機器學習期末報告

Team name : NTU_r05922105_伯恩好可愛

Member : R05922105 陳俞安 / R06922116 賴柏恩 / R06521608 陳德元

Topic : conversations in TV shows

Work division :

組員	分工
陳俞安	seq2seq model, embedding model
賴柏恩	w2v, embedding model
陳德元	寫期末報告、嘗試去掉 stopword

Preprocessing and Feature Engineering:

- **直接處理**

在資料前處理的部分，先將所有的 train 的 data 讀進一個 list，接著使用 jieba 的 dict.txt.big.txt 當作字典來斷詞，然後將斷詞後的文字存成一個檔案。

```
關 馬西 在 船上
祈禱 未來 會 一帆風順
雅信 也 一樣 衷心 冀望
多 年來 努力 認真 苦讀
可以 回家 鄉 服務
其實 雅信 不 知道
在 台灣
早就 有 很多 人 在 等待 他 了
丘雅信 小姐
你 現在 是 台灣 第一個 女醫生
請問 你 現在 有 什麼 感想
對 啊
```

接著斷詞後的檔案設為 word2vec 的語料，並且在 word2vec 的參數部分，將 size 設為 200，workers 設為 6，min_count 設為 10，最後得到一個 vector，維度為 (10604, 200)。

```
10604 200
我 0.072628 -1.347799 0.575301 0.095566 -0.195999 -0.642068 0.740479 0.883721 0.523954 -0.535598 1.020898
-0.890323 0.588435 -0.152363 1.131345 -0.821836 0.000818 0.758039 -0.420794 -0.559153 0.916122 0.360538 0.127898
0.825286 -1.357690 -0.091644 -0.035131 -0.921019 -0.687381 1.259387 -0.106640 -1.460779 0.302979 -0.628452
-0.089607 0.005212 0.332428 -1.581605 -0.377129 0.303077 0.636528 -0.180141 -0.243081 -0.367409 0.801051
-1.029965 -0.101190 -0.484638 0.384970 -0.261740 -0.790270 -0.491136 -1.277653 0.355807 0.072287 -0.593390
-0.034184 0.345597 0.253162 0.667239 -0.460363 0.845036 1.041493 0.709984 0.398015 -0.450982 0.318923 0.267751
-0.169912 0.255529 0.027753 0.900194 0.321050 -0.260597 0.546762 -0.103632 0.949596 -0.061489 0.359374 1.577850
1.065443 -0.229279 -0.855360 -0.288091 -1.014157 0.538516 -0.057155 0.197768 -0.241269 0.135742 -0.055616
-0.399291 -0.107609 -0.261741 -0.377296 -0.325136 -0.243209 0.035762 -0.381000 0.301369 -0.275797 0.523241
0.281884 0.584165 -0.245653 -0.469023 0.350239 0.393346 -0.679181 -0.060180 -0.333157 0.216681 0.167943 0.955605
-0.302320 -0.122617 -0.520392 0.696014 0.313379 0.258449 0.130764 0.253498 0.333186 -0.331362 -0.548809
-0.452727 0.854633 -0.540062 0.601891 0.608953 1.322406 0.940792 -0.637315 0.826114 0.157637 0.784318 0.758337
0.324782 -0.018008 -0.530141 -0.439802 0.118043 -0.364238 0.251393 -0.811842 0.167610 -1.056907 -0.185674
0.092581 -0.494326 -1.484417 0.662319 0.007397 0.287075 -0.538704 0.649572 -0.264696 -1.421648 -0.214873
0.560131 -1.123631 0.915009 1.189667 -0.640724 -1.425268 0.016735 0.844437 -0.036545 -0.225219 0.137392 0.731023
-0.678821 0.145976 0.393750 -0.204145 0.369687 -1.524096 0.103738 -0.191148 -0.192671 -0.182533 -0.093615
0.047408 0.774865 0.070974 0.777376 0.101789 -0.247654 -0.486639 -0.533285 0.341167 0.356576 0.356852 -0.218587
0.469876 -0.125478 0.066011 -0.923654 1.117537 -0.144740
你 0.326104 -0.767131 1.095685 0.089041 0.135161 -0.065756 1.284174 -0.607628 0.265471 0.284653 1.475685
-0.679533 0.174579 -0.117975 1.158374 -0.183534 -0.037934 0.356463 -0.868445 0.398541 -0.354997 0.332392
0.662316 -0.550197 -1.344265 -0.159647 -0.381546 -1.265241 -0.076609 1.454863 -0.192957 -0.510961 -0.026984
-0.399373 1.093368 1.096221 0.664404 -0.046128 -0.651600 1.117330 1.151578 -0.045700 -0.318102 -0.602610
```

