CS 224V Assignment 1

Due 2nd October 2024

Instructions: Use this Colab Notebook in conjunction with this write-up. Make sure to "Save a copy in Drive" before running the notebook. Submit your answers through Gradescope and attach your Google Colab notebook. In red, we label how each question in this writeup corresponds to a Gradescope question

Action Item

Go to TogetherAI and create a free account. This should give you $5 free credit that will be sufficient for your homework. Once you create your account and fill in your details on the first page, you should receive a new API key. If the page does not redirect you to your API, go to the dashboard and copy the API key.
You will need the API key for Task 4.

We expect heavy loads on the OVAL machine used in this assignment close to the deadline. **Thus, we highly recommend you start this assignment early and do not wait until the last minute.** This assignment is designed to be completed in **groups of 2**. Please submit as a group to Gradescope.

Extensions: You are granted an automatic 24-hour extension to either Assignment 1 or Assignment 2, but not both. If you submit Assignment 1 later than the deadline, this 24-hour extension will be automatically applied to Assignment 1. Each individual student has one 24-hour extension, so both partners will require an available extension to take one on Assignment 2.

1 Introduction

This assignment is designed to give you hands-on experience with how we can leverage LLMs to curate knowledge grounded on the web and any custom knowledge corpus. You will learn:

- how to leverage LLMs to create intelligent systems to perform complicated tasks;
- concepts of giving personas to LLMs and how to bring different perspectives;
- why and how to use retrieval systems to augment LLMs for grounding;
- how to observe weaknesses and strengths of existing systems and propose improvements.

2 STORM

LLMs have shown remarkable performance in generating responses range of user queries from simple factual questions such as "What is the capital of Italy?" to more complicated questions from International Mathematical Olympiad. Yet, writing grounded and organized long-form articles from scratch has been a challenge. Recent works such as STORM can generate Wikipedia-like articles by:

1. discovering diverse perspectives in researching the given topic,
2. simulating conversations where writers carrying different perspectives pose questions to a topic expert grounded on trusted Internet sources,
3. curating the collected information to create an outline.
4. generating full length report leveraging section titles and collected information.

STORM is an open-sourced project, and we host the STORM research preview website, which allows the public to generate articles across multiple domains. As of September 23, more than 100,000 articles have been generated through the website. You are encouraged to explore it.

Action Item

Learning Goal: Dive deep into STORM. Get familiar with core design choices.

Task 1 Read the STORM paper and answer following questions (Gradescope Q1)

- In Figure 2 of the paper, the authors implement a filtering method to exclude untrustworthy sources (step 5). How do they define trusted sources?
- How do the authors identify different perspectives on a given topic? Consider why directly prompting an LLM to generate a list of perspectives without any context might be ineffective.
- Besides automatic evaluation, what other evaluation methods do the authors use?
- (Optional) Try STORM UI online. (Please switch toggle to the left to choose STORM mode). If you have tried the STORM website, paste the link to the article you generated. (Gradescope 1.4)
- (Optional) Any feedback on the article you generated above would be appreciated! (Gradescope 1.5)
  - (1) How would you rate the generated article on a Likert scale of 1 to 5 (with 1 being the worst and 5 the best)?
  - (2) What are the strengths (e.g., comprehensive outline, accurate information) and limitations (e.g., improper handling of time-sensitive information, associating unrelated sources) of the article?
  - (3) STORM organizes information using a hierarchical outline (see the "Table of Contents" panel on the left). Is there any additional content you expected to be included? Briefly describe it or share any follow-up questions you have about the topic.
  - (4) Is there anything else you'd like to share about your experience with STORM?

3 Co-STORM

STORM automates knowledge curation by generating a full-length report based on a user's input topic. During lectures, we discussed how users may want to engage more deeply with knowledge curation systems to guide the information-seeking process and tailor the generated report to their needs. This is where Co-STORM comes into play.

Co-STORM extends STORM's capabilities by enabling a more interactive, collaborative approach to knowledge discovery. While language model (LM)-powered tools like chatbots and search engines excel at answering well-posed questions, they struggle when it comes to helping users explore unknown unknowns—information they didn't even know they needed to find.

