

實務專題計畫成果摘要報告 網路評論內容評分與分析

專題編號：112-CSIE-S007

執行期限：111 年第 1 學期至 112 年第 1 學期

指導教授：王正豪

專題參與人員：109590023 廖堃霖

109590030 郭廷弦

109590045 李慶桓

一、緣由與目的

現在人們能夠在網路上輕易的發表自己的言論與意見，比如說 google 地圖上店家的評論，又或者是電子商務裡商品的評論，因此希望能夠藉由此實務專題，讓我們在閱讀評論時能更簡單且方便。本小組所研究的技術，主要是利用巨量分析及預測給予這整段評論的好壞，且能標示出評論內所用詞彙的優劣，並以此為基礎去做功能上的延伸。

二、使用技術方法

(一)Cuda:

以 gpu 的形式去跑訓練或執行一些 function，使整體程式更加快速。

(二)pytorch_lightning:

用來訓練神經網路，使用此工具可以將 bert 微調成可以進行 muti-class 的預測模型。

(三)nltk- sentiwordnet:

是 nltk 中的工具，為一個詞彙數據庫中的每個詞彙提供情感分數。

(四)nltk-punkt:

將一個字串進行分詞。

(五)nltk-averaged_perceptron_tagger:

能預測此分詞的詞性。

(六)bert:

一個可以看懂前後文的預訓練模型。

(七)dataset,dataloader:

用來做批次訓練的工具，可以一次訓練一個 batch size 的資料量

三、架構流程

(一)網頁流程：

1. 買家：

- (1)留言與評分
- (2)預測 predict score
- (3)將資料傳進資料庫
- (4)重整頁面
- (5)顯示評論

2. 賣家：

- (1)進入分析頁面
- (2)抓取所需 comment
- (3)標示優劣詞彙
- (4)更新原始、預測平均分數
- (5)顯示分析結果

(二)模型訓練流程：

1. 資料轉換與採樣
2. 生成訓練集並批次訓練
3. LightingModule
4. BertModel
5. Dropout
6. Linear
7. sigmoid
8. LossFunction
9. Optimizer
10. 訓練及儲存結果

四、工具說明

(一)模擬網站：

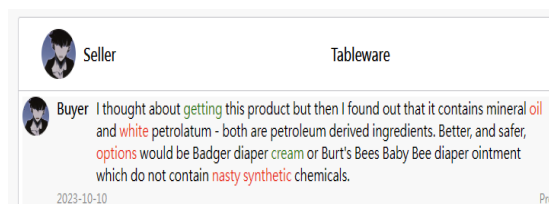
1. 前端：Vue.js、axios
2. 後端：Flask
3. 資料庫：mariaDB

(二)模型訓練

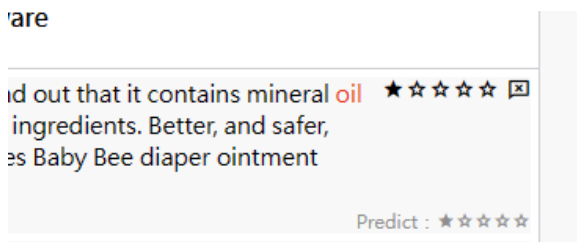
1. pytorch_lighting

五、實驗結果

最終我們決定經由自己設計的網頁進行模擬測試，經由第一張圖(圖一)可以看到留言上的詞彙有以不同的顏色進行標示，再來第二張圖(圖二)也有我們所預測的 predict score，而最後一張圖(圖三)分別使用者們評分的平均值(圖三-左)以及我們所預測評分的平均值(圖三-右)。



(圖一)



(圖二)

Average of comment star : 4.01850	Average of predict star : 4.01199
-----------------------------------	-----------------------------------

(圖三)

六、結論

首先非常感謝我們的指導教授-王正豪教授耐心的指點，一開始像無頭蒼蠅一樣，完全沒有方向，但經由教授的指導下漸漸有了雛形，過程中給予的意見，也使整個專題的進行流暢了許多，再來就是這個實務專題對於我們在團體的項目上幫助很大，雖然並不是非常艱深的大項目，但在這其中所得到的收穫也是裨載而歸。

七、參考文獻

[1]tomorrow”. (2023). 理清 GPU、CUDA、CUDA Toolkit、cuDNN 關係及下載安裝.

Retrieved from

https://blog.csdn.net/qq_42406643/article/details/109545766

[2]進擊的 BERT:NLP 界的巨人之力與遷移學習。(2019)。

檢自

https://leemeng.tw/attack_on_bert_transfer_learning_in_nlp.html

[3] LIGHTNINGMODULE-PyTorch.

Retrieved from

https://lightning.ai/docs/pytorch/stable/common/lightning_module.html

[4] Pytorch-DATASETS &
DATALOADERS.

Retrieved from

https://pytorch.org/tutorials/beginner/basics/data_tutorial.html

[5]

Sentiment Classification Using
SentiWordNet.

Retrieved from

<https://notebook.community/AC209ConsumerConfidence/AC209ConsumerConfidence.github.io/SentiWordNet>

[6]

Multi-label Text Classification with
BERT and PyTorch Lightning.

Retrieved from

<https://curiously.com/posts/multi-label-text-classification-with-bert-and-pytorch-lightning/>