report of AI HW2 109550027 紀竺均

a. Give an input and output example after applying each preprocessing method.

i. remove_stopwords( )

input: “Here is the dog.”

output: “Here dog.”

ii. remove_punctuation( )

input: "Here! Here? Where is the dog ...
"

output: “Here Here Where is the dog”

iii. remove_digits( )

input: “55Here is the dog.666”

output: “Here is the dog.”

iv. stemSentence( )

input: “Cooks cooking a cooked cook”

output: “cook cook a cook cook”

b. result, performance and a few conclusion

feature_num = 500

```
Perplexity of ngram: 92.10366930392443
end train sentiment
F1 score: 0.7057, Precision: 0.7088, Recall: 0.7065
```

```
Perplexity of ngram: 230.1559625669742
end train sentiment
F1 score: 0.6894, Precision: 0.6975, Recall: 0.6917
```

```
Epoch: 0, F1 score: 0.9298, Precision: 0.93, Recall: 0.9298, Loss: 0.2335
```

```
Epoch: 0, F1 score: 0.8898, Precision: 0.8926, Recall: 0.89, Loss: 0.3162
```

feature_num = 400

```
Perplexity of ngram: 92.10366930392443
end train sentiment
F1 score: 0.6891, Precision: 0.6932, Recall: 0.6903
```

```
Perplexity of ngram: 230.1559625669742
end train sentiment
F1 score: 0.6798, Precision: 0.6907, Recall: 0.683
```

```
Epoch: 0, F1 score: 0.9318, Precision: 0.9319, Recall: 0.9318, Loss: 0.2304
```

```
Epoch: 0, F1 score: 0.8907, Precision: 0.8932, Recall: 0.8909, Loss: 0.3165
```

N gram/

Feature_num	Preprocessed	perplexity	F1-score	precision	recall
400	N	92.1036	0.7057	0.7088	0.7065
400	Y	230.1559	0.6894	0.6975	0.6917
500	N	92.1036	0.6891	0.6932	0.6903
500	Y	230.1559	0.6798	0.6907	0.683

Bert/

Feature_num	Preprocessed	Loss	F1-score	precision	recall
400	N	0.2335	0.9298	0.93	0.9298
400	Y	0.3162	0.8898	0.8926	0.89
500	N	0.2304	0.9318	0.9319	0.9318
500	Y	0.3165	0.8907	0.8932	0.8909

Given the result above,

1. Bert model outperform ngram model in all conditions.
2. For different feature numbers, in Ngram model, featureNum=400 outperform featureNum=500; In Bert model, feature_num=500 outperform feature_num=400.
3. For being preprocessed or not, in Ngram model, NO preprocess has higher f1-score and higher perplexity; In Bert model, NO preprocess has higher f1-score and lower loss.

c. Discuss the difference between the bi-gram model and DistilBert

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

A: BERT stands for Bidierctional Encoder Representations from Transformers, the designer use Transfomer Encoder + a great amount of data + two types of training: masked-LM and Next-Sentence-Prediction to train the model. The model comprehend NLP in their way. On the other hand, N gram (bigram here) only calculate the conditional possibilities between two connected words, the model don't have pre-knowledge in any form.

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

A: Continued from (a), bi-gram can't consider long-term dependencies because it only considers the relationship between two connected word, besides, it doesn't have any pre-knowledge. But English is very complex and the order of words is not the most important thing to consider while judging its sentiment.

c. Would the preprocessing methods improve the performance of the bi-gram model? Why or why not?

A: From my result, No preprocessed model get the higher f1-score, which means preprocessing doesn't help improve the performance. My guess is that preprocess remove some words and thus change the origin meaning. For example: I test the text "This is not a great film." and after preprocessing, the output is "great film", which actually talk black into white!

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]? Why or why not?

A: If I convert words appeared less than 10 times as [UNK], the number of lower frequency words will decrease (decrease more than convert words appeared 1 time) and the conditional probabilities will be bigger, entropy being smaller and so does perplexity. So, yes, it will decrease the perplexity more than only convert words appeared 1 time.

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

原本用 git bash 跑 run.sh，跑了十五個小時只跑出不到一半，而且電腦快要燒壞了，後來我研究了一下，用 google colab 跑，才明白工欲善其事，必先利其器，用 GPU 跑真的快好多…。

以下是我怎麼用 colab 跑.sh file 的：

```
[19] from google.colab import drive
     drive.mount('/content/gdrive')
     !cp -r /content/gdrive/MyDrive/HW2/* .

     Drive already mounted at /content/gdrive; to attempt to forcibly remount, call drive.mount("/content/gdrive", force_remount=True).

     !chmod 755 /content/gdrive/MyDrive/HW2/run.sh
     !/content/gdrive/MyDrive/HW2/run.sh
```

結論是真的不要害怕去嘗試不熟悉的東西，否則電腦可能會比腦袋還要先燒壞。