明道中學高中部資訊類專題

113學年度
專題作品說明書

題目

高二三班**21S082**黃翊修
高二三班**21S269**陳禹睿
高二三班**21S032**許晏誌

指導老師:陳郁弦 老師

中 華 民 國 一一三 年 二 月 二十六 日

目錄

摘要

我們發現，消費者在選購商品時，通常依賴評論來判斷商品的好壞。然而，隨著假評論的數量不斷增加，以及目前評分方式過於單調，這些問題大大影響了使用者的決策品質。因此，我們設計了一個創新的評論分析系統，試圖優化現有的購物體驗。我們透過爬蟲技術獲取大量數據，並利用人工智慧模型進行訓練，最終開發出一款Google插件。這款外掛能快速分析評論內容，篩選出真實有價值的意見，並提供更具參考價值的評分方式，幫助使用者更輕鬆判斷商品的好壞。我們希望這個工具能有效提升消費者的購物效率，並創造更透明的購物環境。

關鍵字：LSTM、網路爬蟲、Google插件

壹、前言

一、研究動機

在現今數位時代，網路評價已成為消費者決策的重要參考依據。然而，現有的評分機制大多採用一到五顆星的簡單量化方式，這種方式往往無法準確反映使用者的真實體驗，甚至可能出現評分與內容不符的情況。例如，一則四星評論可能實際上充滿負面意見，而某些高分評價則可能來自機器人或非真實使用者，導致我們消費者難以獲得準確資訊，商家也無法獲取有價值的回饋。

為了解決這個問題，我們運用人工智慧技術，結合文字情感分析，開發一套更精確的評論評估系統。我們設計的系統將透過深入分析評論內容的情緒與語意，提供比傳統評分機制更具代表性與參考價值的評估結果。這不僅有助於消費者做出更明智的購買決策，也能讓商家獲得更具洞察力的產品反饋，並且提升市場透明度與產品品質。

二、研究目的

本實作的研究目的是設計一款Google插件，幫助使用者快速分析評論，提升對商品評價的判斷效率與準確性。此外，透過本實作，我們希望學習文本情感分析的基本概念，並深入理解如何運用人工智慧技術來開發情感分析專案。同時，我們也將學習Google插件的開發技巧與完整流程，包括插件的設計、功能實現與測試，藉此累積經驗，為未來相關專案打下堅實基礎。

貳、文獻探討

1、 相關文獻

表一:相關文獻

主題	文章標題	作者與年份	摘要	心得
LSTM	Long short-term memory	Graves, A. (2012)	該論文分析了1990年代深度學習模型的演進、RNN模型遇到的問題與LSTM如何解決困境。其也詳細說明LSTM如何「保有記憶」，以及模型的應用。	該研究以詳盡對比強調LSTM模型的優勢，分析運作原理，對於本次選擇與建構模型有參考價值
網路爬蟲	Study of Web Crawler and its Different Types	Udapure, T. V., Kale, R. D., & Dharmik, R. C (2014).	該論文探究網路爬蟲的起源，以及數種現今被廣泛使用的網路爬蟲種類。	了解網路爬蟲的運作原理對於我們在前端與網頁溝通，取得評論的操作有很大幫助。
網路爬蟲與反爬蟲	網路爬蟲與反爬蟲相關研究	楊杰淮 (2022)	該研究分析python對網路爬蟲的影響與應用，以beautifulsoup插件探究python與html的結合和互動。	該研究解決了我們實作時遇到的在網頁環境調用python程式的問題，在技

				術細節上解釋詳細。
LSTM應用	A review on the long short-term memory model	houdt, G. V. (2020)	該研究總結LSTM的算法優化與應用，如語音辨識與解決梯度爆炸問題。論文中也說明了LSTM模型的建構和訓練。	該研究使我們更了解如何訓練LSTM模型，在實作時遇到的問題在論文中得到解答。

