AI 資安情報員 - 但好像少了些什麼

113 年度 AIS3 新型態資安暑期課程情資運用及防禦組 B1

張簡雲翔
資訊工程學系
國立暨南國際大學
南投, 臺灣
ase12345636@gmail.com

莊舒涵
資訊工程學系
國立臺灣大學
臺北, 臺灣
eeeeeecsh@gmail.com

蕭愷宸
資訊科
臺北市立松山高級工農職業學校
臺北, 臺灣
Henry970124@gmail.com

陸紹祺
資訊工程學系
國立中央大學
桃園, 臺灣
shaoqilu870@gmail.com

Abstract

在資訊爆炸的時代，資訊安全情資分散且更新迅速，為了因應這樣的時代，本專題提出透過整合 Retrieval-Augmented Generation（RAG）技術及 Line Bot，實現可以簡單利用的情資檢索平台。透過預先將資料收集並做處理建立成 database，並利用 Llama3 模型進行檢索或 VirusTotal API 進行資料查詢與分析。並實作出了一個簡易且可用的 Line Bot，雖然仍有許多不完備之處，但是仍舊提出了一個資安情資運用的可能性。

Keywords—情資運用、Llama3、RAG、Line bot

I. INTRODUCTION

在資訊爆炸的時代，有來自各個不同地方的資安情資，但這些資訊分散在各個角落，且更新速度快，讓人難以全面掌握。傳統的資訊安全防護方法往往需要耗費大量的人力和時間來蒐集和分析這些分散的資訊，這使得我們難以在第一時間內做出有效的防禦反應。因此，如何利用先進的技術來統整和分析這些分散的資安情資，並及時提供有效的防禦建議，成為一個亟待解決的問題。

希望可以透過收集來自各個地方的情資，並結合機器學習的技術進行情資分析與整理，同時結合惡意檔案查詢功能與攻擊技術，提供相關的建議及防禦方式。這不僅能夠提高資訊安全的防護效率，還能幫助使用者及時掌握最新的威脅動態，並採取相應的防禦措施。我們希望通過這樣的嘗試，能夠為資訊安全領域提供一個高效、實用的解決方案，從而提升整體的資安防護能力。

為此，我們想要實作出一個聊天機器人，盡可能地收集來自不同平台的資訊，並使用 Retrieval-Augmented Generation（RAG）的技術進行資料的檢索，並在結合 Line Bot，使使用者可以方便且快速的了解到現在最新的資安情報，以利使用者可以提前因應或進行研究或分析。

II. BACKGROUND

隨著機器學習技術的快速發展，資訊安全領域中，如何使用機器學習的技術來協助工作的相關研究逐漸受到重視。在當前的一個高度資訊化的時代，惡意軟體和網絡攻擊的數量與日俱增，每日都有各式各樣新的攻擊手法出現，並發佈在各種不同的網站中，為此如何製作一個有效、快速的資安情資收集與分析平台，變成一個相當重要的事情。

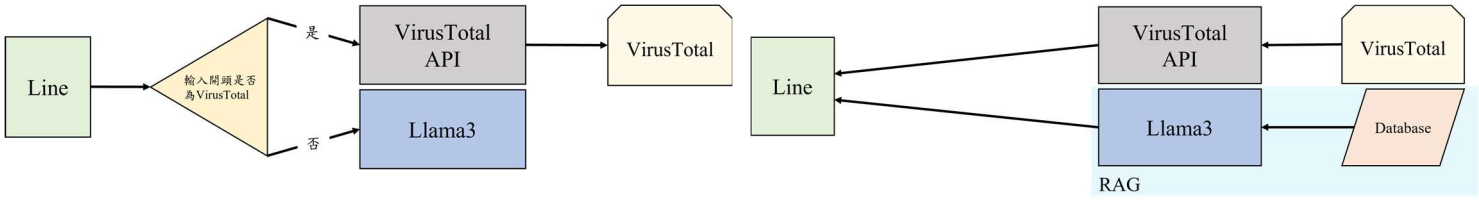
近年來，有各式各樣不同的資安情資分析與收集整合平台已經出現，並不乏利用大型語言模型來進行分析，像是 HackBot [1]就是一個經典的例子，透過使用大量的 CVE 的相關報告，針對 Llama3 的模型進行 fine tuning，使使用者不用花費大量的時間閱讀相關文件，也可以掌握到相關攻擊手法與漏洞。

