

# Assignment 4: Retrieval-Augmented Generation with LangChain

2025 NTHU Natural Language Processing

Hung-Yu Kao

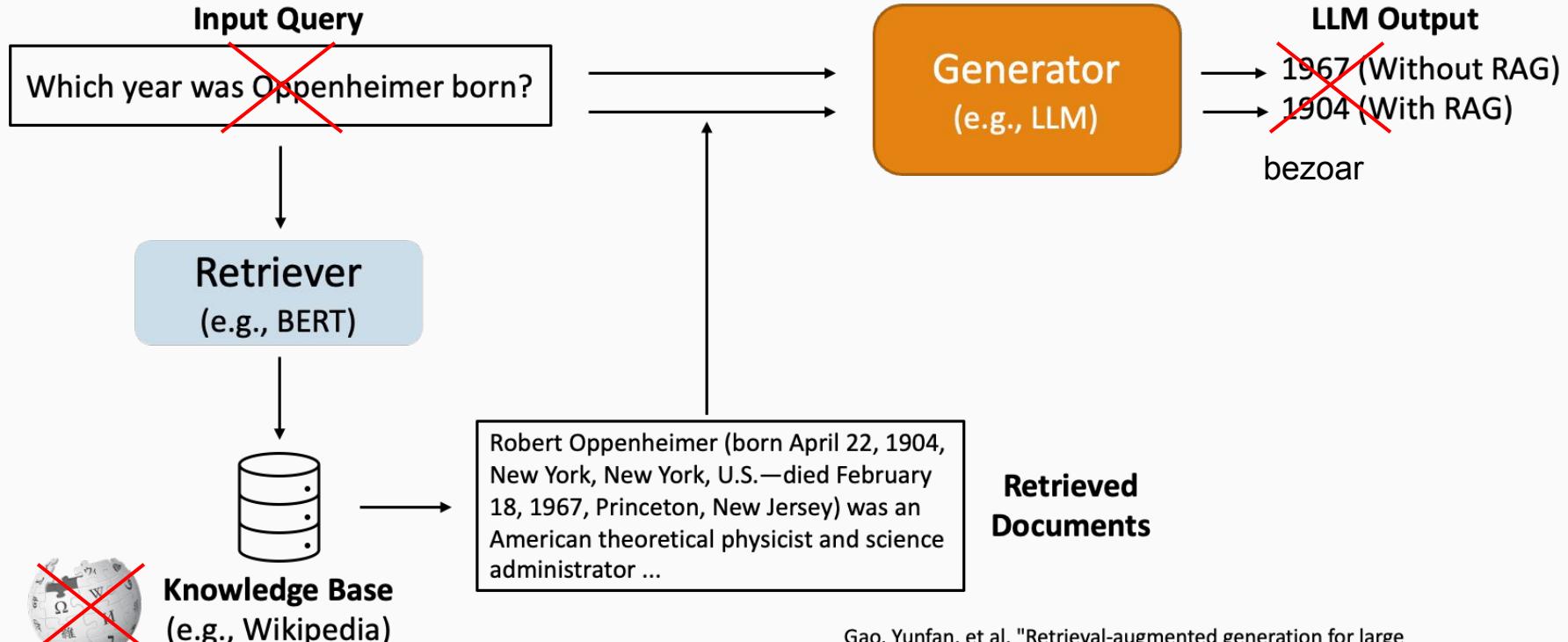
IKM Lab TAs

# Assignment Description

- Task: Retrieval-Augmented Generation (RAG) for **Question Answering**
- Database: Cats' knowledge in texts
- Inputs: **Questions** about cats' knowledge
- Answer: Short answers for the questions (evaluation: exact matching)
- Restricted models
  - Generator: Llama-3.2-1B (freezed)