表1資料來源:研究者自行彙整

二、LSTM

長短期記憶(Long Short-Term Memory, LSTM)網路是一種特殊的循環神經網路(Recurrent Neural Network, RNN)，專門用來解決傳統RNN在處理長期依賴關係時的限制。楊宗恩(**2017**)學者指出:「**LSTM** 可以被視為一種具有時間性處理架構的雙層類神經網路模型。」此外，**Graves**(**2012**)學者提到:「循環神經網路的一大優勢是能夠在處理輸入與輸出序列對應時，運用前後文資訊來提升準確性。」LSTM 透過獨特的記憶單元設計，使其能夠在時間序列資料中有效保留關鍵資訊，並根據需求選擇性地遺忘不必要的資訊，因此在文字情緒分析等應用領域表現優異。

在情緒分析領域，LSTM被廣泛應用於辨識文字內容的情緒傾向，例如:正向、負向或中立情緒。其主要優勢在於能夠掌握較長句子的前後文關係，進而更準確地判斷情緒的變化。LSTM的記憶單元包含三個核心組件—輸入閘、遺忘閘和輸出閘，這些閘門機制負責調節資訊的流動。例如，遺忘閘可根據需求移除無關資訊，使LSTM更專注於關鍵的前後文內容，進而提升情緒分析的準確度。

LSTM的運作流程包括以下步驟:首先，輸入閘決定當前輸入對記憶狀態的影響;接著，遺忘閘根據前一個時間點的記憶狀態，篩選需要保留或移除的資訊;最後，輸出閘決定當前時間步驟的輸出結果。透過這些機制，LSTM能夠有效辨識並擷取文字內容中的情緒資訊，即使這些資訊分散在較長的語句中，也能準確捕捉其關聯性。

總而言之，LSTM具備強大的長期依賴處理能力，在情緒分析方面展現出顯著優勢，並已成為文字情感分類的主流方法之一。**Van Houdt**(**2020**)學者等人指出，「**LSTM**已經深刻影響機器學習與神經運算領域。」此外，LSTM的應用範圍廣泛，不僅限於情緒分析，還在各類自然語言處理任務中發揮重要作用。如下表一所示，遞歸神經網路比較表。

表二:遞歸神經網路比較表

特徵	RNN (Recurrent Neural Network)	LSTM (Long Short-Term Memory)	Contextual LSTM
處理數據類型	序列數據	序列數據	序列數據
時間依賴性建模	能夠建模短期依賴，但對於長期依賴無效	使用記憶單元有效建模長期依賴	建模長期依賴，並加入上下文關係
記憶能力	記憶短期依賴，容易忘記長期依賴	長短期記憶結構使其擅長長期依賴建模	不僅記憶長期依賴，還結合語境和上下文
梯度消失/爆炸問題	容易出現梯度消失和梯度爆炸	透過「遺忘門」等結構減輕梯度消失和爆炸問題	與LSTM相同，並加強語境相關信息
網絡結構	單一隱藏層循環結構	增加了「遺忘門」、「輸入門」和「輸出門」	在LSTM基礎上引入上下文信息，如注意力機制
擅長的應用	短序列數據處理，如簡單時間序列預測	長序列數據處理，如語音識別、機器翻譯	需要上下文感知的應用，如機器翻譯、情感分析
上下文感知能力	無，僅依賴於序列中當前和先前的輸入	通過長期依賴部分可以部分捕捉上下文信息	明確加入上下文建模，能夠捕捉更廣泛的語境關係
適用場景	簡單序列處理任務（如基礎時間序列預測）	複雜序列處理任務（如長文本處理、語音分析）	需要充分上下文理解的任務（如對話系統、NLP）

表二資料來源:研究者自行彙整分析

三、網路爬蟲

網路爬蟲是一種自動化程式，專門用來瀏覽網頁並擷取網站上的資料。其主要功能包括抓取特定資訊，例如：文字內容、圖片、連結等，並將這些資料儲存下來，以便後續進行分析。爬蟲透過模擬使用者的操作，逐頁訪問網站，並依照設定的規則擷取所需的內容。

