

YouTube 留言多模組分析與視覺化系統之設計與實作

Design and Implementation of a Multi-Module YouTube Comment Analysis and Visualization System

指導教授：辛錫進 教授

學生：劉祐伸、洪立俊、陳聖明、邱梓奕、林祺翔

國立聯合大學 資訊工程學系

苗栗市南勢里聯大2號

hsin@nuu.edu.tw

[{U1224051, U1224024, U1224001, U1224048, U1113007}@365.nuu.edu.tw](#)

摘要

本專題設計並實作一套 YouTube 留言多模組分析與視覺化系統，旨在協助使用者快速掌握影片留言區的整體風向、熱門討論主題與觀眾回饋。系統透過 YouTube Data API 蒐集留言資料，並進行資料前處理後，分別執行摘要、關鍵詞、主題、情緒、批評回饋、時間軸熱點與影片內容脈絡等分析模組。最後，系統將各子分析結果整合為綜合分析，並以前端視覺化儀表板呈現與情溫度、情緒分布、熱門主題、關鍵詞與高能時間點等資訊。實作結果顯示，本系統能有效降低人工閱讀大量留言的負擔，並提供創作者與使用者更直觀的留言分析參考。

關鍵詞：YouTube 留言分析、自然語言處理、情緒分析、資料視覺化、主題分析

Abstract

This project designs and implements a multi-module YouTube comment analysis and visualization system to help users quickly understand the overall sentiment, popular discussion topics, and audience feedback in video comment sections. The system collects comment data through the YouTube Data API and performs data preprocessing before applying several analysis modules, including summarization, keyword extraction, topic analysis, sentiment analysis, criticism feedback analysis, timeline hotspot detection, and video content context analysis. The results of these modules are further integrated into a comprehensive analysis and presented through a web-based visualization dashboard, including public opinion scores, emotion distribution, popular topics, keywords, and frequently mentioned video timestamps. The implementation shows that the system can reduce the effort required to manually read large volumes of comments and provide creators and users with a more intuitive reference for understanding audience responses.

Keywords: YouTube Comment Analysis, Natural Language Processing, Sentiment Analysis, Data Visualization, Topic Analysis

第一章、緒論

隨著 YouTube 等影音平台的普及，影片留言區已成為觀眾表達意見、情緒反應與內容回饋的

重要來源。對創作者而言，留言能反映觀眾對影片主題、內容安排與觀看體驗的看法；對一般使用者而言，留言區也能作為理解影片討論熱度與觀眾觀點的參考。然而，當影片留言數量龐大時，若僅依靠人工閱讀，往往難以快速掌握整體風向、熱門討論主題、觀眾情緒與具代表性的回饋內容。

為了解決上述問題，本專題設計並實作一套 YouTube 留言多模組分析與視覺化系統。系統透過 YouTube Data API 蒐集留言資料^[1]，並結合自然語言處理與大型語言模型，進行摘要、關鍵詞、主題、情緒、批評回饋、時間軸熱點與影片內容脈絡等分析。分析結果進一步整合為綜合分析，並透過前端視覺化儀表板呈現與情溫度、熱門主題、情緒分布、關鍵詞與高能時間點等資訊。

本專題期望透過自動化分析與視覺化呈現，降低人工閱讀大量留言的負擔，使使用者能更快速理解影片留言區的重點內容，並提供創作者作為整理觀眾回饋與改善影片內容的參考。

第二章、相關技術

本專題主要結合 YouTube 留言資料取得、自然語言處理、大型語言模型、本地推論與資料視覺化等技術，建立一套可自動蒐集、分析並呈現 YouTube 留言結果的系統。

2.1 留言資料取得

YouTube Data API v3 是本系統取得影片留言資料的主要來源。系統會從使用者輸入的 YouTube 影片連結中解析影片 ID，並透過 API 取得指定影片的留言內容、留言按讚數與影片基本資訊。取得的留言資料會作為後續摘要、關鍵詞、主題、情緒、批評與時間軸分析的基礎。

