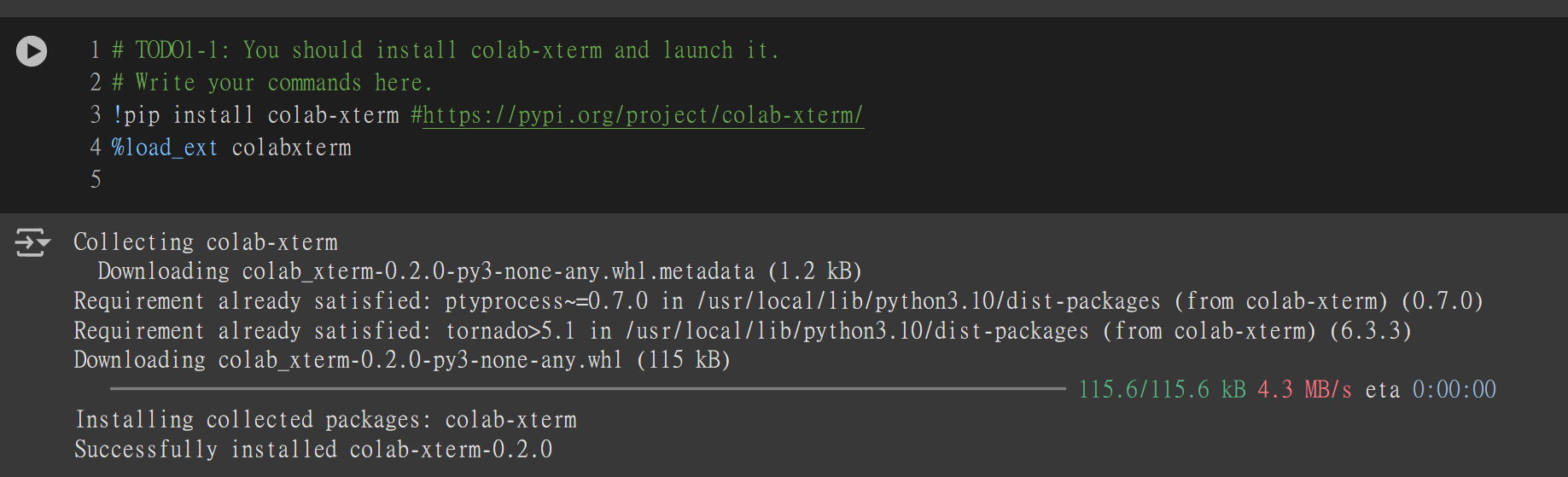
**2024 NTHU Natural Language Processing Assignment 4 : Retrieval-Augmented Generation with LangChain** Python Version : Colab **NTU** r12631055 Running Environment : Colab GPU : Colab T4

