句子情緒辨識

Group2: 110034552 陳庚鼎

目錄

01

背景介紹

背景介紹

現代人長時間處於高壓力的生活節奏以及工作中，會一直處於緊張狀態，難免會出現不同程度的心理問題。當壓力積累的越來越多時，會形成定時炸彈，周遭的人們來不及察覺並提供協助，隨時都有可能會被引爆。

希望利用語言情緒辨識系統，蒐集人們平常於網路及社群平台上發布的內容，藉由分析這些內容來了解情緒狀態

5W1H

What
現代人生活壓力大
情緒不穩定的發生情況上升

When
當表現異常時

Why
觀察人們日常中的情緒
起伏 可以及時提供協助

Who
每個人都可以

Where
社群網路 留言板 聊天室

How
利用BERT模型去進行
預測

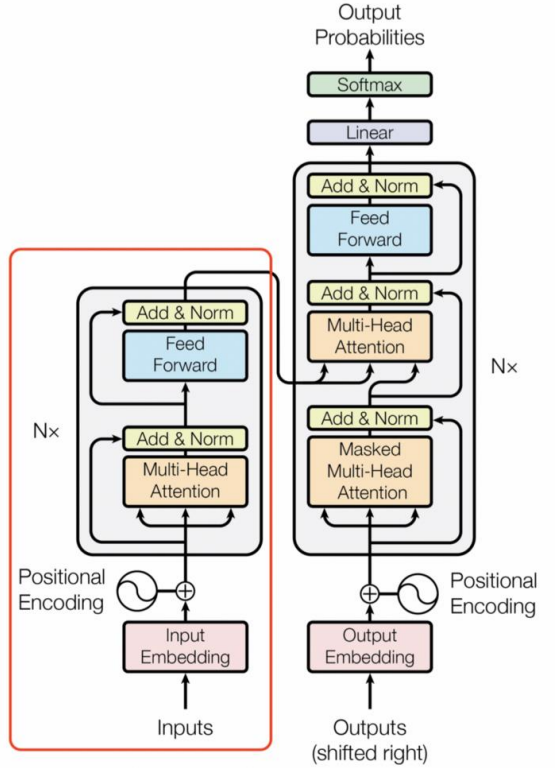
02

研究方法

BERT 介紹

- Bidirectional Encoder Representations from Transformers
- 雙向編碼器表示技術
- 2018年Google發布的預訓練模型
  - 克漏字填空
  - 下個句子預測
- 源自2017年提出的transformer架構
  - 平行計算
  - 防止梯度消失

Transformer 架構



- 六個編碼器和六個解碼器組成，可將輸入字元轉換為不同維度上的向量
- 此架構也廣泛應用在，計算機視覺、語音、生物、化學等領域

遷移學習

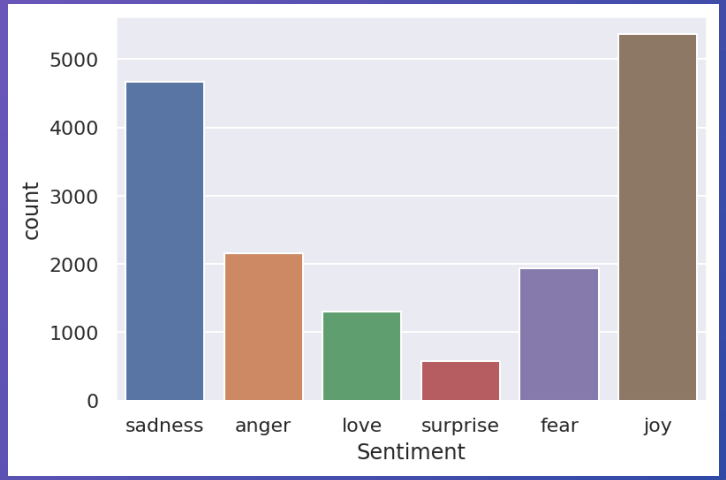
- 將已經在一個特定資料集上訓練好的模型拿來用於另一個資料集的訓練
- 當目標域數據量較小或者數據的標籤很難獲取，可以通過數據量充足或者容易獲取標簽且和該任務相似的任務(源域)來遷移學習
- 自然語言處理就是一個非常缺乏標註資料，而公開資料豐富的領域

03 模型訓練與績效

資料介紹

資料來源

使用Kaggle網站中， Emotions dataset for NLP的資料集，總共有6個情緒標籤，分別為joy、love、surprise、anger、sadness、fear