最早的網路爬蟲可以追溯到**1993**年，由**Matthew Gray**所編寫的**World Wide Web Wanderer**。**(Udapure et al., 2014)**。這是網路爬蟲技術的起點，當時的主要目標是遍歷所有網頁，以探索互聯網的發展。然而，由於當時網路結構的不穩定性及技術限制，該爬蟲並未能完全實現其預期功能。

網路爬蟲的運作過程通常包括三個主要步驟：首先，爬蟲會訪問一個初始的URL，下載該頁面的HTML代碼；接下來，它會解析網頁結構，提取所需數據或識別新的URL，並將這些新URL加入爬取隊列；最後，爬蟲會重複此過程，直到所有目標數據被擷取，或達到預設的條件限制。獲取的數據可儲存為Excel檔、CSV檔（逗號分隔值）或純文字（TXT）檔案，以供進一步分析。

網路爬蟲的應用十分廣泛，包括搜尋引擎索引的建立、價格比較網站的數據收集、新聞與社交媒體內容分析等。然而，這項技術也可能被濫用，例如大規模數據抓取、侵犯個資隱私，或違反網站使用條款。因此，許多網站會設置反爬蟲機制，例如驗證碼、人機驗證（CAPTCHA）、IP存取頻率限制等，以防止未經授權的數據擷取行為。

參、研究方法

1、 研究架構

圖一:研究架構圖

圖片來源:研究者使用gitmind自行繪製

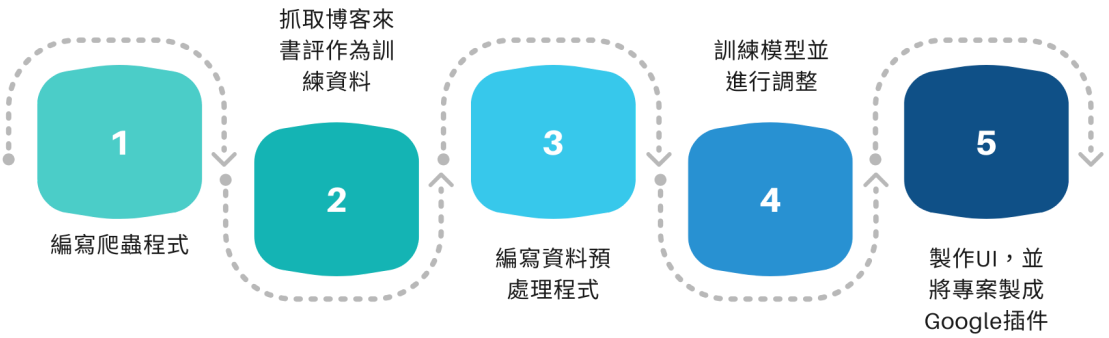
2、 研究方法

(一)文獻分析法:本研究透過網路資料蒐集與文獻閱讀，讓我們了解當前情緒辨識技術的發展現況、所使用的模型架構，以及爬蟲技術的最新進展。我們分析不同情緒辨識模型的特點與應用，並評估爬蟲技術在資料擷取與處理上的發展程度，期望提供更全面的理解與參考。

(二)實作研究法:研究以LSTM模型為核心，結合網路爬蟲技術，開發一款具備直觀且簡潔介面的Google插件，讓使用者能夠輕鬆運用該插件進行直觀的評論分析。

3、 研究流程

圖二:研究流程圖



圖片來源:研究者使用canva.com自行繪製

(一)收集訓練資料

我們利用Python的requests函式庫編寫爬蟲程式，針對博客來網站進行數據擷取。爬蟲的主要功能包括獲取網站上的所有內容，透過標籤識別並篩選出評論部分，最後將整理後的數據儲存為CSV格式(逗號分隔值），以利後續的處理與分析。如下圖三所示，爬蟲之程式碼。