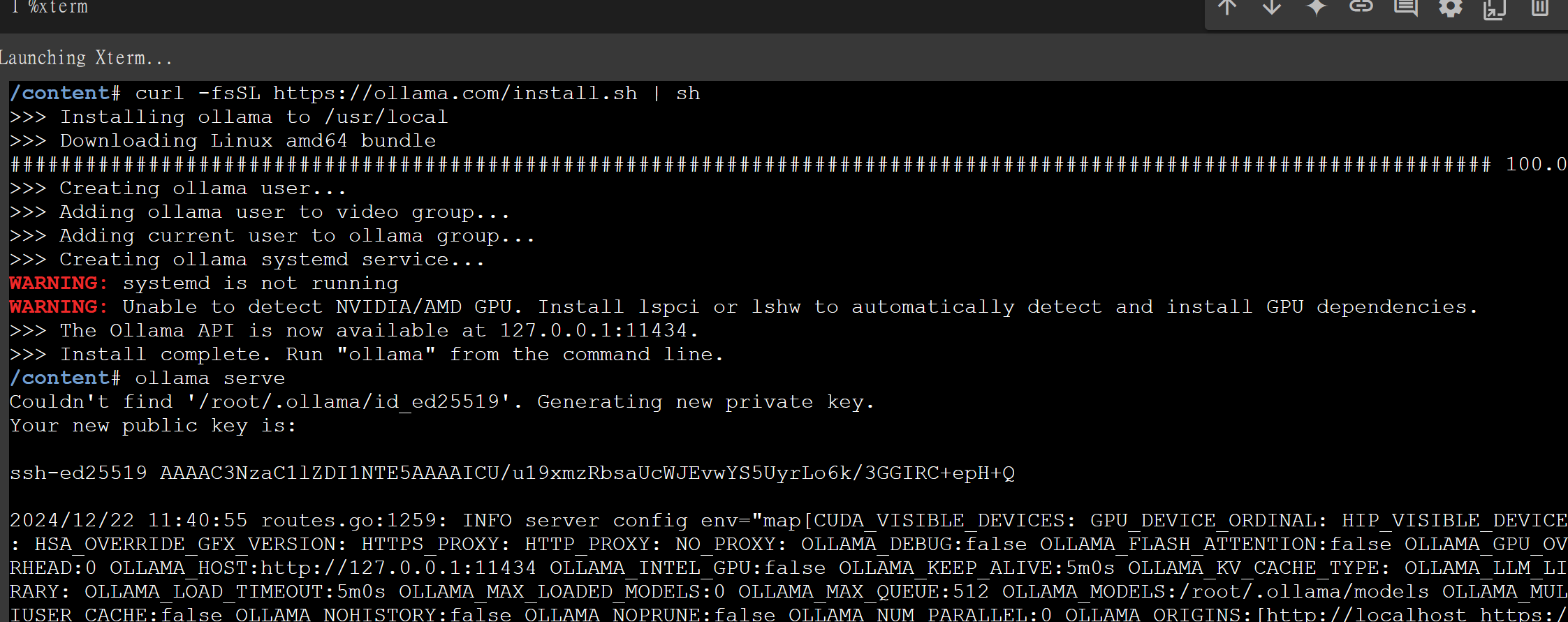
**TODO1-1**



在安裝完需要套件後，依照之前的實作課程內容安裝colab-xterm。

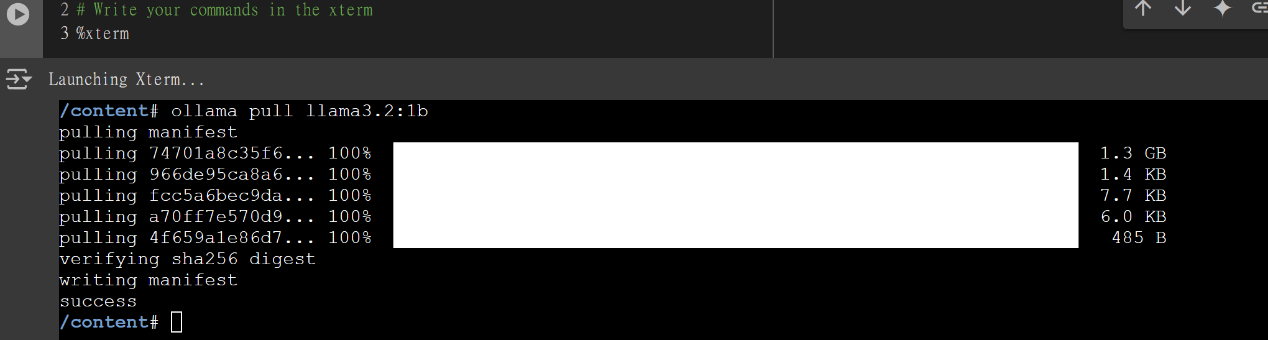
最後開啟測試一下是否成功；最終結果如下，順利開啟。

**TODO1-2**

****

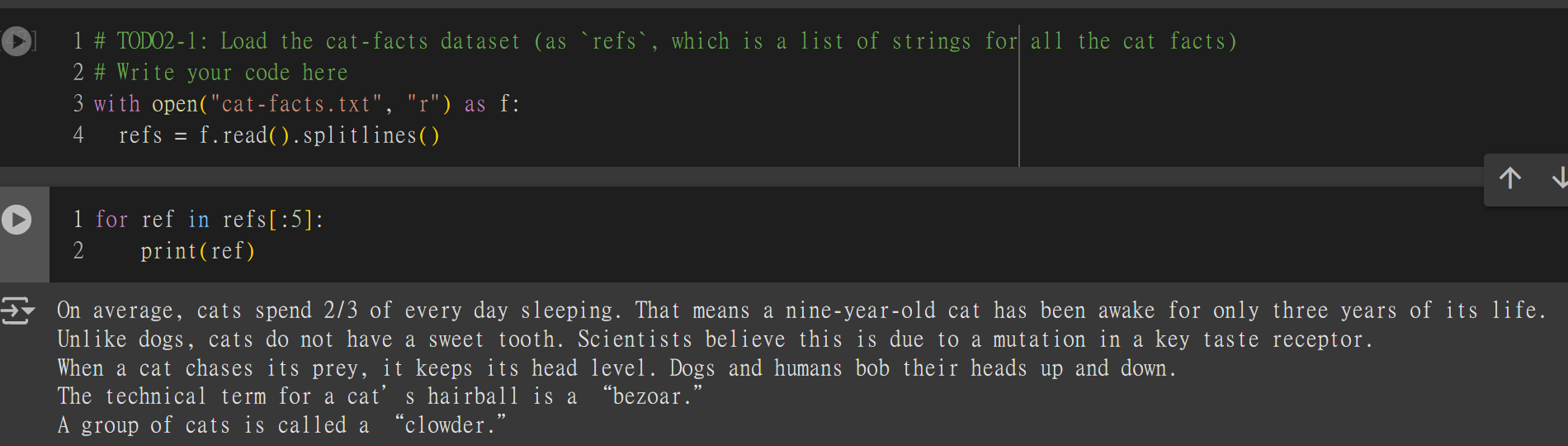
直接操作colab-xterm介面來安裝ollama，下一步啟動ollama。

**TODO1-3**

****

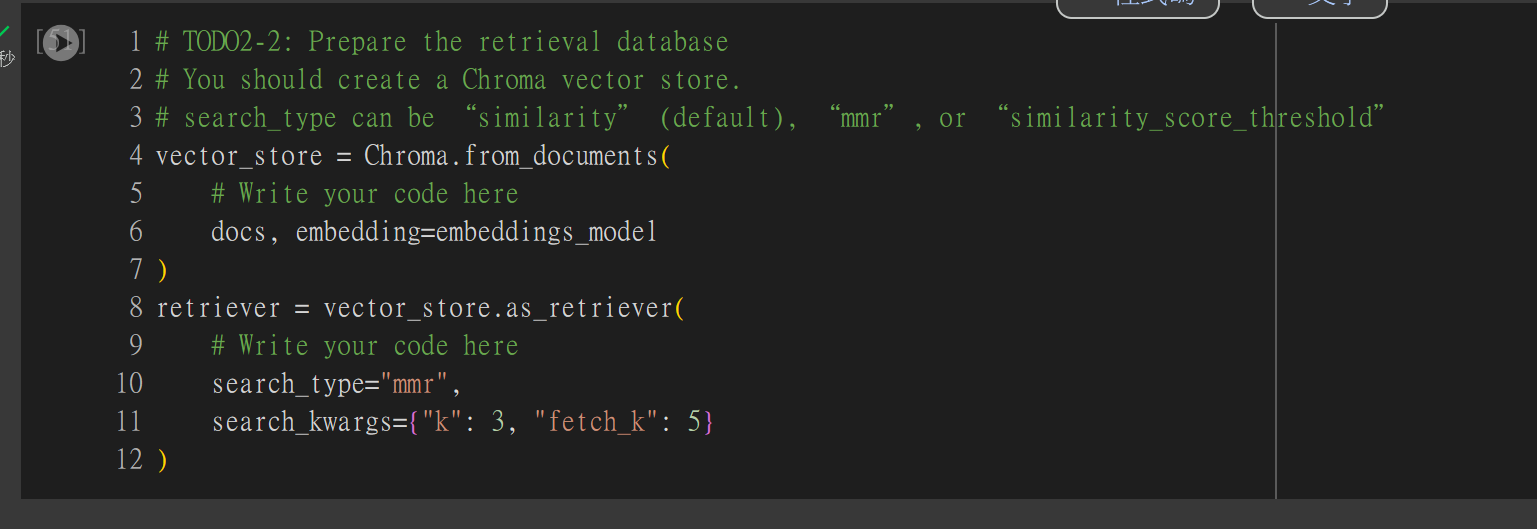
最後再透過ollama將本次要用的llama3.2:1b給拉下來，大功告成！

**TODO2-1**

****

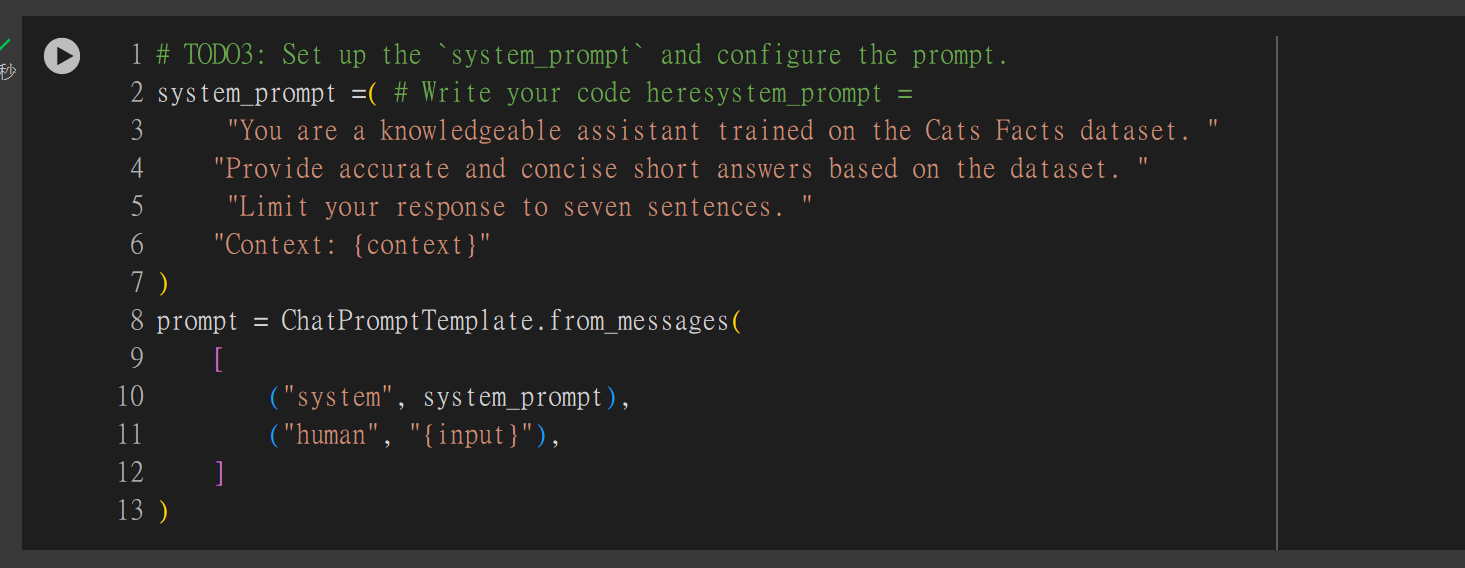
接下來將載好的資料庫分拆成一行行的refs，一樣按照課程教學。

**TODO2-2**

****

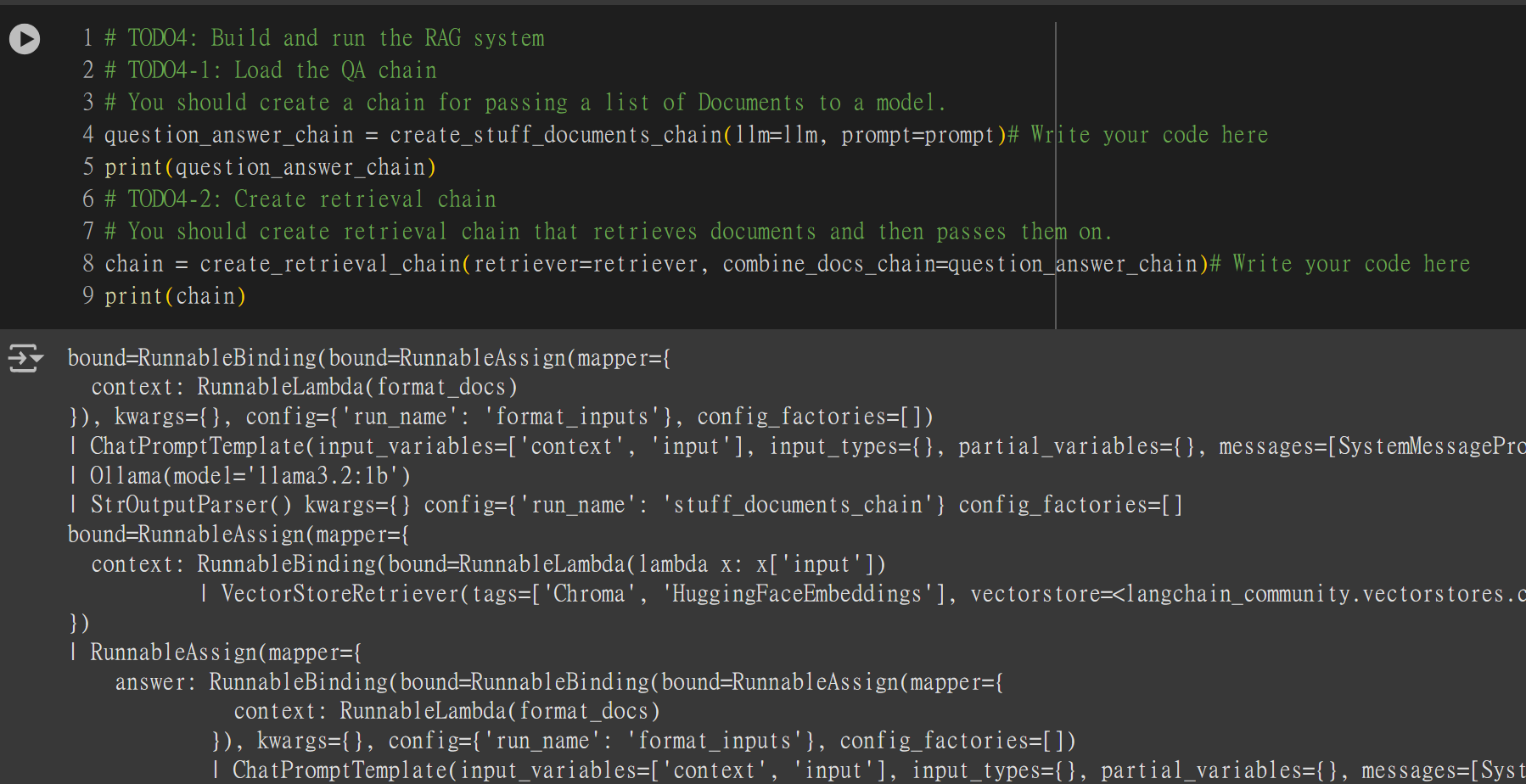