原始資料集

包含train、test和val三個資料集
訓練資料集：16000
測試資料集：2000
驗證資料集：2000

資料前處理

數據增強

利用回譯的方式，將英文轉換成中文及法文、再轉換回來，擴增資料集

主要擴增對象為surprise. love兩個資料集

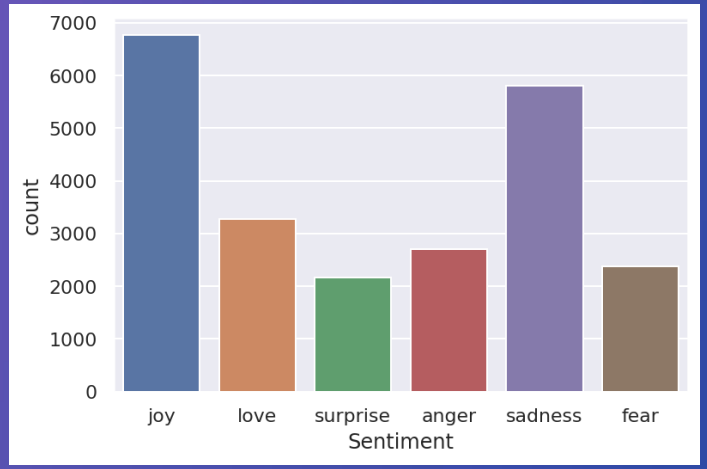
Ex.

i am now nearly finished the week detox and i feel amazing

I have almost done a week, I feel great.

A2067 | fx | =GOOGLETRANSLATE(GOOGLETRANSLATE(A629,"en","fr"),"fr","en")

	A	B	C	D	E	F	G
2060	I've blogged and I feel s	surprise					
2061	I can not even start expr	surprise					
2062	I did not think it was pc	surprise					
2063	I wanted to skate quickly	surprise					
2064	Loading...	surprise					
2065	I feel shocked you ss far	surprise					
2066	I feel so curious with wh	surprise					
2067	I also miss the old curid	surprise					



資料前處理 (text_hammer)

資料檢查

檢查每個句子是否都有標籤

將文字轉成小寫

A => a

將縮寫還原

You're => You are

刪除不相關句子

Email 網址

刪除特殊文字

% + $ #

資料前處理

將label =>數字 => one hot型態

Anger : 0
fear : 1
joy: 2
love : 3
sadness : 4
surprise : 5

資料分割

將訓練、測試和驗證集混和再以7:3重新分割訓練與測試集

模型建立

載入**BERT**預訓練模型

```
from transformers import BertTokenizer, TFBertModel, BertConfig,TFDistilBertModel,DistilBertTokenizer,DistilBertConfig
dbert_model = TFDistilBertModel.from_pretrained('distilbert-base-uncased')
```

加入全局平均池化層和**dropout**層

最後使用**sigmoid**函數進行分類

```
embeddings = bert(input_ids,attention_mask = input_mask)[0]
out = tf.keras.layers.GlobalMaxPool1D()(embeddings)
out = Dense(128, activation='relu')(out)
out = tf.keras.layers.Dropout(0.1)(out)
out = Dense(32,activation = 'relu')(out)

y = Dense(6,activation = 'sigmoid')(out)
```

使用 **balanced_accuracy** 評估模型

```
metric = CategoricalAccuracy('balanced_accuracy'),
# Compile the model
model.compile(
        optimizer = optimizer,
        loss = loss,
        metrics = metric)
```

參數優化

利用了實驗設計中的田口方法，減少調整參數的總次數，並獲得相同的結果。

選擇了上述所提到的四項參數作為四個因子，並使用三水準，應用L9直交表來幫助參數優化。

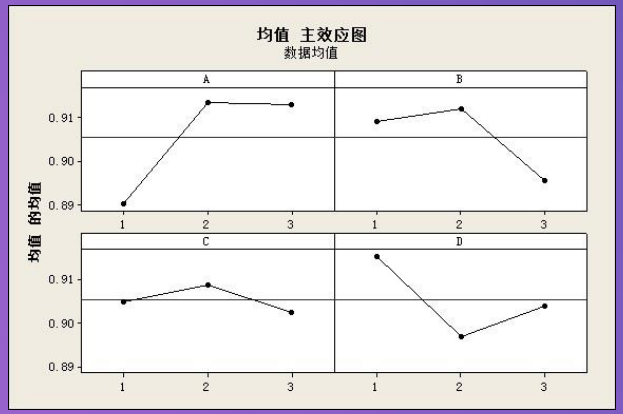
因子	說明	水準1	水準2	水準3
A	epoch	1	2	3
B	dropout	0.1	0.2	0.3
C	Batch Size	8	16	32
D	learning rate	5.00E-05	3.00E-05	2.00E-05

