基於階層式 mBERT 與可解釋性 AI 之假新聞即時辨識 Chrome 擴充套件

A Real-Time Fake News Detection Chrome Extension Based on Hierarchical mBERT and Explainable AI

指導教師：温育瑋 博士

學生：范文爾 毛彥霖 蘇景賢

國立聯合大學 資訊工程學系

苗栗市南勢里聯大2號

摘要

近年來，數位媒體與即時通訊軟體普及，使假新聞與錯誤資訊更容易快速擴散。傳統事實查核多屬事後查證，且需使用者離開閱讀情境，難以及時降低誤信風險。因此，本研究設計並實作一套低延遲假新聞辨識 Chrome Extension，讓使用者在瀏覽新聞或可疑連結時，即時取得判定結果與輔助說明。

系統以前端擷取 URL 與頁面資訊，後端整合 Firecrawl、MongoDB、文本前處理與階層式 mBERT 分類模型，完成網址查詢、未知網址擷取、文本分析與推論，並透過可解釋性 AI 標示關鍵句段與可疑詞彙。實作結果顯示，本系統可降低查核門檻，並提升使用者對可疑資訊的警覺。

關鍵詞：假新聞辨識、Chrome Extension、階層式 mBERT、可解釋性 AI、跨語言學習

Abstract

In recent years, digital media and instant messaging platforms have accelerated the spread of fake news and misinformation. Traditional fact-checking is often delayed and requires users to leave the browsing page, limiting its immediacy. Therefore, this study implements a Chrome Extension for low-latency fake news detection during web browsing.

The system collects webpage URLs through the extension, while the backend integrates Firecrawl, MongoDB, text processing, and a hierarchical mBERT model for URL querying, unknown URL crawling, and classification. It also highlights key sentences and suspicious terms to support explainable results. The system embeds fake news detection into daily browsing, reduces the barrier to fact-checking, and improves user awareness of suspicious information.

Keywords：Fake News Detection, Chrome Extension, Hierarchicalm BERT, Explainable AI,Cross-lingual Learning

一、前言

近年來，社群媒體與即時通訊軟體的普及，使新聞與網路資訊能在短時間內快速傳播，但也提高假新聞與錯誤資訊擴散的風險。目前常見的事實查核方式多仰賴人工查證或事後查核平台，雖能提供較完整的查核結果，但使用者通常需離開原本的閱讀頁面另行搜尋資料，流程較為耗時，且不易在資訊接收當下即時降低誤信與轉傳風險。

因此，本專題設計並實作一套假新聞辨識系統，透過 Chrome Extension 將查核功能整合至使用者的日常瀏覽情境中。系統後端結合 Firecrawl 網頁擷取、MongoDB 資料庫與階層式 mBERT 分類模型，可自動取得使用者瀏覽之 URL，並依據資料庫既有結果或即時爬取內容進行推論分析。系統最終提供真偽判定、信心分數與關鍵句段標示，協助使用者在閱讀過程中快速評估資訊可信度，降低假新聞造成的誤導與傳播風險。

二、文獻探討

2.1 長文本分類與階層式模型

新聞文章通常包含標題、導言、事件背景與後續補充資訊，文本長度可能超過 BERT 類模型的輸入限制。若採用截斷策略，模型可能忽略後段的重要語境或矛盾資訊。為改善此問題，Yang et al. [1] 提出 Hierarchical Attention Networks，透過詞、句子與文件層級的階層結構聚合語意資訊。Pappagari et al. [2] 則提出階層式 Transformer，將長文本切分為多個片段後進行高層語意整合，以降低輸入長度限制對分類結果的影響。

上述研究顯示，長文本分類應避免僅依賴單一片段或簡單截斷。因此，本研究延續階層式長文本建模概念，將新聞文章切分為多個重疊 chunks，分別輸入 mBERT 萃取區塊表徵，再透過跨區塊 self-attention 建立文章層級語意關係，以更完整地處理長篇新聞文本。

