人工智慧概論 HW2 report

109550025謝翔丞

1. Three Give an input and output example after applying each preprocessing method in the report. For example, the input sentence is “Here is the dog.” and the output after removing stopwords is “Here dog.” .

上方 : input

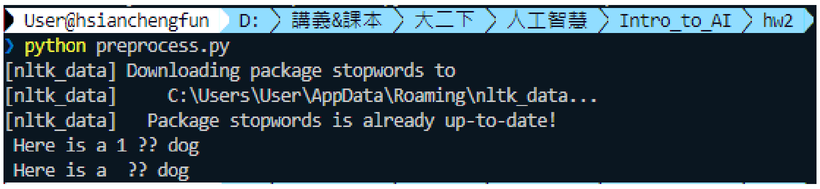
下方 : output

這次的preprocessing我大概使用以下幾種作法

甲、 Remove_number

i. 去除文中所有的數字

ii.

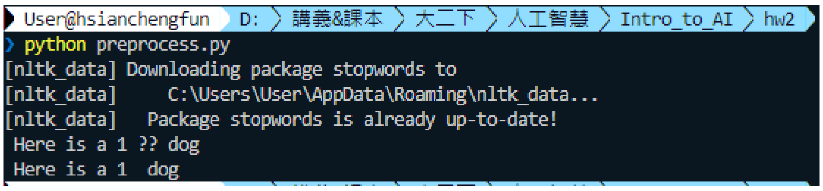


```
User@hsianchengfun  D: 〉講義&課本 〉大二下 〉人工智慧 〉Intro_to_AI 〉hw2
〉 python preprocess.py
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\User\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
 Here is a 1 ?? dog
 Here is a  ?? dog
```

乙、 Remove_punctuation

i. 去除文中所有標點符號

ii.

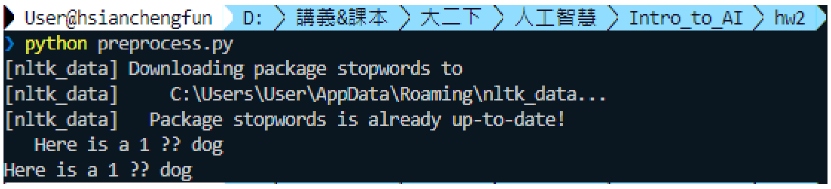


```
User@hsianchengfun  D: 〉講義&課本 〉大二下 〉人工智慧 〉Intro_to_AI 〉hw2
〉 python preprocess.py
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\User\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
 Here is a 1 ?? dog
 Here is a 1  dog
```

丙、 Remove_whitespace

i. 去除文中多餘(不必要的空白)

ii.

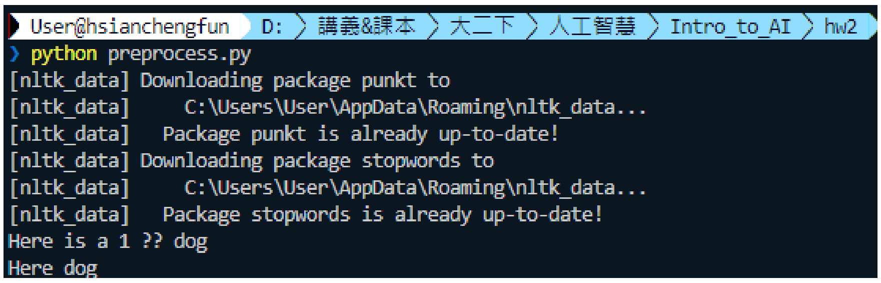


```
User@hsianchengfun  D: 〉講義&課本 〉大二下 〉人工智慧 〉Intro_to_AI 〉hw2
〉 python preprocess.py
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\User\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
   Here is a 1 ?? dog
Here is a 1 ?? dog
```

丁、 Remove_stem

i. 提取文中詞幹

ii.



```
User@hsianchengfun  D:  講義&課本  大二下  人工智慧  Intro_to_AI  hw2
❯ python preprocess.py
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\User\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\User\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
Here is a 1 ?? dog
Here dog
```

2. The performance of using different numbers of feature_num in sentiment classification.

甲、 這次我使用 n=500 及 n=200 來做比較，由下表可以發現無論Bi-gram還是DistilBert，隨著n越大都會使得F1_score越靠近1，也就是越精準的意思。

Bi-Gram

	Perplexity	F1 score	Precision	Recall
N = 500 without preprocessing	94.42507940407172	0.7062	0.7092	0.7070
N = 500 with preprocessing	244.95213433506714	0.7122	0.7233	0.7149
N = 200 without preprocessing	94.42507940407172	0.6713	0.6716	0.6714
N = 200 with preprocessing	244.95213433506714	0.6705	0.6893	0.6762