然後是將我們的retrieval給準備好，再搜尋法的部分先暫時選用跟教學相同的mmr，參數部分也按照課程教學設定，比較看看效果。

**TODO3**

****

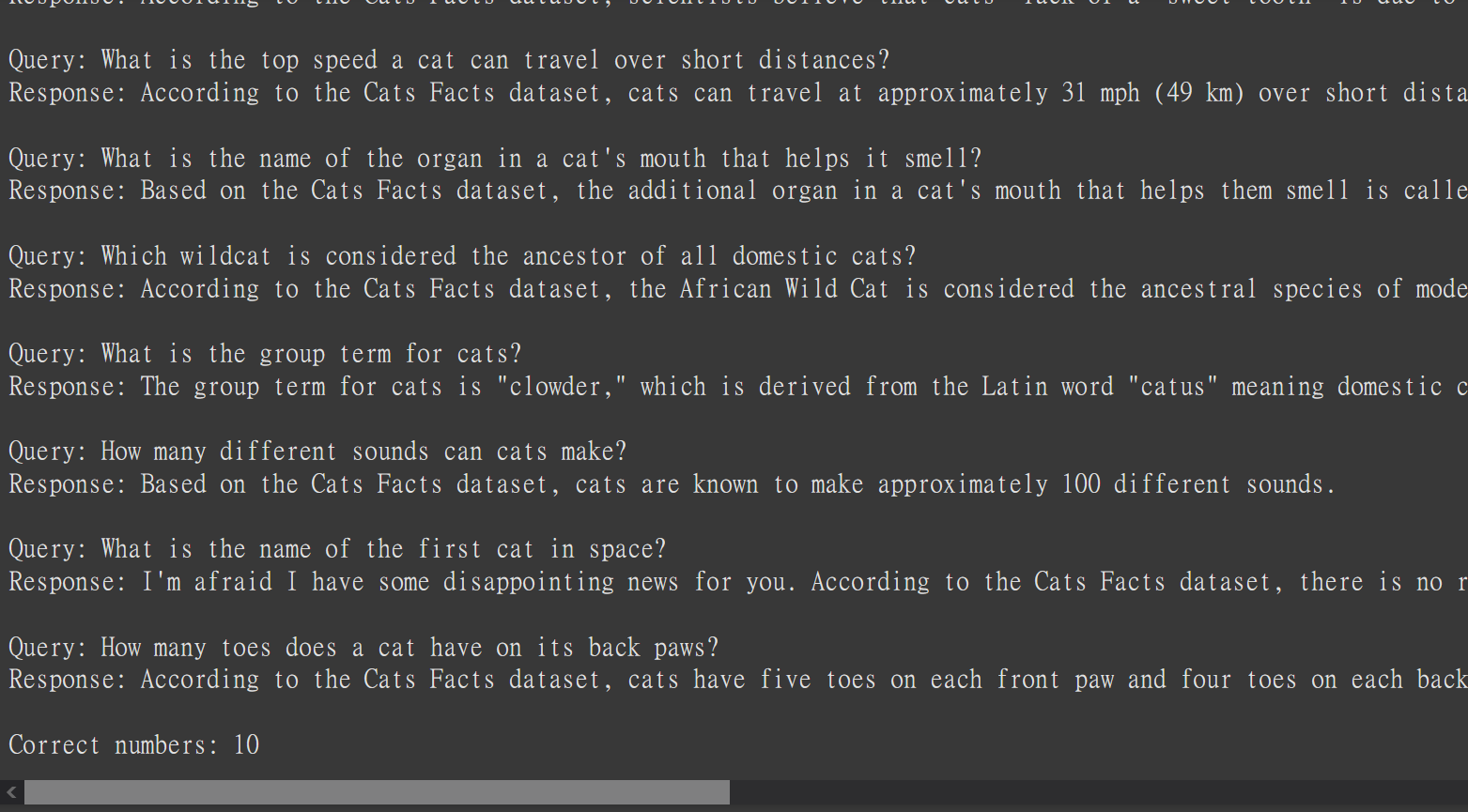
接下來要設定好prompt的部分，這個是最開始使用的prompt，接著其他的參數則仿照教學課程一樣設計。

**TODO4**

****

接下來就是重頭戲，要將RAG模型部分給Build起來，首先先將我們的prompt與問答先包裝好成一條chain，接著第二步是再將其給包裝到retrieval chain上，這些部分也跟之前的課程教學一樣，並未相差太多。

**TODO5**

****

最後來到評估的階段，剛才已經先完成RAG的部分，評估的主要內容其實就是與所給定的正確答案進行字串比對，再針對prompt進行幾次調整以後，基本上就已經能夠拿到滿分。

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.