2.2 可解釋性分析與信心校準

深度學習模型雖能在文本分類任務中取得良好表現，但其判斷依據通常不易理解。於假新聞偵測情境中，若系統僅輸出分類結果而缺乏依據說明，可能降低使用者信任。因此，本研究設計區塊層級與句段層級之可解釋性分析方法，利用跨區塊 attention matrix 的 column sum 找出 key chunk，並結合 masked max pooling 回溯結果與分類器權重，擷取具影響力句段作為模型關注區域的近似解釋。

為評估內部訊號解釋方法，本研究採用 Ribeiro et al. [4] 提出之 LIME 作為對照。LIME 透過局部擾動觀察模型輸出變化，並訓練可解釋的局部代理模型，以估計不同詞彙或句段對預測結果的貢獻。由於 LIME 具較高忠實度但需反覆推論，因此本研究以其作為基準，比較內部訊號方法在解釋忠實度與推論延遲上的差異。

此外，softmax 輸出機率不一定能準確反映模型可靠度。Guo et al. [3] 指出，現代神經網路即使具備高準確率，仍可能產生過度自信或信心不足的校準問題。由於假新聞即時辨識系統需將信心分數對應至前端風險提示等級，本研究於模型訓練完成後加入 Temperature Scaling，透過驗證集 NLL 尋找最佳溫度參數，並以 ECE 評估校準前後的信心可靠度。

三、方法與系統設計

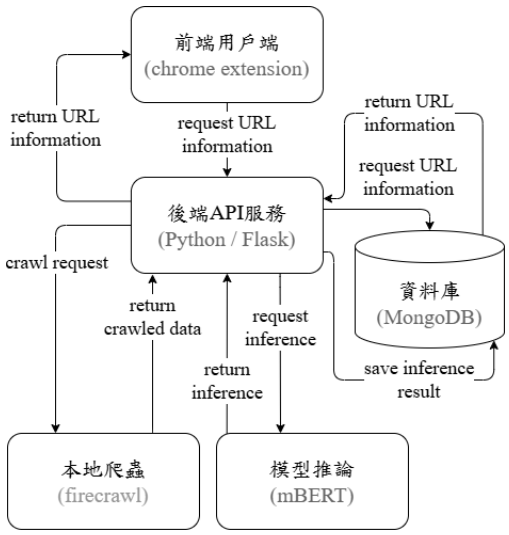
3.1 系統架構

本系統前端設計以非侵入式設計為目的，讓使用者可以在不用點進連結，即可知曉該連結是否可能為假新聞。前端擴充功能主要負責使用者介面互動、目前網址偵測、快取控制與網頁元素高亮標記。後端伺服器基於 Python Flask 框架開發，採用非同步執行緒與事件監控設計，整合機器學習模型推論與本地 Firecrawl。資料庫方面系統採用 MongoDB 作為儲存引擎，資料包含前端需要的資訊、以及未知資料處理過程中的處理狀態。為應對現代網站廣泛採用動態加載與反向代理服務，爬蟲系統採用了本地部署的 Firecrawl 服務作為核心。透過 crontab 每8小時自動執行，主動式爬取預設新聞網站文章並推論後，存入資料庫；若是在前端使用時遇到未收錄於資料庫的 URL，則會即時爬取並推論，被動式擴充資料。

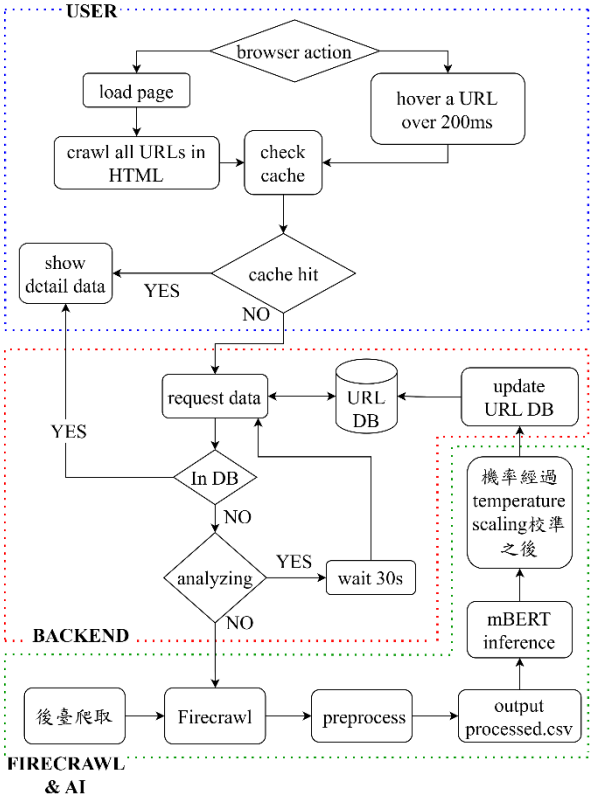
本系統以後端為中樞，除了接收與回傳來自前端的請求以外，也作為資料中繼站的角色。當偵測到未知資料，會由後端呼叫爬蟲爬取，並以同樣的方式呼叫前處理、推論模型，除此以外，所有需要存資料庫的動作皆由後端完成。

