陽明交通大學 資工系大三 111550177 吳定霖

NLP HW3 Report

執行環境

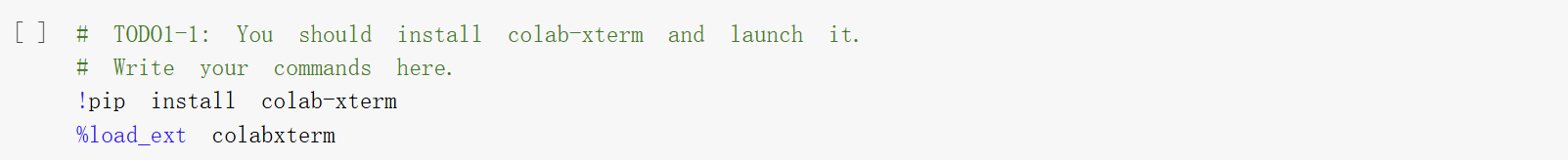
我直接在 Colab 上面執行我的程式碼，使用 T4 GPU 和 大量 VRAM。使用套件請看 requirements.txt，大部分與助教的範例 code 相同，不過我在套件安裝那邊自己加入了 “!pip install transformers -U” 和 “!pip install faiss-cpu”，以確保程是執行不會出錯。

Generative AI and reference information’s specification

1. Colab AI : for code generation
2. ChatGPT : for data analysis and report writing
3. [Langchain Document](https://api.python.langchain.com/en/latest/vectorstores/langchain_community.vectorstores.chroma.Chroma.html#langchain_community.vectorstores.chroma.Chroma.as_retriever) : for writing code
4. [RAG tutorial](https://jayant017.medium.com/rag-using-langchain-part-3-vector-stores-and-retrievers-a75f4d14cbf3) : for writing code

Coding Work

Todo1-1:



Todo1-2:

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

Todo1-3:

一張含有 文字, 螢幕擷取畫面, 軟體, 多媒體軟體 的圖片

自動產生的描述

Todo2-1:

一張含有 文字, 字型, 行, 螢幕擷取畫面 的圖片

自動產生的描述

Todo2-2:

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

Todo3:

一張含有 文字, 字型, 螢幕擷取畫面, 筆跡 的圖片

自動產生的描述

Todo4:

一張含有 文字, 字型, 螢幕擷取畫面 的圖片

自動產生的描述

Todo5:

The Best correct number :

I reach correct number 10.

一張含有 文字, 螢幕擷取畫面, 文件, 字型 的圖片

自動產生的描述

一張含有 文字, 軟體, 螢幕擷取畫面, 網頁 的圖片

自動產生的描述

Report

**(10%) Please describe the details of your implementation for the RAG system (please tell us 1. What’s in your RAG system? 2. Which retrieval model you use? 3. What’s your prompt? 4. What’s new**

**in your code in comparison with the code from our lab course?) in this assignment and list your best score for the ten questions.**

我要先聲明，雖然我現在的這個 code 並沒有非常多和助教不同的新方法，而且也不是我後來做 ablation study 的時候觀察到 performance 最好的方法。但是，它的確能做到 correct number 10，因此我這邊是以這個 RAG implementation 來作為 best score 範例。其他的方法探討會放在後面的部分。

1. My RAG system with the best score is ChromaDB, same as TA.
2. My retrieval model is mmr(maximal marginal relevance), but I modify the hyperparameters “k” and “fetch\_k” into “4”and “6”.  
   一張含有 文字, 字型, 螢幕擷取畫面 的圖片

   自動產生的描述
3. My prompt is   
   “Use the given context to answer the question.

Context: {context}

If you don't know the answer, guess it based on the context.

Use three sentences maximum and keep the answer concise.”  
一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述

1. My difference with lab course is (1) I modify hyperparameters “k” and “fetch\_k” into “4”and “6”. (2)I modify the lab course prompt.

My best score : 10 (all) correct

一張含有 文字, 螢幕擷取畫面, 文件, 字型 的圖片

自動產生的描述

一張含有 文字, 軟體, 螢幕擷取畫面, 網頁 的圖片

自動產生的描述

**(10%) Please provide analysis for the RAG performance using different prompts.**

**我對於每一種不同的 prompt 都做了 20 次的 question answer 實驗，計算每一次的 correct number 並將這 20 個 correct number 做平均，以減少 LLM 回的隨機性造成的偏差。**

(a)

Same as TA

說明 : 我使用了和 TA 在 lab course 中一模一樣的 prompt 來做測試。

Prompt :

Use the given context to answer the question.