接著，會利用 tokenize，將所有的 text 轉成 sequences，然後 padding 到 15，成為一個(757000, 15)維的 vector。

- **去掉 stopword**

在資料前處理的部分，先將所有的 train 的 data 讀進一個 list，接著使用 jieba 的 dict.txt.big.txt 當作字典來斷詞，和上面不同的是會去判斷是否存在 stopwords.txt，如果存在在 stopwords 中的詞就會被去掉，然後再將斷詞後且留下的文字存成一個檔案。

```
關 馬西 船上
祈禱 未來 一帆風順
雅信 衷心 冀望
年來 努力 認真 苦讀
回家 鄉 服務
雅信 知道
台灣
早就 人 等待
丘雅信 小姐
現在 台灣 第一個 女醫生
請問 現在 感想

讀 名校
目前 日本 最 有名 醫科大學
請問 未來 規劃
```

接著斷詞後的檔案設為 word2vec 的語料，並且在 word2vec 的參數部分，將 size 設為 200，workers 設為 6，min_count 設為 10，最後得到一個 vector，維度為 (10107, 200)。

```
10107 200
妳 -0.124628 -0.597709 0.216859 -0.066966 0.221622 0.485266 -0.396728 -0.527779 0.498452 0.577613 0.228037
-0.522646 -0.159667 -1.306582 0.593134 0.779091 0.537914 -0.108663 -1.014304 -0.945851 -0.041138 0.632111
0.795537 0.076089 0.011596 0.496457 -0.419233 0.143283 -0.183964 0.502058 -0.594432 -0.246767 0.255292 0.113238
0.912178 -0.220466 0.778383 -0.148615 -0.410981 0.624446 0.094890 -0.244456 -0.155298 -0.441112 -0.025100
-0.275594 -0.362581 -0.120101 1.024653 -0.008882 -0.152186 0.778449 0.442569 -0.122729 0.416512 -0.539468
-0.552222 0.207444 -0.702191 0.553076 -0.132903 -0.027574 -0.189001 0.402531 0.149433 0.565004 -0.095234
0.994427 0.109720 -0.401390 -0.690475 0.016349 -0.720352 -0.198113 0.160568 -0.488681 0.959844 -0.425108
-0.828152 0.200076 0.623213 0.087917 -0.777769 -0.351133 0.058462 -0.150044 -0.083443 -0.228013 -0.134377
0.354822 0.107499 0.449937 0.273595 -0.504170 -0.508083 0.142950 -0.026240 0.167495 -0.139217 -0.096210
-0.546915 0.805401 -0.239831 0.244715 -0.021343 0.891495 0.039727 -0.216486 -0.429379 1.024805 0.159177 0.252850
0.381380 -0.597580 -0.147407 0.685221 0.637433 -0.348849 -0.370656 -0.293609 0.373667 0.413803 -0.652473
0.578893 0.015002 0.610579 -0.839281 0.044987 -0.031550 0.260232 0.397303 -0.752416 0.237808 -0.400743 0.962163
-0.144254 -0.422161 0.036515 0.157378 0.945791 0.461447 -0.471993 0.356508 0.146367 -0.563661 -0.302844
-0.141892 0.158441 -0.006421 -0.639262 0.432083 -0.253843 0.063004 0.116940 -0.147455 0.058160 0.324088
-0.218928 0.035911 -0.076194 -0.917385 -0.361787 0.002731 -0.549342 -0.171706 0.248238 -0.426274 -0.649543
-0.402325 0.382411 -0.337990 -0.960078 0.122957 -0.643495 -0.405518 -0.299225 0.003169 -0.462884 0.429642
-0.397447 -0.054270 0.724764 -0.311498 0.105256 -0.458283 0.169156 -0.062629 -0.230695 -0.022895 -0.061039
-0.898026 -0.379671 -0.376027 0.069949 -0.582945 0.319161 0.812192 0.063889 0.671696 -0.231854
好 -0.968900 -0.250145 -0.262732 -0.052316 0.274161 -0.066086 -0.189812 -0.598116 0.101688 0.435065 0.774310
0.747583 -0.512335 -0.383058 -0.136212 1.379231 -0.788330 0.407177 -0.516594 -0.057916 0.499333 0.246653
-0.404311 0.468585 0.199784 -0.242891 -0.178889 -0.238206 0.360026 -0.004906 -0.037358 -0.112142 0.535262
-0.114210 0.534239 0.530127 -0.793022 0.471771 -0.378736 0.598097 -0.050706 0.536620 0.005202 -0.045946
```

