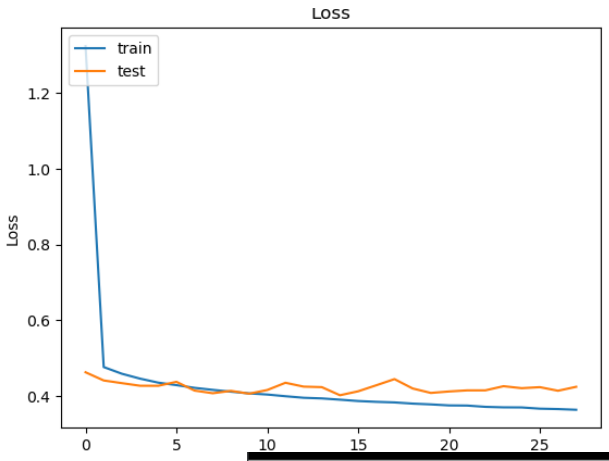
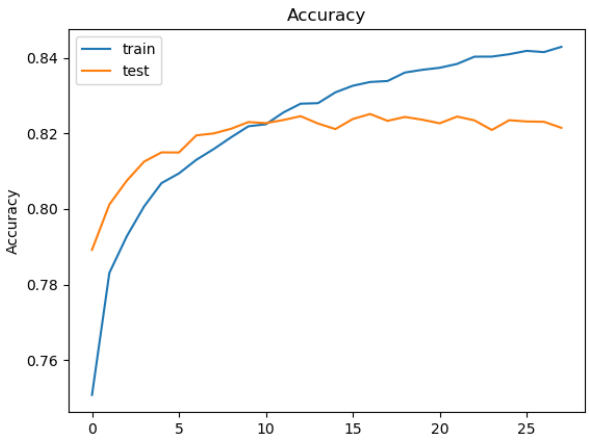
學號：R06942048 系級： 電信碩一 姓名：林彥伯

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？



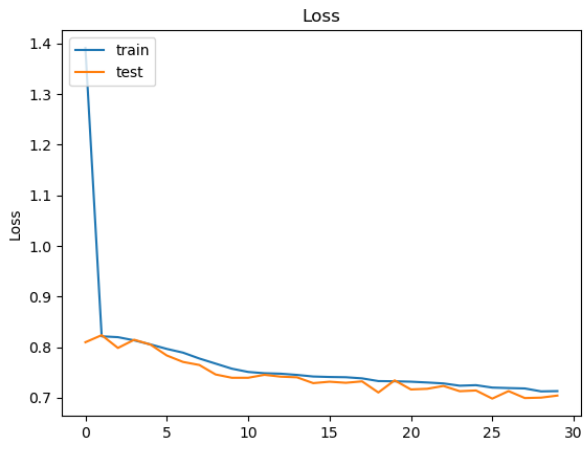


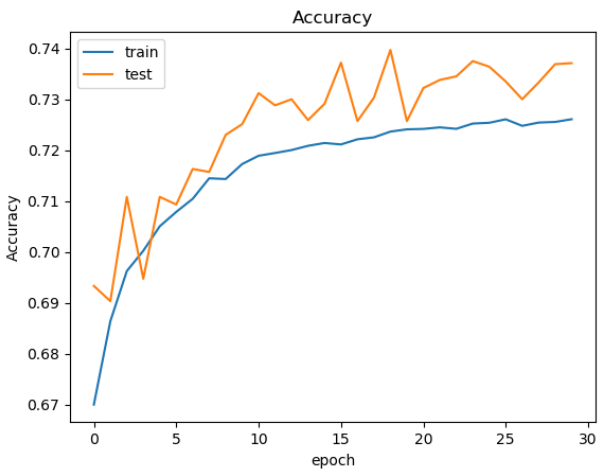
Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 40)	0
embedding_1 (Embedding)	(None, 40, 100)	2500000
lstm_1 (LSTM)	(None, 40, 256)	365568
lstm_2 (LSTM)	(None, 64)	82176
dense_1 (Dense)	(None, 512)	33280
leaky_re_lu_1 (LeakyReLU)	(None, 512)	0
dropout_1 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 1)	513