III. SYSTEM STRUCTURE

A. Overview

系統分為前端與後端，前端是使用 Line Bot 作為使用者輸入以及顯示輸出，後端則由專題成員的電腦作為運算以及傳送資料的伺服器（圖一）。

在資料輸入部分（圖一(a)），用戶通過 Line Bot 進行互動。當使用者輸入資料時，會送到後端的伺服器，首先檢查輸入內容的開頭是否為 VirusTotal。如果是，系統會調用 VirusTotal API 並送到 VirusTotal 網站進行查詢資料。如果輸入內容並非以 VirusTotal 作為開頭，則會當作是 Llama3 的輸入。



(a)資料輸入　　(b)資料輸出

圖一：系統架構圖

當資料輸入資後，經過處理後則會通過 Line Bot 返回給用戶（圖一(b)）。如果是 VirusTotal 的查詢結果，直接將 VirusTotal API 返回的資料直接回傳給使用者。如果不是，Llama3 將會與已經預先處理好的情資資料庫進行交互，利用 RAG 技術檢索相關資料並生成回應，並通過 Line 返回給使用者。

B. Llama3

Llama3 是 Meta 於 2023 年推出的大型語言模型，模型主要架構是基於 Transformer 上進行修改，與原先 Transformer 主要不同的地方在於 position embedding、self-attention 的 Q K V 的算法上的差異，以及 normalization 的做法不同[2]。

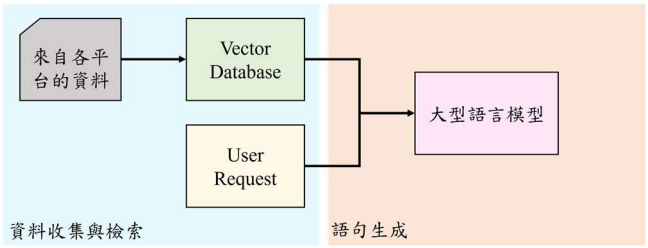
Llama 3 的設計概念是希望能夠支援更廣泛的應用場景，包括但不限於聊天機器人、內容生成、回答使用者等。Llama3 在訓練過程中，相較於其他大型語言模型採用了更大規模的數據集，並且一定程度的改善了原先 Transformer 的特性，使得它在處理複雜語言任務時表現得更加出色。

為了方便使用 Llama3，我們使用 Ollama 這個軟體，Ollama 可以很方便的下載以及將 Llama 或一些其他已經訓練好的大型語言模型載入到程式當中，並考量算力選用最小的 Llama3 的模型作為我們後續實作所使用的大型語言模型。

C. RAG

大型語言模型在訓練的時候，所選用的資料集會受限於訓練時當下的時空背景，以及主要用於使模型回應答案較為順暢的資料集作為訓練用的資料。在現在這個快速發展的時代下，資料的迭代與變化十分迅速，也有許多資料是屬於模型訓練的研究者無法取得。在一些特殊的任務上，還是會有希望模型可以回應或檢索原先模型並沒有訓練過的資料來回應問題，或即時的去更新已經被訓練過的資料，為此發展出了 Retrieval-Augmented Generation(RAG)這項技術[3]，相較於 fine tuning 而言，可以再不調整本身模型權重的情況下，使模型的回答更符合使用者的需求，比較容易達到即時且快速的調整，也能達到提供模型原先沒有收集到的資料，用於回應使用者的問題。

RAG 的技術分成兩個部分 （圖二），一個部分為資料的收集以及檢索（圖二淺藍色處），另一部分大型語言模型讀入使用者的需求以及資料庫的內容後，生成回答的部分（圖二淺橘色底處）。在資料已經收集完成後，先將資料經過 Llama 的 embedding layer，將原本的文本資料轉換成向量的形式，並儲存起來建立成一個 database。在使用者輸入需求時，同時將使用者的需求以及資料庫輸入到大型語言模型中，進行回答的生成。



圖二：RAG 運作簡易示意圖

我們在實作的時候，主要是透過 LangChain 這個套件來實作，LangChain 包含很多各式各樣方便大型語言模型串接各種東西的工具，包含但不限於專門處理以及建立 vector database 的 Chroma，我們這次主要用到的也是 Chroma 這個工具來將已經轉換成向量形式的資料，儲存成一個 SQLite 的形式的 database。在模型需要使用的時候將 database 讀進程式中，並轉換成 retriever，同時將 retriever 以及使用者的輸入結合再一起丟給大型語言模型進行預測。

