學號：B04901102 系級： 電機三 姓名：簡仲明

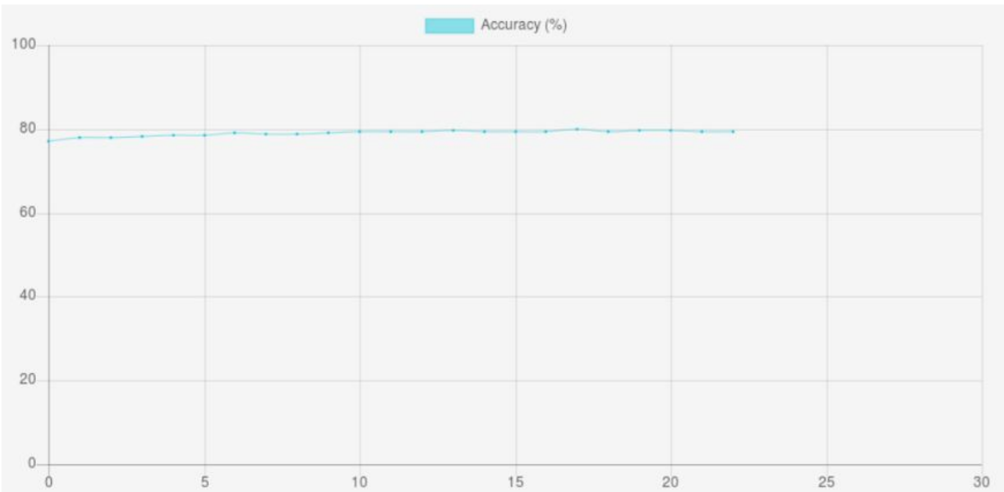
1. (1%) 請說明你實作的 RNN model，其模型架構、訓練過程和準確率為何？(Collaborators: 無)

答：

summary

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 37, 64)            128000
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 37, 128)           24704
_________________________________________________________________
bidirectional_1 (Bidirection (None, 128)               98816
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 129
=================================================================
Total params: 251,649
Trainable params: 251,649
Non-trainable params: 0
```

accuracy on validation data(%)，橫軸為epoch



參數：epoch = 30(但有patience = 5的early stop，對象為validation accuracy)

batchsize = 100

Embedding：num_words = 2000(為了配合BOW做比較)

標點符號：移除

padding length = 37(training set和testing set中最長的句子)

單層CNN：activation function = relu，無drop out

單層Bidirectional LSTM：drop_out = 0.2

單層DNN：activation function = sigmoid，無drop out

loss funciton = binary cross entropy

optimizer = adadelta

validation set ：shuffle過後的十分之一training data

validation accuracy	public accuracy	private accuracy
0.7983	0.79807	0.79936
(private + public) / 2 = 0.79872		

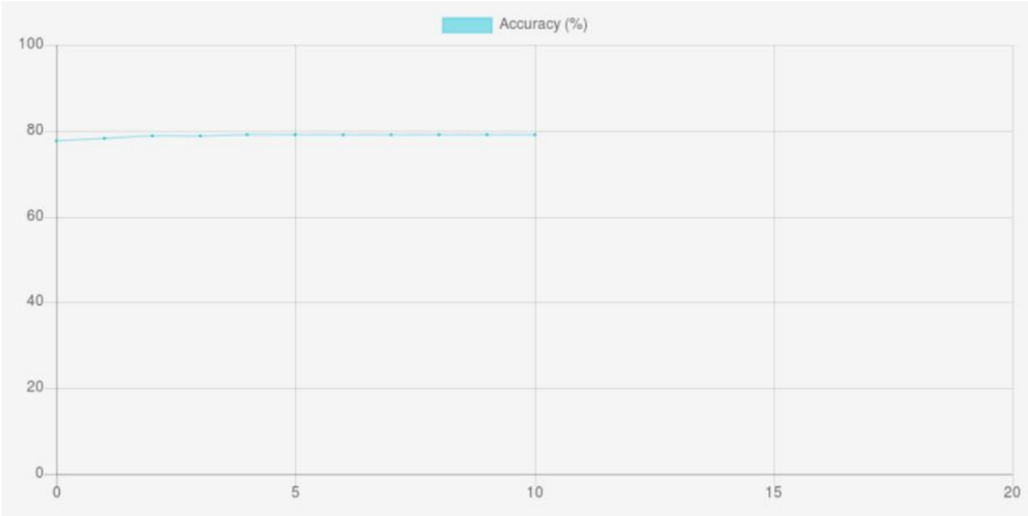
2. (1%) 請說明你實作的 BOW model，其模型架構、訓練過程和準確率為何？(Collaborators: 無)

答：

summary

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 1024)              2049024
_________________________________________________________________
dropout_1 (Dropout)          (None, 1024)              0
_________________________________________________________________
dense_2 (Dense)              (None, 128)               131200
_________________________________________________________________
dropout_2 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_3 (Dense)              (None, 16)                2064
_________________________________________________________________
dropout_3 (Dropout)          (None, 16)                0
_________________________________________________________________
dense_4 (Dense)              (None, 1)                 17
=================================================================
Total params: 2,182,305
Trainable params: 2,182,305
Non-trainable params: 0
```