Total params: 2,981,537
Trainable params: 481,537
Non-trainable params: 2,500,000

本次實作是先拿 testing data, semi-data, training data，來 train tokenizer(只取前 25000)跟 embedding layer weight，接著放入兩層 LSTM 然後 fully connected layer。準確率可達 82.4%左右 (不 pre-train 的 embedding layer 只有 79%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





Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 5000)	0
dense_1 (Dense)	(None, 512)	2560512
leaky_re_lu_1 (LeakyReLU)	(None, 512)	0
dropout_1 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 1)	513

Total params: 2,561,025
Trainable params: 2,561,025
Non-trainable params: 0

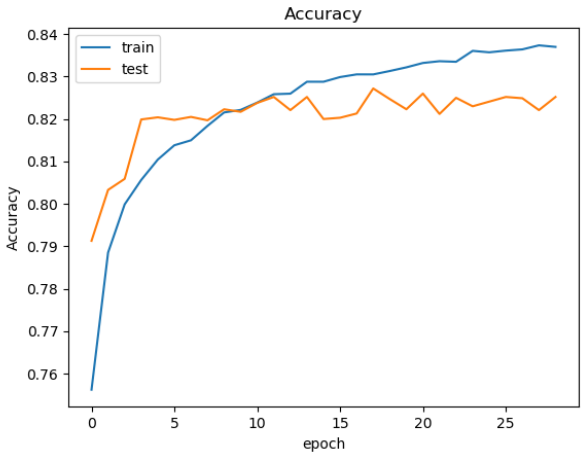
把原本 texts_to_sequences 換成 texts_to_matrix，但要注意 vocabulary size 不能太大，因為這個 matrix 的 row 是資料筆數、column 是 vocabulary size，轉完之後再放入 fully connected net 做訓練，準確度大概在 73%左右。

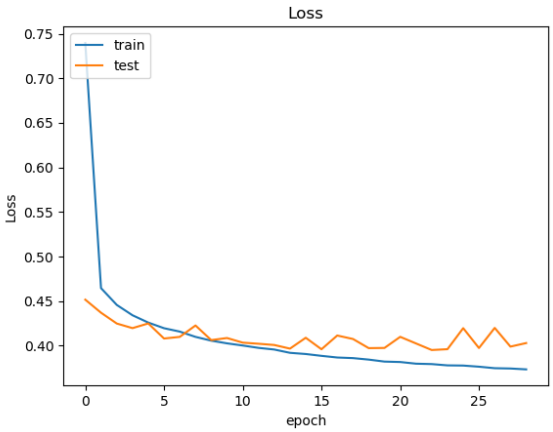
3. **(1%) 請比較 bag of word 與 RNN 兩種不同 model 對於"today is a good day, but it is hot"與"today is hot, but it is a good day"這兩句的情緒分數，並討論造成差異的原因。**

	today is a good day, but it is hot	today is hot, but it is a good day
BOW	0.6508	0.6508
LSTM	0.0658	0.9634