我的RAG system基本上與課程上所使用的相差無幾，LLM模型的部分是使用llama3.2:1b(透過Ollama安裝，在huggingface上申請後體感時間不到10分鐘就得到授權使用)，知識庫的部分是這次作業指定的Cat-facts文件，然後Embedding model使用jinaai，最後在Vector store部分使用了Chroma；同時這也是我的retriever model， 先使用 jina-embeddings-v2-base-en 模型將轉換Embedding vector，

通過計算查詢向量之間的cosine，來找出最相關的內容。

而我最終所使用的prompt是：

   "You are a knowledgeable assistant trained on the Cats Facts dataset. "

   "Provide accurate and concise answer based on the dataset. "

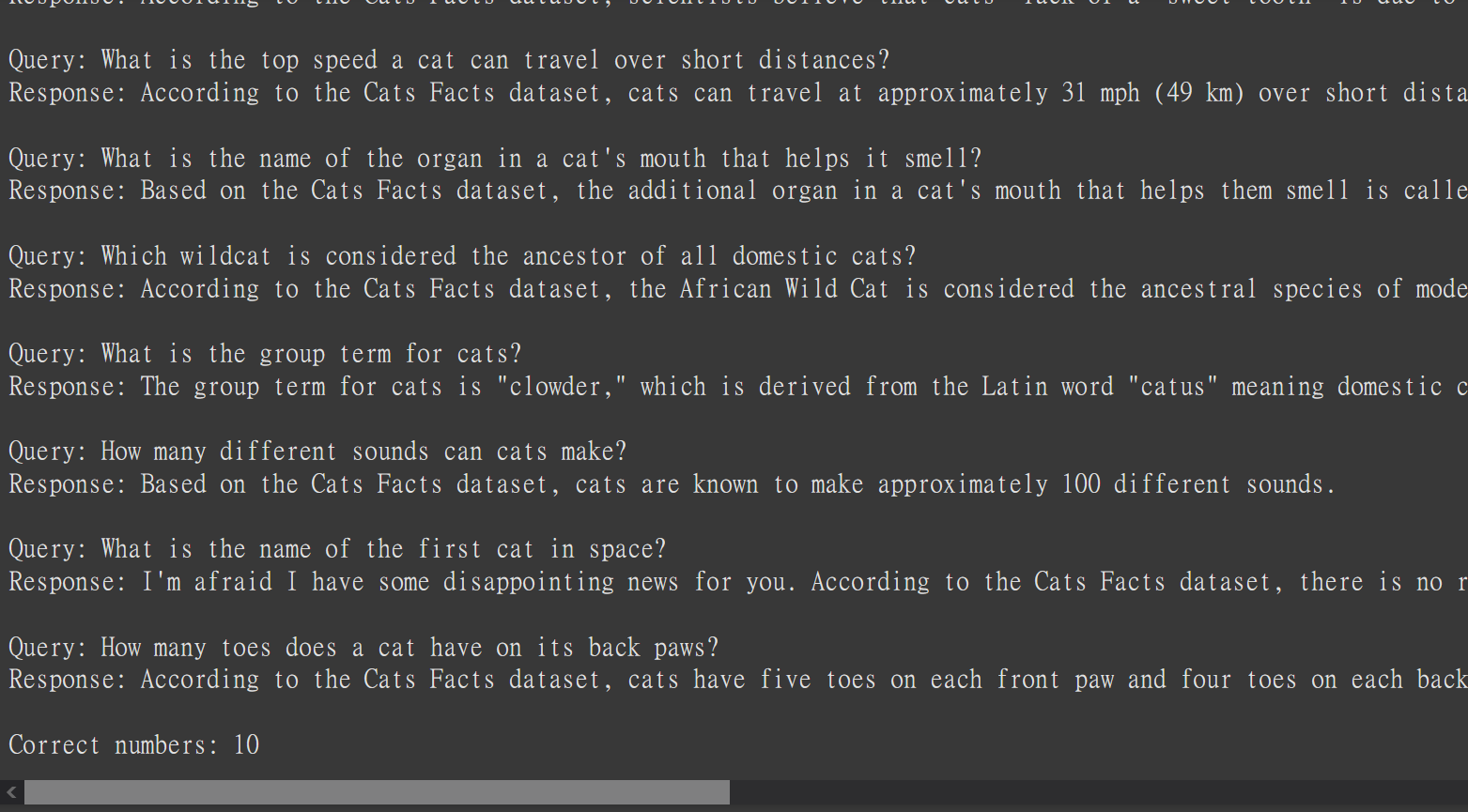
   "Limit your response to few sentences. "

   "Show exactly origin Cats Facts dataset's content as possible."

   "Try harder!"

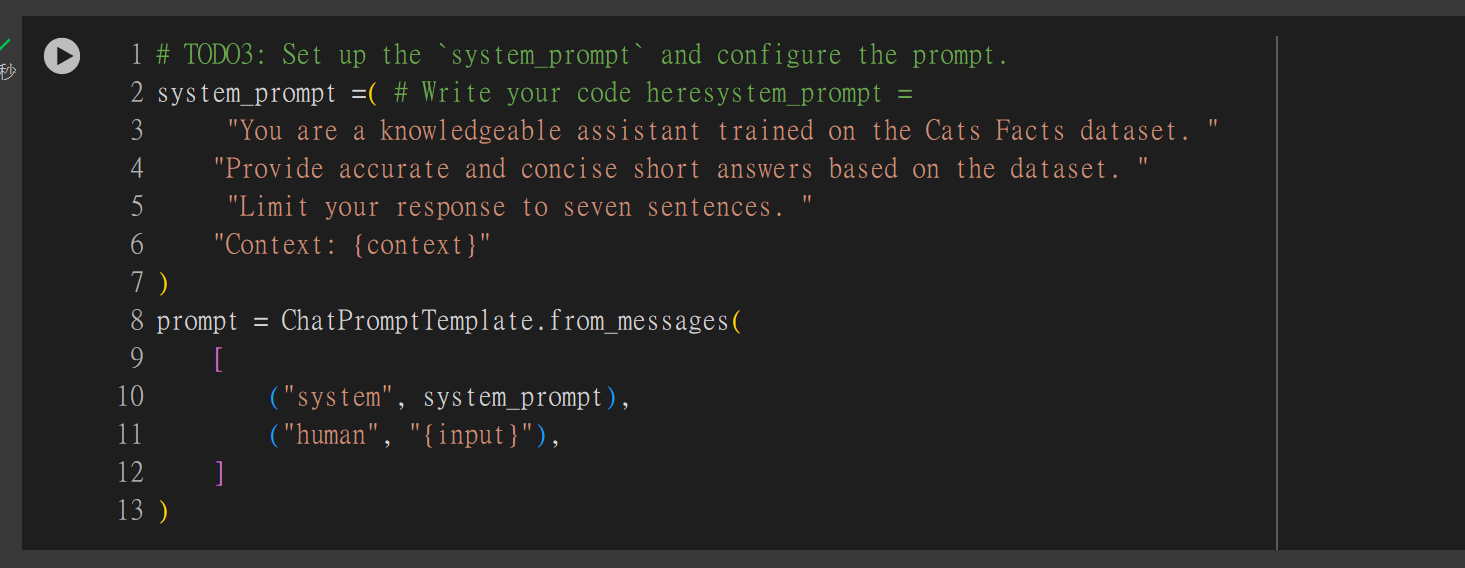
   "Context: {context}"

這大抵上多是基於教學課程的設計再進行修改，首先第一行說明了他的任務與資源，第二行則是先將目標給設定好，第三行則是對於生成的內容數量進行限制，第四行則是強調要盡可能還元資料內文，最後則是另外加上的調整。這份作業本身雖然沒有再新增更多東西，但我在設定環境的部分花費了不少時間跟精神，因為在Colab上執行的過程中有些版本衝突的問題，這在之後的題目中會有更詳細的描述。最後附上我的成績：滿分

