Conversational Book Search System using LLMs

Vahid Rahimzadeh
rahimzade@ut.ac.ir

Danial Baledi
Danial.baledi@ut.ac.ir

1 Problem statement

Emergence of large language models and their potential in combination with information retrieval methods has provided a tremendous outlook for work and research in the field of recommender systems and search chatbots. Such perspectives were not previously accessible. Despite this, the created capacity has not been fully utilized. In this project, we aim to address a small portion of these capacities using existing research. Book search is a task that has often been traditionally carried out, ideally using pre-defined features alongside the book's title. Many times, such searches result in a flood of different books, many of which may not align with our preferences. In such situations (and in many others), we want to find our desired book by providing a description. In this project, we intend to design a chatbot that, through conversation with the user, suggests suitable books. Additionally, if possible and with appropriate data, the chatbot can also consider the user's history in its recommendations.

2 What you proposed vs. what you accomplished

We initiated our project with a comprehensive pipeline tailored to our objectives. Following the initial presentation in our class, and considering the feedback from our instructors, we opted to streamline our approach by removing certain modules, such as clarification and query rewriting. Originally, we commenced our work using the CMU dataset. However, given the system's immediate impressive performance, we sought to enhance the depth of our challenge.

Subsequently, we curated a dataset sourced from Ketabrah Ketabrah, focusing on Persian books. This dataset comprised 4719 books, each featuring over 15 columns, with 12 columns specifically utilized in our project. Further details on the dataset are outlined in the dedicated section. An additional refinement to our approach involved the creation of a query-document relevance dataset, serving as an evaluation benchmark for our retriever module. Furthermore, to ensure a holistic assessment, we conducted a human-driven overall quality check of the system. This evaluation was carried out collaboratively by the authors and a member of the Intelligent Information Systems' Lab at the University of Tehran.

3 Related work

In light of the multi-faceted nature of the designated project, we review the works associated with each component individually.

3.1 Large Language Models

Over the past few years, Large Language Models built upon the transformer architecture (14) have brought about a paradigm shift in the realm of artificial intelligence. These models have consistently achieved state-of-the-art performance across diverse natural language understanding and dialogue tasks (2).Notably, their prowess is evident in zero or few-shot learning scenarios, where, with well-crafted prompts, they demonstrate adaptability to new tasks without the need for modifying underlying model parameters (11).

3.2 Retrieving

It has been shown that endowing Large Language Models with the capability to retrieve information from external corpora can enhance their performance in tasks such as question answering and reduce hallucinations (1). A stand-alone query can be used to retrieve documents using any ad-hoc retrieval system. Depending on how queries (and documents) are represented. Retrieval systems can be grouped into two categories: sparse retrieval systems that use sparse vector store present queries (and documents) and dense retrieval systems that use dense vector representations. The most commonly used query and document vectors are based on bag-of-words (BOW) representations. Sparse retrieval methods are based on lexical matching. Although simple, effective, and computationally efficient, lexical matching has an intrinsic challenge that a concept is often expressed using different vocabularies and language styles in documents and queries, resulting in the vocabulary mismatch problem. Dense retrieval methods provide a new path to address the vocabulary mismatch problem, by using learned vector representations (4). One approach is to use two BERT encoders: the first one to encode the query, and the second one to encode documents into dense vectors. The both encoders or one of them should be trained to maximize the similarity between related queries and documents (7) (9).
In conversational status, the query could be rewritten based on previous turns (queries) (10) (15) or incorporate all queries to compute a dense vector (15).
After the retrieval module retrieves a few hundred candidate documents from a large document collection, the document ranking step can afford to use more sophisticated models than those used for document retrieval to (re-)rank these retrieved candidates. A common approach to document ranking is to learn a ranking model to score each retrieved candidate document with respect to input query (4).