系統流程以使用者為流程起始點，當使用者載入頁面成功 hover 一個 URL 時，前端會向後端請求未知 URL 的基本資訊，後端會到資料庫中查找該 URL 的資料，如果該 URL 沒有存在資料庫中，後端隨即呼叫爬蟲爬取未知的 URL 內文，並開啟未知 URL 的處理流程。

爬蟲不會在瞬間完成爬取，因此未知 URL 處理須採用非同步執行來確保前端可即時回應，處理流程尚未完成時，前端會回應 processing 來讓使用者知道資料正在處理中。爬取成功後，會將工作編號、URL、處理時間、純文字內容、處理狀態等資訊存成 csv 檔，送入前處理並推論。推論後會將資料傳回後端並存進資料庫。



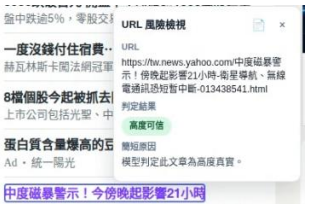
圖一、系統架構圖



圖二、系統流程圖

3.2 前端系統功能說明

提示框:使用者 hover 一個 URL，前端會顯示提示框，並向後端請求提示框所需要的資訊，如果使用者只是滑過某一個 URL 不會向後端請求資訊。提示框中包含 URL 的基本資訊與互動按鈕。



圖三、前端提示框

側邊欄:當使用者按下取得詳細資訊，前端會在旁邊顯示側邊欄，除基本資訊外側邊欄中還會顯示文章內文，並將包含可疑句段區塊渲染。

圖四、前端側邊欄及渲染結果

3.3 即時與背景爬取模組設計

本研究之爬取模組包含即時爬取與背景定時爬取。即時爬取用於處理資料庫中尚未收錄的未知 URL，透過 User-Agent 輪換、請求標頭設定、Referer 偽裝、渲染等待、Proxy 輪換與重試機制，降低 CDN 與反爬蟲防護造成的擷取失敗。背景定時爬取則是選擇採用 Frontier BFS 架構並整合 Firecrawl，定期從預設新聞網站的分類頁與文章連結中發現並蒐集新聞內容，並依網站防護特性調整擷取策略，避免取得空白頁面或不完整文章，以提升資料收集的穩定性與完整性。

3.4 階層式滑動視窗 mBERT 長文本分類與機率校準架構

由於 mBERT 單次輸入長度限制為 512 個子詞 tokens，若直接將長篇新聞截斷後輸入模型，模型僅能接收文章前段內容，可能導致後段的重要事件背景、補充說明、語意轉折或矛盾資訊被捨棄。此問題在新聞文本分類任務中尤其明顯，因為新聞文章的真偽判斷往往不只依賴標題或開頭段落，而可能需要綜合全文脈絡進行判斷。為降低長文本截斷所造成的資訊流失，本研究參考 HAN 與長文本 Transformer 分段建模概念，設計階層式滑動視窗分類架構，如圖五所示。

首先，系統以滑動視窗將新聞文本切分為多個重疊 chunks，每個 chunk 經 tokenizer 處理後不超過 512 tokens，並設定 stride 為 128 tokens，以保留相鄰區塊之間的上下文連續性。透過重疊切分，可降低重要句段剛好落在切分邊界而造成語意不完整的問題。接著，各 chunk 分別輸入 mBERT 進行語義特徵萃取，取得 token-level hidden states。