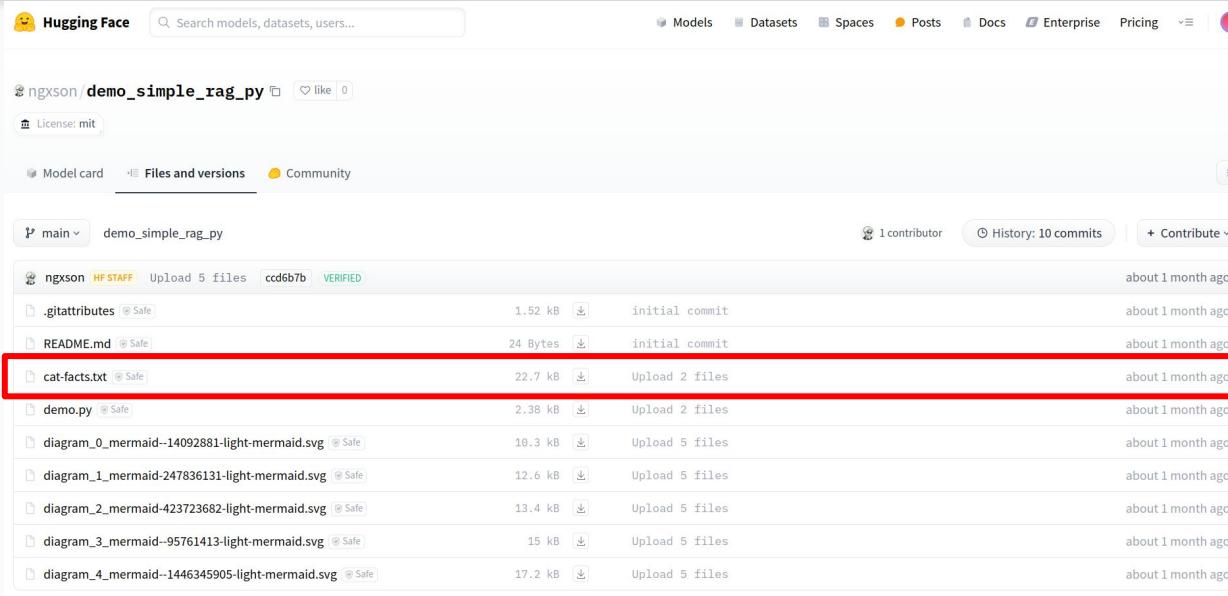
# Retrieval-Augmented Generation (RAG)

❓ What is the technical term for a cat's hairball?



Gao, Yunfan, et al. "Retrieval-augmented generation for large language models: A survey." *arXiv preprint arXiv:2312.10997* (2023).

# Dataset: Cat-facts



The screenshot shows the Hugging Face Model card for the repository `demo_simple_rag_py`. The card includes a search bar, navigation links for Models, Datasets, Spaces, Posts, Docs, Enterprise, and Pricing, and a user profile icon.

The repository details shown are:

- Owner: `ngxson`
- License: `mit`
- Contributors: 1 contributor
- Commits: History: 10 commits | + Contribute

The `Files and versions` tab is selected, showing the following file list:

File	Size	Last Commit
<code>.gitattributes</code>	1.52 kB	initial commit
<code>README.md</code>	24 Bytes	initial commit
<code>cat-facts.txt</code>	22.7 kB	Upload 2 files
<code>demo.py</code>	2.38 kB	Upload 2 files
<code>diagram_0_mermaid-14092881-light-mermaid.svg</code>	10.3 kB	Upload 5 files
<code>diagram_1_mermaid-247836131-light-mermaid.svg</code>	12.6 kB	Upload 5 files
<code>diagram_2_mermaid-423723682-light-mermaid.svg</code>	13.4 kB	Upload 5 files
<code>diagram_3_mermaid-95761413-light-mermaid.svg</code>	15 kB	Upload 5 files
<code>diagram_4_mermaid-1446345905-light-mermaid.svg</code>	17.2 kB	Upload 5 files

The file `cat-facts.txt` is highlighted with a red rectangle.

The dataset is presented in `ngxson/demo_simple_rag_py`

# Dataset

## Cat-facts dataset

- Database size: 150 facts
- Test QA: 150 QA pairs (generated by GPT-5)
- Each fact is represented as a sentence.
  - E.g., When a cat chases its prey, it keeps its head level. Dogs and humans bob their heads up and down.
  - E.g., The technical term for a cat's hairball is a “bezoar.”
  - ...