圖三:爬蟲程式碼

```
    # 從網站擷取評論數據
    headers = {"User-Agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.169 Safari/537.36"}
    product_id = str(url.split('/products/')[1].split('?')[0])
    comments_url = f"https://www.books.com.tw/booksComment/getCommemt/{product_id}"
    res = requests.get(comments_url, headers=headers)
    soup = BeautifulSoup(res.text, 'html5lib')
    comments = soup.find_all("span", calss="comment-content")

    # 以CSV檔保存評論
    csv_output = io.StringIO()
    csv_writer = csv.writer(csv_output)
    for comment in comments:
        csv_writer.writerow([comment.getText().strip()])

    csv_output.seek(0)
    df = pd.read_csv(io.StringIO(csv_output.getvalue()), header=None)
    comment_list = df[0].tolist()
```

圖片來源:研究者於 replit 環境實作並截圖

(二)清洗資料

我們採用Jieba作為中文分詞工具。由於中文文字不像英文單詞那樣具有明確的界限，因此分詞成為模型建構中的關鍵步驟。Jieba能夠將長篇中文評論拆解為具備語義意義的詞彙，並提供可擴充的中文停用詞庫。在人工標註資料時，常見但無情感意義的詞語與標點符號會被加入停用詞列表，以提升模型的辨識準確度。如下圖四所示，導入停用詞語料庫截圖。

圖四:導入停用詞語料庫

```
#  停用詞載入
def load_stopwords(file_path):
    with open(file_path, 'r', encoding='utf-8') as file:
        stopWords = set(line.strip() for line in file)
    return stopWords
```

圖片來源:研究者於 replit 環境實作並截圖

為了確保評論內容在資料處理過程中的一致性，並避免繁體與簡體中文混用，我們採用了OpenCC（Open Chinese Convert）工具，對所有CSV檔案進行統一的文字轉換，將其中的內容全數轉換為繁體中文。如下圖五所示，OpenCC轉換中文部份程式碼。

圖五中的內容是讀取爬蟲儲存的CSV檔。我們定義正面評論為0，負面評論為1，並且將爬蟲獲取的資料人工標註正負面，完成資料準備。

圖五:opencc轉換程式部分

```
with open('C:/Users/Your-Username/AppData/Local/Google/Chrome/User Data/Default/Extensions/test flask/popup/tokenizer.pkl', 'rb') as handle:
    token = pickle.load(handle, encoding='latin1')

model_path = 'C:/Users/Your-Username/AppData/Local/Google/Chrome/User Data/Default/Extensions/test flask/popup/weights/boston_model.h5'
model = load_model(model_path)
print("Model loaded successfully.")
```

圖片來源:研究者於 replit 環境實作並截圖

圖中的內容是讀取爬蟲儲存的CSV檔。我們定義正面評論為0，負面評論為1，並且將爬蟲獲得的資料手動標註正負面，完成資料準備。

(三)訓練模型

我們採用了LSTM模型，其優勢在於能夠保留長時間的依賴關係，使其適用於文字分析、語音辨識等涉及長篇內容的任務。特別是在處理較長或情感較為複雜的評論時，例如：書評，其中可能同時包含正面與負面的評價詞彙，LSTM 能夠更有效地理解並處理這類內容。

在模型建構方面，我們設定輸出維度（output_dim）為128，並使用256個神經元。模型建置完成後，即進行訓練，並將訓練內容的最大長度設為300個字，以避免內容過長影響模型效能。如下圖六所示，LSTM參數設定值。

圖六：LSTM參數設定

```
def predict_sentiment(texts):

processed_texts = [list(jieba.cut(text, cut_all=True)) for text in texts]
processed_texts = [list(filter(lambda a: a not in stopWords, content)) for content in processed_texts]


sequences = token.texts_to_sequences(processed_texts)
sequences = sequence.pad_sequences(sequences, maxlen=300)


predictions = model.predict(sequences)

return predictions
```