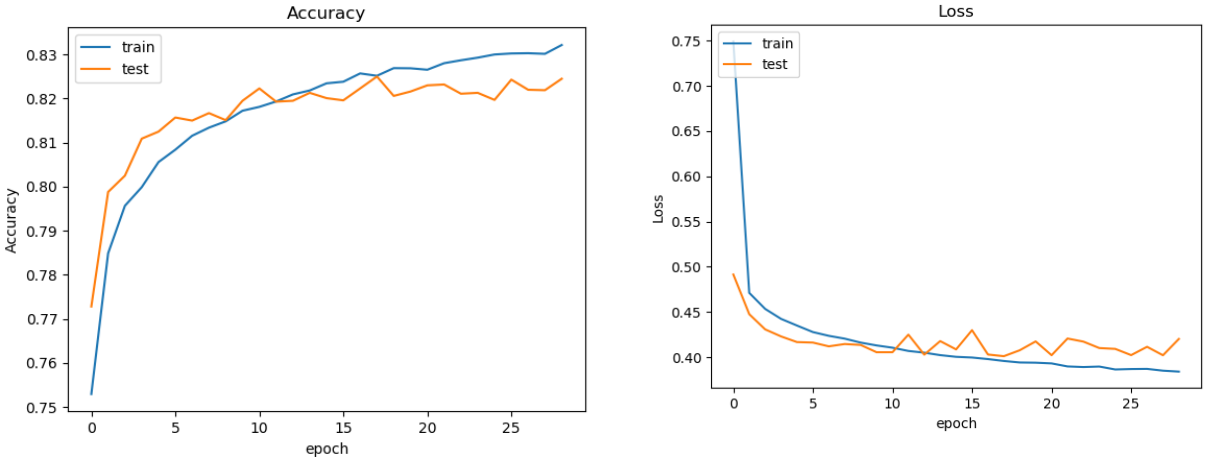
在 bow 的狀況下，會把所有的順序打斷。所以這兩個的句子的詞是都一樣的，所以 texts_to_matrix 出來的數據會是一樣的，所以 predict 後的分數也是相同的。 反之 LSTM 還有保留詞的順序，所以分數不同，不僅不同之外、分數差異也非常大(接近 0 跟 1)。

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。**





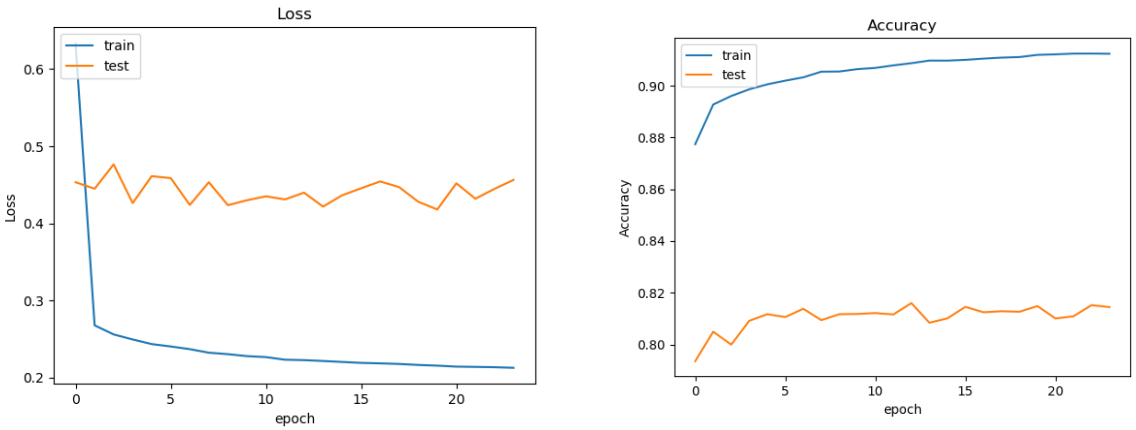
(上圖)含有標點符號的 tokenize



(上圖)無標點符號的 tokenize

含有標點符號的 tokenize 的方式的準確度會稍微較高，可以到 82.7%左右，但不包含標點符號只有 80%左右。因為這些 twitter 的用語有一些人會習慣加很多驚嘆號(!!!!!)來表示一些強烈的反應(多數 label 是 1)，如果有保留標點符號的話可以使情緒判斷更加明白。

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。



先利用 train 好的 model(準確率約 82%左右，2 層 LSTM)來標記 semi-data 的 label(情緒分數需>95%或<5%)的方式來標 label，但可以發現到 train 的準確度會變極高但 testing 卻沒有提升。原因在於拿來標 label 的 model 與後來新 train 的 model 架構太像了，會導致有點在自 high 的感覺(訓練一個可以順利標出標出資料的 model 沒甚麼意義)。所以我試著新 train 的 model 架構盡量與標 label 的不一樣，先進行 1D convolution 後進行 2 層 LSTM，不過結果還是沒有提升，看來只多一層 convolution 並不會使這個 model 與原本的 model 差異性變得太大。

有試著使用 73%左右的 bow model 來標 label，但 semi-data 太多筆了，即便我將 vocabulary size 調小一樣 out of memory。