# Construction for Questions and Answers

Index	Question	Answer Sentence in the database	Answer we set
1	Why don't cats have a sweet tooth?	Unlike dogs, cats do not have a sweet tooth. Scientists believe this is due to a <b>mutation</b> in a key <b>taste</b> receptor.	Taste mutation
2	How many sounds do cats make?	Cats make <b>about 100</b> different sounds. Dogs make only about 10.	About 100
3	How many domestic cats exist worldwide?	There are more than <b>500 million</b> domestic cats in the world, with approximately 40 recognized breeds.	500 million
4	How many people are bitten by cats annually in the U.S.?	Approximately <b>40,000</b> people are bitten by cats in the U.S. annually.	40,000
5	What is a cat's top running speed?	A cat can travel at a top speed of approximately <b>31 mph</b> (49 km) over a short distance.	31 mph

# Code (hints only)

You can modify the code template this time!

# Connect to Huggingface (1/4)

Step1

The screenshot shows the Huggingface homepage. At the top right, there is a user profile icon with a red arrow pointing to it. A red box highlights the "Profile" section, which displays the name "ZhiQuanFeng". Below this, there is a sidebar with various options like "Trending", "All", "Models", "Datasets", "Inbox (0)", and "Sign Out". On the right side, there is a large circular profile picture placeholder. Below the picture, the user's name is displayed as "Zhi-Quan Feng" and their handle as "ZhiQuanFeng". There are two buttons: "Edit profile" and "Settings". Red arrows point from the "Profile" text in the sidebar to the "Profile" section in the top right, and from the "Edit profile" button to the "Edit profile" button on the right.

Step2

## Connect to Huggingface (2/4)

Step4



Zhi-Quan Feng  
ZhiQuanFeng

- Profile
- Account
- Authentication
- Organizations
- Billing
- Access Tokens**
- SSH and GPG Keys
- Webhooks
- Papers
- Notifications
- Local Apps and Hardware NEW
- Gated Repositories

Step3

### Access Tokens

#### User Access Tokens

+ Create new token

Access tokens authenticate your identity to the Hugging Face Hub and allow applications to perform actions based on token permissions. ⚠ Do not share your **Access Tokens** with anyone; we regularly check for leaked Access Tokens and remove them immediately.

Name	Value	Last Refreshed Date	Last Used Date	Permissions	⋮
NLP_HW	hf...pgJF	about 16 hours ago	about 15 hours ago	FINEGRAINED	⋮
my_token	hf...kEJe	Sep 24	5 days ago	FINEGRAINED	⋮

# Connect to Huggingface (3/4)

Create new Access Token

Token type  
 Fine-grained    Read    Write  
⚠️ This cannot be changed after token creation.

Token name

User permissions (ZhiQuanFeng)

**Repositories**

- Read access to contents of all repos under your personal namespace
- Read access to contents of all public gated repos you can access
- Write access to contents/settings of all repos under your personal namespace

**Inference**

- Make calls to the serverless Inference API
- Make calls to Inference Endpoints
- Manage Inference Endpoints

**Webhooks**

- Access webhooks data
- Create and manage webhooks

**Collections**

- Read access to all collections under your personal namespace
- Write access to all collections under your personal namespace

Step5  
Type a name here  
(Can be arbitrary)

Org permissions  
None if not specified.

Search for orgs

**Repositories**

- Read access to contents of all repos in selected organizations
- Interact with discussions / Open pull requests on repos in selected organizations
- Write access to contents/settings of all repos in selected organizations

**Inference endpoints** ⓘ

- Make calls to inference endpoints
- Manage inference endpoints

**Org settings**

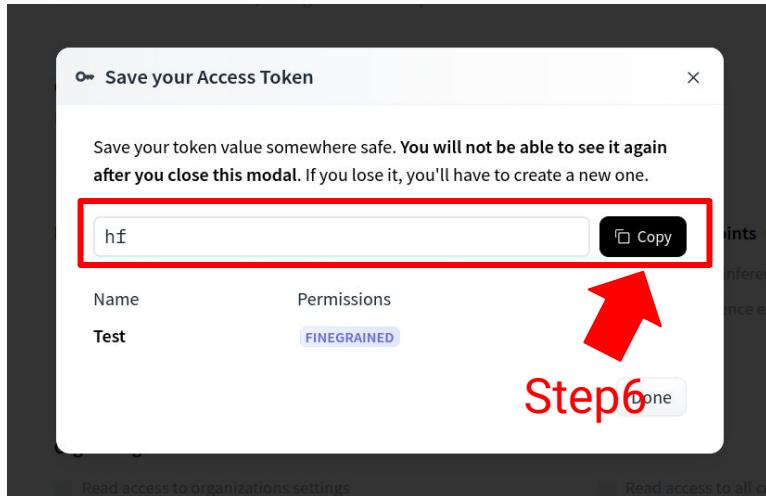
- Read access to organizations settings
- Write access to organizations settings / member management