圖片來源：研究者於 replit 環境實作並截圖

在第五至第六次訓練中，模型的準確率已逐漸趨於穩定，未再出現顯著提升，這表示模型的訓練已接近收斂，整體性能達到最佳狀態。如下圖七所示，訓練次數與結果準確率輸出。

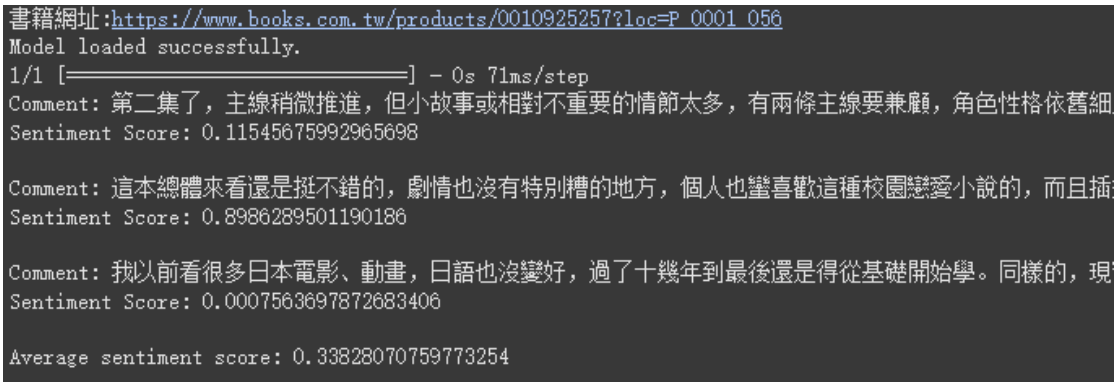
圖七：訓練次數與結果準確率輸出

```
loss: 0.3626 - accuracy: 0.8679 - val_loss: 0.3077 - val_accuracy: 0.8898 - 13s/epoch - 428ms/step

loss: 0.2194 - accuracy: 0.9224 - val_loss: 0.2789 - val_accuracy: 0.8936 - 10s/epoch - 323ms/step

loss: 0.1319 - accuracy: 0.9575 - val_loss: 0.2485 - val_accuracy: 0.9066 - 12s/epoch - 381ms/step

loss: 0.0629 - accuracy: 0.9825 - val_loss: 0.2528 - val_accuracy: 0.9092 - 15s/epoch - 483ms/step

loss: 0.0247 - accuracy: 0.9935 - val_loss: 0.2931 - val_accuracy: 0.9105 - 19s/epoch - 623ms/step

loss: 0.0076 - accuracy: 0.9997 - val_loss: 0.2943 - val_accuracy: 0.9157 - 12s/epoch - 393ms/step

loss: 0.0032 - accuracy: 1.0000 - val_loss: 0.3187 - val_accuracy: 0.9092 - 12s/epoch - 400ms/step

loss: 0.0020 - accuracy: 1.0000 - val_loss: 0.3427 - val_accuracy: 0.9105 - 10s/epoch - 328ms/step

loss: 0.0012 - accuracy: 1.0000 - val_loss: 0.3520 - val_accuracy: 0.9118 - 12s/epoch - 393ms/step

loss: 8.9993e-04 - accuracy: 1.0000 - val_loss: 0.3600 - val_accuracy: 0.9105 - 12s/epoch - 398ms/step
/python3.10/dist-packages/keras/src/engine/training.py:3000: UserWarning: You are saving your model as an HDF5 file via `model.save()`.
ave_model(
```

圖片來源：研究者以google colab 實作並截圖

模型的輸出結果包含Comment和Sentiment Score兩個部分。其中，Comment為爬蟲程式從網站擷取的文字內容，而Sentiment Score則是模型對該文字進行情感分析後的評分結果。當評論表達正面情緒時，分數越接近0；當情緒較為負面時，分數則趨近1；若評論未明確包含正面或負面的情感詞彙，分數則接近中間值。從下圖八可以觀察到，書評的情感分數大多偏向正面。

圖八：模型輸出格式



圖片來源：研究者以google colab 實作並截圖

(四)製作成google插件

為了讓使用者更方便使用，我們最終選擇以Google Chrome插件來呈現模型。Google 插件的開發主要使用HTML、CSS和JavaScript，並且必須包含一個名為manifest.json的關鍵檔案。該檔案的作用是讓Chrome瀏覽器識別並執行該插件，使其能夠與瀏覽器進行互動，進而實現所需功能。如下圖九所示，manifest.json檔案內容。

