

Basic setups of RAG system on Hotpot QA dataset

Week 9 - LGT

Dr Lin Gui

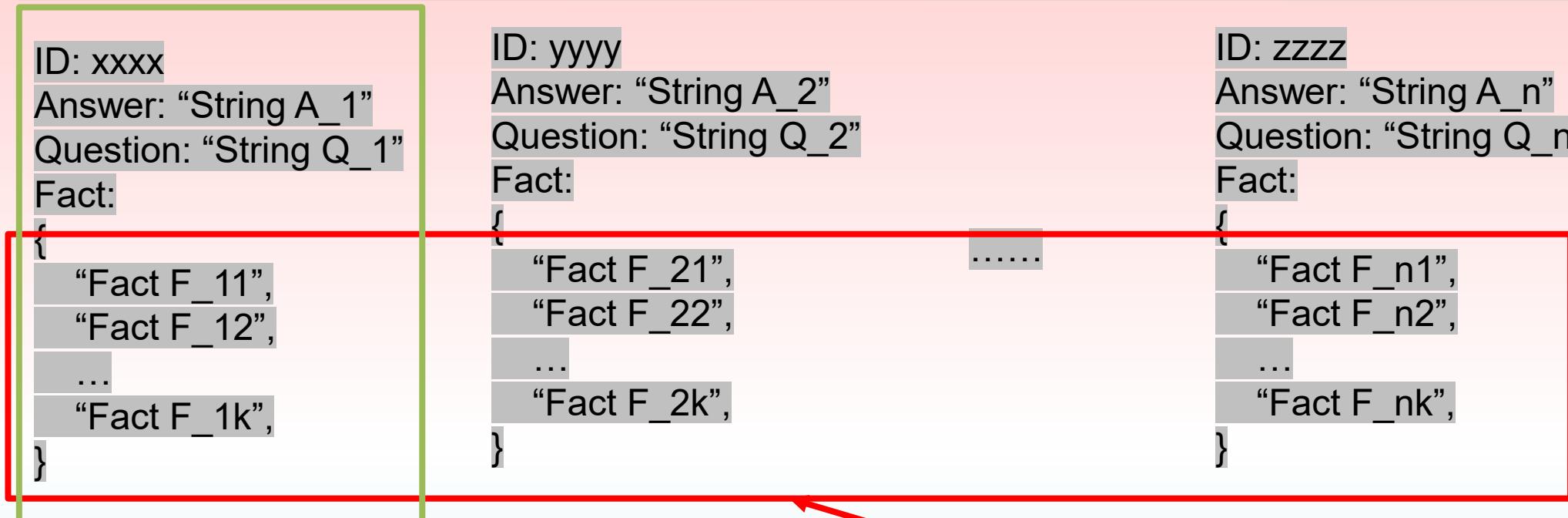
Lin.1.gui@kcl.ac.uk



Quick Takeaways

- In this week we introduced RAG system for QA and some other related tasks.
- However, as a benchmark for RAG system, the Hotpot QA dataset has already provided the evidences list.
- You do not need to design a full database/knowledge-base IR system on all evidence.
- Only focus on the ranking of evidence within each question-answer pair should be enough.

Example



Only index the candidate evidences under the question

Do not need to index all evidence

This hotspot QA dataset has already filtered the non-relevant documents; there is no necessary to build a full indexing

Example

ID: xxxx
Answer: "String A_1"
Question: "String Q_1"
Fact:
{
 "Fact F_11",
 "Fact F_12",
 ...
 "Fact F_1k",
}

ID: yyyy
Answer: "String A_2"
Question: "String Q_2"
Fact:
{
 "Fact F_21",
 "Fact F_22",
 ...
 "Fact F_2k",
}

ID: zzzz
Answer: "String A_n"
Question: "String Q_n"
Fact:
{
 "Fact F_n1",
 "Fact F_n2",
 ...
 "Fact F_nk",
}

Pipeline A -
For each give question:
 Collect all the facts
 Indexing on the collection
 Searching, ranking, obtain the top k
 RAG(top_k_facts;Q)
 Release the collection → Ø



Pipeline B -
For each give question:
 Collect all the facts
 Indexing on the collection
For each given question:
 Searching, ranking, obtain the top k
 RAG(top_k_facts;Q)



Some references might be helpful

- https://github.com/bbuing9/ICLR24_SuRe
- SuRe: Official Code for the paper "SuRe: Summarizing Retrievals using Answer Candidates for Open-domain QA of LLMs" (ICLR 2024)
- In this project, three different methods are used to build an IR system on the HotpotQA dataset:
 - BM25
 - Contrival
 - DRP

Some references might be helpful

- https://github.com/bbuing9/ICLR24_SuRe
- SuRe: Official Code for the paper "SuRe: Summarizing Retrievals using Answer Candidates for Open-domain QA of LLMs" (ICLR 2024)
- In this project, three different methods are used to build an IR system on the HotpotQA dataset
- It is recommended that you examine the provided code to gain insight or to employ it as a baseline model

Some references might be helpful

- [GitHub - dmis-lab/CompAct:](#)
- [\[EMNLP 2024\] CompAct: Compressing Retrieved Documents Actively for Question Answering](#)
- This project provides a very convenient way to compress docs into embeddings

Some references might be helpful

- Other classical method for text embedding:
- <https://github.com/princeton-nlp/SimCSE>
- <https://huggingface.co/sentence-transformers>