3.3 Clarification

In the absence of confidence in the retrieved documents, our model should pose clarification questions. A clarification question may be a single inquiry seeking more information from the user, or it could be selected from a set of predefined questions. We can enhance the generation of clarification questions using Large Language Models, instructing the model to assess whether the query requires clarification and, if necessary, generate a suitable clarification question. Hashemi et al. propose GuidedTransformer that leverages information from conversation history, retrieved documents, and potential clarifying questions (6). Their approach yields significant improvements in the question selection task on Qulac, a dataset consisting of question-answer pairs for faceted and ambiguous queries. (16). Rosset et al. tackle the task of question suggestion in a “People Also Ask” search engine setting. They argue that a useful question is not simply related to the topic of a user’s query, but should also be “conversation leading” and provide meaningful in for-

mation for the user's next step. They propose a BERT-based and a generative GPT-2-based model for question suggestion. They find that questions generated by GPT-2 are syntactically correct, but less useful than the ones selected by BERT from a pre-defined pool of questions. The authors suggest that a reason for the inferior performance of GPT-2 might be due to the lack of explicit guidance in semantics (12). propose a model that generates clarifying questions with respect to the user query and query facets. They work focused more in facets extraction. They fine-tune the GPT-2 language model to generate questions related to the query and one of the extracted query facets. Sekulić et al. propose a model that generates clarifying questions regarding the user query and its facets. Their work also focused on facet extraction. They fine-tune the GPT-2 language model to generate questions related to the query and one of the extracted query facets (13).

3.4 Evaluation

A key challenge in these kind systems is evaluation, to compare different systems and approaches and carefully consider the different forms of evaluation to draw conclusions on what works best. There are two main approaches for capturing the human element that are system-oriented evaluation and user-oriented evaluation. A) System-oriented evaluation captures the user's requests and preferences in a fixed dataset that can be used for comparison of different systems. When using this form of evaluation, researchers focus on developing models and systems that best match the user preferences that were captured in the dataset. B) User-oriented evaluation observes the interactions of a real user with the search system. The users can be in a lab, which allows detailed instrumentation and interpretation of their use of the system (5).

For this project we focus on the first approach. The most useful and popular metrics for this approach are:

- Recall
- Mean Reciprocal Rank (MRR)
- Mean Average Precision (MAP)
- Normalized Discounted Cumulative Gain (NDCG)

Despite their simplicity, these metrics are among the most widely used for examining Information Retrieval systems. In the field of Conversational Information Retrieval, we could use Conversation Information Retrieval-specific datasets in conjunction with the above metrics (Carnevali).

4 Dataset

4.1 Books Dataset

For the dataset, we initially considered utilizing existing datasets. After reviewing various datasets, we opted for the "goodbooks-10k" dataset. The rationale behind this choice lies in the balance between the completeness of its data cells and the appropriate number of records. This dataset comprises more than 29 columns and 10,000 rows. Despite its relative completeness compared to other datasets, many of its cells remain empty. Subsequently, following an examination of the retrieval system (which will be elaborated upon later) and the parametric knowledge of the language model, we concluded that employing a Persian dataset is more appropriate. Due to the unavailability of a suitable Persian dataset, we decided to conduct our own data crawling. To achieve this, we extracted information from book details on the "Ketabrah" website.[1] Due to hardware limitations for embedding step (which will be elaborated upon

