

國立清華大學

智慧化企業整合

Intelligent Integration of Enterprise

利用 BERT 進行句子情緒辨識

指導教授：邱銘傳 教授

學生：110034552 陳庚鼎

## 摘要

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。

關鍵字：情緒辨識、NLP、BERT、社群媒體

## 目錄

1. 背景.....	4
1.1 背景說明.....	4
1.2 問題描述 5W1H.....	4
2. 文獻回顧.....	5
2.1 BERT(Bidirectional Encoder Representations from Transformers).....	5
2.2 Transformer.....	5
3. 研究方法.....	6
3.1 資料介紹.....	6
3.2 資料前處理.....	7
3.3 模型建立.....	8
3.4 參數優化.....	10
4. 研究結果.....	12
5. 參考資料.....	13

## 1、背景介紹

### 1.1 背景說明

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。本次實驗欲透過遷移學習的方式將六種情緒(傷心、生氣、喜愛、開心、驚訝、恐懼)的句子進行分類訓練，並希望之後能夠藉由蒐集人們於社群網站上面發布的句子，利用本模型去進行觀察人們的情緒狀態，並於情緒異於正常時能夠觀察到警訊，並且適時的在現實中對於當事者進行協助。

### 1.2 問題描述 (5W1H)

針對本次研究主題，5W1H 分析如下：

表 1、5W1H

What	現代人生活壓力大，情緒不穩定的發生情況上升。
Why	透過深度學習的模型，可以藉由人們在社群網路上發布的句子，觀察人們日常中的情緒變動，並且能夠及時發現並提供協助。
Where	社群網路、留言板、聊天室。
When	當人們於日常中表現異常時。
Who	每個人都可以。
How	先將句子進行資料前處理後，利用 BERT 模型訓練，學習判斷句子情緒的能力。

## 2、文獻回顧

### 2.1 BERT (Bidirectional Encoder Representations from Transformers)

基於變換器的雙向編碼器表示技術 (Bidirectional Encoder Representations from Transformers, BERT) 是 Google 於 2018 年所提出，其為一種自然語言處理 (Natural Language Processing, NLP) 的預訓練技術，利用 BERT 可以使文章語意理解更為容易。BERT 是一種完全雙向的非監督式模型，BERT 整體框架分成兩個階段，預訓練以及微調，透過前期大量資料的無監督訓練，學習語言、文法、詞意等知識，在利用有標註的資料進行微調，可以解決自然語言處理上資料標註不足的問題。預訓練時主要是利用兩個任務去進行，第一個為克漏字填空，將句子中的某個單字遮住，並將單字的隱藏向量輸入，利用模型去進行單字預測，第二個為下個句子預測基於對兩個文本句子之間關係的理解，這種關係並非通過語言建模直接獲得。在多項 NLP 任務上取得了出色的表現。

### 2.2 Transformer

Transformer 模型由 Google 在 2017 年在中提出。拋棄了傳統 CNN 和 RNN，使用 Attention 替換了原先 Seq2Seq 模型中的迴圈結構，由 self-Attention 和 Feed Forward Neural Network 組成。Transformer 等相關技術也逐漸由 NLP 流向其他領域，例如計算機視覺 (Computer Vision)、語音、生物、化學等。Transformer 模型結構是藉由六個編碼器(Encoder)和解碼器(Decoder)組成，輸入 (Inputs) 和輸出 (Outputs) 通過同一個訓練好的詞嵌入層 (Word Embedding) 將輸入字元轉換為維度為  $d$  的向量。Transformer 利用 attention 的方式，有利於平行運算，除了有更好的學習表現外，還提升了訓練速度，對各種領域都很很大的影響。

### 3.研究方法

#### 3.1 資料介紹

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。資料集分為訓練及、測驗及和驗證集，訓練集中有 16000 筆訓練資料，測驗集和驗證集中各有 2000 筆資料，總共是 20000 筆的資料，資料分布情形如圖 1、圖 2 和圖 3