由於 YouTube 留言多為非結構化文字，內容可能包含表情符號、網址、換行、多語言文字與影片時間戳記，因此取得資料後仍需進行資料清理、欄位整理、語言判斷與時間戳辨識，使後續分析能以一致格式處理資料，並降低雜訊對分析結果造成的影響。

2.2 自然語言處理與留言分析

自然語言處理技術可用於從大量文字資料中萃取具代表性的資訊。本系統依照不同分析目的設計多個留言分析功能，包含摘要分析、關鍵詞

分析、主題分析、情緒分析、批評回饋分析與時間軸分析。各分析模組會依據任務特性採用不同模型或方法，使系統能從多個角度理解 YouTube 留言內容。

2.2.1 摘要分析模型

在摘要分析方面，本系統曾使用 BERTSUM 作為摘要模型之訓練與實驗基礎。BERTSUM 是以 BERT 語意表示為基礎的摘要方法，能夠擷取文本中的重要句子或重點內容，適合用於將大量留言濃縮為較精簡的摘要結果。在本專題中，摘要分析主要用於整理留言區的整體討論方向，使使用者不需逐一閱讀留言即可快速掌握主要內容。

2.2.2 關鍵詞與主題分析模型

在語意相似度與主題分析方面，本系統使用 MiniLM 作為語意向量模型之訓練與應用基礎。MiniLM 能將文字轉換為向量表示，使系統可以比較不同留言之間的語意相似度。透過語意向量，系統可將內容相近的留言歸納為相同主題群，進一步整理出熱門討論焦點、主題關鍵詞與代表留言。相較於單純依靠詞頻統計，語意向量能更好地處理用詞不同但意思相近的留言內容。

在關鍵詞分析方面，本系統使用 KeyBERT 相關方法進行代表詞彙抽取。KeyBERT 會根據文本語意表示計算詞彙與整體內容之間的相關性，並選出較具代表性的關鍵詞。此方法可用於整理留言中反覆出現或具有討論價值的詞彙，並作為熱門關鍵詞、留言標籤與主題輔助說明。系統亦針對中文與英文留言分別進行處理，以提升不同語言留言的關鍵詞抽取效果。

2.2.3 情緒分析方法

在情緒分析方面，系統會判斷留言區的整體風向與情緒組成。除了整理正向、中性與負向比例外，也進一步呈現喜悅、憤怒、悲傷、厭惡、驚訝與恐懼等細部情緒類型，並以前端圖表呈現情緒分布。此分析可協助使用者了解留言區是偏向支持、批評、疑惑或情緒分歧，並作為綜合分析中判斷輿情溫度的重要依據。

2.2.4 批評與時間軸分析方法

在批評回饋分析方面，系統會整理留言中的不滿原因、批評內容與改善建議。由於批評與建議往往需要根據上下文判斷，若僅使用關鍵詞命中容易造成誤判，因此本系統會搭配大型語言模型輔助語意歸納，使批評分析能更接近留言實際語意，而非單純依靠字面詞彙分類。

在時間軸分析方面，系統主要透過時間戳記擷取與統計方式進行處理。當留言中出現如「1:23」、「12:45」等影片時間格式時，系統會擷取該時間點並統計其出現頻率，用以判斷觀眾較常提及的影片片段。此方法可找出留言區中的高討論度時間點，並搭配代表留言呈現觀眾關注原因。

綜合而言，本系統結合 BERTSUM、MiniLM

、KeyBERT、情緒分類方法、時間戳規則擷取與大型語言模型輔助分析等技術，分別處理摘要、語意分群、關鍵詞抽取、情緒判斷、批評整理與時間軸熱點偵測等任務。透過多模型與多模組的設計，系統能避免僅依靠單一分析方法判斷留言區狀態，並從不同面向提供較完整的留言分析結果。