**Collections**

- Read access to all collections in selected organizations
- Write access to all collections in selected organizations

Step6

## Connect to Huggingface (4/4)



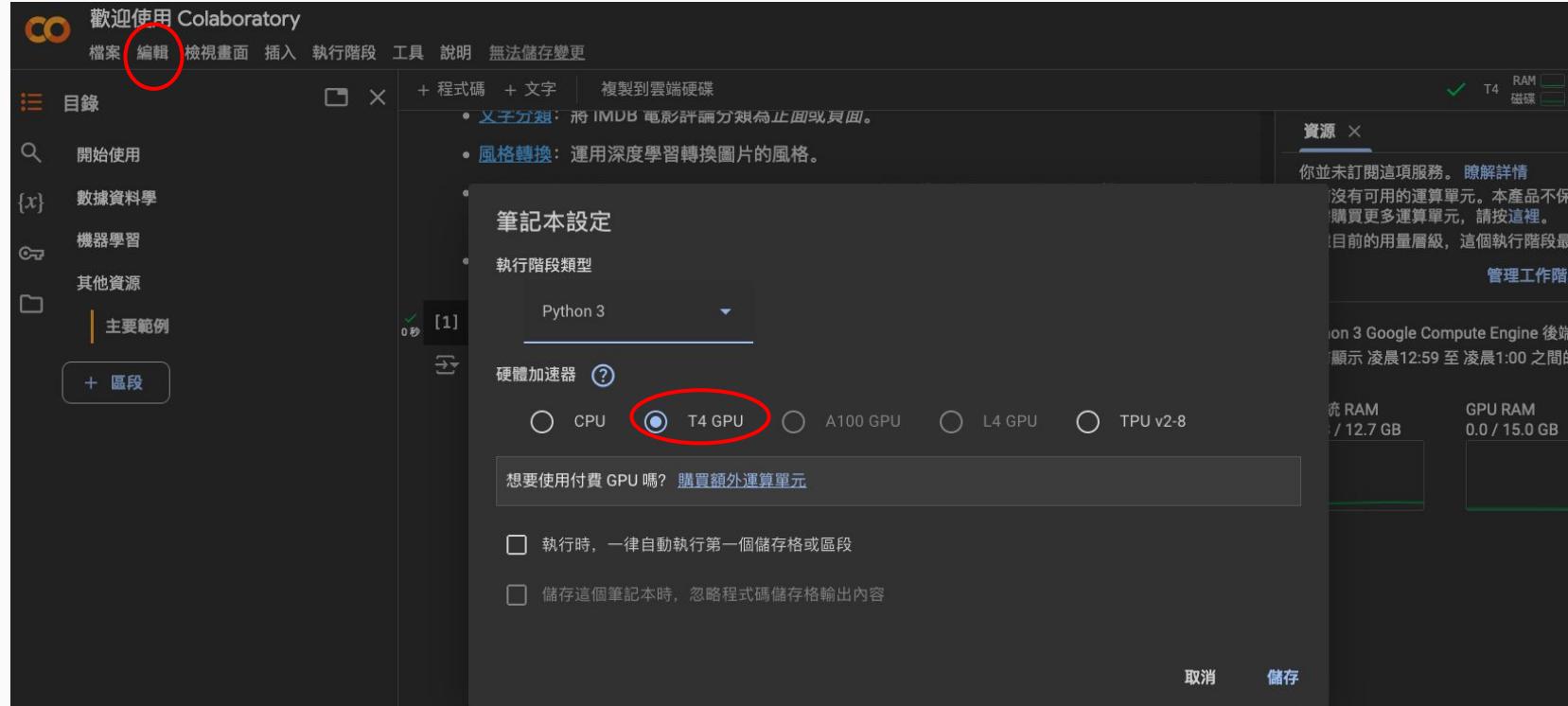
```
▶ 1 from huggingface_hub import login  
2  
3 hf_token = "  
4 login(token=hf_token, add_to_git_credential=True)  
  
[6] 1 !huggingface-cli whoami  
↳ ZhiQuanFeng
```