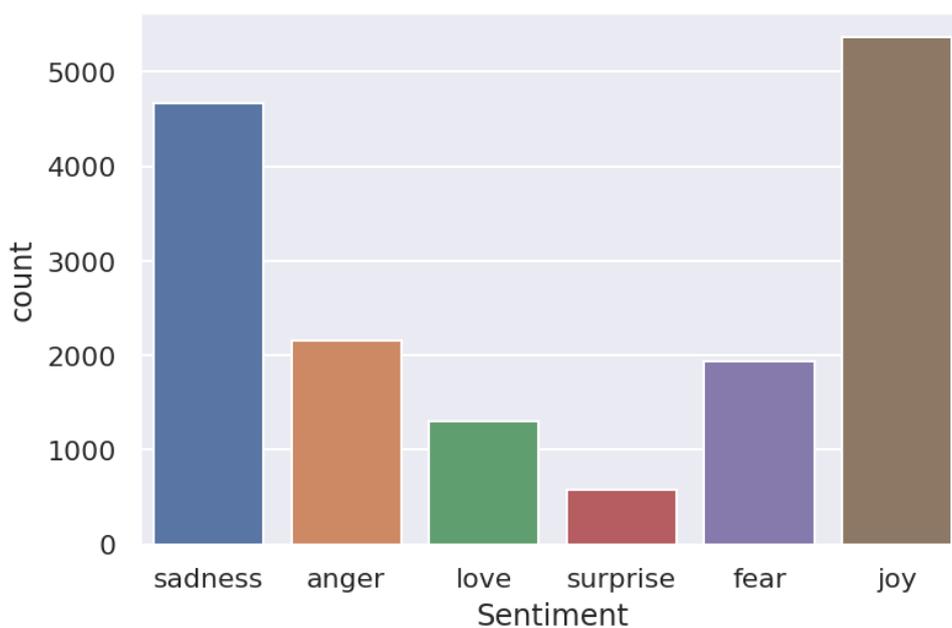


圖 1、訓練集資料分布

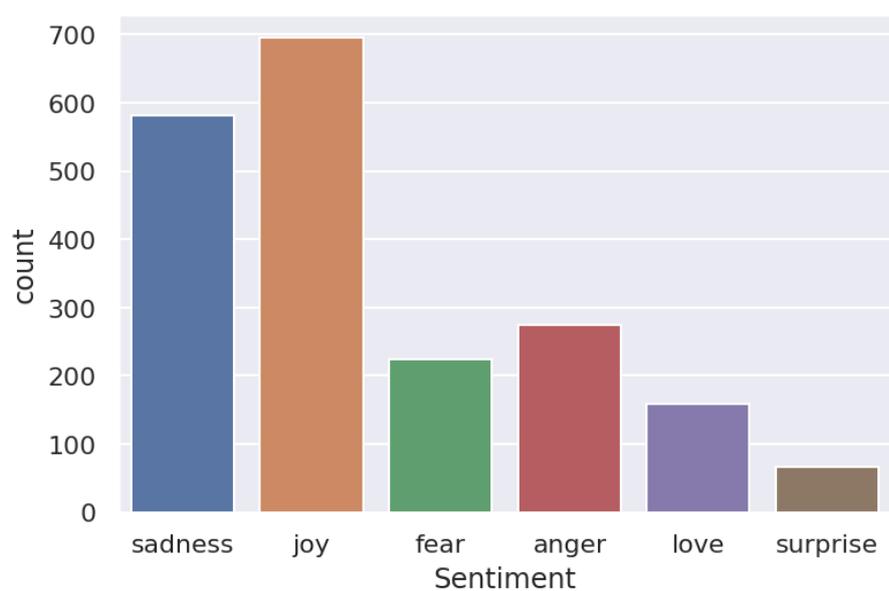


圖 2、測驗集資料分布

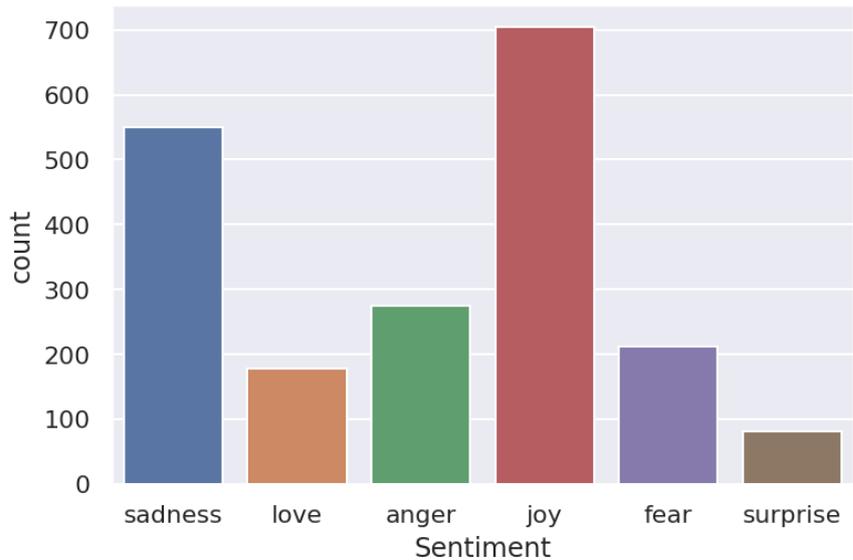


圖 3、驗證集資料分布

從以上圖中可以得知，六個標籤的資料分布有點不平均，所以在進行實驗之前，會先進行一次資料擴增，使得最多與最少的資料量之間數量差距可以縮小。

### 3.2 資料前處理

由於資料分布不平均的關係，擔心會影響到訓練結果，所以在進行資料處理前，先將資料集內的資料做資料擴增。這次選擇的擴增方法為回譯，將原始的句子利用翻譯的方式，轉譯為其他的語言，再重新翻譯回原本的語言，使資料集在語意不影響的情況下，得到其他種表達方式。主要是對於 love, surprise 兩個資料集裡用 Google 翻譯做資料擴增，使用了中文及法文作為回譯目標，讓數量最多與最少的落差可以減少，圖四為翻譯的過程。

	A	B	C	D	E	F	G
A2067	=GOOGLETRANSLATE(GOOGLETRANSLATE(A629,"en","fr"),"fr","en")						
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 curio	surprise					

圖 4、使用 Google 翻譯進行資料擴增

先將三個資料集的資料做合併，接著利用 text\_hammer 中的函數對句子做資料前處理，總共進行了五個步驟，資料空缺檢查、將句子內文字都轉成小寫、將縮寫還原成原始型態、刪除不相關的句子、刪除特殊符號，接著在將原本的標籤轉換成 one hot 的型態。前處理完畢之後將資料以 7:3 的比例分割為訓練集和測試集，圖 7、圖 8、圖 9 為程式碼參考。

```
def text_preprocessing(df, col_name):
    column = col_name
    df[column] = df[column].progress_apply(lambda x: str(x).lower())
    df[column] = df[column].progress_apply(lambda x: th.cont_exp(x)) #you're -> you are; i'm -> i am
    df[column] = df[column].progress_apply(lambda x: th.remove_emails(x))
    df[column] = df[column].progress_apply(lambda x: th.remove_html_tags(x))
    # df[column] = df[column].progress_apply(lambda x: ps.remove_stopwords(x))

    df[column] = df[column].progress_apply(lambda x: th.remove_special_chars(x))
    df[column] = df[column].progress_apply(lambda x: th.remove_accented_chars(x))
    # df[column] = df[column].progress_apply(lambda x: th.make_base(x)) #ran -> run,
    return(df)
```