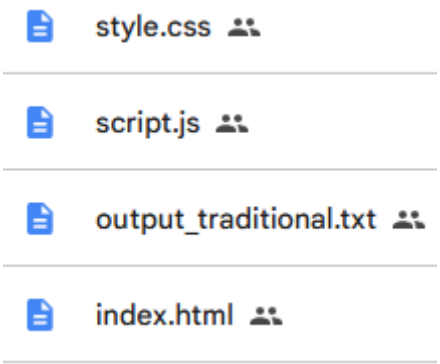
圖九：manifest.json檔案內容

圖片來源：研究者於 replit 環境實作並截圖

在檔案結構中，popup/index.html 被定義為插件的彈出視窗，因此需要建立

一個名為 popup 的資料夾，並在其中建立 index.html 來設計視窗內容，以及 style.css 用於樣式設定，以確保視覺呈現的統一性與使用體驗的優化。

圖十：資料夾內容，第一、三、四項檔案為UI介面



圖片來源：研究者從電腦資料夾內自行截圖

HTML 是構建 UI 介面的基礎，首先需要建立基本的頁面結構，並透過適當的標籤來定義內容與外觀。在此過程中，除了使用 <link> 標籤來連接 CSS 檔案，和 <script> 標籤來引入 JavaScript 以實現互動功能外，還需要設計 UI 元素以確保介面符合使用者需求。

首先是標題與按鈕，頁面的主標題使用 <h1> 標籤來呈現，使使用者能夠直觀地了解頁面的用途。此外，透過 <button> 標籤建立按鈕，讓使用者可以透過點擊來觸發相關功能。

接著是頁面內容，在頁面設計中，<p> 標籤主要用於顯示和修改文本內容，特別是在需要動態更新資訊時。例如，頁面可以根據使用者操作，透過 <p> 標籤即時顯示分析結果、處理進度等重要資訊。

再來是UI 介面架構，基本的 UI 介面通常包括頁面容器、標題、按鈕、顯示區域等元素，這些元素可透過 CSS 設定樣式，使介面更美觀且符合使用者體驗需求。

最後是互動與動態更新，在 UI 設計完成後，JavaScript 可用於處理頁面的互動與內容變更。例如，當使用者點擊按鈕時，JavaScript 可即時修改 <p> 標籤的內容，顯示系統分析結果或運行狀態。此外，JavaScript 也可用來控制過渡效果、顯示進度條等，以提升使用者體驗。

圖十一：popup/index.html 內容

```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link href="https://fonts.googleapis.com/css2?family=Noto+Sans+TC:wght@400;700&display=swap"
rel="stylesheet">
    <link rel="stylesheet" href="./style.css" />
  </head>
  <body>
    <div class="top-line"></div>

    <div class="container">
      <header>評論分析系統</header>
      <button id="analyzeBtn">進行該頁面商品評論分析</button>
      <div id="status" style="display: none;">等待中...</div>
    </div>

    <div class="bottom-line"></div>

    <script src="./script.js"></script>
  </body>
</html>
```

圖片來源:研究者於 replit 環境實作並截圖

接下來，我們將撰寫 CSS 程式碼，以設計和優化網頁的視覺呈現。針對關鍵字顯示的螢光效果，我們將使用 CSS 設定背景高亮，使其在使用者互動時更加醒目。這樣的設計不僅有助於提升資訊的可讀性，也能增強整體介面的互動感與專業度，使頁面呈現更直觀且符合使用需求。

圖十二、十三: style.css 內容

```css
body {
  font-family: 'Noto Sans TC', sans-serif;
  background-color: #f9f9f9; /* 恢復原本的背景色 */
  color: #333;
  display: flex;
  justify-content: center;
  align-items: center;
  flex-direction: column;
  position: relative; /* 用於讓線條定在頂部和底部 */

  margin: 50px;
  width: 400px;
  height: 150px;
}
header {
  font-size: 2rem;
  font-weight: 700;
  margin-bottom: 20px;
}
button {
  background-color: #4CAF50;
  color: white;
  border: none;
  padding: 15px 30px;
  font-size: 1rem;
  cursor: pointer;
  border-radius: 5px;
  transition: background-color 0.3s ease;
}
```

