**Report：NLP\_HW4 成功大學\_N96134645\_許宸華**

|  |  |
| --- | --- |
| **Running environment：Colab** | **Python version：Colab** |
| **GPU(s) used：Tesla T4** | **CUDA 版本: 12.2** |

**以下的chatgpt皆為chatgpt-4o**

**● (10%) Please describe the details of your implementation for the RAG system (please tell us**

**1. What’s in your RAG system?**

LLM：llama3.2:1b

retrieval model：jinaai/jina-embeddings-v2-base-en

RAG data：https://huggingface.co/ngxson/demo\_simple\_rag\_py/resolve/main/cat-facts.txt

model\_kwargs = {'trust\_remote\_code': True}

encode\_kwargs = {'normalize\_embeddings': False}

search\_type="mmr"

search\_kwargs={"k": 3, "fetch\_k": 100}

**2. Which retrieval model you use?**

**一張含有 文字, 字型, 螢幕擷取畫面, 行 的圖片

自動產生的描述**

經過測試，在mmr的情況下search\_kwargs={"k": 3, "fetch\_k": 100}

jinaai/jina-embeddings-v2-base-en做為retrieval模型表現還是最好。

**3. What’s your prompt?**

**一張含有 文字, 字型, 螢幕擷取畫面 的圖片

自動產生的描述**

**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.\*\***

模型：有嘗試過其他embedding模型但原先的jinaai/jina-embeddings-v2-base-en在特定參數下還是表現比較好。

參數：有試過similarity, mmr, similarity\_score\_threshold，similarity和mmr表現差不多，都比但similarity\_score\_threshold 還要好，但mmr在經過fetch\_k的調整也明顯比similarity還強，所以還是一樣用mmr。

Prompt：改用較為穩定的prompt，準確度也比較高。

**一張含有 文字, 字型, 螢幕擷取畫面 的圖片

自動產生的描述**

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

一開始使用助教課給的範例prompt，答對數量最多在6

**一張含有 文字, 字型, 螢幕擷取畫面, 行 的圖片

自動產生的描述**

**一張含有 文字, 字型, 螢幕擷取畫面, 代數 的圖片

自動產生的描述**

從這個prompt可以感覺到模型很常回覆I don’t know，即使他後面的解釋很可能是對的。以及他生成的文字有經過自己模型的修飾，可能會對不到答案。

所以我更改prompt 希望他認為自己是專家，更有自信的回答問題，並且希望他能更依賴RAG的資料。  
一張含有 文字, 字型, 螢幕擷取畫面, 行 的圖片

自動產生的描述

一張含有 文字, 字型, 螢幕擷取畫面, 白色 的圖片

自動產生的描述

Correct number從6-8 但還是很喜歡回覆I don't know，並且他會分析很多與答案無關的資訊。

所以我使用以下的prompt讓模型找到正確答案時直接去使用他，並且清楚知道正確回答比提供過多不必要資訊更重要。

**一張含有 文字, 字型, 螢幕擷取畫面, 行 的圖片

自動產生的描述**

**一張含有 文字, 字型, 螢幕擷取畫面, 代數 的圖片

自動產生的描述**

雖然說Correct number達到9，但是這個prompt很不穩定，甚至有出現過Correct number只有3的情況。

當模型回覆I don't know，我認為上下文可能包含間接答案，只是他還沒找到或思考的還不夠多，但就直接回答了，所以我希望模型更積極地嘗試回答問題，在上下文中找不到答案時，不是簡單地回應I don't know，而是重新檢索相關資訊進行推理，也希望這樣可以增加穩定度。

一張含有 文字, 字型, 螢幕擷取畫面 的圖片

自動產生的描述

一張含有 文字, 字型, 代數, 收據 的圖片

自動產生的描述

這個prompt確實達到很穩定的效果，Correct number大約都在7跟8。而錯誤題多為1、3、5題，以下是分析原因：

第1題：Query: How much of a day do cats spend sleeping on average?