接著，會利用 tokenize，將所有的 text 轉成 sequences，然後 padding 到 15，成為一個(757000, 15)維的 vector。

Model Description:

我們嘗試了兩種 model，分別是 seq2seq 和 embedding，下面會一一介紹。

Seq2seq：

第一種是直接丟進一句話，然後直接預測下一句，並且去和答案算距離，句立最小的則為解答。

Layer (type)	Output Shape	Param #	Connected to
input_3 (InputLayer)	(None, 15)	0	
input_4 (InputLayer)	(None, 15)	0	
embedding_3 (Embedding)	(None, 15, 200)	9825200	input_3[0][0]
embedding_4 (Embedding)	(None, 15, 200)	9825200	input_4[0][0]
lstm_1 (LSTM)	[(None, 256), (None,	467968	embedding_3[0][0]
lstm_2 (LSTM)	(None, 15, 256)	467968	embedding_4[0][0] lstm_1[0][1] lstm_1[0][2]
dense_2 (Dense)	(None, 15, 1)	257	lstm_2[0][0]

Total params: 20,586,593
Trainable params: 936,193
Non-trainable params: 19,650,400

Embedding：

第二種則是直接從文本出發，如果在 training data 中為上下文關係，則給予 label 為 1，如果相距十分遙遠，則給予 label 為 0，訓練出來的 model 在去預測 testing data，預測出來的值最接近 1 則為解答。

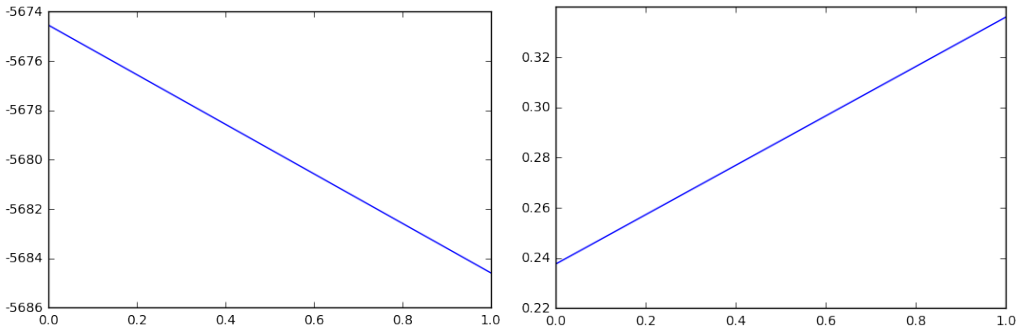
Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 15)	0	
input_2 (InputLayer)	(None, 15)	0	
embedding_1 (Embedding)	(None, 15, 200)	9825200	input_1[0][0]
embedding_2 (Embedding)	(None, 15, 200)	9825200	input_2[0][0]
bidirectional_1 (Bidirectional)	(None, 512)	701952	embedding_1[0][0]
bidirectional_2 (Bidirectional)	(None, 512)	701952	embedding_2[0][0]
dot_1 (Dot)	(None, 1)	0	bidirectional_1[0][0] bidirectional_2[0][0]
dense_1 (Dense)	(None, 1)	2	dot_1[0][0]