accuracy on validation data(%)，橫軸為epoch



參數：epoch、batchsize、Embedding、loss funciton、optimizer、validation set皆與RNN相同

前三層DNN：activation function = relu，drop out = 0.5

output layer：activation function = sigmoid，無drop out

validation accuracy	public accuracy	private accuracy
0.7916	0.78856	0.78910
(private + public) / 2 = 0.78883		

與RNN相比，雖使用了十倍多的參數，但epoch數大約只用了三分之一，且training time只有二十分之一(5分鐘vs106分鐘，甚至還未扣除讀檔時間)，但accuracy只差了1%左右，是相當不錯的表現。

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: 無)

答：

	sentence 1	sentence 2
BOW	0.751442	0.751442
RNN	0.326962	0.935344

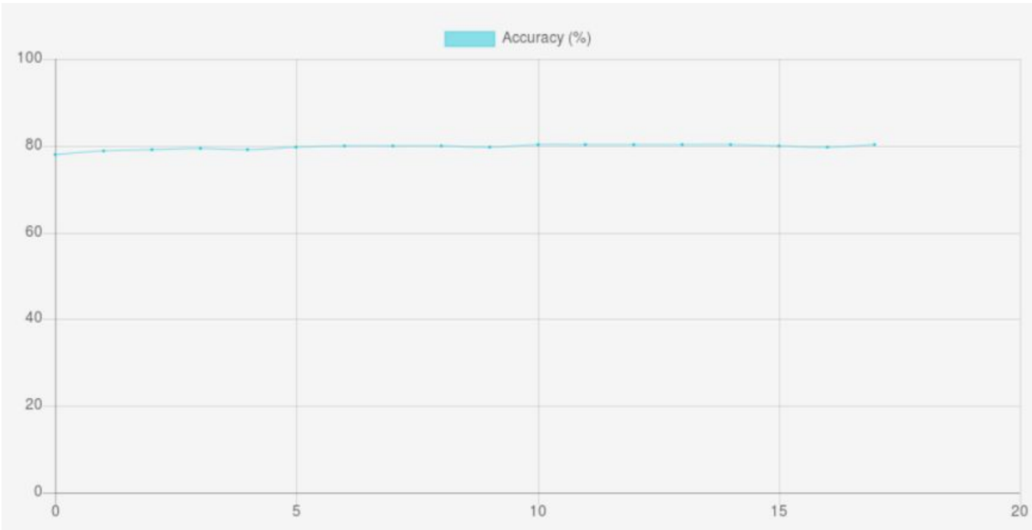
一如預期，這兩句話對於BOW model來說是一模一樣的，這兩句話的關鍵字應為good，含有這個單字的句子在預期中應該較可能是正面含意的，而轉折點but對於BOW模型來說應該不具有任何意義(因為沒辦法分辨是 好but壞或是 壞but好，兩者情緒相反但用字相同)，故應該會被BOW直接忽略，或是只會造成BOW的預測值往中間(0.5)靠攏，來達到減低loss的目標。