A large red box highlights the string value of the variable 'hf\_token' in the first line of code. Two red arrows point upwards from the bottom of the code area towards this highlighted box. To the right of the code, the text 'Step7 Your access token here' is displayed.

Step7  
Your access token here

If you succeed, your Hugging Face ID will be displayed here.

# How to use GPU on Colab?



Type in a code block: `!nvidia-smi`

# TODO1: Set up the environment of Ollama (1/3)

## Introduction to Ollama

- Ollama is a platform designed for running and managing large language models (LLMs) directly **on local devices**, providing a balance between performance, privacy, and control.
- There are also other tools support users to manage LLM on local devices and accelerate it like *vllm*, *Llamofile*, *GPT4ALL*...etc.

### ▼ Launch colabxterm

```
[ ] 1 # TOD01-1: You should install colab-xterm and launch it.  
2 # Write your commands here.
```

```
[ ] 1 # TOD01-2: You should install Ollama.  
2 # You may need root privileges if you use a local machine instead of Colab.
```

Install the colab-xterm package and use “load\_ext” to use it.

# TOD01: Set up the environment of Ollama (2/3)

```
[ ] 1 %xterm
```

```
[ ] 1 # TOD01-3: Pull Llama3.2:1b via Ollama and start the Ollama service in the xterm  
2 # Write your commands in the xterm
```

Use "%xterm" and the followint command to the command line:

- curl -fsSL https://ollama.com/install.sh | sh
- Download Llama3.2-1b using pull command in Ollama
- ollama serve

# A simple test of Ollama

## ▼ Ollama testing

You can test your Ollama status with the following cells.

```
[ ] 1 # Setting up the model that this tutorial will use
2 MODEL = "llama3.2:1b" # https://ollama.com/library/llama3.2:3b
3 EMBED_MODEL = "jinaai/jina-embeddings-v2-base-en"
```

```
[ ] 1 # Initialize an instance of the Ollama model
2 llm = Ollama(model=MODEL)
3 # Invoke the model to generate responses
4 response = llm.invoke("What is the capital of Taiwan?")
5 print(response)
```

- Here we define the retrieval model and document reader model.
- We can perform a simple test for checking Ollama's status.
- The .invoke API is to input sequence to the model and obtain the outputs.

## Define the retriever model

```
[ ] 1 # Create an embedding model
2 model_kwargs = {'trust_remote_code': True}
3 encode_kwargs = {'normalize_embeddings': False}
4 embeddings_model = HuggingFaceEmbeddings(
5     model_name=EMBED_MODEL,
6     model_kwargs=model_kwargs,
7     encode_kwargs=encode_kwargs
8 )
```

Build a simple document retriever using LangChain (HuggingFaceEmbeddings).

# TOD02: Load the cat-facts dataset and prepare the retrieval database

```
[ ] 1 !wget https://huggingface.co/ngxson/demo_simple_rag_py/resolve/main/cat-facts.txt  
  
[ ] 1 # TODO2-1: Load the cat-facts dataset (as `refs`, which is a list of strings for all the cat facts)  
2 # Write your code here  
  
[ ] 1 from langchain_core.documents import Document  
2 docs = [Document(page_content=doc, metadata={"id": i}) for i, doc in enumerate(refs)]
```

Load the data to be retrieved and use the `langchain.docstore.document.Document` API to construct a database.

