NLP-Final 書面報告

Group 21 108062656 顧翔予 110065503 劉庭銘 110065532 曾天裕

動機：

在自然語言處理，或者是在任何機器學習的領域中，學習用的資料都是決定模型是否成功的關鍵因素之一。若是在資料量極為稀少的情況下強硬地進行訓練，可能會造成模型的準確率極為低落。為了解決這個問題，我們小組嘗試在訓練資料集短少的情況下使用Data augmentation的方式增加訓練資料量，並嘗試提升模型的預測準確率。

實作過程：

資料來源：

我們使用課堂上week9的作業所提供的訓練資料集，也就是判斷一個包含party的英文句子中，該party屬於下列三種senses的哪一種：

```
SENSE = {
    1: 'a social event at which a group of people meet to talk, eat, drink, dance, etc.', # 派對
    2: 'an organization of people with particular political beliefs', # 政黨
    3: 'a single entity which can be identified as one for the purposes of the law' # （法庭）當事人；……方
}
```

該資料集由助教提供，包含707筆資料，每一筆資料是一句英文句子，其中必定包含party這個英文字，並且包含其正確的標籤。

我們只取其中**前50筆資料**作為我們的training data set，並將剩餘的資料作為test data set。

模型訓練：

因為模型本身的品質並不是我們考量的重點，我們僅使用一個簡單的LSTM模型來預測：

```
model = Sequential()
model.add(tf.keras.layers.LSTM(4))
model.add(tf.keras.layers.Dense(128,
            kernel_initializer='random_normal',
            bias_initializer='zeros'))
model.add(tf.keras.layers.Dense(OUTPUT_CATEGORY, activation='sigmoid'))
model.compile(optimizer='Adam',
                loss=tf.keras.losses.CategoricalCrossentropy(from_logits=False),
                metrics=['accuracy'])
```

Data augmentation方法：

我們使用的data augmentation方法共有四種：

- Swap random two word:
  此方法會在一段句子隨機選擇兩個字，並將其對調，得到轉換後的新句子。我們重複三次隨機對調，並因此生成三個句子。另外每次對調都是使用前面已對調過的句子，並無reset。
- Delete Random word:
  此方法與第一種相同，首先會先從句子隨機選擇某些字，但接下來是將選擇到的文字刪除，以此得到簡化版的相近句子。我們的做法為先隨機選兩個字a,b，並排列刪除，當然每一次的刪除都會回復成原本的句子。也就是(1.)只刪掉a(2.)只刪掉b(3.)a,b全刪除，最後便能一次加入三段新句子。
- Back Translation:
  此方法為利用google translation的技術，利用google翻譯先翻成他國語言，再翻譯回英文，生成意思相近的不同句子。至於Google translation則需要申請google cloud帳號才能使用。實作中，我們使用到的語言有繁體中文，波蘭文及泰文，因此能一次性多生成三個句子。
- Random words replacing with similar word:
  在一段句子中，隨機選擇某些字，將之替換為意思相近的其他用字，以此增加句子數量。在我們的實作中，我們會在一個句子中隨機選擇四個字並作替換，以此多得到一個新句子。

結果：

比較：

Random words replacing with similar word		V				V	V	V				V	V	V		V
Back Translation			V			V			V	V		V	V		V	V
Delete random word				V			V		V		V	V		V	V	V
Swap random two word					V			V		V	V		V	V	V	V
Train size/ Accuracy(%)	50 56.65	215 72.02	200 75.72	200 73.91	200 74.50	859 77.01	876 75.37	876 75.35	800 77.54	800 75.28	800 76.48	3512 76.05	3524 76.95	3600 77.72	3200 76.01	14112 75.92

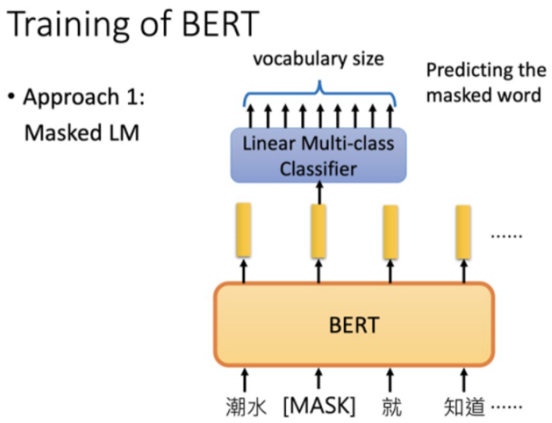
以上為應用四種data augmentation方式所產生的資料集。
初始的result accuracy為56.65%，並沒有使用任何data augmentation的training size=50。
而隨著data augmentation的方法數增加，生成更多句子，training size自然也跟著變大，最後使用全部方法得到的size為14112筆data。
而比較結果中，最好的是使用第一，三，四種的混和方法，training size擴大到3600筆，accuracy則為最高的77.72%。

在執行速度方面，Back Translation因為需要google cloud所提供的翻譯服務，因此執行速度明顯比其他三者要慢。但在資料集很少的情況下，執行時間都在可接受的範圍之內。

結論：

可以看到應用了以上這四種方式之後所產生的資料集，其模型預測準確度都從原先的**56%左右**提升到了**75%左右**。雖然方法間的混用確實有產生差異，但其差異並不大。可以確認的事情是在資料量極為稀少的情況下，data augmentaion是一個能夠快速實作並執行的提升模型預測準確率的有效方法。

其他方式：



利用Bert Masked LM方法將我們要做的特定詞 ”party” 替換為 “[MASK]” ，然後藉由BERT進行transfer learning。

> a naked party, also known as a nude party, is a party where the participants are required to be nude.

> a naked [MASK], also known as nude [MASK], is a [MASK] where the participants are required to be nude.

實驗結果：
training set size: 50, test set size: 657, accuracy: 89.34%
經由這個實驗，我們可以看出BERT pretrain的強大，在training set很小的情況下也可以達到很好的準確率。