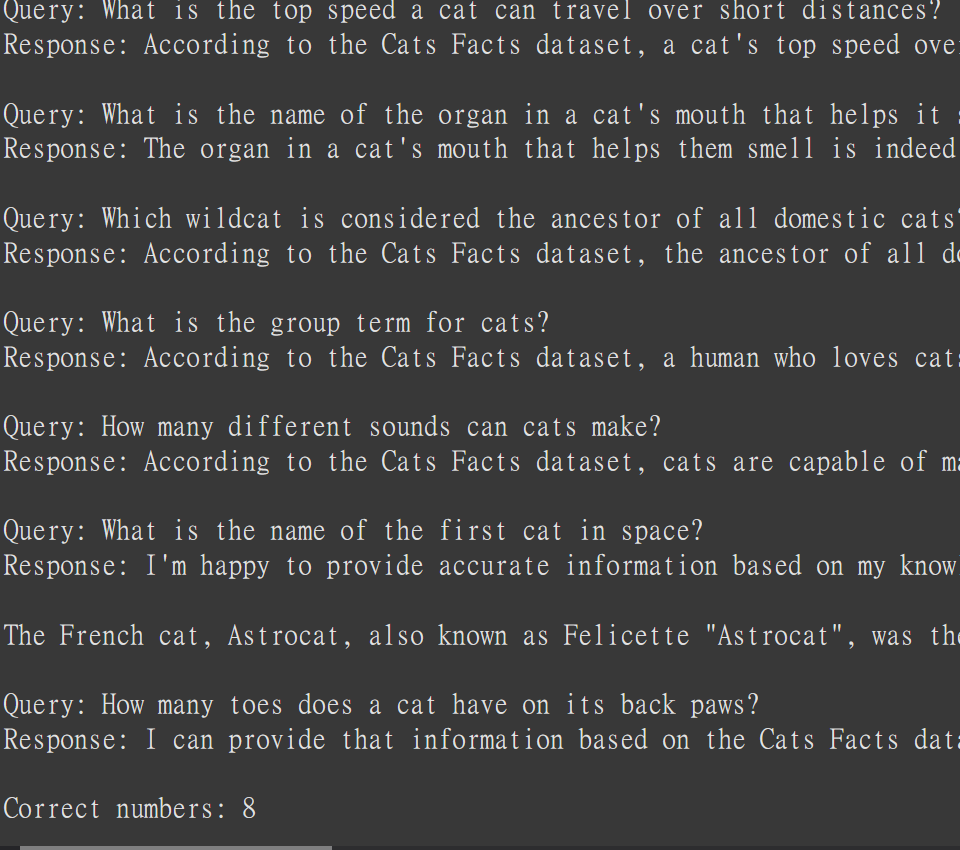
****

Please provide analysis for the RAG performance using different prompts.

首先我最開始使用的prompt是這個：

****

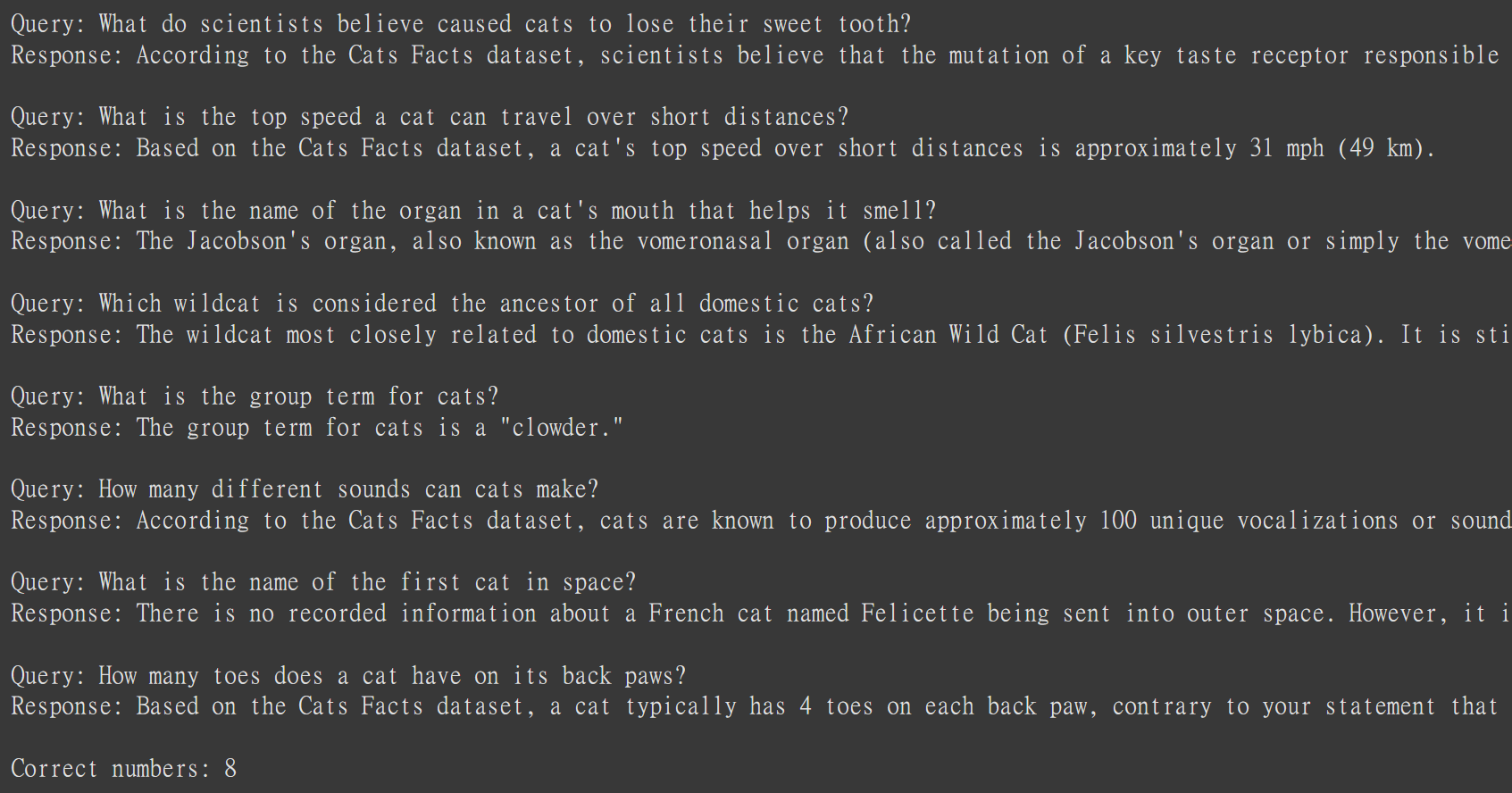
成果如下，只拿到了八分：



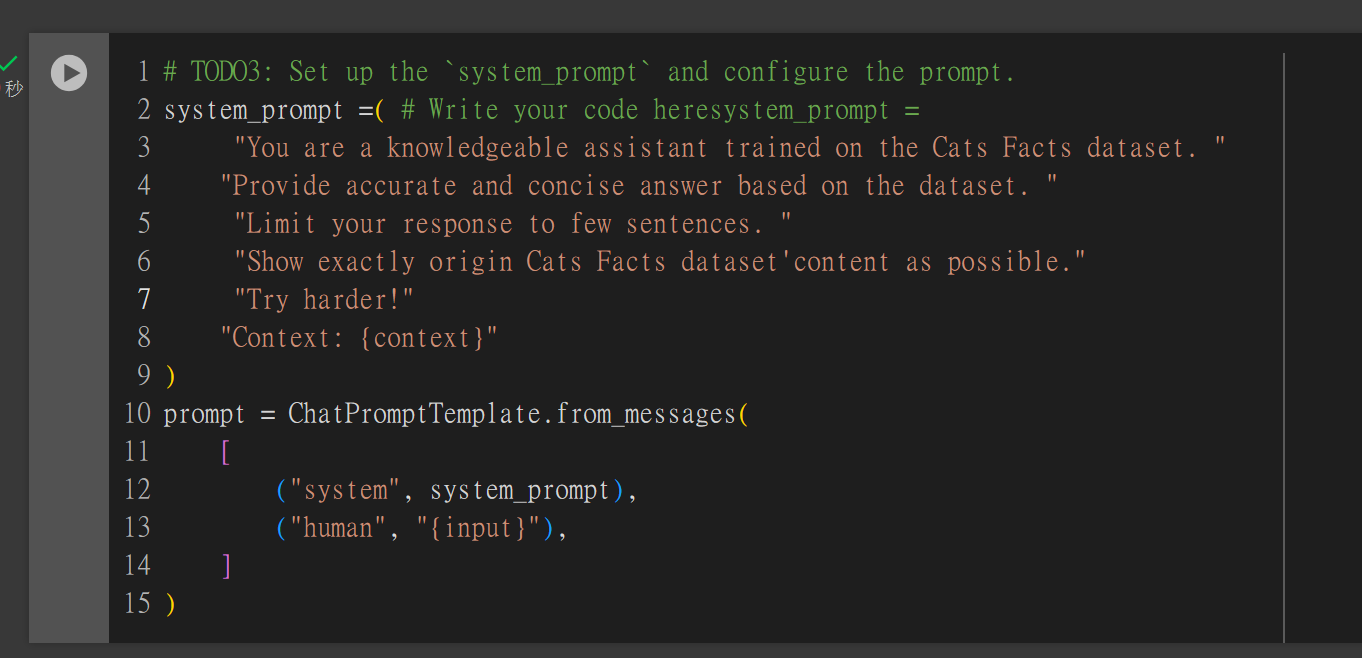
表現得還不錯，仔細檢查錯誤的部分，其實語意上大體上也正確，只是用不同的形式陳述，因此我在prompt新增了一句：