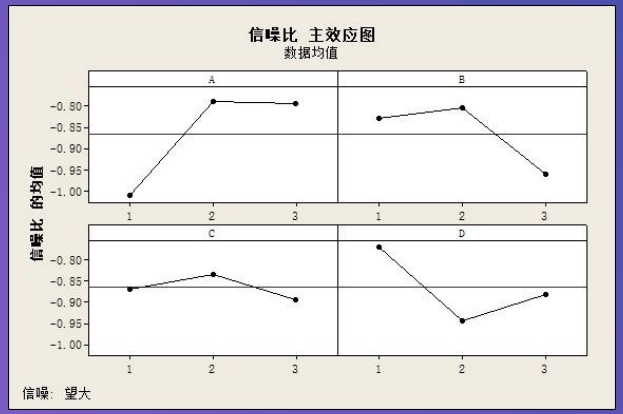
參數優化

L9 實驗參數組合

實驗	epoch	dropout	Batch Size	learning rate	Accuracy
1	1	0.1	8	5.00E-05	0.9032
2	1	0.2	16	3.00E-05	0.8915
3	1	0.3	32	2.00E-05	0.8759
4	2	0.1	16	2.00E-05	0.9184
5	2	0.2	32	5.00E-05	0.9266
6	2	0.3	8	3.00E-05	0.8944
7	3	0.1	32	3.00E-05	0.905
8	3	0.2	8	2.00E-05	0.9171
9	3	0.3	16	5.00E-05	0.9159

參數優化





水平	A	B	C	D
1	-1.0109	-0.8302	-0.8685	-0.7698
2	-0.7903	-0.8038	-0.8333	-0.9447
3	-0.7939	-0.9611	-0.8934	-0.8806
Delta	0.2206	0.1573	0.0600	0.1748
排秩	1	3	4	2

均值响应表

水平	A	B	C	D
1	0.8902	0.9089	0.9049	0.9152
2	0.9131	0.9117	0.9086	0.8970
3	0.9127	0.8954	0.9025	0.9038
Delta	0.0229	0.0163	0.0061	0.0183
排秩	1	3	4	2

參數優化(最佳)

實驗	epoch	dropout	Batch Size	learning rate	Accuracy
10	2	0.2	16	5.00E-05	0.9252

	precision	recall	f1-score	support
0	0.92	0.96	0.94	813
1	0.91	0.86	0.89	712
2	0.94	0.94	0.94	2028
3	0.87	0.88	0.87	985
4	0.97	0.96	0.97	1739
5	0.86	0.90	0.88	647
accuracy			0.93	6924
macro avg	0.91	0.91	0.91	6924
weighted avg	0.93	0.93	0.93	6924

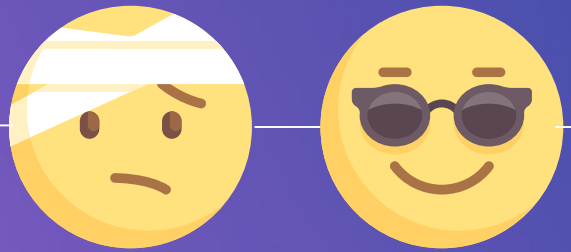
參數優化(加碼)

實驗	epoch	dropout	Batch Siz	learning rate	Accuracy
11	2	0	16	5.00E-05	0.9284

	precision	recall	f1-score	support
0	0.95	0.93	0.94	813
1	0.89	0.91	0.90	712
2	0.95	0.93	0.94	2028
3	0.87	0.89	0.88	985
4	0.96	0.97	0.97	1739
5	0.87	0.88	0.88	647
accuracy			0.93	6924
macro avg	0.92	0.92	0.92	6924
weighted avg	0.93	0.93	0.93	6924

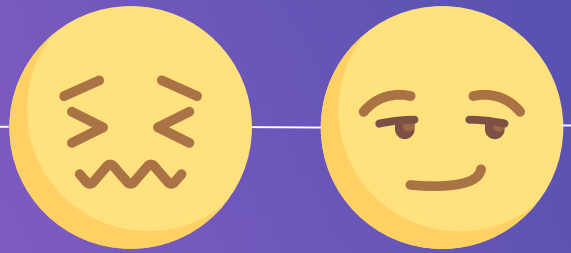
結論

可以再思考還有甚麼方法，可以幫助模型增進正確率，如資料集蒐集或是改善模型



資料集的數量差異對於準確度的影響並不高

可能會出現兩種情緒都符合的句子，這時就會造成判斷有些偏誤，可以考慮其他更有分別度的情緒標籤



未來可以實際收集社群平台上的句子，讓模型更貼近現實

自動化爬蟲，蒐集用戶留言並分析，自動偵測特殊用戶

Thank You