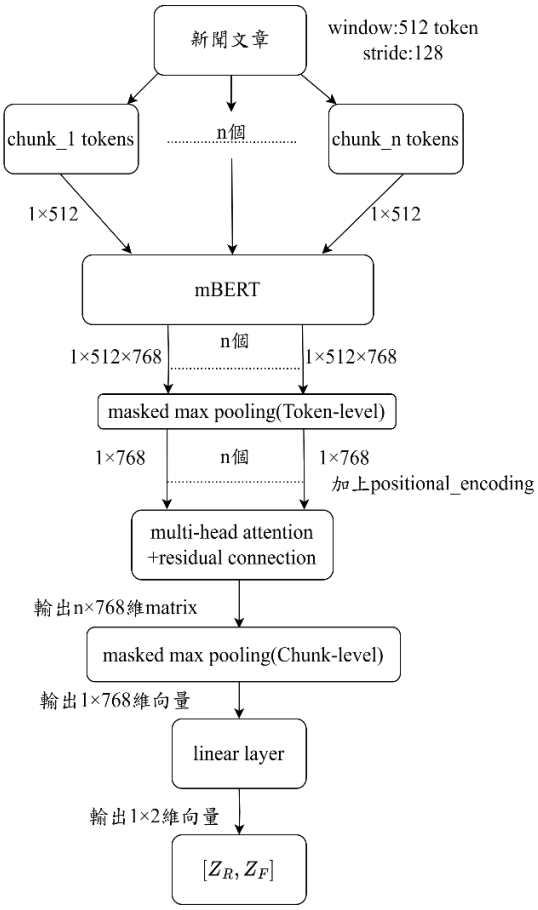
為將不同長度的 token 序列轉換為固定長度的區塊表示，本研究採用 masked max pooling 聚合 token-level 特徵，並透過 attention mask 排除 padding token 的影響，使 chunk 向量能更穩定地反映該區塊中較具代表性的語義訊號。

在取得各 chunk 的語義向量後，本研究進一步於 chunk-level 序列加入位置編碼，使模型能保留不同區塊在原文中的相對順序資訊。接著，透過多頭自注意力機制學習不同 chunks 之間的全域語義關係，使模型不僅能分別理解各區塊內容，也能建模跨段落之間的關聯、呼應與潛在矛盾。注意力輸出經 residual connection、LayerNorm 與 dropout 整合後，再透過 masked max pooling 聚合為文件級向量，以形成整篇新聞文章的整體語義表示。最後，文件級向量經全連接層輸出二元分類 logits $[Z_R, Z_F]$，分別對應真實新聞與虛假新聞，並以交叉熵損失進行端對端微調。

此外，考量本系統部署於即時假新聞辨識情境中，模型輸出的信心分數將直接影響前端風險

提示的呈現，因此僅有分類正確率仍不足以完整反映系統可靠度。為提升模型信心分數與實際正確率之間的一致性，本研究於模型訓練完成後進行 Temperature Scaling。

具體而言，透過驗證集最小化負對數概似損失（Negative Log-Likelihood, NLL）取得最佳溫度參數 T^*，再將原始 logits 調整為 Z/T^* 後計算 softmax 機率。此方法不改變模型原始分類結果，只調整輸出機率分布，使信心分數更適合作為前端風險等級與使用者提示依據，並進一步使用 ECE 評估校準前後的信心可靠度。



圖五、模型架構圖

四、實驗結果

4.1 對照模型設計與整體測試結果

本研究針對三種模型分別使用五組 random seed 進行重複訓練。各組實驗皆固定相同訓練集、驗證集與測試集切分，僅改變模型初始化與訓練過程中的隨機因素，以比較不同架構在相同資料條件下的平均效能與穩定性。最終結果以各 seed 於 200 筆測試集上之準確率平均值與標準差呈現。

表一 、不同模型之平均 Accuracy 與 Macro-F1 表現

模型架構	Accuracy Avg ± Std	Macro-F1 Avg ± Std
single mBERT	92.30% ±1.44%	92.29% ±1.44%
sliding-window mBERT	90.60% ±1.95%	90.58% ±1.96%
hierarchical mBERT	**93.30% ±1.60%**	**93.30% ±1.61%**