圖 5、資料前處理

```
to_categorical(data_train.Sentiment)

array([[0., 0., 1., 0., 0., 0.],
       [0., 0., 1., 0., 0., 0.],
       [0., 0., 0., 0., 0., 1.],
       ...,
       [0., 0., 1., 0., 0., 0.],
       [0., 0., 1., 0., 0., 0.],
       [0., 0., 0., 0., 1., 0.]], dtype=float32)
```

圖 6 label 轉碼程式

```
from sklearn.model_selection import train_test_split
data_train, data_test = train_test_split(df_cleaned, test_size = 0.3, random_state = 42, stratify = df_cleaned.Sentiment)
```

圖 7 重新分割訓練與資料集

### 3.3 模型建立與參數調整

本研究利用 BERT 預訓練好的參數來協助訓練。因此我們除了引入網路外，並再最後加入全局平均池化層減少模型參數，避免過擬合；以及加入 Dropout 層減少訓練的時間，也能避免過擬合；最後再利用 Sigmoid 函數進行分類。模型建立的程式碼如圖 8、圖 9：

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

圖 9、載入 BERT 預訓練模型

```

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)

```

圖 10、模型架構

判斷模型準確率的標準部分，使用的 `balanced_accuracy` 來確保不會有其中一個分類準確率低落的情形，以平均準確率來提升整體的準確率可以一起成長，如圖 11。

```

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

```

圖 11、學習率調整

Layer (type)	Output Shape	Param #	Connected to
input_ids (InputLayer)	[(None, 70)]	0	[]
attention_mask (InputLayer)	[(None, 70)]	0	[]
tf_bert_model_1 (TFBertModel)	TFBaseModelOutputWithPoolingAndCrossAttentions(last_hidden_state=(None, 70, 768), pooler_output=(None, 768), past_key_values=None, hidden_states=None, attentions=None, cross_attentions=None)	108310272	['input_ids[0][0]', 'attention_mask[0][0]']
global_max_pooling1d_2 (GlobalMaxPooling1D)	(None, 768)	0	['tf_bert_model_1[0][0]']
dense_6 (Dense)	(None, 128)	98432	['global_max_pooling1d_2[0][0]']
dropout_114 (Dropout)	(None, 128)	0	['dense_6[0][0]']
dense_7 (Dense)	(None, 32)	4128	['dropout_114[0][0]']
dense_8 (Dense)	(None, 6)	198	['dense_7[0][0]']

圖 12、最終模型架構

### 3.4 參數優化

除了預訓練時所獲得的最佳模型參數外，其餘的參數皆是使用相同的數據，還有超參數優化的空間。希望可以在有限的實驗次數中找到最佳模型參數來達到最高的準確率，節省時間以及減少人為調整的因素。挑選出的因子以及水準如表 2：

表 2、實驗因子及水準

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

我們利用了實驗設計中的田口方法，有效減少調整參數的總次數，並獲得相同的結果。我們選擇了上述所提到的四項參數作為四個因子，並使用三個水準，應用 L9 實驗設計參數組合來幫助參數優化。實驗設計參數組合及結果詳情如表 3：

表 3、L9 實驗設計參數組合

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

準確度最高的為實驗 5，準確度達到了 0.9266，將這 9 次實驗結果透過統計分析後可以得到各個因子對於準確率的影響程度，圖 13、圖 14、圖 15 為利用 Minitab 計算出的實驗結果。我們又從中挑選出了各個因子中最好的水準組合來做測試，希望可以得到更好的結果，最佳組合為表 4，準