If you don't know the answer, say you don't know.

Use three sentences maximum and keep the answer concise.

Context: {context}

Average correct numbers: 4.3

分析 : 我把 TA 的這個 prompt 當作 baseline，用來跟下面不同的 prompt 作為對照組。

(b)

Force LLM to answer questions if it doesn't know

說明 : 我讓 LLM 在它認為它自己無法根據 RAG 檢索到的資料來回答問題時，強迫 LLM 要根據檢索到的資料來猜一個答案，不管它對於它猜的答案是否有信心。

Prompt :

Use the given context to answer the question.

If you don't know the answer, guess it based on the context.

Use three sentences maximum and keep the answer concise.

Context: {context}

Average correct numbers: 4.8

分析 : 由於這裡強迫了 LLM 一定要根據檢索到的資料來回達一個答案，所以當 LLM 對於自己的回應沒有信心，可是 RAG 確實有正確檢索到可以用來回答問題的資訊時，LLM 可能就能根據檢索到的資料來猜對答案，而不是回答「I don't know.」 而直接放棄作答。因此，能夠增進些許正確率。然而，LLM 會對於自己回答沒有信心的情況，大多是 RAG 的確沒有檢索到回答問題必要的資訊，所以通常強迫回答也不能回答正確、正確率提升有限。

(c)

Put the context right after the “Use the given context to answer the question.” sentence.

說明 : 原本 TA 在 lab course 的 prompt，把 Context: {context} 這行有關 RAG 提取出來的資訊內容放在 prompt 的最後一行。這邊把 Context: {context} 移到 Use the given context to answer the question. 的下一行。會有這個想法是因為，Use the given context to answer the question.描述了接下來會有可以用來參考的 context，但是原本的 lab course prompt 並沒有馬上接續提供 context 的內容，而是繼續寫 system prompt 的其他部分，我認為這或許導致了 LLM 無法識別 context 的真正內容是什麼，故做了此實驗。

Prompt :

Use the given context to answer the question.

Context: {context}

If you don't know the answer, say you don't know.

Use three sentences maximum and keep the answer concise.

Average correct numbers: 7.1

分析 : 將 context 置於開頭大幅提高了平均正確數字。這代表 LLM 對於 context 上下文的關聯性理解有所增強，Use the given context to answer the question.描述了接下來會有可以用來參考的 context，並能夠直接在下一行得到 context 的實際內容，上下文更緊密地跟問題結合，而非受其他 system prompt 的誤導，因此能夠大幅提升正確率。

(d)

Submission prompt

說明 : 我把前面「Force LLM to answer questions if it doesn't know

」還有「Put the context right after the “Use the given context to answer the question.” sentence.」兩個 prompt 技巧結合到一起。會叫 submission prompt 是因為我前面的 best score code 也是使用這樣的 prompt 組合。

Use the given context to answer the question.

Context: {context}

If you don't know the answer, guess it based on the context.

Use three sentences maximum and keep the answer concise.

Average correct numbers: 7.9

分析 : 此 prompt 結合了 Force LLM to answer questions 與 Use the given context to answer the question.，讓 LLM 同時獲得兩者的優點，進而提升正確率。可以觀察到，Force LLM to answer questions 在這裡的提升比之前使用 lab course prompt 的提升更加明顯，我認為這是因為，在這裡 RAG 的提取能力更強了，因此包含回答問題的必須資訊的機率更高，因此 LLM 在猜答案的時候有更高的機率猜中。

(e)

Give an example of the question and its answer in the system prompt. (One shot learning)

說明 : 在 prompt 裡面，我利用了 150 個 cat fact 中非 10 個問題的答案的 text 來自己創造了一個 QA 的範例，提供給 LLM 作為參照。這就是 one-shot learning。預期 LLM 看過問題及其回答的範例之後，可以增強它在回答這種類型問題上的能力。

Prompt:

There will be a user input question like :

<How many whiskers on each side of a cat's face?>

You need to answer the question like :

<A cat usually has about 12 whiskers on each side of its face.>

Use the given context to answer the question.

If you don't know the answer, say you don't know.

Use three sentences maximum and keep the answer concise.