而對於RNN來說，but這個字應該會對前半句造成反轉的效果(A but B會造成預測值產生類似not(predict(A)) && predict(B)的效果)，因此第一句的good在前半句，造成預測值小於0.5，而第二句的good在後，因此預測值大於0.5。但若忽略but這個字(或是換為and)，這兩個句子應該都是正面含意的，因此可以看到兩者的平均值仍然大於0.5。

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: 無)

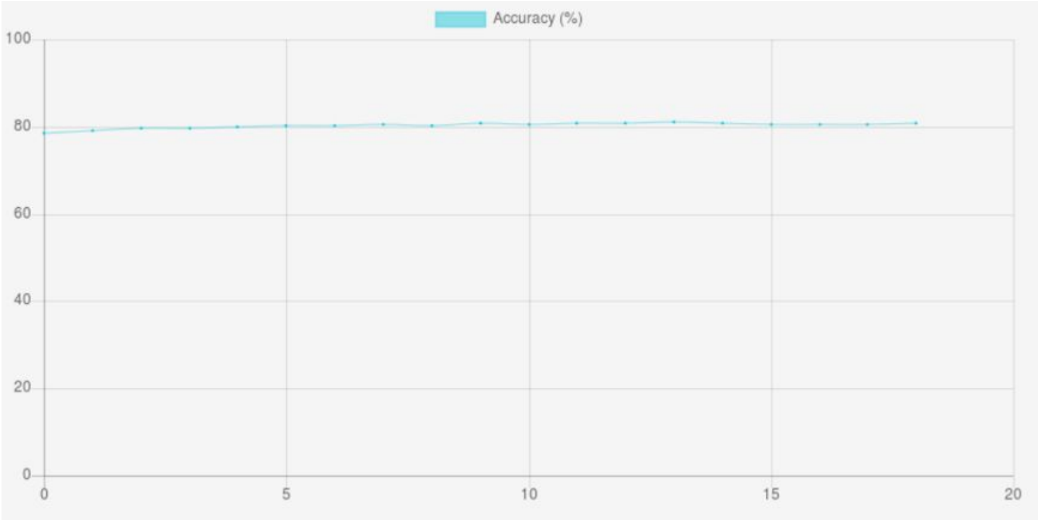
答：

with symbol



validation accuracy	public accuracy	private accuracy
0.8023	0.80857	0.80634
(private + public) / 2 = 0.80846		

without symbol



validation accuracy	public accuracy	private accuracy
0.8111	0.81016	0.81068
(private + public) / 2 = 0.81042		

令人驚訝地，不刪去標點符號時的準確率居然高了0.196%左右，推測是因為驚嘆號、問號、刪節號等等標點符號某種程度上來說也具有表達情緒的功能。但我仍認為適度刪減標點符號(如縮寫撇、引號或是其他奇怪的符號)可以增加預測的準確度。