確度為 0.9252。

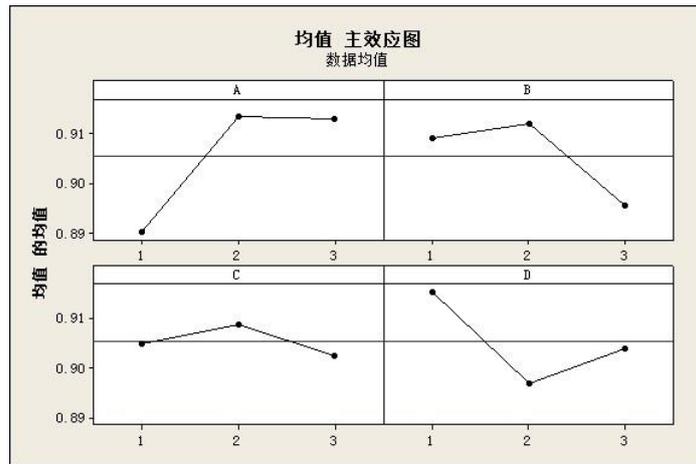
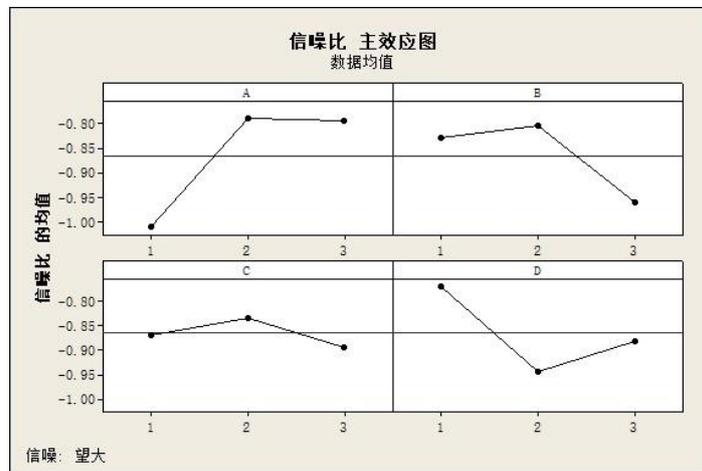


圖 13



信噪：望大

圖 14

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

圖 15

表 4、最佳組合

	實驗	epoch	dropout	Batch Size	learning rate	Accuracy
最佳組合	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

圖 16、最佳參數調整的各情緒預測準確度

#### 4.研究結果

由於最佳參數調整之後，發覺預測準確度與實驗設計中的最高準確度差距不高，所以又在進行了一次實驗，這次我們將 dropout 層刪除掉，結果如表 5。顯示出目前參數調整的最好結果大概就是落在 93%附近，顯示出假如我們想要追求更高的準確度，我們應該從其他方面下手，例如資料集或是模型架構，才有可能突破這個限制，將準確度拉到更高的地方。

表 5

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

於模型上進行泛化度測驗的時候，可能會出現一句話被判斷符合兩個情緒的可能性差不多，經觀察後發現假如是正面以及反面的情緒類別可以判斷正確，但在正面情緒細分為 joy, love, surprise 的部分，有時會出現 joy 和 love 都很高或是 joy 和 surprise 都很高的情況，可能是因為這兩者情緒本身就是會互相參雜，或許之後可以考慮其他分類更明確的情緒標籤。

未來展望希望可以結合網路爬蟲，蒐集社群網站上實際發布的句子，對模型再

進行訓練，將模型的泛化程度可以更加提升，或是增加更自動化的步驟，自動蒐集網站上的用戶留言，進行分析後針對情緒不穩定的人們進行關照。

## 5.參考資料

- (一) <https://www.kaggle.com/praveengovi/emotions-dataset-for-nlp>
- (二) <https://zhuanlan.zhihu.com/p/48508221>
- (三) <https://medium.com/ching-i/transformer-attention-is-all-you-need-c7967f38af14>
- (四) <https://www.gushiciku.cn/pl/gdg7/zh-tw>
- (五) [https://paddlepedia.readthedocs.io/en/latest/tutorials/pretrain\\_model/bert.html](https://paddlepedia.readthedocs.io/en/latest/tutorials/pretrain_model/bert.html)
- (六) [https://blog.csdn.net/weixin\\_44799217/article/details/115374101](https://blog.csdn.net/weixin_44799217/article/details/115374101)
- (七) <https://zhuanlan.zhihu.com/p/339894411>