2.3 大型語言模型與本地推論

大型語言模型具備文字理解、語意歸納與自然語言生成能力，適合用於處理較複雜的文字整理任務。本系統使用本地大型語言模型作為輔助推論工具，並透過 Ollama 執行本地模型推論，例如 Gemma 系列模型。相較於僅依靠關鍵詞比對或規則式分類，大型語言模型能根據上下文理解留言含義，較適合用於整理批評原因、觀眾建議與綜合分析等需要語意判斷的任務。

在本系統中，大型語言模型主要用於摘要整理、批評回饋歸納、影片內容脈絡分析與綜合分析。批評回饋分析會透過模型整理留言中的不滿原因與改善建議，降低單純關鍵詞命中造成的誤判。影片內容脈絡分析則會依據字幕或文字內容整理影片章節與摘要。綜合分析會進一步整合摘要、主題、情緒、關鍵詞、批評、時間軸與影片內容脈絡等結果，產生 AI 智慧快報、創作者行動建議與觀眾觀看提示。

由於本系統採用本地推論方式，分析流程可在本機環境中完成，降低對外部生成式服務的依賴。然而，大型語言模型推論需要較多運算資源，因此分析時間會受到模型大小、輸入文字長度與硬體規格影響。

2.4 資料視覺化與互動式介面

為提升分析結果的可讀性，本系統使用 Next.js 與 React 建立 Web 前端介面，提供影片連結輸入、分析模式選擇、任務狀態顯示、分析結果呈現與歷史紀錄查詢等功能。前端介面以 Tailwind CSS 設計深色系儀表板風格，並透過 Recharts 與自製圖表元件呈現分析結果。

視覺化內容包含輿情溫度計、熱門主題分布、情緒圖譜、關鍵詞圖表、時間軸熱點與影片內容章節等。主分析頁面負責提供整體總覽，子分析頁面則提供更細部的模組結果，使使用者能由整體趨勢進一步查看特定分析內容。

此外，本專題亦實作 Discord Bot 作為開發與測試階段的互動式介面。開發者可透過 Discord 指令觸發分析流程，並以 Embed 格式快速檢視各分析模組的輸出內容，以利在 Web 前端完成前驗證分析結果與輸出格式。

2.5 資料儲存與容器化環境

本系統使用 Prisma 與 Supabase 管理分析紀錄資料。當分析完成後，系統可將分析結果、影片資訊、分析模式與結果內容儲存至資料庫，使使用者能於歷史紀錄頁面查詢先前分析結果。此設計可提升系統的重複使用性，避免每次查看結果

都必須重新執行分析流程。

在執行環境方面，本系統使用 Docker 管理系統環境與相關依賴，包含前端、後端、分析流程與推論環境等。透過容器化方式，可降低不同開發環境之間的版本差異，並提升部署與重現系統環境的便利性。

第三章、系統設計

3.1 系統架構

本系統採用前後端分離架構，主要包含前端視覺化介面、後端分析服務、留言分析模組與資料儲存模組。使用者可於前端輸入 YouTube 影片連結並選擇分析模式，前端會將請求送至後端服務，由後端負責建立分析任務、取得留言資料、執行各項分析模組，最後將分析結果回傳至前端進行視覺化呈現。此外，系統亦設計歷史紀錄功能，用於保存分析結果，方便後續查詢。