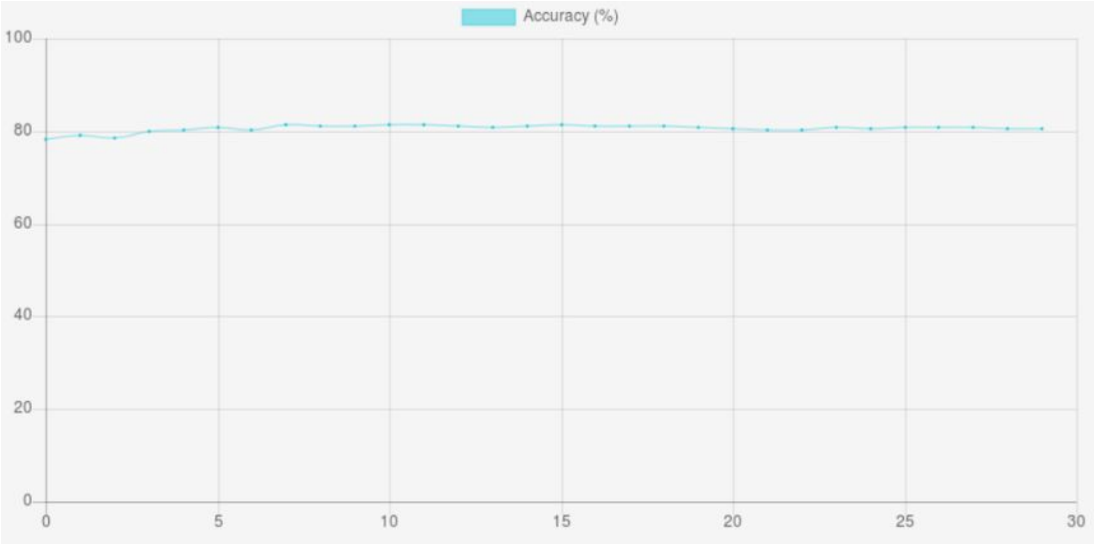
5. (1%) 請描述在你的semi-supervised方法是如何標記label，並比較有無semi-surpervised training對準確率的影響。
(Collaborators: 無)

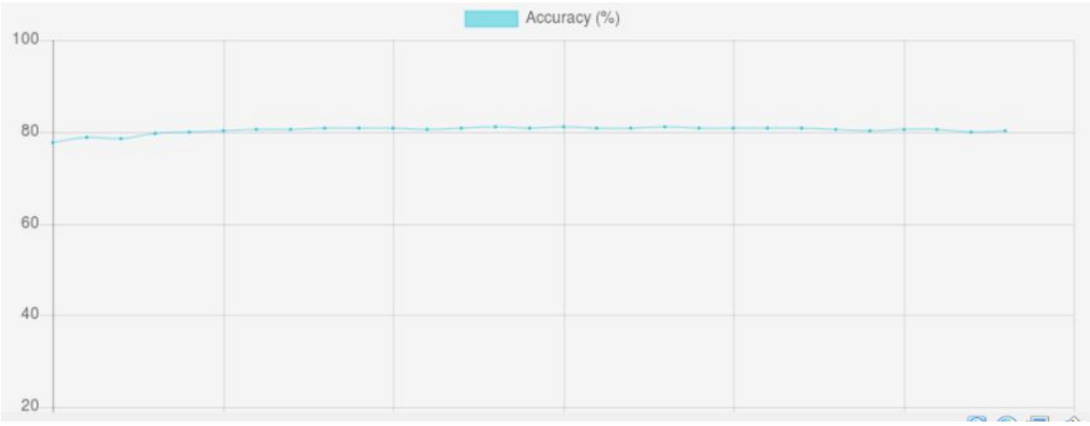
答：我會先做一個pretrained model，接著對nolabel data做預測，取預測值小於0.1及大於0.9的部份標記為0和1，接著再對pretrained model做進一步的training來得到最後版本的model。

由於我是使用glove來做word embedding的，因此再pretrain時就必須先將nolabel data也納入word embedding的corpus中，而在第二階段的training時就不再做: embedding layer的training。