Total params: 21,054,306
Trainable params: 1,403,906
Non-trainable params: 19,650,400

None

Experiments:

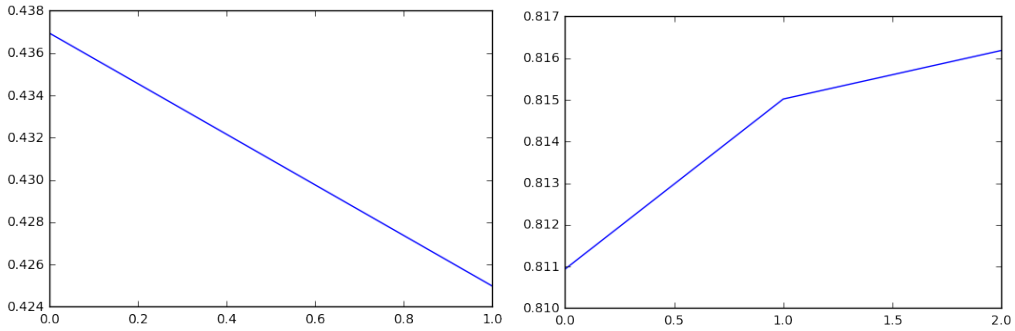
Seq2seq:



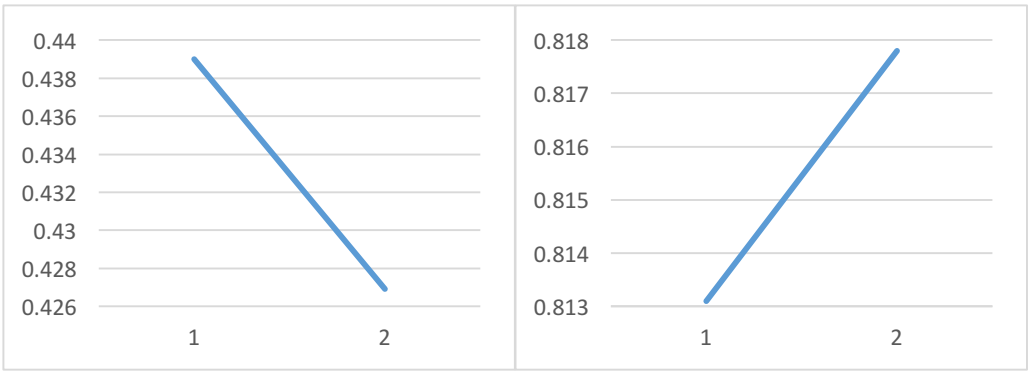
Embedding:

在參數部分，我們設計了兩種參數，第一是將 training data 複製的次數，希望產生多一點 training data 以增加準確率，嘗試的範圍大約是 3-4 次，第二則是將複製的 data 往後推移的距離，亦是希望增加 training data 的多樣性以增加模型的準確率，嘗試的範圍大約為 250 - 400。

