

# Artificial-Intelligence

## homework2

0816023 張紳濡

## 1 Introduction

使用 Bi-gram 與 DistilBert 偵測正面與負面評論

## 2 Data

- train: 20000 positive and 20000 negative review.
- test: 5000 positive and 5000 negative review.

## 3 What I do

### 3.1 Preprocessing

- function 1: remove chinese words:  
e.g.:  
'War movie' is a Hollywood genre that has been done and redone so many times that cliché dialogue  
-> ' War movie ' Hollywood genre done redone many times cliché dialogue
- function 2: remove Punctuation:  
e.g.:  
.<br /><br />My intention to see it was certainly JJL being one of my favourite actresses.  
->My intention see certainly JJL one favourite actresses
- function 3: remove number:  
e.g.:  
I saw this movie when I was about 12 when it came out.  
->saw movie came

All of example have removed stop words

### 3.2 Different numbers of feature-num

這邊的 feature-num 我使用 500 與 250，結果如下表：

	perplexity	F1	Precision	Recall	Loss
250bigram1	276.6	0.67	0.67	0.67	
250bigram2	4865.4	0.67	0.69	0.67	
250 bert1		0.93	0.93	0.93	0.22
250 bert2		0.9	0.9	0.9	0.28
500bigram1	276.6	0.71	0.71	0.71	
500bigram2	4865.4	0.7	0.71	0.7	
500 bert1		0.93	0.93	0.93	0.23
500 bert2		0.9	0.9	0.9	0.28

- perplexity: 在 perplexity 的部分，可以明顯發現在經過 preprocessing 後 perplexity 明顯上升，我想是因為去除了 stop word 與標點符號所以相似的 bigram 變少了 (perplexity 越小越好) 這邊為了處理 0 的問題我把等於 0 的改成 0.00001(smoothing)
- F1 score: 這邊在 F1 score 的部分，在 bi gram 比較能看得出差異對於 feautre-num 為 500 的 model F1 score 略高一些代表對於 bi gram 取比較多的 feature 會比較好，而 bert 方面我覺得可能 250 就足夠他判斷了。
- precision,recall: 可以發現 precesion 與 recall 還蠻平均的，代表 model 不會太過謹慎也不會太過寬鬆。

### 3.3 Bi-gram VS DistilBert

- bi-gram cannot outperform DistilBert: 我認為是因為 bert 會考慮整個語句而 bi-gram 只考慮部分語句，而且 bert 有比較多層
- Can bi-gram consider long-term dependencies: 我覺得可能可以把 perplexity 加入 model 的訓練內可能可以，因為 perplexity 有整句的資訊。
- Would the preprocessing methods improve the performance of the bi-gram model: 我覺得這次訓練下來的結果沒有增進結果，從上面的表格可以發現並沒有增進多少，我考慮過可能是我 preprocessing 沒做好因此跑過只移除 stop word 的版本如下表：

	perplexity	F1	Precision	Recall	Loss
250bigram1	276.6	0.67	0.67	0.67	
250bigram2	2330.3	0.65	0.67	0.67	
250 bert1		0.93	0.93	0.93	0.22
250 bert2		0.93	0.93	0.93	0.26
500bigram1	276.6	0.71	0.71	0.71	
500bigram2	2330.3	0.68	0.7	0.69	
500 bert1		0.93	0.93	0.93	0.23
500 bert2		0.92	0.92	0.92	0.26

可以發現也沒有增加多少。

- UNK: 我認為 perplexity 會減少，因為原本不同 bigram 但機率很低的 bigram 會整合在一起因而減少 perplexity。

## 4 Problem I meet

### 4.1 run too long

在 part 1 的部分若要做到 example 的結果需要跑很久，因為要生成每個 word 對映下一個 word 的機率表，而我在做其他部分的時候也沒有用到這個部分，因此我在跑的時候把這邊註解掉了 (model.py line 60-65) 建議助教若要跑把這邊也註解掉。

### 4.2 preprocessing

我認為 preprocessing 應該要讓結果變好，但實際上的結果並沒有，因此我認為可能有更好的 preprocessing function 可以實作使結果變好。

## 5 Code

All code can be find in

<https://github.com/Sakuya0229/Artificial-Intelligence-HW2>