Context: {context}

Average correct numbers: 4.5

分析 : 實驗結果可知，one-shot learning 無法提高太多正確率，這是因為，這個問答很簡單，並不需要 one-shot learning QA範例來引導 LLM 來回答問題。這個實驗的主要 bottleneck 在 RAG 是否能夠正確的提取出資訊，而不是 LLM 的回應能力。

(f)

One shot learning but the answer in example only have the answer keywords

說明 : 由於答案只要包含到關鍵字 keywords 就算回答正確了，所以我常是讓 LLM 看到的範例，只包含它必須回答的關鍵字。

Prompt :

There will be a user input question like :

<How many whiskers on each side of a cat's face?>

You need to answer the question like :

<12>

Use the given context to answer the question.

If you don't know the answer, say you don't know.

Use three sentences maximum and keep the answer concise.

Context: {context}

Average correct numbers: 4.3

分析 : 和 (e) 的結果差不多，正確率比 (e) 還要低可能是因為，只要求 LLM 回答關鍵字，LLM 只要回答的少許關鍵字沒包含到正確答案，就算答錯。而回答整個句子則有更多機會包含到涵蓋正確答案的關鍵字，所以正確率稍微高一點點。

(g)

Answer with one sentence

說明 : 原本 TA 的 lab course 的 prompt 是要求 LLM 回答使用 three sentence。這裡我嘗試讓 LLM 回答 one sentence。

Prompt :

Use the given context to answer the question.

If you don't know the answer, say you don't know.

Use one sentences maximum and keep the answer concise.

Context: {context}

Average correct numbers: 3.9

分析 : 限制一句回答減少了 LLM 的表達空間，導致表現下降。

(h)

Answer with two sentence

說明 : 原本 TA 的 lab course 的 prompt 是要求 LLM 回答使用 three sentence。這裡我嘗試讓 LLM 回答 two sentence。

Prompt :

Use the given context to answer the question.

If you don't know the answer, say you don't know.

Use two sentences maximum and keep the answer concise.

Context: {context}

Average correct numbers: 4.2

分析: 限制為兩句的效果接近原始 prompt，表明兩句回答能平衡簡潔與完整性，但提升幅度有限。

(i)

Answer with four sentence

說明 : 原本 TA 的 lab course 的 prompt 是要求 LLM 回答使用 three sentence。這裡我嘗試讓 LLM 回答 four sentence。

Prompt :

Use the given context to answer the question.

If you don't know the answer, say you don't know.

Use four sentences maximum and keep the answer concise.

Context: {context}

Average correct numbers: 4.0

分析: 限制為四句未能提升表現，甚至略低於三句的效果。可能是因為多句回答導致資訊冗餘或干擾，使 LLM 無法集中回答問題核心。

(j)

Answer without specifying number of sentences

說明 : 原本 TA 的 lab course 的 prompt 是要求 LLM 回答使用 three sentence。這裡我嘗試不指定 LLM 回答的 sentence 數量。

Prompt :

Use the given context to answer the question.

If you don't know the answer, say you don't know.

keep the answer concise.

Context: {context}

Average correct numbers: 1.6

分析 : 當沒有限制回答的 sentence 數量時，LLM 傾向所有問題都直接回答「I don’t know」，因此正確率分常低。

**(10%) Please compare the RAG performance with different retrieval models and the performance without using RAG (note that Llama 3.2 should not be fine-tuned in this assignment).**

ChromaDB with MMR (maximal marginal relevence)

說明 : 這個實驗使用了 ChromaDB 以及 MMR 演算法，對於 MMR 中不同 k 和 fetch\_k 值來做實驗，探討不同 k 和 fetch\_k 值對於 RAG 問題回答有什麼影響。

一張含有 文字, 螢幕擷取畫面, 數字, 字型 的圖片

自動產生的描述

一張含有 文字, 圖表, 行, 繪圖 的圖片

自動產生的描述

結果分析 :

整體上， ChromaDB的正確率落在 6.8-7.7 的區間。

k 較小時（如 k=2），隨著 fetch\_k 的增加，正確率在一段範圍內有所提升，但超過某一點（例如 fetch\_k > 8），正確率反而開始下降。

k 適中或偏大時（如 k=4 或 k=5），正確率的穩定性相對較高，且在某些情況下，較大的 fetch\_k（如 9 或 10）能顯著提升表現。

k 過大時（如 k=7），整體正確率會略有下降，尤其當 fetch\_k 過高（如 fetch\_k > 12），正確率甚至可能大幅降低。

接下來，觀察每一個 k 值分別的正確率趨勢 :

