學號：r06521705 系級： 土木所營管組碩一 姓名：陳思愷

1. (1%) 請說明你實作的 RNN model，其模型架構、訓練過程和準確率為何？ (Collaborators:)

   答：

	Loss	Acc	Val_loss	Val_acc
Epoch1	0.5338	0.7320	0.4556	0.7922
Epoch2	0.4688	0.7815	0.4362	0.8000
Epoch3	0.4499	0.7908	0.4313	0.8034
Epoch4	0.4392	0.7971	0.4205	0.8074
Epoch5	0.4286	0.8031	0.4153	0.8118
Epoch6	0.4218	0.8015	0.4156	0.8099

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_2 (LSTM)                (None, 64)                42240
_________________________________________________________________
dropout_2 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_2 (Dense)              (None, 2)                 130
=================================================================
Total params: 42,370
Trainable params: 42,370
Non-trainable params: 0
```

🡐 word2vec

這部分我嘗試了兩種方式，第一種是利用 word2vector 將字詞轉成向量後丟入 model 當中(上面的圖)，第二種是利用 tokenizer 將字詞排序後再丟入含有 embedding 層的 model 進行訓練，兩者的表現差不多。

2. (1%) 請說明你實作的 BOW model，其模型架構、訓練過程和準確率為何？
(Collaborators:)

答：

建立的 BOW 字典有 15000 字，丟到與 tokenizer 差不多的 model 當中，訓練結果如下感覺從 validation 部分來看很快分數就上不去了，整體表現也沒有之前 RNN 的做法來的好，詞的順序對於語意的判別感覺還是很重要的。

	Loss	Acc	Val_loss	Val_acc
Epoch1	0.4930	0.7750	0.4521	0.7915
Epoch2	0.4142	0.7854	0.4484	0.7901
Epoch3	0.3895	0.8074	0.4689	0.7889
Epoch4	0.3652	0.8364	0.4724	0.7945
Epoch5	0.3158	0.8516	0.4802	0.7886
Epoch6	0.3021	0.8745	0.5201	0.7910

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 100, 128)          2560000
_________________________________________________________________
lstm_1 (LSTM)                (None, 64)                49408
_________________________________________________________________
dropout_1 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_1 (Dense)              (None, 2)                 130
=================================================================
Total params: 2,609,538
Trainable params: 2,609,538
Non-trainable params: 0
_________________________________________________________________
```

3. (1%) 請比較 bag of word 與 RNN 兩種不同 model 對於"today is a good day, but it is hot"與"today is hot, but it is a good day"這兩句的情緒分數，並討論造成差異的原因。

   (Collaborators:)

   答：

"today is a good day, but it is hot":

在 RNN 當 中 : score of label[0] : 0.21   score of label[1] : 0.79

在 BOW 當 中 : score of label[0] : 0.412   score of label[1] : 0.588

"today is hot, but it is a good day":

在 RNN 當 中 : score of label[0] : 0.124   score of label[1] : 0.876

在 BOW 當 中 : score of label[0] : 0.412   score of label[1] : 0.588

BOW 主要是紀錄了詞語出現的頻率，捨去了字與字之間的順序關係，所以這兩句話對於 BOW 的模型中是一樣的故分數會相同，反之，RNN 的模型會將先後順序考慮進去， 故在分數上會有所不同。

4. (1%) 請比較"有無"包含標點符號兩種不同 tokenize 的方式，並討論兩者對準確率的影響。

   (Collaborators:)

   答：

無標點符號(只留下文字和空格) : 0.81292

留下標點符號(留下標點符號其他東西易清除乾淨) : 0.81804

感覺標點符號對於預測的準確率是有相當的影響程度的，像是句號或驚嘆號或…根句子結合後能夠呈現出相當明顯的正反情緒

5. (1%) 請描述在你的 semi-supervised 方法是如何標記 label，並比較有無 semi- surpervised training 對準確率的影響。

   (Collaborators:)

   答：

我使用的是 self-training 的方式，先利用 label data train 出一個 model，在將 unlabel

data 丟入進行預測，將門檻設為 : 0.8 (高於 0.8 的 data 我們把它做標記後加入 training data ，且從 unlabel data 中移除)，爾後再利用更新過後的 data 再 train 一次新的 model，重複上述步驟直到 unlabel data 被全數編入。

上述是施作的方式，但實際施作起來並沒有讓表現變好，我認為可能的原因是 data 的預處理做得不夠完整，上網有找到一些可能的解決方式例如，stopword 或 大小寫一致等等，但這次並沒有順利嘗試出更好的表現。