# TOD02: Load the cat-facts dataset and prepare the retrieval database

```
[ ] 1 # TOD02-2: Prepare the retrieval database
2 # You should create a Chroma vector store.
3 # search_type can be "similarity" (default), "mmr", or "similarity_score_threshold"
4 vector_store = Chroma.from_documents(
5     # Write your code here
6 )
7 retriever = vector_store.as_retriever(
8     # Write your code here
9 )
```

## TODO3: Set up the `system\_prompt` and configure the prompt

### ▼ Prompt setting

```
[ ] 1 # TODO3: Set up the `system_prompt` and configure the prompt.  
2 system_prompt = # Write your code here  
3 prompt = ChatPromptTemplate.from_messages(  
4     [  
5         ("system", system_prompt),  
6         ("human", "{input}"),  
7     ]  
8 )
```

- For the vectorspace, the common algorithm would be used like Faiss, Chroma...(<https://python.langchain.com/docs/integrations/vectorstores/>) to deal with the extreme huge database.

Write a system prompt to guide the document reader model in performing the desired task.

# TODO4: Build and run the RAG system

```
[ ] 1 # TODO4: Build and run the RAG system
2 # TODO4-1: Load the QA chain
3 # You should create a chain for passing a list of Documents to a model.
4 question_answer_chain = # Write your code here
5                         stuff_documents_chain
6 # TODO4-2: Create retrieval chain
7 # You should create retrieval chain that retrieves documents and then passes them on.
8 chain = # Write your code here
9          retrieval_chain
```

# The Question and Answer Pairs

Load

```
How much of a day do cats spend sleeping?  
Two thirds  
  
Why don't cats have a sweet tooth?  
Taste mutation  
  
How do cats keep their heads when chasing prey?  
Head level  
  
What is the technical term for a cat's hairball?  
Bezoar  
  
What is a group of cats called?  
Clowder|  
  
Which paw do most female cats prefer?  
Right paw  
  
Why can't cats climb down trees head first?  
Claw direction  
  
How many sounds do cats make?  
About 100
```

```
1 # Question (queries) and answer pairs  
2 # Write your code here  
3 # Please load the questions_answers.txt file and prepare the `queries` and `answers` lists.  
4 queries = [  
5 # Questions queries  
6 ]  
7 answers = [  
8 # Corresponding answers  
9 ]
```

The queries (the ten questions) and the correct answers are provided in the notebook file.

There are 150 QA pairs, each corresponding to an entry in the Cat-Fact dataset, and they appear in the same order as in the original dataset.

# The Question and Answer Pairs

- 
- [1] On average, cats spend 2/3 of every day sleeping. That means a nine-year-old cat has been awake for only three years of its life.
  - [2] Unlike dogs, cats do not have a sweet tooth. Scientists believe this is due to a mutation in a key taste receptor.
  - [3] When a cat chases its prey, it keeps its head level. Dogs and humans bob their heads up and down.
  - [4] The technical term for a cat's hairball is a "bezoar."
  - [5] A group of cats is called a "clowder."

QA pair for [1] How much of a day do cats spend sleeping?  
Two thirds

The first QA pair corresponds to the first data,

QA pair for [2] Why don't cats have a sweet tooth?  
Taste mutation

the second QA pair corresponds to the second data...

QA pair for [3] How do cats keep their heads when chasing prey?  
Head level

The data in cat-fact and QA pairs are in the same order.

QA pair for [4] What is the technical term for a cat's hairball?  
Bezoar

QA pair for [5] What is a group of cats called?  
Clowder

TOD04-3 and TOD05: Improve to let the LLM correctly answer the ten questions.

Input the query to the RAG system and get the response

```
1 for i, query in tqdm(enumerate(queries), total=len(queries)):  
2     # TOD04-3: Run the RAG system  
3     response = # Write your code here  
4     # The following lines perform evaluations.  
5     # if the answer shows up in your response, the response is considered correct.  
6     # Compute recall@1, recall@5 and Accuracy.  
7     # Store the questions, ground-truths and answers in a json file.  
8  
9 # TOD05: Improve to let the LLM correctly answer the ten questions.
```

Evaluate the response

Print the scores

# Submission

# Scoring for TODOs

TODO	Score
TODO1: Set up the environment of Ollama	5%
TODO2: Load the cat-facts dataset and prepare the retrieval database	10%
TODO3: Set up the `system_prompt` and configure the prompt.	10%
TODO4: Build and run the RAG system <small>(Your submitted verion should use jina-embeddings-v2-base-en and Llama3.2-1b model, if not, your score for TODO4 will be reduced by 50%)</small>	10%
TODO5: Improve to let the LLM correctly answer the 150 questions. <small>(Use Llama3.2-1b)</small>	10%