DistilBert

	F1 score	Precision	Recall	Loss
N = 500 without preprocessing	0.9325	0.9330	0.9325	0.2282
N = 500 with preprocessing	0.8773	0.8824	0.8777	0.3289
N = 200 without preprocessing	0.9336	0.9337	0.9336	0.2265
N = 200 with preprocessing	0.8763	0.8828	0.8768	0.3299

3. Discuss what you observed with perplexity, F1-score, precision, and recall of different methods in the report.

   甲、　這次我採取以下四種比較方式：

   i. All removed
   ii. Not remove stem
   iii. Not remove punctuation and not remove stem
   iv. Not remove punctuation and not remove stem and not remove numbers

Remove_stopwords	Remove_stem	Remove_punctuation	Remove_numbers	Perplexity	F1_score
V				135.117	0.6298
V			V	135.453	0.6300
V		V	V	181.132	0.6914
V	V	V	V	244.952	0.7110

我們可以發現，隨著越多的preprocessing method被使用，F1_score也越來越靠近1，由此可知，最好的”配方”應該是這四種remove方法都實用上去。並且也能發現，perplexity與F1_socre並非正或負相關，因此不能指稱f1的降低必伴隨perplexity的升高。

4. Discuss the difference between the bi-gram model and DistilBert (a lightening variant of BERT). You might need to run extra experiments if you give some hypotheses. Some required discussions are given below.

   a. What are the reasons you think bi-gram cannot outperform DistilBert?

      i. 我認為是因為bigram只考慮相鄰兩個詞彙之間的關係，然而BERT能去偵測劇中有哪些詞彙是必要、和答案相關的，也能辨識前後語意，因此BERT在計算量和精確度上都比bigram高上許多。

b. Can bi-gram consider long-term dependencies? Why or why not? Not

這點其實很好理解，假設有兩個句子

1. He is good ,she is bad
2. he is bad, she is good

以上在bigram看來都是相同的句子，但是兩者其實在語意上是相反的，這就是bigram無法判斷出來的部分。

c. Would the preprocessing methods improve the performance of the bi-gram model? Yes

透過適當的預處理，刪除不必要的文字，是可以有效幫助模型增加他的精確率的。

d. If you convert all words that appeared less than 10 times as [UNK] (a special symbol for out-of-vocabulary words), would it in general increase or decrease the perplexity on the previously unseen data compared to an approach that converts only a fraction of the words that appeared just once as [UNK]?

從原先將只出現一次的詞彙轉為UNK改成小於10次都轉為UNK，如此一來模型在運行時會降低他遇到不認識的單字，並且提高probability，換句話說就會降低entropy，而我們知道perplexity是2^(entropy)，所以perplexity也會隨之降低。

5. Describe problems you meet and how you solve them.

   a. 基本上，這次做這份作業是我上大學以來最灰心的一次作業，光是看懂這次作業的主軸跟目的就已經花了我一個禮拜，好不容易搞懂要做甚麼之後，打開code發現我完全看不懂在幹嘛。像是preprocessing，我原本一直很疑惑助教都已經提供remove stopwords了我們是還有甚麼好做的，後來才知道標點符號、數字等也都是可以移除的，這些都還好處理，到了model.py的才是我最崩潰的地方。

   我從上次作業就不太懂self這個東西的意義跟身分，再加上這次他又有很多奇奇怪怪的屬性讓我越來越頭暈。重點是，光從各個function的input 名字我看不出來他們的意義，所以花很多時間在搞清楚寫這個函式的用意跟作用在哪裡，問題在於我又不知道要輸出、回傳甚麼東西，所以一直卡關、像隻無頭蒼蠅每天都在問同學一樣的問題，問到自己都覺得很愧疚，幫不了別人就算了還一直打擾別人;另外就是，有些部分我跟同學的寫法一樣，但是卻會遇到我有報錯他的不會的狀況，不斷比較也找不到我們的方法有甚麼不同，花了很多時間應付這種無奈的問題最後也沒找出bug真的很讓人心力交瘁。

   這門課做到第二份作業了，似乎越來越感受到自己在這門課堂實作上的無力，儘管助教真的很熱心的頻繁在討論區幫忙解

答疑問，但是連問題都提不出來的人，別人想救也難，我就是這種狀態。但還是很感謝助教在上面幫忙解惑，也很感謝許多讓我一直問問題的同學，要不是大家的幫忙，我是絕對沒辦法完成這份作業，謝謝你們。