5328, Recall: 0.5254, Loss: 0.0006
Epoch: 3, F1 score: 0.4463, Precision: 0.5389, Recall: 0.5186, Loss: 0.0005
Epoch: 4, F1 score: 0.4076, Precision: 0.5618, Recall: 0.5164, Loss: 0.0005 0001
Epoch: 5, F1 score: 0.6708, Precision: 0.6711, Recall: 0.6709, Loss: 0.0005
Epoch: 6, F1 score: 0.8782, Precision: 0.8805, Recall: 0.8784, Loss: 0.0002
Epoch: 7, F1 score: 0.8984, Precision: 0.8988, Recall: 0.8984, Loss: 0.0001
Epoch: 8, F1 score: 0.8911, Precision: 0.894, Recall: 0.8913, Loss: 0.0002
Epoch: 9, F1 score: 0.8983, Precision: 0.8992, Recall: 0.8984, Loss: 0.0
```

2. with remove stop words

```
Epoch: 0, F1 score: 0.3342, Precision: 0.4374, Recall: 0.4996, Loss: 0.0006
Epoch: 1, F1 score: 0.3333, Precision: 0.75, Recall: 0.5, Loss: 0.0006
Epoch: 2, F1 score: 0.3337, Precision: 0.5834, Recall: 0.5001, Loss: 0.0006
Epoch: 3, F1 score: 0.556, Precision: 0.5943, Recall: 0.5767, Loss: 0.0006
Epoch: 4, F1 score: 0.84, Precision: 0.8517, Recall: 0.8412, Loss: 0.0002
Epoch: 5, F1 score: 0.8691, Precision: 0.8749, Recall: 0.8696, Loss: 0.0004
Epoch: 6, F1 score: 0.8744, Precision: 0.8823, Recall: 0.875, Loss: 0.0001
Epoch: 7, F1 score: 0.8862, Precision: 0.8875, Recall: 0.8863, Loss: 0.0
Epoch: 8, F1 score: 0.8883, Precision: 0.8883, Recall: 0.8883, Loss: 0.0001
Epoch: 9, F1 score: 0.8543, Precision: 0.8725, Recall: 0.8559, Loss: 0.0
```

Discussion

■ Discuss the innovation of the NLP field and your thoughts of why the technique is evolving from ngram -> LSTM -> BERT.

一開始使用 n-gram 來預測在 n 個單字之後的下一個為何，但有些時候，要預測的字其實是要基於往前超過 n 個的那個字，而 n-gram 的定義使得它沒有辦法有這類長期記憶。因此就研發出 LSTM，它透過三種閘門，得以處理這種長期依賴關係。但其實 LSTM 也沒法處裡太長的記憶，而 BERT 基於 Transformer 的架構，讓它可以更有效的處裡長期依賴關係。