1. k = 2

最佳組合：fetch\_k=7（7.1 正確率）

整體趨勢：隨著 fetch\_k 從 2 增加到 7，正確率逐漸提高，達到頂峰後開始回落。

分析：較小的 k 代表 LLM 更注重初始的 RAG 檢索資訊，但過高的 fetch\_k 可能會提取過多不必要的冗餘資訊，導致 LLM 受到誤導，正確率下降。

(2) k = 3

最佳組合：fetch\_k=9 或 fetch\_k=10（7.35 正確率）

整體趨勢：在 fetch\_k=3 至 fetch\_k=10 範圍內，表現相對穩定，並在 9 和 10 取得最高分數。

分析：適度增加的 k 和較高的 fetch\_k 似乎提供了較平衡的 RAG 檢索能力。

(3) k = 4

最佳組合：fetch\_k=9 或 fetch\_k=11（7.7 正確率）

整體趨勢：表現穩定，fetch\_k=4 至 fetch\_k=8 表現幾乎一致，直到 fetch\_k=9 和 fetch\_k=11 有顯著提升。

分析：這可能表示在 k=4 的情況下，RAG 檢索結果的準確度達到了不錯的平衡。也因此我選擇此作為我的 submission code prompt。

(4) k = 5

最佳組合：fetch\_k=9 或 fetch\_k=12（7.65 正確率）

整體趨勢：隨著 fetch\_k 增加，正確率出現一些波動，但在 9 和 12 時達到高峰。

分析：這代表，k=5 時更高的 fetch\_k 有助於擴展 RAG 檢索範圍，但過多的數量可能會帶來不必要的冗餘資訊，混淆 LLM 的回應。

(5) k = 6

最佳組合：fetch\_k=10（7.45 正確率）

整體趨勢：表現較穩定，但在 fetch\_k=12 和更高時正確率略有下降。

分析：在 k 足夠大時，較大的 fetch\_k 比較不會使得正確率下降非常多，可能是因為，被 fetch\_k 帶入的資訊有較多都提交給 LLM，而不會產生必要資訊被冗餘資訊混淆而剔除的情況。

(6) k = 7

最佳組合：fetch\_k=7 或 fetch\_k=11（7.4 和 7.3 正確率）

整體趨勢：正確率在中低範圍表現尚可，但當 fetch\_k 過大時（如 12、13、14），正確率急劇下降。

分析：過大的 k 和 fetch\_k 組合會讓 RAG 檢索過於分散，降低檢索相關性。

總結:

k 的最佳值在 4 或 5 左右，這樣可以兼顧 RAG 檢索的多樣性與生成的準確性。

fetch\_k 不能過高， k 的大小最好限制在 10 以下，以避免不必要的冗餘資訊。

在更強調精度的情況下，可以選擇較小的 fetch\_k。

FAISS with MMR (maximal marginal relevence)

說明 : 這個實驗使用了 FAISS 以及 MMR 演算法，對於 MMR 中不同 k 和 fetch\_k 值來做實驗，探討不同 k 和 fetch\_k 值對於 RAG 問題回答有什麼影響。

一張含有 文字, 數字, 字型, 螢幕擷取畫面 的圖片

自動產生的描述

一張含有 文字, 圖表, 行, 繪圖 的圖片

自動產生的描述

結果分析 :

整體上，FAISS 正確率比 ChromaDB 的效果還要好。大部分的正確率落在 7.0-8.5 的區間。

較小的 k 值（例如 k=2 和 k=3）在大部分情況下，能夠保持更高的平均正確數量，表現相對穩定且高效。

中等 fetch\_k 的表現最佳：當 fetch\_k 適度增加時，模型的正確率往往能達到高峰，但一旦超過一定範圍，正確率可能會下降。

隨著 k 的增大，模型的表現開始變得不穩定：當 k 達到 6 或 7 時，即使在某些情況下有不錯的表現，但整體趨勢逐漸呈現下降，代表 RAG 檢索可能導致更多的冗餘資訊進入生成過程。

接下來，觀察每一個 k 值分別的正確率趨勢 :

(1) k = 2

最佳組合：fetch\_k = 7或 fetch\_k = 9（8.15 和 8.1正確率）

隨著 fetch\_k 的增加，正確率從 7.6 稍微提升，並在 7 和 9 達到高峰。然而，當 fetch\_k 超過 9 時，正確率開始回落到 7.65 左右。