錯誤原因：沒用分數顯示2/3，可能是小數或回答1/3時間醒著。

第3題：Query: What do scientists believe caused cats to lose their sweet tooth?

錯誤原因：答案沒與語句完全相同

第5題：Query: What is the name of the organ in a cat's mouth that helps it smell?

錯誤原因：Jacobson's organ 和 Jacobson’s organ的逗號不同

這邊也可以用prompt去限制回答的形式，但我覺得這種做法有點太overffit了，因此我認為這個prompt夠穩定已經達到我要的效果。

**● (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).**

我替換了三種不同的embedding model ( chatgpt做推薦 )，分別與原先的jinaai/jina-embeddings-v2-base-en模型做比較。(mmr：k = 3, fetch\_k = 5)

1. **sentence-transformers/paraphrase-mpnet-base-v2**

一張含有 文字, 字型, 行 的圖片

自動產生的描述

**一張含有 文字, 字型, 螢幕擷取畫面, 代數 的圖片

自動產生的描述**

**Correct number = 7**

這個模型主要針對句子語意相似性進行訓練，擅長處理句子的語意表示，由於這次evaluation的方式是要與答案一模一樣，即使語意相同有些微不一樣就不算，因此這個模型表現的稍差也是合理的。

1. **intfloat/e5-base**

**一張含有 文字, 字型, 行, 白色 的圖片

自動產生的描述**

**一張含有 文字, 字型, 螢幕擷取畫面, 代數 的圖片

自動產生的描述**

**Correct number = 6**

這個模型在檢索方面的能力比較強大，但在語意方面我測試的結果還是原本的比較好，即便調整一些參數，Correct number也大概在6。

1. **sentence-transformers/all-distilroberta-v1**

**一張含有 文字, 字型, 白色 的圖片

自動產生的描述**

**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**

這個模型是一種bert的輕量化模型，適合簡單的問題和直接的回答，他的表現其實也很不錯，固定在9左右，我有試著調整參數去improve，但影響都不大，最後還是輸給jinaai/jina-embeddings-v2-base-en (mmr：k = 3, fetch\_k = 100)。

1. **without RAG**

以下是沒有使用RAG直接使用LLM去解題，可以看到效果超級差，這也很符合預期，畢竟RAG也算是讓LLM看答案吧。

**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**

**一張含有 文字, 螢幕擷取畫面, 字型, 代數 的圖片

自動產生的描述**

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

1. 為了方便分析我有加入問題的編號以及顯示出答錯的題目編號

**一張含有 文字, 字型, 螢幕擷取畫面, 行 的圖片

自動產生的描述**

**一張含有 文字, 字型, 代數, 螢幕擷取畫面 的圖片

自動產生的描述**

2. 我有向chatgpt提問mmr參數設定的影響：

**”””chatgpt提供”””**

fetch\_k：這是初步檢索文檔的數量。在檢索時，會先從向量庫中檢索 fetch\_k 個文檔，再從中選擇最符合查詢的 k 個文檔。例如，fetch\_k=5 表示會首先檢索 5 個文檔，然後從中挑選最相似的 3 個返回。

fetch\_k 的設置能夠提高檢索的準確度，尤其是在文檔庫較大時，首先檢索更多文檔，能增加返回最相關文檔的機會。

**”””chatgpt提供”””**

因此我嘗試著將fetch\_k調大，結果發現準確率越來越高。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| fetch\_k | 5 | 10 | 30 | 100 |
| Average  Correct number | 4-8 | 5-8 | 6-8 | 10 |

我嘗試fetch\_k到30後就直接想說調到100看看，沒想到幾乎每次Correct number都等於10 (少數幾次9)。甚至順帶解決了Jacobson's organ 和 Jacobson’s organ的逗號不同的問題，我其實覺得很神奇，但這也代表說mmr的第一次檢索其實超爛，要檢索到100個才能達到完全正確，反之挑選最相似的第二次檢索差異就都不大(但k不能調到1也會變差)。