由表一測試結果可知，Hierarchical mBERT 在五組 random seed 下取得最高平均 Accuracy 與 Macro-F1。此結果客觀顯示，在本實驗的資料切分與訓練設定下，透過跨區塊注意力機制進行文件級語義整合的 hierarchical mBERT，於整體測試集上具備最佳的分類表現。

4.2 模型對外部測試集的泛化能力分析

由於本研究所提出之 hierarchical mBERT 主要用於處理 mBERT 輸入長度限制所造成的長文本資訊流失問題，因此進一步使用外部測試集評估不同模型架構在不同文本長度下的泛化能力。外部測試集依 token 長度分為三組：隨機抽取資料、513–1280 tokens 之中長文本，以及 1281 tokens 以上之超長文本各 600 筆。三種模型皆使用相同測試資料，並以 Accuracy 與 Macro-F1 作為評估指標。

表二、不同文本長度下之模型 Accuracy 表現

模型架構	隨機	中長文本	超長文本
hierarchical mBERT	92.28% ±1.37%	**94.30% ±1.03%**	90.93% ±1.28%
single mBERT	92.62% ±1.11%	93.70% ±2.13%	89.57% ±1.72%
sliding-window mBERT	**93.17% ±2.47%**	94.17% ±1.76%	**92.83% ±2.77%**

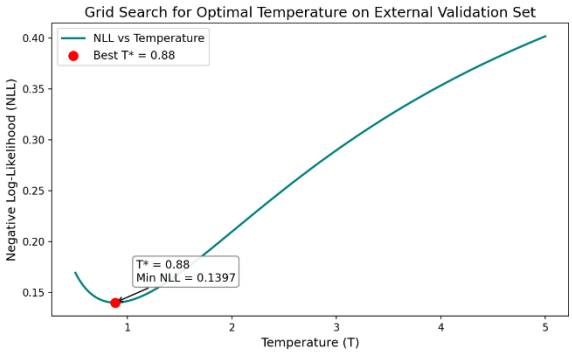
表三、不同文本長度下之模型 Macro-F1表現

模型架構	隨機	中長文本	超長文本
hierarchical mBERT	92.25% ±1.39%	**94.30% ±1.03%**	90.93% ±1.29%
single mBERT	92.59% ±1.10%	93.69% ±2.15%	89.54% ±1.74%
sliding-window mBERT	**93.16% ±2.46%**	94.16% ±1.77%	**92.82% ±2.80%**

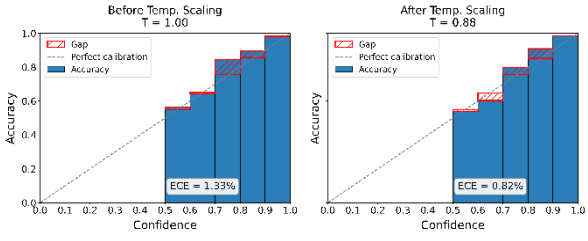
由表二可知，hierarchical mBERT 於中長文區間取得最高 Accuracy（94.30%），略高於 sliding-window mBERT（94.17%）與 single mBERT（93.70%），顯示跨區塊 self-attention 有助於整合 chunk 間語意資訊。雖然 sliding-window mBERT 在 Random 與超長文區間表現較佳，可能受 logits average 的平滑效果影響，但本研究仍採用 hierarchical mBERT 作為主要模型，原因在於其同時具備中長文本分類效能與後續 key chunk、影響力句段擷取所需的可解釋性結構。

4.3 Temperature Scaling 與機率校準實驗結果

由於最終系統部署時僅對外部新聞進行文字前處理與模型判斷，並不會逐篇進行 Gemma4 重寫，因此本研究採用只經前處理之外部資料作為校準依據。具體而言，隨機抽取 2000 筆外部前處理資料作為 calibration set，透過最小化 NLL 取得最佳溫度參數 = 0.88（圖六），再以未參與校準的隨機 2000 筆外部資料作為 test set 進行 ECE 評估。結果顯示，校準後 ECE 由 1.33% 降低至 0.82%（圖七），表示 Temperature Scaling 可改善模型於外部新聞資料上的信心可靠度。