圖片來源:研究者於 replit 環境實作並截圖

在建立所有必要的檔案後，開啟Google瀏覽器並進入擴充功能管理頁面，接著啟用開發人員模式。啟用後，點擊「載入未封裝項目」，選擇包含擴充功能的資料夾並載入。完成這些步驟後，插件即可正常運行並開始使用。如下圖十三所示，Google插件內顯示狀態。

圖十四:google插件內顯示狀態

圖片來源:研究者從google管理擴充模式頁面內自行截圖

這個程式的主要功能包含三個部分:首先是提供一個直觀易用的UI介面,使用者可以透過這個介面與程式互動;其次,程式能夠取得當前網頁的網址,並且根據當前頁面的內容進行處理;最後,程式會接收並顯示從模型回傳的數據,讓使用者可以清楚地看到模型的結果或建議。

圖十五:網頁UI

圖片來源:研究者從google網頁自行截圖

四、研究設備與器材

表三:系統之研究設備與器材

用途	採用工具

主要使用的程式語言，用於數據處理和模型開發。	Python
深度學習框架，用於構建和訓練情感分析模型。	TensorFlow
製作UI介面以及處理前後端交互。	HTML CSS JS
開發環境	replit

表三資料來源:研究者自行彙整分析

肆、實作成果

一、初始介面

當使用者按下啟用插件的按鈕後，系統將自動彈出初始介面，該介面會顯示一個開始按鈕。當使用者點擊該按鈕後，插件將啟動並開始分析評論內容。

圖十六:初始介面

圖片來源:研究者從google網頁自行截圖

二、分析中畫面

當使用者按下開始按鈕後，系統將進入分析階段，這個過程可能需要一段時間才能完成。為了提升使用者體驗，畫面將切換至專用的過渡介面，以明確顯示系統正在執行分析。同時，畫面上會顯示提示文字「分析中，請稍候...」，確保使用者能夠清楚了解當前進程，並耐心等待結果產生。

圖十七、十八:分析中畫面

圖片來源:研究者從google網頁自行截圖

三、分析結果顯示

當分析結束後，介面將顯示以下關鍵資訊，幫助使用者快速理解分析結果，並進行驗證或解讀:

書名:顯示系統所分析的書籍名稱，讓使用者確認目標書籍是否正確，避免誤將其他書籍的評論納入分析範圍。

評論數量:顯示系統分析的總評論數量，幫助使用者檢查數據範圍是否準確，例如是否遺漏部分評論，或包含過多無關評論。

平均分數:根據評論內容計算出的平均評分，範圍為 0 到 5 分，0 分代表最低評價，5 分代表最高評價。這個分數可讓使用者快速掌握該書的整體評價趨勢。

關鍵字:顯示系統從評論中提取之高頻率出現的詞彙，這些關鍵字能反映讀者關注的重點或書籍的主要特色，幫助使用者更直觀地了解其優缺點。

這些結果將以清晰的方式呈現，使使用者能夠直觀理解並據此進行後續決策或操作。

圖十九、二十:分析結果畫面

開始分析

書籍資訊

書名: 多賢的祕密貼文：我的真心，僅限本人閱讀

評論數: 38 則

平均評分: 4.5

熱門關鍵字: 青春、人際關係

圖片來源:研究者從google網頁自行截圖

伍、討論

一、對於專案進行**SWOT**分析

表四:系統之SWOT分析

Strengths 優勢	Weakness 劣勢
1.提供博客來使用者一種快速找好書的方式。 2.介面簡潔, 使用者容易上手。	1.安裝流程對部分用戶來說仍有困難。 2.此專案只能在博客來擷取評論並分析, 無法適用於任意網站。
Opportunities 機會	**Threats 威脅**
1.若有更合適的模型出現, 情緒辨識AI的效率與準確率皆可能進一步上升。 2.該模型可延伸擴充至其他網站, 如社群媒體、外送平台、購物網站等。	1.網站的反爬蟲程式不斷改進, 爬蟲功能可能因此無法使用或需要降低爬取速度。