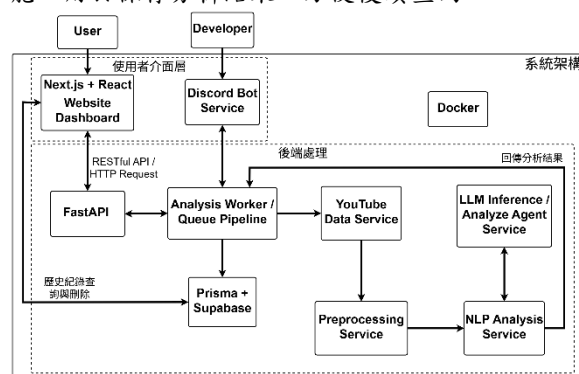


圖 1 系統服務架構圖

3.2 資料處理流程

使用者輸入影片連結後，系統會先解析影片 ID，並由 Analysis Worker 呼叫 YouTube Data Service 取得留言、影片資訊與字幕資料。取得資料後，Preprocessing Service 會進行留言清理、欄位整理、語言判斷與時間戳辨識，並將處理後的資料送入各分析模組。各子分析完成後，結果會回傳至 Worker 進行彙整，最後提供前端顯示或儲存至資料庫作為歷史紀錄。

3.3 分析模組設計

本系統採用多模組分析設計，將留言內容依不同分析目的拆分處理，如表 1 所示。各子分析分別負責摘要、關鍵詞、主題、情緒、批評、時間軸與影片內容脈絡等任務，最後由綜合分析整合各模組結果，產生整體風向、智慧快報、創作者建議與觀眾觀看提示。

模組	設計目的	輸出內容
摘要分析	濃縮大量留言內容	留言重點摘要
關鍵詞分析	找出高頻與代表詞彙	熱門關鍵詞、標籤
主題分析	歸納留言討論焦點	主題群、代表留言
情緒分析	判斷留言區風向	輿情分數、情緒分布

批評分析	整理觀眾不滿與建議	批評原因、改善建議
時間軸分析	找出留言提及的影片片段	高能時間點、代表留言
影片內容脈絡分析	補充影片本身脈絡	影片章節、內容摘要
綜合分析	整合各子分析結果	智慧快報、創作者建議

表 1 分析模組表

3.4 視覺化設計

為提升分析結果的可讀性，本系統以前端儀表板呈現各項分析結果。主分析頁面負責提供整體總覽，包含輿情溫度計、熱門討論焦點、時間軸熱點、情緒分布、批評與改善訊號、熱門關鍵詞與影片內容脈絡等區塊。各子分析頁面則提供更詳細的模組結果，例如主題分析頁顯示各主題群的關鍵詞與代表留言，情緒分析頁顯示情緒比例與代表留言，時間軸分析頁則呈現留言提及的影片時間點與高能片段。

此設計將主分析定位為快速總覽，子分析則作為詳細檢視頁面，使使用者能依需求由整體結果進一步查看特定分析內容。

第四章、系統實作與測試結果

4.1 開發環境與工具

本系統以前後端分離方式進行實作，前端負責使用者操作介面與分析結果視覺化，後端負責建立分析任務、執行留言蒐集與多模組分析流程。系統亦透過資料庫保存分析紀錄，並使用 Docker 管理系統執行環境^[8]，以提升部署與開發環境的一致性。主要使用技術如表 2 所示。

類別	使用技術	說明
前端介面	Next.js ^[3] 、React ^[4]	建立影片輸入頁、分析結果頁與歷史紀錄頁
視覺設計	Tailwind CSS	設計響應式介面與深色系儀表板
圖表呈現	Recharts ^[7]	呈現輿情溫度、主題分布、情緒圖譜與時間軸熱點
後端服務	FastAPI	提供分析任務建立、狀態查詢與結果回傳 API
任務處理	Analysis Queue / Worker	管理分析任務與執行多模組分析流程
資料來源	YouTube Data API v3	取得指定影片的留言、影片資訊與字幕資料
文字分析	NLP 模型、LLM	執行摘要、關鍵詞、主題、情緒、批評與綜合分析
互動測試	Discord Bot	透過指令觸發分析並以 Embed 回傳結果
資料	Prisma ^[5] 、	儲存分析紀錄並提供