D. VirusTotal API

VirusTotal 提供了一個官方的 API，可以用來嵌入在程式中，透過該 API 去與 VirusTotal 的網站上進行查詢，可以傳送給 API 的檔案形式有很多種，像是一段 URL 或者是一段可執行檔的 hash 值，在 API 得到了使用者的輸入後，就會到 VirusTotal 的網站查詢相關的資料並回傳給使用者，回傳的形式可以是 html 或者 json 的格式，提供給使用者進行分析以及利用。

我們這次將該 API 加入我們的實作中，考量到時間以及算力資源，我們利用傳送 hash 值給 VirusTotal，並回傳 json 的形式給我們，回傳的資料是該 hash 值所查詢到的最近一次的掃描紀錄，再將回傳的資料直接回傳給使用者，不做其他的處理。

E. Line Bot

Line Bot 是一個 Line 官方所提供的一個與用戶互動的機器人，並會創立一個新的帳號，可以讓任何人去加這隻機器人為好友，並可以使用一般傳送 Line 的方式互動，常見於各種不同的地方，向是餐廳的點餐系統，或者協助政府政令宣導的帳號，抑或是學校的官方推播帳號。

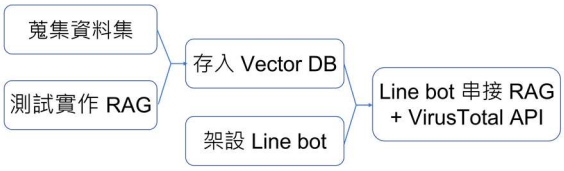
Line 官方沒有提供任何可以去建立 Line Bot 行為模式的後台伺服器，但是可以透過取得 channel access key 以及 channel secret，並在設定伺服器所在的網址，就可以去撰寫後端運作的程式，所以我們將 Line Bot 的後端伺服器架設在實驗成員的電腦上，並透過 Django 去建立一個有辦法收取 Line Bot 所傳送的資料，並在將其資料送給大型語言模型，或 VirusTotal 的 API 去做查詢，得到任何相關結果後，在回傳給 Line Bot，顯示給使用者。

礙於成本與時間空間的考量，我們沒有特別申請一個網域，所以透過 Ngrok 這個軟體，這是一個可以將本地的服務提供至公網的軟體，透過這個軟體就可以在不用特別申請一個網域的情況下，就可以在本地建設一個服務，並公布在公網上方，讓所有人來存取或使用。

IV. IMPLEMENTATION AND DIFFICULT

A. 實驗流程

我們全體實驗成員進行分工，一部分的人去收集以及將資料整理成 csv 檔並去研究 VirusTotal API 的用法，另一部分的人去研究 RAG 的實作方式以及 Line Bot 的架設方法。首先需要先收集到資料，在經過簡單測試 RAG 的程式碼可以使用後，再將資料存入 vector database，以供後續使用，同時去架設 Line Bot 所需要的 code，最後在將所有的 code 整合起來（圖三）。透過這樣的分工與流程來進行後續的實作。



圖三：實驗流程圖

B. 資料收集

我們主要選用三個網站上的新聞作為我們的資料來源，分別是：Broadcom.com、darkreading.com、cyware.com 三個網站上的資料，並利用 EasyScraper 這個軟體把網站上的資料透過爬蟲的技術抓下來，主要抓的時間範圍是這三個月內的新聞內容，抓取新聞標題以及新聞內容，並整理成 csv 檔，第一列為新聞標題，第二列為新聞內容，資料約 1000 多筆，每筆資料長度平均約為 750 的字元，資料來源的內容主要語言都是以英文為主。

C. 小型測試

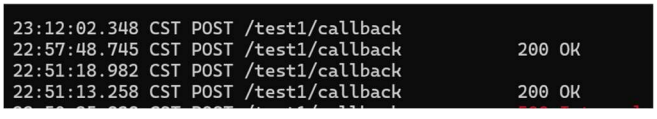
在將所有的資料進行處理前，先進行一個小規模的測試，根據我們的統計發現，資料的平均長度為 750 的字元，所以 chunck size 設為 750，chunck overlap 設為 150，我們先輸入隨機抽選的三筆資料，透過 LangChain 所提供的 Chroma 將資料轉換成 vector database，之後再透過 Llama3 進行搜尋資料的內容，並加以顯示出來，經過測試可以正常運作，回答的內容明顯與我們所提供的資料有高度相關，所以我們將所有的資料都經過 Chroma 儲存成 database，以供後續的利用。