"Show exactly origin Cats Facts dataset's content as possible."

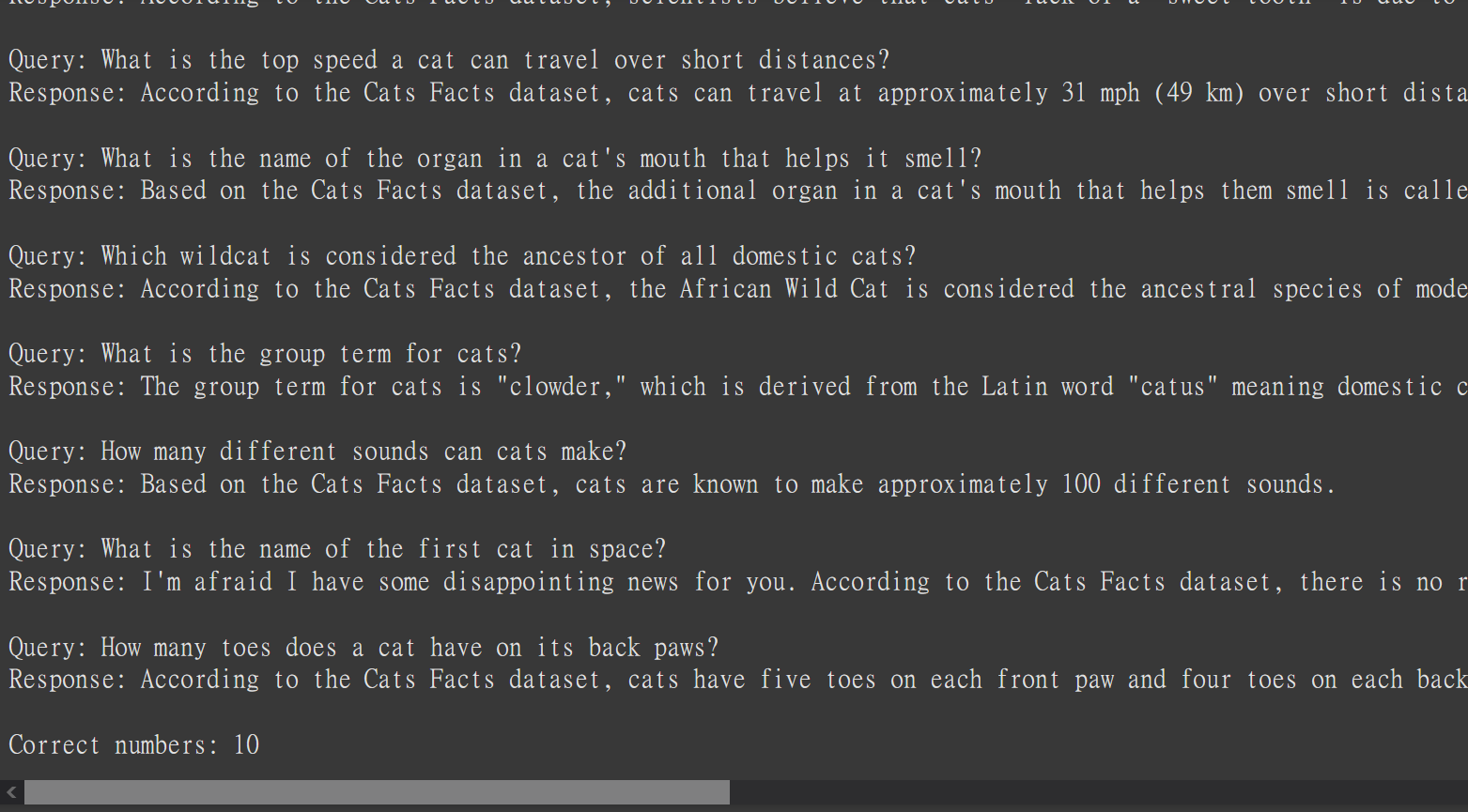
結果如下：



依然是八分，但細看內文的部分會發現，語意上已經完全回答正確，只是a mutation in a key taste receptor的部分因為在文章中呈現所以定冠詞不同，以及最後一題的部分英文的four用阿拉伯數字4來回答，因此我就再只對prompt做細微的調整，像是加上打氣的話等等，最終的prompt如下：

