Conversation in TV Shows

- **Team name & Members:**

NTU_r06521705_GLU

學號	姓名
R06521705	陳思愷
R05521704	王冠人
R05521112	蕭承熙

- **Introduction & Motivation:**

會選擇這個主題的原因在於組員都對於自然語言處理有興趣，希望不只有HW5，也能藉由 final project 的部分對於這領域有著更加深入的理解，並嘗試在未來研究領域之中試著與之結合。

- **Data Preprocessing/Feature Engineering :**

在資料的預處理之中，我們先將五份 data 串在一起做為訓練用的資料 training data，由於文本剛開始到手都是”句子”，因次我們先用中文斷句套件—結巴(jieba)，來幫文本進行斷句，但由於 jieba 中的預設辭典為簡體中文，如果直接使用效果會不佳，故我們先把辭典轉換成繁體中文專用的

dict.txt.big，有利後續的成果表現，另外，在 jieba 斷句的模式中分成:

(1)全模式 (2)精準模式 (3)搜尋模式，在這次的 project 中 jieba 的操作我們都是使用預設的精準模式。

接下來是對於”stopword”的處理，中文中有許多很常出現但其實對於語意並沒有太多理解上幫助的字詞，例如: 哎呀、了、喔...等等，其實不會對語意也太大的影響，但對於訓練過程中如果存在會在預測中造成不精準的情況，所以希望可以在訓練之前把文本中的這些字詞拿掉。在設法取得 stopword.txt (集合大多數斷詞的一個文檔)後，對文本中的 stopword 進行去除，如此可以讓 word2vec model 在訓練之後詞與詞之間的關聯性更佳的精確。

- **Model Description (At least two different models) :**

在此我們是利用 gensim 中的 word2vec 去訓練詞向量的模型，並嘗試兩種不同的參數設計設計出 model，並比較優劣或做 ensemble。在此我們特別對於 word2vec 的模型參數做簡單的介紹以及以什麼評斷標準來決定詞向量之間的關聯性高或低。

1.Word2vec:

對於”句子”或”句子”之間，若想藉由電腦來觀察彼此之間的關聯性，勢必需要將其作轉換，依此便可藉於”詞向量”的概念去運作，將詞語轉換成高維空間中的一個向量。

Word2vec 這個套件便是用於此，介於設定辭典、詞向量維度等等參數去訓練 model，爾後便可利用訓練出來的 model 將 input 的詞句轉換成詞向量。

(1) size :設定訓練出的 word vector(詞向量) 的維度。

(2) window : 控制在訓練的時候一次能往前後看幾個字。

(3) min_count : 詞出現的數字要高於這個參數才會視為訓練對象。

(4) iteration : 訓練時更新參數的週數。

(5) sg : 控制要使用 cbow 或是 skip-gram model，我們使用的是 skip-gram。

(6) alpha : 也就是 learning rate，我們使用 default (0.025)。

2.評斷詞句的相似度:

我們在決定選項應該要選擇哪一個，是依照選項和問句之間的” 相似度” 來決斷，至於該如何判斷這種相似度，我們認為觀察詞向量之間的距離是可行的:

(1) Euclidean distance:

利用 jieba 將 testing data 的問句和選項分詞後，丟入訓練好的 w2v model 之中轉化成詞向量，在利用選項和問句的向量去判斷遠近關係，這邊我們利用到第一種計算距離的方式是歐氏幾何距離(Euclidean distance)。

也就是將其當作空間中的一個點，直接計算點與點之間的距離，計算公式

如下:

$$d(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2} = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

但課程中有說過，在高維空間之中，歐氏幾何距離的計算會失真，可能關連到了高維空間中所會出現的維度災難，導致大多數在高維空間中要計算距離時並不會以歐氏幾何距離來計算。

(2) Cosine similarity:

我們使用的第二種方式是去計算題目和選項之間的餘弦相似度(Cosine similarity)，作法如下:

$$\text{Similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}}$$

在高維空間之中，這是比較準去且可信的計算方式。

- **Experiment and Discussion :**

決定好了在高維空間中評斷相似度的方法（Cosine similarity）後，我們開始對於 Word2vec 的各項參數進行試驗，嘗試找出表現較佳的參數數值組合:

(1) Model1:

Size: 128

Window: 10

Iteration: 50

Min_count: 1

(2) Model2:

Size : 64

Window: 20

Iteration: 100

Min_count: 1

另外，我們也嘗試利用 NN model 來訓練:

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 128)               16512
_________________________________________________________________
dense_2 (Dense)              (None, 256)               33024
_________________________________________________________________
dropout_1 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_3 (Dense)              (None, 512)               131584
_________________________________________________________________
dense_4 (Dense)              (None, 1024)              525312
_________________________________________________________________
dropout_2 (Dropout)          (None, 1024)              0
_________________________________________________________________
dense_5 (Dense)              (None, 512)               524800
_________________________________________________________________
dense_6 (Dense)              (None, 256)               131328
_________________________________________________________________
dropout_3 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_7 (Dense)              (None, 128)               32896
=================================================================
Total params: 1,395,456
Trainable params: 1,395,456
Non-trainable params: 0
_________________________________________________________________
```

- **Conclusion :**

	public	private
W2v model1	0.51620	0.53043
W2v model2	0.51620	0.51027
NN model	0.50079	0.52727

上述是個別於 Kaggle 上的表現分數

而後我們利用 voting 去做結果的 ensemble，後在 Kaggle 的 private 也取得了較為進步的成績 0.5415，public 分數為 0.54031。

- **Reference :**

https://radimrehurek.com/gensim/models/word2vec.html

https://github.com/castanan/w2v

https://taylorwhitten.github.io/blog/word2vec

http://www.52nlp.cn/tag/word2vec

https://www.zhihu.com/question/29978268

https://www.jianshu.com/p/0702495e21de