儲存	Supabase ^[6]	歷史查詢
執行環境	Docker	管理系統執行環境與 依賴套件

表 2 開發環境與工具

4.2 系統功能實作

本系統實作 Web 前端與 Discord Bot 兩種互動方式。Web 前端作為主要操作介面，使用者可輸入 YouTube 影片連結並選擇分析模式，系統建立分析任務後會持續查詢任務狀態，並於完成後顯示分析結果。Discord Bot 則作為開發與測試階段的輔助介面，開發者可透過指令觸發分析流程，並以 Embed 格式快速檢視各模組輸出。後端部分透過 FastAPI 提供任務建立、狀態查詢、結果取得與取消任務等功能，並由 Analysis Worker 執行完整分析流程。主要功能實作如表 3 所示。

功能	實作內容
綜合分析	整合各子分析結果，產生 AI 智慧快報、創作者建議與觀眾提示
摘要分析	將大量留言整理為重點摘要
關鍵詞分析	擷取留言中的熱門關鍵詞與代表標籤
主題分析	將留言分群並整理熱門討論焦點與代表留言
情緒分析	計算輿情分數、正中負比例與情緒分布
批評分析	歸納觀眾不滿原因、批評內容與改善建議
時間軸分析	從含時間戳留言中找出高討論度片段
影片內容分析	透過字幕整理影片章節與內容脈絡
歷史紀錄	將分析結果儲存至資料庫，供前端查詢
Discord 測試	透過指令與 Embed 快速驗證分析結果

表 3 功能實作表

4.3 系統畫面展示

系統首頁提供影片連結輸入與分析模式選擇，如圖 2 所示。分析完成後，綜合分析頁面會以 Dashboard 方式呈現整體分析結果，如圖 3 所示。子分析頁面則針對特定分析項目提供更細部的結果，例如圖 6 為主題分析子頁，可顯示主題關鍵詞與代表留言。圖 8 為歷史紀錄頁面，可供使用者查詢過去分析結果。



圖 2 系統首頁與分析模式選擇



圖 3 綜合分析 Dashboard-1



圖 4 綜合分析 Dashboard-2

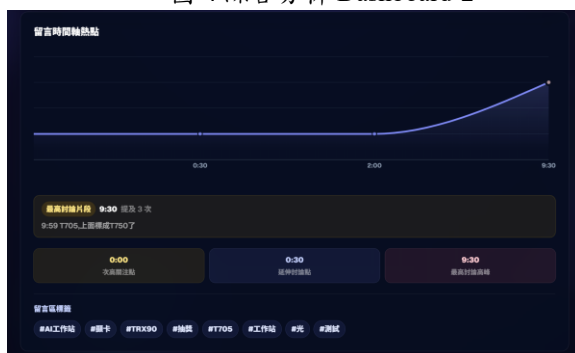


圖 5 綜合分析 Dashboard-3



圖 6 主題分析子頁-1



圖 7 主題分析子頁-2



圖 8 歷史紀錄頁面-1

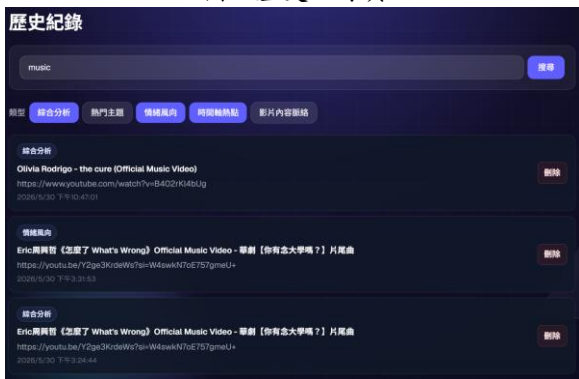


圖 9 歷史紀錄頁面-2

4.4 測試結果與效能分析

為驗證系統功能與分析效率，本專題選取不同長度及留言型態的 YouTube 影片進行綜合分析測試，包含短影片、中長影片與長影片。測試項目以系統的綜合分析流程為主，該流程會整合留言蒐集、各子分析模組、影片內容脈絡分析、綜合分析與前端視覺化結果呈現。由於時間軸分析會受到留言中是否包含時間戳記影響，影片內容分析則會受到字幕可取得性與影片長度影響，因此不同影片的分析時間與主要耗時項目會有所差異。