D. 困難挑戰

在這段實驗的過程中，其實我們面臨到許多的挑戰，像是 Chroma 的資料量，不能夠超過 166 筆，需要分批處理，經過 Chroma 官方的 github 上的 issue 表明，可能是因為 Chroma 是利用 SQLite 的方式進行儲存所以導致的原因，只有說後續他們會再進行更新，但目前是還沒有看到後續的下文。

在 LangChain 的官方文件中表明，如果重新執行資料的處理，預設會將以前已經建立的 vector database 給覆蓋掉，所以在實驗過程中每次測試，並沒有特別去刪除舊有的 database，但是後來發現他是一直往後面疊加，並沒有特別進行刪除。

因為我們的算力資源僅有一張 2GB VRAM 的 MX450，又因為我們需要串接 Line Bot，無法直接架設在 Colab 上方，導致我們每次模型在檢索以及生成回覆的時候其實所耗費的時間都十分的久，時間長達 Ngork 在監控傳送情況的時候，會直接忘記持續監控到我們回傳的封包（圖四），雖然仍舊有正常運作回傳給 Line Bot。



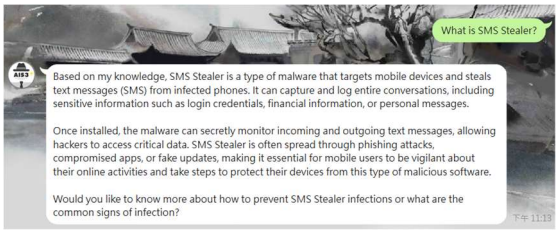
圖四：Ngork 監控情況

在小型測試的時候，在本機端的結果都有正常運作，但是在我們將所有資料轉換並儲存到 vector database 時發現，無論我們怎麼詢問模型都會回應我們所提供的資料中，並沒有相關的東西，根據我們上網查詢以及推測，可能是因為我們所選用的模型能力太弱，導致模型檢索能力不足，以至於模型無法檢索到 database 裡面的資料。所以最後我們僅使用三筆資料所構成的 database，作為我們最後的展示成果。

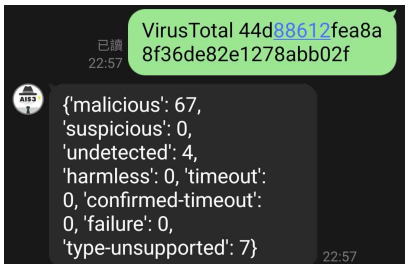
E. 成果展示

我們分成兩個部分進行測試，首先先詢問有關我們所提供給模型的內容（圖五(a)），可以看到模型有正常的給予回覆，雖然給予的回覆與我們原先在本機端直接測試的有明顯差異，也好像有點不是從我們所提供的資料做的回應，但礙於時間考量，就沒有多做其他測試。

再來針對開頭為 VirusTotal 的訊息，有正常的呼叫 VirusTotal 的 API，並有正確得查詢到我們所想要知道的內容（圖五(b)），效果與我們預期的完全相同。



(a)開頭非 VirusTotal



(b)開頭是 VirusTotal

圖五：Line Bot 回傳結果

V. CONCLUSION

在這次的專題中，我們完成了一個可以查詢三筆資料的 RAG 架構，雖然與 Line Bot 串接上似乎有問題，但是還是可以正常地從大型語言模型中得到回覆的答案，也有同時的串接 VirusTotal 的 API，並能夠查詢使用者所提供的 hash 值在 VirusTotal 的掃描結果。

雖然本次成果並非十分豐碩，但也是對資安情資收集與分析提供了一種可能的發展方向，對於未來如果在時間與算力允許的情況下，可以多做其他大型語言模型的測試，或者對 VirusTotal 可以傳送回來的資料多做其他處理與分析，或者開放更多的查詢方法。

在現在資訊爆炸的時代下，有越來越多不同的資安情資分析與收集的平台出現，不乏有很多也加上了大型語言模型作為分析，也可以觀察到近年來相關應用與研究日益重要，希望在未來的某日我們可以看到大型語言模型在資安領域中的研究發展到一個無與倫比的地步。

REFERENCES

[1] Chiranjeevi g. HackBot - AI Cybersecurity Chatbot. 2024, https://github.com/morpheuslord/HackBot?tab=readme-ov-file

[2] Dubey, Abhimanyu et al. “The Llama 3 Herd of Models.” (2024).

[3] Amazon Web Services. 什麼是 RAG (檢索增強生成)？. https://aws.amazon.com/tw/what-is/retrieval-augmented-generation/.