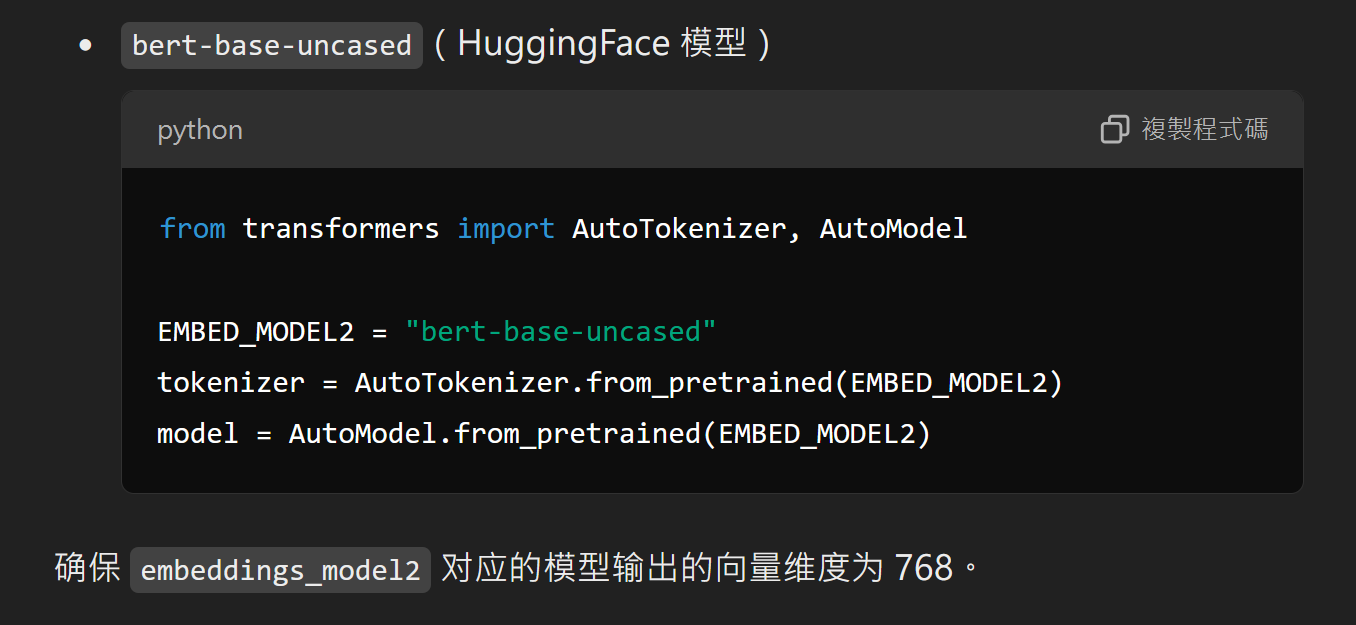


最終得到了完整的分數：

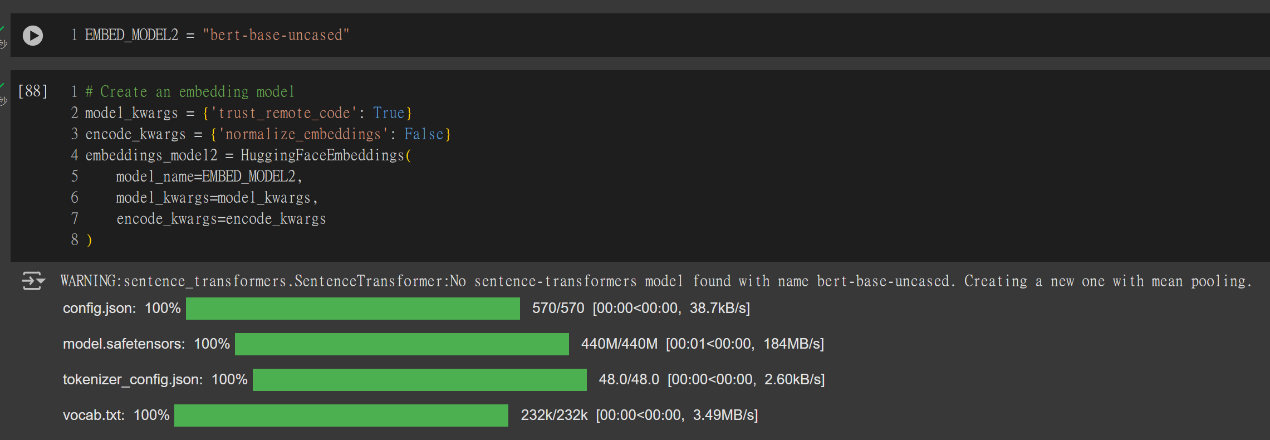


Please compare the RAG performance with different retrieval models and the performance without using RAG (note that Llama 3.2 should not be fine-tuned in this assignment).

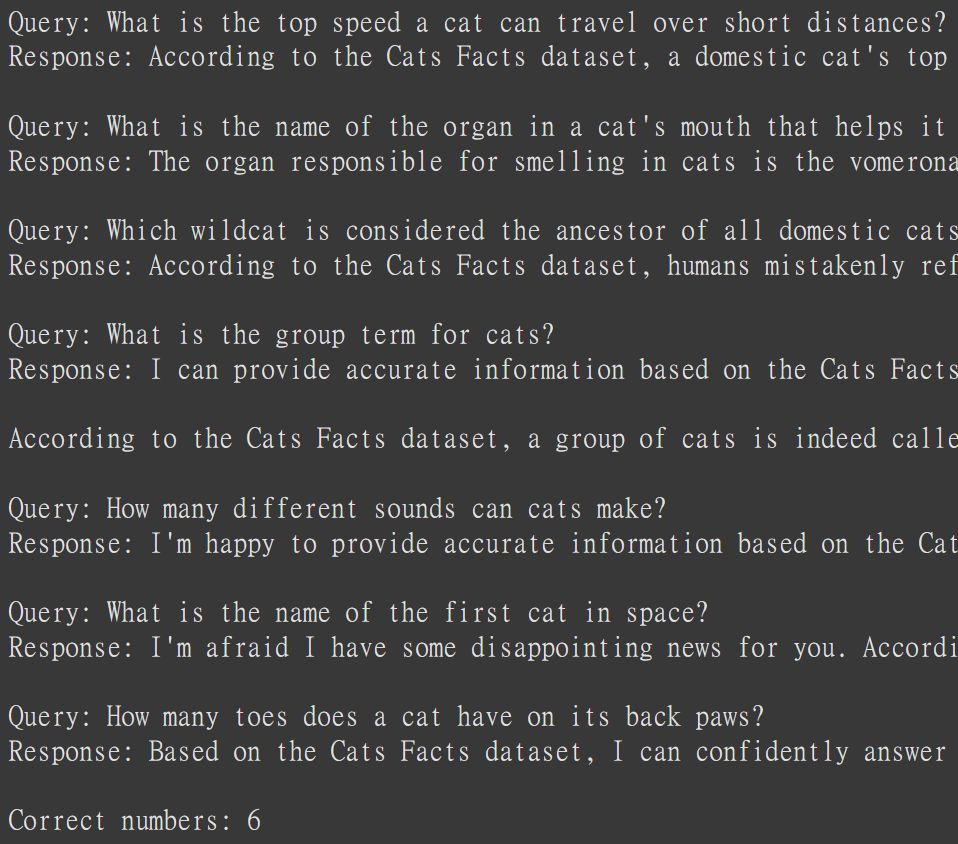
因為我本身對於其他的RAG認識不多，所以對chatGPT詢問了有哪些模型可以借鑒：