項目	規格
CPU	Intel Core i7-13700
GPU	NVIDIA GeForce RTX 5070 Ti
RAM	32GB
OS	Windows 11

表 3 測試環境規格

影片類型	影片長度	分析前三瓶頸耗時	綜合分析總耗時
恐怖短片	約 3 分鐘	影片內容脈絡：58.8 秒 批評回饋：27.9 秒 綜合整合分析：25.8 秒	約 139.8 秒
評判	約 20 分鐘	影片內容脈絡：93.9 秒 綜合整合分析：19.9 秒 批評回饋：19.3 秒	約 167 秒
紀錄片專題討論	約 1 小時 15 分	影片內容脈絡：210.2 秒 批評回饋：27.3 秒 綜合整合分析：23.7 秒	約 271.6 秒
純音樂	約 1 小時	批評回饋：40.1 秒 綜合整合分析：30.8 秒 影片內容脈絡：19.2 秒	約 134.2 秒

表 4 效能測試結果

由測試結果可知，三組測試樣本之綜合分析總耗時介於約 140 秒至 270 秒之間。測試結果顯示，分析時間不僅受到影片長度影響，也會受到留言數量、字幕長度、留言內容複雜度與 LLM 推論速度影響。例如，約 20 分鐘影片的耗時高於約 1 小時純音樂影片，表示系統總耗時並非單純與影片長度成正比，而是與實際需要處理與生成的文字內容量有關。

從主要耗時項目可觀察到，影片內容脈絡分析、批評回饋分析與綜合分析較常成為耗時瓶頸。原因在於這些分析較依賴 LLM 進行語意整理與文字生成，例如影片內容脈絡分析需根據字幕產生章節與摘要，批評回饋分析需歸納留言中的不滿原因與改善建議，綜合分析則需整合各子分析結果並產生最終說明。由於本系統採用本地 LLM 推論，文字輸出的 token 生成速度會直接影響分析時間，因此當輸入內容較長或輸出文字較多時，整體耗時也會增加。整體而言，本系統可於約 2 至 5 分鐘內完成不同類型影片的綜合分析，並回傳可視化分析結果。

第五章、結論與未來展望

本專題完成一套 YouTube 留言多模組分析與視覺化系統，能自動取得影片留言並進行摘要、關鍵詞、主題、情緒、批評、時間軸與影片內容脈絡分析。系統透過綜合分析整合各子分析結果，並以前端儀表板呈現與情溫度、熱門討論焦點、情緒分布、關鍵詞與高能時間點等資訊，協助使用者快速掌握留言區重點。

測試結果顯示，本系統可完成不同類型影片的綜合分析流程，並產生可視化結果。然而，分析時間仍會受到留言數量、字幕長度、影片內容複雜度與 LLM 推論時間影響；時間軸分析亦依賴留言中是否包含時間戳，影片內容分析則受字幕可取得性影響。未來可朝推論效率優化、多語言與諷刺語氣判斷、多影片比較、雲端部署與權限管理等方向改進，以提升系統的實用性與擴充性。

參考文獻

- [1] Google Developers, “YouTube Data API - Comments: list,” Google for Developers.
<https://developers.google.com/youtube/v3/docs/comments/list>
- [2] FastAPI, “FastAPI Documentation.”
<https://fastapi.tiangolo.com/>
- [3] Vercel, “Next.js Documentation.”
<https://nextjs.org/docs>
- [4] React, “React Reference.”
<https://react.dev/reference/react>
- [5] Prisma, “Prisma Documentation.”
<https://www.prisma.io/docs>
- [6] Supabase, “Supabase Documentation.”
<https://supabase.com/docs>
- [7] Recharts, “Recharts Documentation.”
<https://recharts.org/>
- [8] Docker Inc., “Docker Documentation.”
<https://docs.docker.com/>