圖六、Temperature Scaling 於校準集之 NLL 曲線



圖七、Temperature Scaling 前後之 ECE 比較

4.4 模型可解釋性的實驗結果

為評估解釋結果是否反映模型實際預測依據，本研究採用 comprehensiveness 作為忠實度指標，模型對原文之預測類別信心為 p_{target}，遮蔽句段後之預測信心為 p'_{target}，則信心下降幅度定義為：

$$\Delta p = p_{target} - p'_{target}$$

其中，Δp 越大，表示被遮蔽之句段對模型原始預測影響越大。本研究比較三種方法。第一為本研究提出之內部訊號方法，該方法根據 key chunk、masked max pooling 回溯結果與分類器權重選出候選句段。第二為 LIME，其透過擾動取樣估計各句段對模型預測之貢獻。第三為隨機基線，亦即從相同句段池中隨機抽取相同數量之句段，並重複 5 次取平均，以降低隨機抽樣造成的變異。實驗結果如表四所示，本研究提出之內部訊號方法平均選句時間僅 9.5 毫秒，明顯低於 LIME 的 606.8 毫秒，顯示其具備低延遲優勢，較符合瀏覽器擴充功能即時提示需求。

然而，在忠實度方面，本方法之彙總 Δp 為 0.0575，僅略高於隨機基線的 0.0474，且逐篇樣本中僅有 41.5% 高於隨機基線。相較之下，LIME 的彙總 Δp 達 0.1105，且有 86.5% 樣本高於隨機基線，顯示其解釋結果較穩定。綜合而言，本方法雖適合作為低延遲前端提示，但僅依賴 attention 權重或內部表徵訊號仍不足以保證解釋忠實度，需搭配遮蔽式或擾動式方法進行驗證。

表四、不同解釋方法於 comprehensiveness 指標與選句時間之比較

方法	平均 Δp	高於隨機比例	平均選句時間
本方法	0.0575	41.5%	**9.5 ms**
LIME	**0.1105**	**86.5%**	606.8 ms
隨機	0.0474	————	————

4.5 不同網站防護環境之爬取耗時比較

為評估系統於不同防護環境下的擷取效率，本研究選取 Teepr、ZeroHedge 與中國時報進行測試，分別代表無特殊防護、Google Cloud CDN 與 Cloudflare 環境。每站各測試 50 筆公開新聞 URL，並以相同 Firecrawl 流程記錄成功率與平均耗時。結果顯示，三者成功率皆為 100%，平均每篇耗時分別為 2.79 秒、2.67 秒與 2.42 秒。雖然中國時報採用 Cloudflare 防護，但在本次測試中平均耗時最低，顯示爬取效率不僅受防護機制影響，也與網站回應速度、頁面結構、文章內容載入方式及 Firecrawl 等待設定有關。因此，本實驗結果不代表 Cloudflare 防護強度較低，而是表示在本次測試 URL、等待時間與系統設定下，中國時報頁面取得流程較快。

表五、不同網站防護機制與爬取耗時比較

網站	防護機制	等待時間 (ms)	平均每篇耗時 (s)	總耗時 (s)
Teepr	無	2000	2.79	139.59
ZeroHedge	GCDN	3000	2.67	133.59
中國時報	CF	8000	2.42	120.84

五、參考文獻

[1] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, “Hierarchical attention networks for document classification,” in Proc. 2016 Conf. North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), San Diego, CA, USA, 2016, pp. 1480–1489, doi: 10.18653/v1/N16-1174.

[2] R. Pappagari, P. Żelasko, J. Villalba, Y. Carmiel, and N. Dehak, “Hierarchical transformers for long document classification,” in Proc. 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), Singapore, 2019, pp. 838–844, doi: 10.1109/ASRU46091.2019.9003958.

[3] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, “On calibration of modern neural networks,” in Proc. 34th Int. Conf. Mach. Learn. (ICML), ser. Proceedings of Machine Learning Research, vol. 70, Sydney, NSW, Australia: PMLR, 2017, pp. 1321–1330.

[4] M. T. Ribeiro, S. Singh, and C. Guestrin, “‘Why should I trust you?’: Explaining the predictions of any classifier,” in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD), San Francisco, CA, USA, 2016, pp. 1135–1144, doi: 10.1145/2939672.2939778.