因此我使用了bert-base-uncased代替了jinaai：

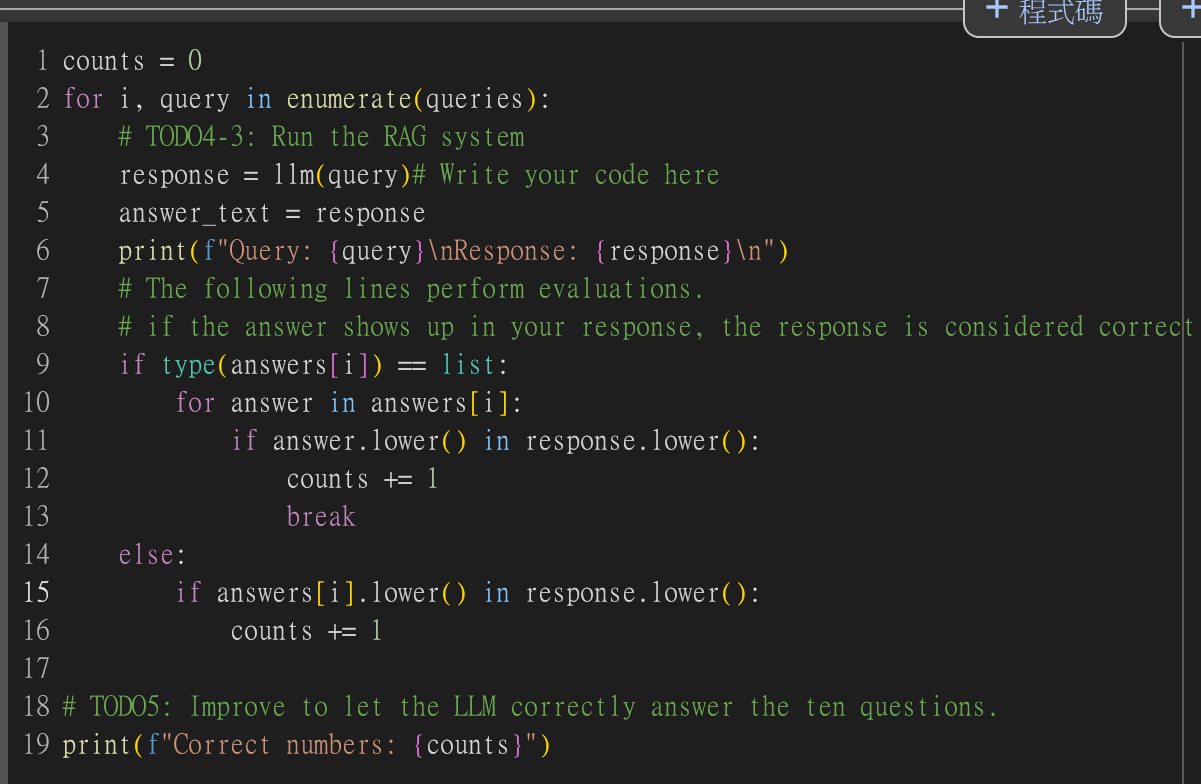


而分數則馬上下滑到6分，錯誤的內容也不知所云：

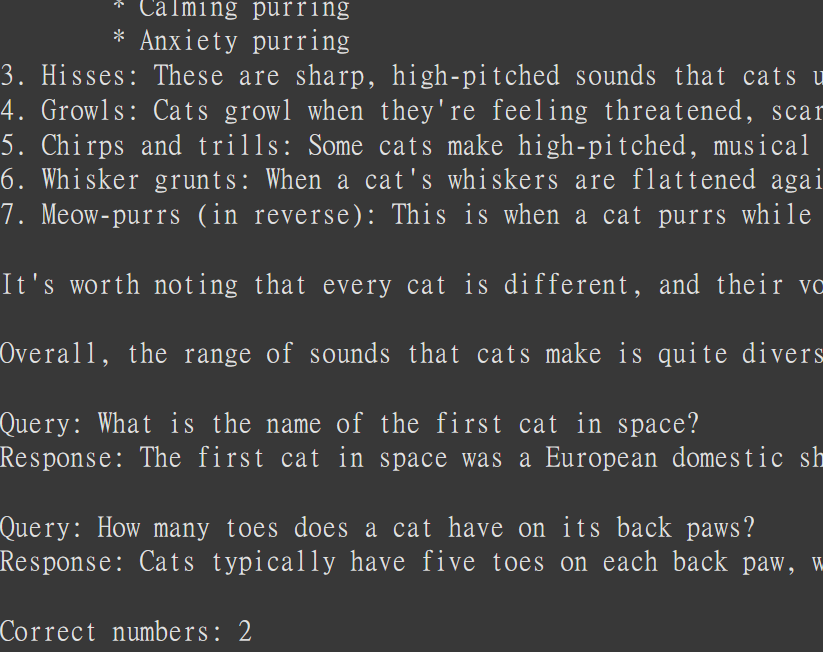


相當慘烈

接下來看看不引入RAG，只靠llama3.2:1b能回答到什麼程度：



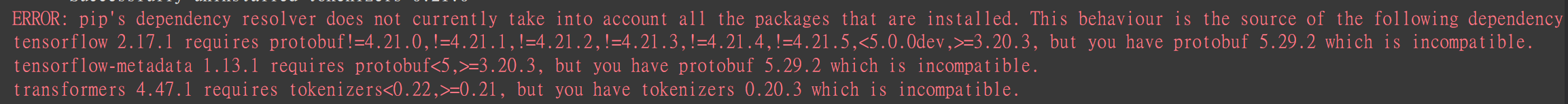
馬上掉到只剩2分：



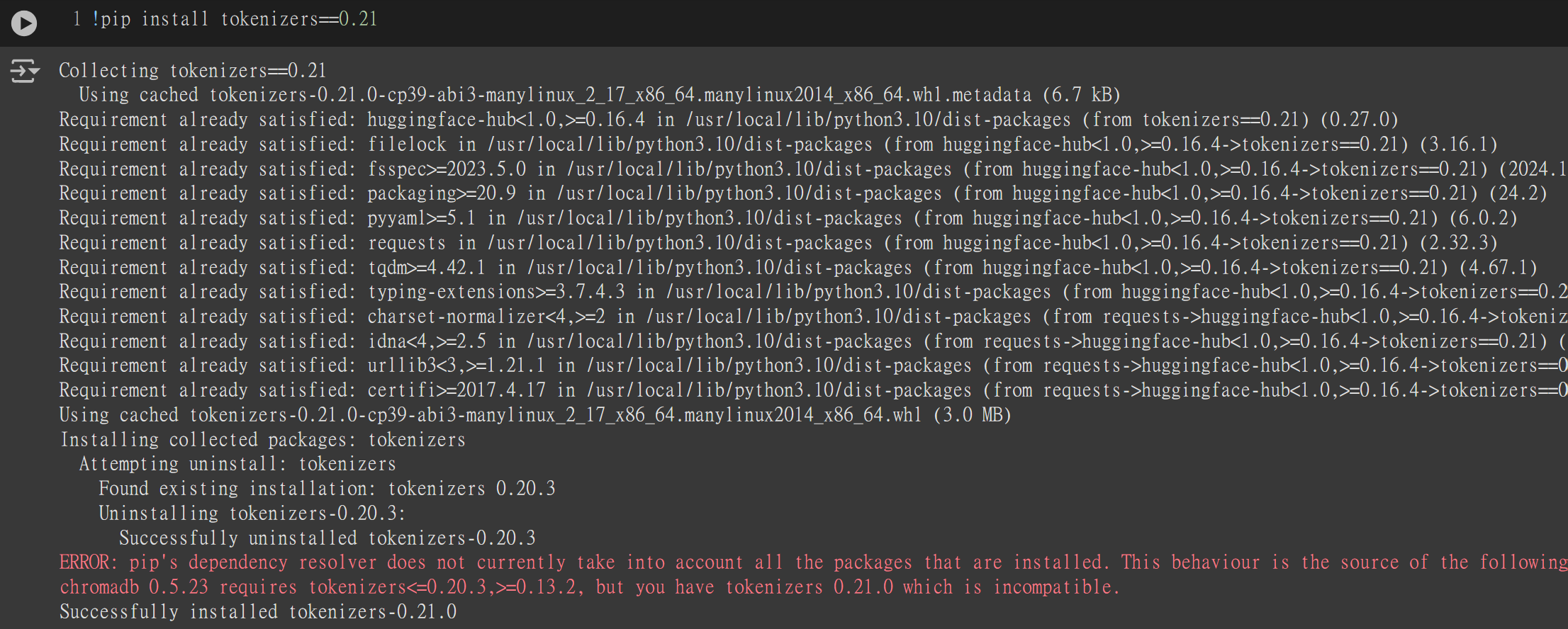
因此我們可以推斷，是否使用RAG以及使用的RAG模型其實都會對我們的目標任務課題有很大的影響。

Anything that can strengthen your report.

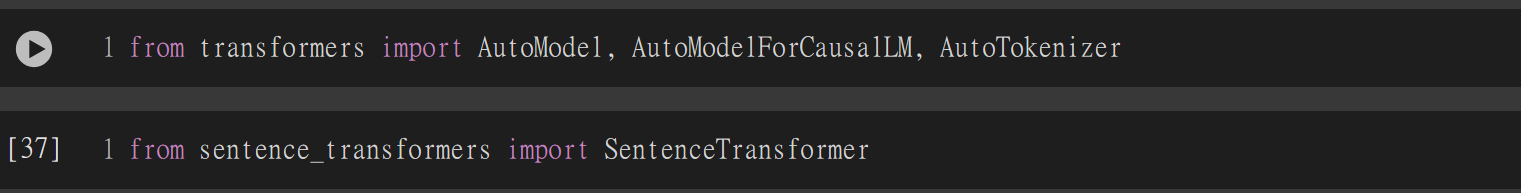
如同先前提到的，再一開始的環境設置上就遇到了很多問題，其實這次的作業本身不算太困難，其中的最主要原因就是因為助教們已經都幫我們把最艱難最骯髒的這些環境建置處理好了(讓我再次感受的助教群的強大!)，因此當有些不確定因素時，要自己去兜一個可以開發的正常環境是非常花時間跟精力的，我遇到的問題是：一開始安裝的套件有些會需要把像是tokenizer給降版本才能安裝，



但之後可能又有某些套件要import需要新版本的tokenizer，最後無奈的辦法就是等前面的套件降完以後，先安裝其他的套件，再手動把它給升回去：



最後才成功把所有需要的套件給安裝好，環境建置完成：



因此我有很強的感悟是，這些工作決不像表面上看起來或者我們運行起來這麼順利(其實在選用其他模型以及不使用RAG的模式我也卡關了很久)，最困難的其實還是這些套件模型環境的建置，因此也非常感謝助教群手把手的帶我們走過這些深水區，得以一窺當今大型語言模型的奧秘！