分析：k=2 時，檢索的初始結果相對精確，適當的 fetch\_k 擴展有助於補充多樣性，但過多的檢索結果可能導致冗餘資訊增加。

(2) k = 3

最佳組合：fetch\_k = 4 或 fetch\_k = 9（8.55 和 8.15正確率）

整體表現穩定，但在 fetch\_k = 4 時達到高峰，隨後有輕微波動。

分析：k=3 增加了 RAG 檢索的多樣性，適當的 fetch\_k 可以進一步提升效果，但過高的 fetch\_k 會降低 RAG 檢索的相關性。

(3) k = 4

最佳組合：fetch\_k = 10（8.3正確率）

從 fetch\_k = 4 開始，正確率逐漸提升，直到 fetch\_k = 10 達到峰值；之後的數據表現相對穩定但略微下降。

分析：這顯示出 k=4 是一個相對平衡的選擇，能夠在 RAG 檢索多樣性與相關性之間取得不錯的平衡。

(4) k = 5

最佳組合：fetch\_k = 7 或 fetch\_k = 12（7.75 和 7.6正確率）

整體波動較大，並且沒有出現明顯的高峰值。當 fetch\_k 超過 11 後，正確率開始下滑。

分析：k=5 時，RAG 檢索的相關性開始受到挑戰，過高的檢索深度（fetch\_k）可能引入過多無關冗餘數據。

(5) k = 6

最佳組合：fetch\_k = 9（8.05正確率）

數據在 fetch\_k = 6 開始逐步提升，並在 fetch\_k = 9 達到最高值。然而，fetch\_k 過高後，正確率又開始下降。

分析：k=6 時，RAG 檢索的多樣性進一步提升，但相關性受到了顯著影響，因此過高的檢索結果數量無法持續提升效果。

(6) k = 7

最佳組合：fetch\_k = 10（8.0正確率）

當 fetch\_k = 7 時表現尚可，但在 fetch\_k = 10 達到短暫的高峰。隨著 fetch\_k 繼續增大，正確率呈現下降趨勢。

分析：過大的 k 和 fetch\_k 組合會導致 RAG 檢索結果過於分散，LLM 難以有效處理過量冗餘資訊。

總結:

較小 k 值（如 2 或 3）能夠確保檢索相關性，而中等的 fetch\_k（如 7 至 10）能夠增強檢索多樣性，兩者結合能夠最大化 RAG 的性能。

當 k 或 fetch\_k 過大（如 fetch\_k > 12 或 k=7）， RAG檢索資訊的相關性開始降低，正確率下降。

ChromaDB with similarity threshold

說明 : 這個實驗使用了 ChromaDB 以及 similarity threshold，對於不同 similarity threshold 值來做實驗，探討不同 similarity threshold 值對於 RAG 問題回答有什麼影響。

一張含有 文字, 行, 圖表, 繪圖 的圖片

自動產生的描述

結果分析 :

整體趨勢上，當 similarity threshold 提高時，LLM 的平均正確回答數量並沒有上升。

0.3 和 0.5 是表現最佳的區間，正確率達到 7.75。

當 threshold 超過 0.5 後，正確率開始逐漸下降，尤其是到 0.9 時，降至僅 3.15。

Threshold 0.9 會造成這麼劇烈的正確率下降是因為，沒有 similarity 超過 0.9 的 text 存在於 dataset 中，因此 RAG 無法提取資訊。

較低的 threshold（如 0.2）表現雖然不差，但整體不如 0.3 和 0.5。

總結 :

最佳的 similarity threshold 不在極端值，而是在一個適中的區間。根據結果，0.3 和 0.5 是目前的最佳選擇。

較大的 similarity threshold 雖然直覺上認為可以提升 RAG 檢索內容的相關性，但實際上會過度壓縮檢索範圍，反而降低模型的表現。

FAISS with similarity threshold

說明 : 這個實驗使用了 FAISS 以及 similarity threshold，對於不同 similarity threshold 值來做實驗，探討不同 similarity threshold 值對於 RAG 問題回答有什麼影響。

一張含有 文字, 行, 繪圖, 圖表 的圖片

自動產生的描述

整體趨勢上，LLM 在大多數 Threshold 值下的表現相對穩定，平均正確回答數量大多維持在 7.9 到 8.25 之間，這顯示 FAISS 相對於 ChromaDB 能在廣泛的 Threshold 範圍內維持不錯的 RAG 檢索能力。然而，當 similarity threshold 超過 0.8 後，模型的表現出現明顯下滑，尤其是 0.9，僅剩 2.65。這同樣是因為 dataset 幾乎沒有 similarity 超過 0.9 的 text。