Inspired by educational scenarios where students learn by listening and participating in conversations with teachers, Co-STORM introduces a novel method that lets users observe and influence discussions between multiple LM agents. Instead of requiring users to ask every question, the agents in Co-STORM take the initiative by asking questions on the user's behalf, allowing for the serendipitous discovery of new knowledge.

To make the interaction more user-friendly, Co-STORM organizes the discourse into a dynamic mind map, helping users track the conversation and ultimately providing a comprehensive report. The mind map visually outlines the key information uncovered, giving users a clear understanding of how the discussion evolved.

Action Item

Learning Goal: Get familiar with human-AI collaborative knowledge curation system.

Task 2 Choose a topic that you want to learn about and generate a report using CoSTORM Web UI. (Gradescope Q2)

TODO items

- Log into CoSTORM Web UI. Note this is not the same as STORM UI.
- After logging in, switch the toggle to the right to activate Co-STORM mode.
- Under the topic input box, select General Internet Search (Bing Search) as your search engine.
- Choose a topic for complex information seeking, anything of your interest! The topic cannot be directly answered by google search, cannot be answered with a short sentence by LLM chatbot, and cannot be math and coding problems. Here are some examples of topics for inspiration (Please do not use any of these topics for generating a report):
  - The Role of AI in Addressing Climate Change
  - Circular Fashion: Sustainability in the Clothing Industry
  - Influence of Social Media on Election Outcomes

  Once you have selected your topic, paste it into Gradescope. (Gradescope 2.1)
- Interact with Co-STORM.
  - **Observe**: Click the "Generate" button to view each turn.
  - **Engage**: Actively engage in the conversation in response to the agent using the input box at the bottom of the page.

  We expect you to observe at least 8 non-consecutive LLM agent utterance turns along with actively engaging for at least 3 non-consective turns.
- Next to each LLM agent's utterance, click the "Eval" button to assess the quality of the turn. Your evaluations will be automatically logged and graded based on feedback quality.
- Generate and download report. Switch to article tab (at top of the page, under title) and click generate article button. View the report (you will be asked for more question on the report in Task 3). Download the PDF version of the report by clicking "Show as PDF" button at the bottom of the page.
- Upload the report to Gradescope (Gradescope 2.2)
- Paste Co-STORM session URL (in the browser URL bar, in the format of https://oval-storm-dev.vercel.app/conversation/topic-id) to Gradescope (Gradescope 2.3)

Action Item

Learning Goal: Analyze strength and weakness of conducting complex information seeking with Co-STORM.

Task 3 Respond to the following questions based on the generated report (Gradescope Q3)

1. Did you find the information you were looking for?
2. Do you like the structure of the outline for the report? What changes would you make?
3. Do you agree with the choice of experts generated? Why or why not?
4. Write the three most surprising and useful citations and three irrelevant and/or off topic citations.
5. Would you like to use Co-STORM for other topics? Please name a few topics and scenario where you would love to use Co-STORM.
6. (Optional) Would you like to be invited to conduct in-depth evaluation of Co-STORM? (Choice will not impact your grade)

4 Co-STORM with a Custom Knowledge Corpus

In the previous section, we used the Co-STORM system to search for knowledge across the web. However, in many cases, users may prefer to work with their own knowledge corpus (data) to generate detailed reports. This section introduces how to use Co-STORM with a custom knowledge corpus.

For this task, we will utilize arXiv as our custom knowledge corpus, instead of relying on the web. arXiv is an open-access repository for electronic preprints and post-prints (known as e-prints), which are moderated for posting but not peer-reviewed. It covers a wide range of subjects. In this exercise, we will focus on the Computer Science section, specifically the Computing Research Repository (CoRR).

Action Item

You will begin by selecting a sub-domain of interest from the provided list of 40 sub-domains within Computer Science. Please ensure that you sign up with your group names on the Google Sheet. Each sub-domain can accommodate a maximum of two groups, and once the limit is reached for a specific sub-domain, you must select an alternative.

4.1 Learn how to Use a Custom Knowledge Corpus

For building a retrieval corpus, there are various methods. Traditionally, TF-IDF based systems Sparck Jones (1972) were used. In recent years, vector embeddings Seo et al. (2019); Cohan et al. (2020); Chen et al. (2024); Zhang et al. (2024) have gained traction because of their better performance, especially when the search query and the text do not have surface-level overlap.