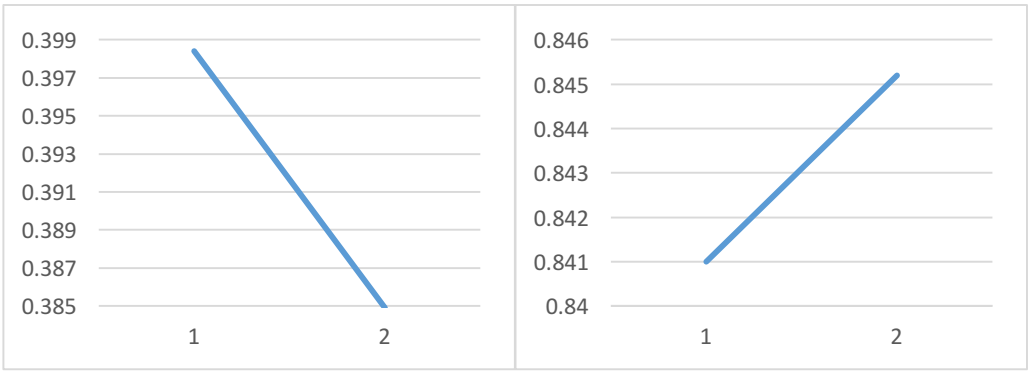
- (4, 250)



- (4, 300)



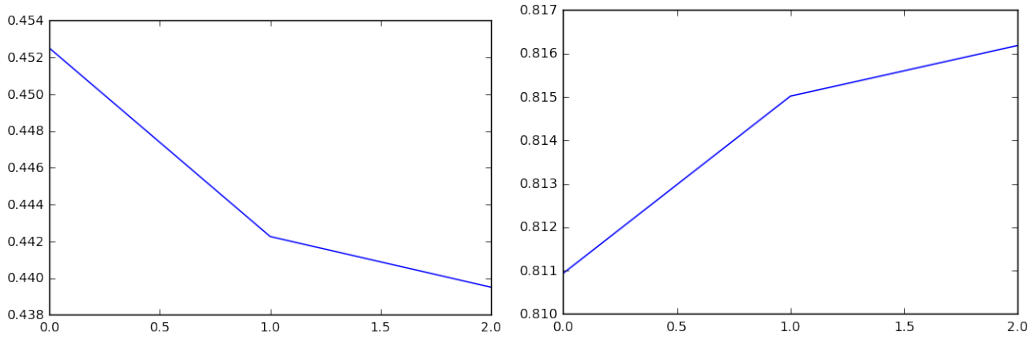
- (5, 400)



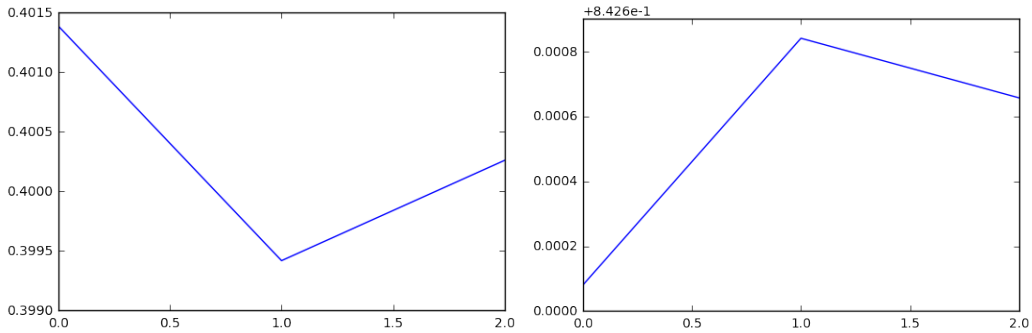
Embedding_nostopword:

在參數部分，和上述相同就不再贅述，複製 training data 的次數範圍大約是 4-5 次，將複製的 data 往後推移的距離嘗試的範圍大約為 250 - 350。

- (4, 250)



- (5, 350)



Discussion:

在嘗試了上述的實驗後，seq2seq 的模型在 kaggle 上面的表現比想像中的低上太多，大概只有 0.08 的正確率，讓我們不得不放棄轉向第二種 model，所幸的是第二種 model 的準確率可到 0.44，終於過了 kaggle 上面的 simple baseline，透過 emsemble3 個 model 的方式，也終於成功突破 strong baseline。

而本來想說實驗將所有 stopword 去掉後，可以得到更佳的結果，讓 kaggle 的排名能夠更進步一些，但不幸的是，將所有的 stopword 去掉後，成果卻沒有想像中的好，在 kaggle 上都只有 0.37 – 0.39 的成績。