實驗顯示，threshold 0.5 到 0.6 是 RAG 檢索效果較佳、較穩定的區間。

BM25 Retriever

說明 : 這個實驗使用了 BM25 Retriever，對於不同 top\_k 值來做實驗，探討不同 top\_k 值對於 RAG 問題回答有什麼影響。

一張含有 行, 繪圖, 圖表, 文字 的圖片

自動產生的描述

實驗分析 :

整體趨勢上，top\_k 在 2 和 9 時達到了最高的平均正確數字，分別為 7.45 和 7.35。top\_k 在 3-6 的範圍內正確率較低，但波動並不劇烈，正確數字穩定在 7.0 左右。當 top\_k 增加到 10 時，平均正確數字下滑到 6.9，顯示 RAG 檢索更多內容並不一定能幫助 LLM 更好地回答問題。

當 top\_k 太低（例如僅為 1），雖然檢索結果可能非常相關，但一旦第一個結果不足以涵蓋完整答案，模型就容易回答錯誤。

隨著 top\_k 增加，檢索到的資訊多樣性雖然提升，但其中的冗餘資訊也會增多，導致模型需要處理更多無關冗餘資訊。

**(10%) Anything that can strengthen your report**

我還有一些額外的實驗方法探討。

Different RAG Document Chunks Size

說明 : 我們之前都是把每一個 cat fact 當作一個 document 來存成 embedding 來做 RAG 檢索，因此我想實驗看看，如果每一個 document chunk size 變大，對於 RAG 檢索能力會有什麼影響?

一張含有 文字, 行, 圖表, 繪圖 的圖片

自動產生的描述

結果分析 :

當 chunk size 為 1 時，正確率，平均正確回答數達到 8.35。

隨著 chunk size 增加到 2，正確率下降到 7.65，但仍保持在較高的分數。

這說明 LLM RAG 在處理更小、更專注的 chunk 時，能獲得更精準的檢索結果，從而更有助於回答問題。每個 chunk 包含的資訊量雖然少，但與問題的相關性可能更高，減少了無關訊息的干擾。

Chunk size 3-5 的正確率在 7.1-7.3 之間。

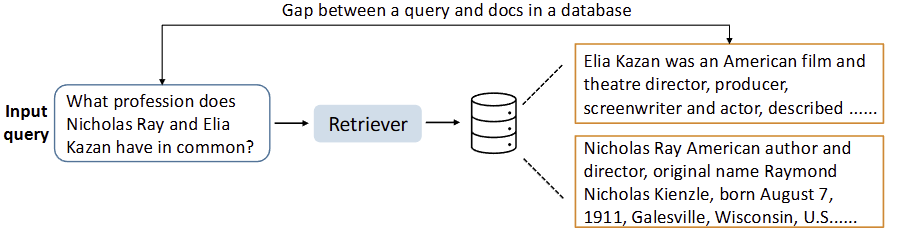
當 chunk size 增加到 6 時，結果稍微回升到 7.25。這是因為，chunk 太大時，資訊密度增加，但過於冗長可能會降低 RAG 檢索結果的精準度。

當 chunk size 增加到 10 時，模型的正確回答數驟降至 6.5。

這表明，過大的 chunk 導致 RAG 檢索結果變得過於模糊，無法準確對應問題的需求。大 chunk 可能包含太多的無關的資訊，影響了檢索的精準部。

**Rewriting Query**

說明 : 這是老師上課提到的，可以把 query 的內容先經過 LLM 重新撰寫過之後，再進行 RAG 檢索。這樣可能可以使得 RAG 的 similarity 計算結果更加精準，得到更好的檢索結果。



Rewriting query prompt :

Rewrite the given user input query to improve retrieval results.

Use concise language for the rewritten query.

一張含有 文字, 行, 圖表, 繪圖 的圖片

自動產生的描述

結果分析 :

由結果趨勢可以看到，Rewrite Query 反而降低了 RAG 的檢索結果。這可能是因為，Rewriting prompt 設計的不夠好，導致產生出來的 Rewrite Query 並沒有捕捉到原本的 Query 所要包含的關鍵意涵，甚至曲解原來的含意。計算 similarity 的時候，反而造成包含正確答案的 text 和 Rewrite Query 之間的 similarity 較低，沒有檢索到該正確答案資訊。