You must upload a JSON file that includes all **test questions**, their **ground-truth answers**, and your corresponding **predictions**. The JSON file is like:

```
[{"Query": ... , "Ground_Truth": ..., "Prediction": ...}, {"Query": ... , "Ground_Truth": ..., "Prediction": ...}, ...]
```

# Scoring

Coding work : **45%**

- You must upload a JSON file that includes all **test questions**, their **ground-truth answers**, and your corresponding **predictions**. Additionally, your report must include a screenshot showing your testing logs and the resulting accuracy.
- For retrieval please report recall@1 and recall@5 scores, for generation, please report the exact match (EM) score.
- If either of these required items is missing, your score for TODO5 will be reduced by 50%.

Report: **55%**

- (5%) Please describe the **details of your implementation for the RAG system** (please tell us 1. What's in your RAG system? 2. What's your prompt? 3. What's new in your code in comparison with the code from our lab course?) in this assignment.
- (10%) Please provide analysis for the RAG performance using different prompts in **generator** model.
- (10%) Please analyze the RAG performance under different input data formats provided to the **retriever**.
- (10%) Please provide analysis for the RAG performance using different **input order** of the retrieved documents to the generator model.
- (10%) Please analyze the generation performance of the model when **counterfactual information** is introduced into the input sequence of the generator.
- (10%) Anything that can strengthen your report.

# Delivery policies: File formats

- Coding work: Python file (.py or .ipynb)
  - Download your script via Colab.
- Predictions: Json file (.json)
- Package list: requirements.txt
  - E.g., numpy==1.26.3
- Report: Microsoft Word (.docx or .pdf)
- **No other formats are allowed.**
- Zip the files above before uploading you assignment.



# Delivery policies: Filenames

- Do not forget the correct formats!

	<b>Filename rule</b>	<b>Filename example</b>
Coding work	NLP_HW4_ <b>school</b> _student_ID.py	NLP_HW4_ <b>NTHU</b> _12345678.py
Predictions	NLP_HW4_ <b>school</b> _student_ID.json	NLP_HW4_ <b>NTHU</b> _12345678.json
Report	NLP_HW4_ <b>school</b> _student_ID.docx	NLP_HW4_ <b>NTHU</b> _12345678.docx
Package list	requirements.txt	
Zipped file	NLP_HW4_ <b>school</b> _student_ID.zip	NLP_HW4_ <b>NTHU</b> _12345678.zip

# Delivery policies: Things You should include

- In your report:

	Example	
Environment types	If Colab or Kaggle	If local
Running environment	Colab	System: Ubuntu 22.04, CPU: Ryzen 7-7800X3D
Python version	Colab	Python 3.10.1

# Delivery policies: Rules of coding

- If you use ChatGPT or Generative AI, please specify your usage **both** in:
  - **Code comments**
  - **Reports**
- **No plagiarism.** You should not copy and paste from your classmates. **Submit duplicate code or report will get 0 point !**
- Please provide links if you take the code from the Internet as reference.
- The following behaviors **will cause loss in the score of the assignment:** **(1) Usage with Generative AI without specifications (2) Internet sources without specifications (3) Plagiarism.**

# Punishments

Rule	Name your code: NLP_HW3_ <b>school_stu</b> <b>dent_ID.py</b> (.ipynb is also acceptable)	Name your report: NLP_HW3_ <b>school_stu</b> <b>dent_ID.docx</b> (.pdf is also acceptable)	Name your file: NLP_HW3_ <b>school_stu</b> <b>dent_ID.zip</b>	Include requirements.txt
Punishment	-5	-5	-5	-5
Rule	Include python version in your report	Do not modify the report template	Your code or report should not shows a high degree of similarity to another student's submission.	
Punishment	-5	-5	-100 for both	

In this assignment, you are allowed to modify the code template!

# Uploading the zipped file

- Please upload your file to NTU COOL.
- You will have three weeks to finish this assignment.
- If you have any question, please e-mail to [nthuikmlab@gmail.com](mailto:nthuikmlab@gmail.com)