表四資料來源:研究者自行彙整分析

二、結論

(一)在本次研究中, 我們所使用的模型在精確度方面尚未達到理想水準, 顯示參數設定與結構仍有進一步優化的空間。首先, 模型的層數與每層節點數可能需要調整, 以更符合數據特性並提升學習能力。此外, 增加高品質的訓練數據, 有助於改善模型的表現, 尤其是在現有數據不足以涵蓋目標任務的多樣性時。另一方面, 資料預處理階段採取更精細的停用詞處理策略將能有效去除無關資訊並突顯關鍵特徵。未來我們可進一步優化這些策略, 以提升模型的準確性與穩定性, 從而建立更高效、更精確的模型。

(二)在實際應用中, 我們發現許多網站為了保護數據安全, 設置了多種反爬蟲機制, 例如 IP 限制、JavaScript 驗證、訪問頻率檢測與 CAPTCHA 驗證等, 這些機制可能影響插件的正常運作, 並降低數據抓取的成功率。例如, 當網站偵測到異常訪問時, 可能會封鎖請求、要求額外驗證, 甚至完全限制數據存取。若請求過於頻繁, 伺服器也可能將插件的 IP 列入黑名單, 進一步影響後續操作。因此, 在插件設計時, 需要考量如何有效應對這些反爬蟲機制, 例如模擬真實使用者行為、適當降低請求頻率, 並針對不同網站的防禦機制制定相應策略, 以確保插件能穩定運行並成功擷取所需數據。

三、未來展望

(一)應對反爬蟲機制, 提升專案泛用性:我們將進一步探索技術手段, 提升插件的功能並降低被反爬蟲機制偵測的風險。例如:優化請求模擬方式、增加隨機化參數或模仿真實使用者行為, 以提高數據抓取的穩定性。此外, 也可考慮尋找合法且有效的替代方案, 例如:利用網站提供的 API 來獲取所需數據, 或與網站合作以取得正式的數據使用授權。這些方法不僅有助於

提升數據擷取的成功率，還能確保操作的合規性與可持續性，從而更有效地支持相關研究與應用發展。

（二）從更多層面去分析評論：為了進一步提升插件的功能性，除了數據收集外，我們還可以讓其具備評估與分析顧客滿意度的能力，涵蓋多個評價指標。例如：除了整體評分外，插件還可以細化滿意度評估，包括顧客對產品或服務的實用性、性價比、設計美觀性及售後服務等方面的評價。這可以透過分析顧客評論內容，自動提取關鍵詞並結合情感分析技術來實現。此外，為了更準確地反映顧客的真實感受，插件可以綜合考量評論的內容，進一步降低單一數據來源可能帶來的片面性。

陸、參考文獻

1. 楊宗恩 (2017)。以**LSTM**深度神經網路語言模型建構英文課程重點摘要。國立屏東科技大學資訊管理學系碩士班：碩士論文。
2. Wai-kin Wong, Wang-chun Woo. (2015, September 19) Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting, from chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://arxiv.org/pdf/1506.0421
3. 楊杰淮 (2022)。網絡爬蟲與反爬蟲相關研究。取自：https://hdl.handle.net/11296/69u8a8
4. Trupti V. Udapure, Ravindra D. Kale, Rajesh C. Dharmik(2014) Study of Web Crawler and its Different Types. IOSR Journal of Computer Engineering (IOSR-JCE) ,Volume 16, Issue 1, Ver. VI (Feb. 2014), PP 01-05
5. Graves, A. (2012, January 1). *Long Short-Term Memory*. Springer Nature Link. https://link.springer.com/chapter/10.1007/978-3-642-24797-2_4
6. houdt, G. V. (2020, May 13). *A Review on the Long Short-Term Memory Model*. Springer Nature Link. https://link.springer.com/article/10.1007/S10462-020-09838-1