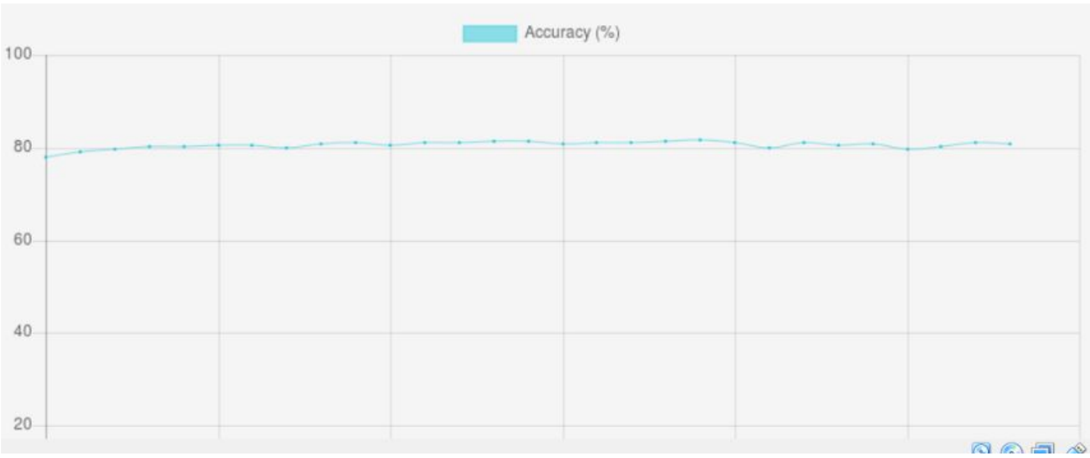
我分別做了兩個semi-supervised及兩個supervised的model來做比較：

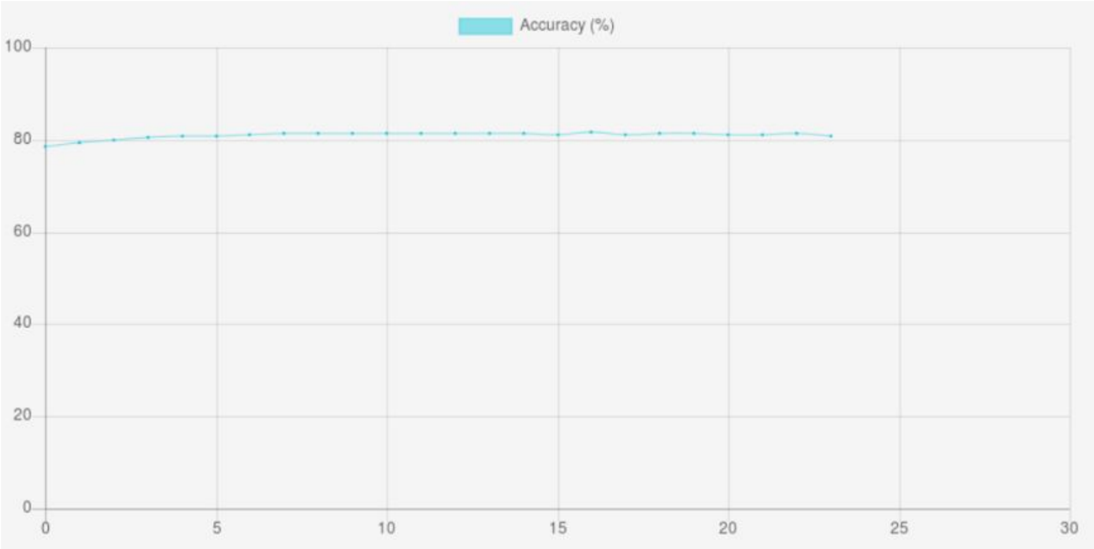
semi-supervised


Accuracy (%)
100
80
60
40
20
0
0
5
10
15
20
25
30


Accuracy (%)
100
80
60
40
20

supervised


Accuracy (%)
100
80
60
40
20



	semi-supervised		supervised	
validation	0.8147	0.8100	0.8156	0.8162
public	0.81165	0.81387	0.81319	0.81543
private	0.81205	0.81484	0.81280	0.81546
(public+private)/2	0.81185	0.81436	0.81300	0.81545
average	0.81310		0.81422	

從表格中可以看到，semi-supervised model並未得到比較好的結果，有可能是pretrained data不夠造成pretrained model不夠精準所致。