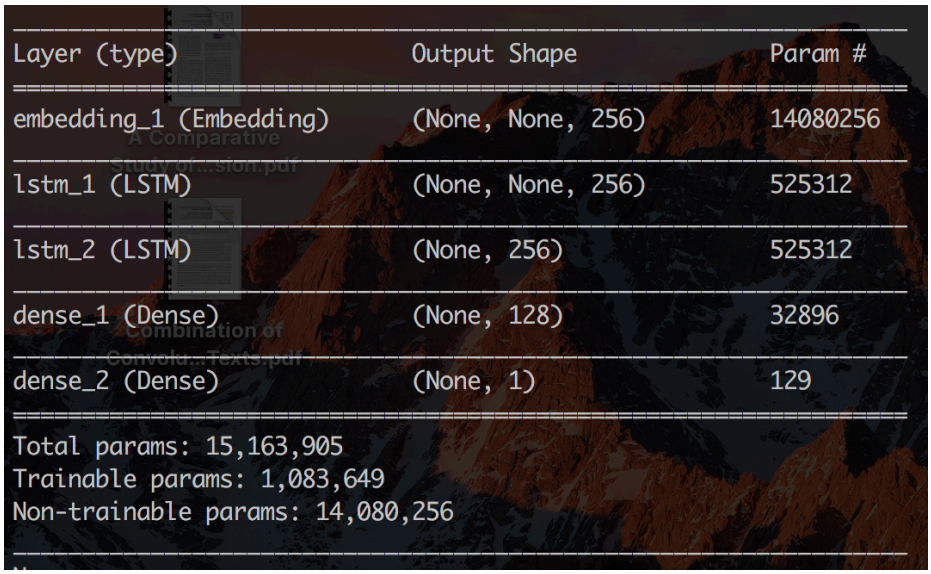
學號：R02922165　　系級： 資工碩二　　姓名：李冠穎

1. RNN 結構如下



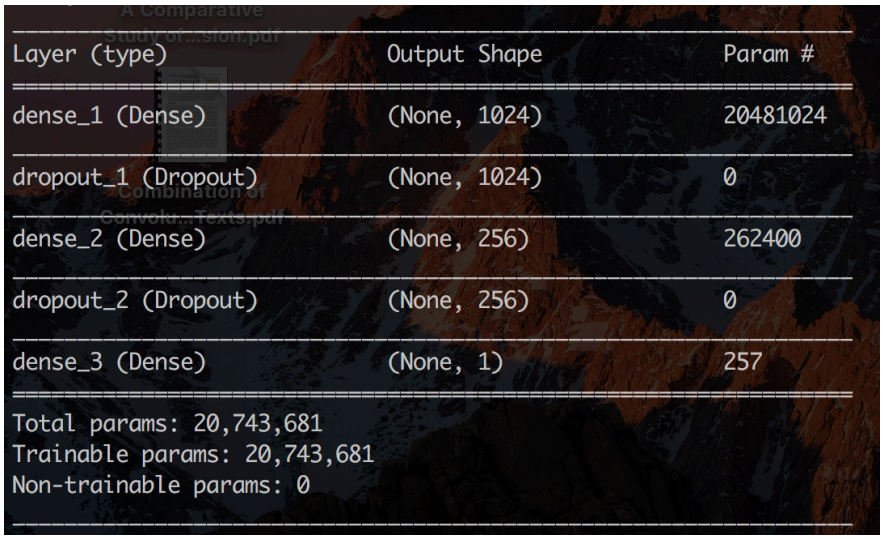
Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, None, 256)	14080256
lstm_1 (LSTM)	(None, None, 256)	525312
lstm_2 (LSTM)	(None, 256)	525312
dense_1 (Dense)	(None, 128)	32896
dense_2 (Dense)	(None, 1)	129

Total params: 15,163,905
Trainable params: 1,083,649
Non-trainable params: 14,080,256

首先將 training_label.txt, training_nolabel.txt 以及 testing_data.txt (去掉 testing_data.txt 裡頭第一行以及各行的 id 以及逗號) 合併之後，留下出現次數大於 5 的字並利用 Gensim 套件 train 維度為 256 的 word vector，接著把 model 裡頭的 Embedding weights 換成 pre-trained 好的這些 word vectors。並把 Embedding layer 設為 non-trainable 訓練 10 個 epoch 並利用 Keras 所提供的 ModelCheckPoint 存下 improving models。

如此可以在 public dataset 上得到 82.04，private dataset 則是 81.92 的準確率。

2. BOW 結構如下



Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 1024)	20481024
dropout_1 (Dropout)	(None, 1024)	0
dense_2 (Dense)	(None, 256)	262400
dropout_2 (Dropout)	(None, 256)	0
dense_3 (Dense)	(None, 1)	257

Total params: 20,743,681
Trainable params: 20,743,681
Non-trainable params: 0

僅使用 count 在前 20000 的 words (16G 記憶體塞不下，因此可能還有進步空間？雖然應該很有限)，不在前 20000 的詞彙直接忽略不算，dropout 使用 0.5，因為若不使用一下就 over-fit 了。

如此在 public 以及 private dataset 上可以達到 79.82 跟 79.63 的準確率。

3. RNN 在"today is a good day, but it is hot"的分數為 0.6306，在" today is hot, but it is a good day"則為 0.9809。BOW 不管在哪個句子都是 0.7114，這很容易想像畢竟兩個的 word count 完全一樣，所以進去的 vector 也一樣，自然分數就一樣了。

   而 RNN 因為有 hidden state 的關係，字進去的順序不一樣，hidden state 就會不一樣，因此可以達到"but it is hot"這句稍微不那麼 positive。

4. follow 一樣的結構，但使用沒有標點符號的所有 data 來 train word vector 並在 train model 的時候一樣使用沒有標點符號的 training data，在 public 及 private dataset 上得到的分數分別為 82.00 以及 81.82。

   或許是因為在辨別文字情緒的時候，標點符號如驚嘆號，或是刪節號 (…) 常常是一個分辨情緒的好媒介。譬如…可能就表示無言，或許較為偏向 negative 的情緒，而驚嘆號雖然在較為強烈的正向及負向情緒都會使用，但正向似乎多了些 (這或許可以用統計的方式來驗證)。

   不夠我認為兩者的準確率並沒有顯著的差別，所以可能需要更多驗證才足以顯示何者為優，或是 case by case。

5. 設定當分數大於 0.9 時，將 data 設為 positive，而分數小於 0.1 時，設定為 negative。但是經過試驗發現這次的 task 使用 semi-supervised 似乎沒有什麼幫助，而且 training time 會增加不少 (跟許多同學討論後發現似乎也都有這樣的結論)，在同一個 model 結構下，以上的方式在 public 以及 private dataset 上的分數為 81.80, 81.56，甚至差了些，不過這小幅度應該是屬於誤差範圍，因此可能還需要多做些實驗才會知道。