To start, we need to select a suitable embedding model. There are benchmarks available for various domains (e.g. medical, scientific, general knowledge etc.), and a widely used one is MTEB [1] Muennighoff et al. (2023). For this assignment, we will be using the GTE-large-en-1.5 model Zhang et al. (2024)[2] for its balance of high quality and relatively low vector embedding size (1024 floating point numbers per vector). We also need a *vector database* Pan et al. (2024). A vector database is a data structure that enables fast calculation of a similarity metric like cosine similarity between a query embedding and tons of corpus embeddings. For this, we will be using Qdrant [3].

To retrieve with vector embeddings, the following steps are usually needed:

1. The entire corpus of text should be preprocessed. This includes *chunking* Kamradt (2024) long text to fit into the context window of an embedding model.
2. Vector embeddings for all chunks are computed by feeding them to an embedding model, one by one.
3. The resulting vector embeddings are stored in a vector database.

Then, to search through the corpus, you need to:

1. Convert the given query to a vector embedding.
2. Calculate the similarity metric over all corpus embeddings, and return the top K most similar vectors as the output.

[1] A live leaderboard is available at https://huggingface.co/spaces/mteb/leaderboard.
[2] Model weights are available at https://huggingface.co/Alibaba-NLP/gte-large-en-v1.5.
[3] https://github.com/qdrant/qdrant

4.2 Conduct knowledge curation on a Custom Knowledge Corpus

Action Item

Learning Goal: Dive Deep into Co-STORM; Get Hands-on!

Task 4: Read Co-STORM paper and follow the provided Google Colab notebook to explore the internal workings of Co-STORM. You will gain insights into what it takes to build a system like Co-STORM.

- How many types of roles are proposed in the collaborative discourse protocol in the paper? (Gradescope Q4.1).
- What are the two operations for maintaining the dynamic mind map discussed in the paper? (Gradescope Q4.2).
- Upload the cs224v_assignment_submission_dir.zip file to (Gradescope Q4.3).
- Upload the Google Colab notebook as a PDF. In Google Colab, click File → Print → "Save as PDF" and upload the downloaded PDF file to (Gradescope Q4.4)

Action Item

Learning Goal: Evaluate the Strengths and Weaknesses of Using Co-STORM for Complex Information-Seeking in the Academic Domain

Task 5: After generating a report using Co-STORM on the academic paper corpus via Google Colab notebook, answer the following questions: (Gradescope Q5)

1. How did Co-STORM respond when you tried to steer the conversation by injecting your own utterance? Did you feel you could effectively change the focus of the discussion? Why or why not?
2. Do you agree with the selection of experts identified in the report? Explain.
3. List the three most surprising and useful citations from the report, as well as three citations you found irrelevant or off-topic.
4. Did the report contain the information you were seeking? Explain your answer.
5. What are the key differences between the reports generated using the web as a data source and the customized academic paper corpus?
6. Propose improvements for knowledge curation systems to enhance their performance when working with custom academic corpora.
7. What additional custom corpora would you find most useful for use on Co-STORM?
8. What are some potential applications that can leverage Co-STORM with arXiv?

References

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 2318–2335, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S. Weld. 2020. SPECTER: Document-level Representation Learning using Citation-informed Transformers. In *ACL*.

Greg Kamradt. 2024. 5 levels of text splitting. https://github.com/FullStackRetrieval-com/RetrievalTutorials/blob/main/tutorials/LevelsOfTextSplitting/5_Levels_Of_Text_Splitting.ipynb. Accessed: 2024-09-14.

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. MTEB: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, Dubrovnik, Croatia. Association for Computational Linguistics.

James Jie Pan, Jianguo Wang, and Guoliang Li. 2024. Survey of vector database management systems. *The VLDB Journal*, 33(5):1591–1615.

Minjoon Seo, Jinhyuk Lee, Tom Kwiatkowski, Ankur Parikh, Ali Farhadi, and Hannaneh Hajishirzi. 2019. Real-time open-domain question answering with dense-sparse phrase index. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4430–4441, Florence, Italy. Association for Computational Linguistics.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.

Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, Meishan Zhang, Wenjie Li, and Min Zhang. 2024. mgte: Generalized long-context text representation and reranking models for multilingual text retrieval.