[1] https://www.ketabrah.ir/

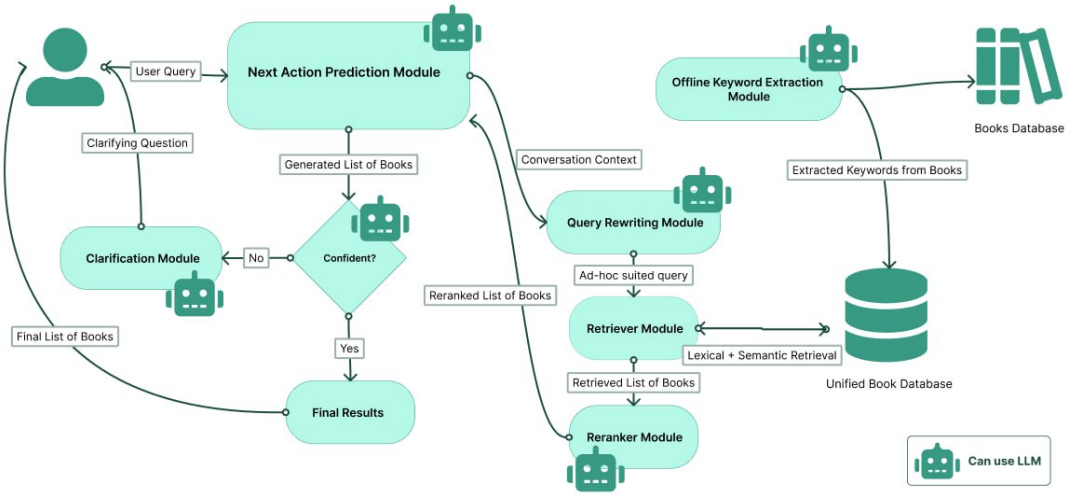

Figure 1: Initially Proposed system

later), we limited the crawling to information related to novels available on the site. The resulting dataset consists of 4,719 rows. The considered specifications for each row include the book's title, its webpage address on the website, the names of authors and translators, the publisher's name, the publication year, the book's table of contents, its rating on the site, the number of users who have rated the book, the number of pages, the language, the book's subject, the total number of pages, the keywords assigned by the website, and the ISBN identifier of the book. It is worth noting that not all specifications were available for every book, leading to empty cells in the dataset.

After constructing the dataset, it is imperative to create a corresponding text vector database. For this purpose, we utilized ChromaDB[2], an embedding database. The data was transformed into chunks of 512 characters, where 256 characters at the end and the beginning of consecutive chunks overlap. In addition to the 512 characters, authors' names, translators, book subjects, and keywords related to the book were appended to the end of each chunk. Furthermore, book title, publisher, rating, number of raters, number of pages, publication year, and book index in the original dataset were added as metadata to each chunk.The associated vector database was also constructed in this manner.

4.2 AI Generated Queries and Query Book Relevancy Datasets

Evaluation of our retrieval modules necessitated the creation of a relevance dataset. Leveraging the keywords column from our crawled book dataset, originally intended for SEO purposes, we employed these keywords as our ground truth for relevancy. Initially, the dataset encompassed 6209 unique keywords. To generate sufficient queries for evaluation, a three-step filtering process was implemented. Firstly, non-Persian keywords, such as writer's names in English, were eliminated, resulting in 3467 remaining keywords. Subsequently, we calculated the frequency of books associated with each keyword, sorting them in descending order of frequency. Although our initial intention was to use the 50

[2]https://www.trychroma.com/

most frequent keywords, we encountered limitations with this approach. Certain frequent keywords were essentially duplicates (e.g., "Persian Stories" and "Persian Romance"), and the selection lacked the desired diversity and representation of the entire set of keywords. To address this, we sought the assistance of GPT-4 and GPT-3.5 (16K) models.

In collaboration with these language models, we provided instructions alongside keyword-frequency tuples, tasking them with selecting the most suitable keywords. Due to context limitations, GPT-4 was provided with keywords having a frequency greater than 10, while GPT-3.5 (16K) received keywords with a frequency greater than 3. The outcome was 50 keywords from each model, with 72 keywords in common between the two selections. These mutually chosen keywords formed the basis for our subsequent steps.

Using the identified seed keywords, we directed GPT-3.5 to generate 10 queries for each keyword, resulting in a total of 720 queries across the 72 book-related keywords, which were utilized for our retrieval module evaluation. We also considered prompting GPT-4 for 5 initial seed queries and use them as context for GPT3.5 to improve outputs, but using prompt engineering we came to conclusion that the results were not that different so we used GPT3.5 directly due to budget restrictions.

5 Baselines

In the initial proposal, we initially proposed evaluation baselines encompassing various components of the pipeline, including keyword extraction, clarification, next action prediction, reranker, and retrieval. However, following constructive feedback from our instructors, we refined our approach and decided to specifically focus on baseline evaluations for the retrieval system and added an end-to-end quality check done by human annotators. The chosen baselines for information retrieval included BM25, a conventional information retrieval technique, and the incorporation of random retrievers to enhance comparative analysis.

In collaboration with members of the Intelligent Information Systems' Lab at the University of Tehran, we facilitated a human-annotated quality check. Annotators were tasked with providing queries to our system and vanilla ChatGPT3.5, subsequently evaluating and comparing the quality of responses while assigning scores based on their assessments. This addition of a human-centric evaluation component, alongside the retrieval baseline assessments, provides a comprehensive and nuanced understanding of system performance, capturing both algorithmic and qualitative dimensions in our research report.

6 Approach

We first proposed a comprehensive pipeline as shown in **??**. Based on the valuable comments of intructures, we changed our approach. In summary, our work consists of two main components: "retrieve" and "chatbot". In the retrieve section, we employed a weighted combination of two types of retrievers. The first retriever utilizes BM25, emphasizing exact similarities between user input and stored data. The second retriever focuses on semantic similarities using vector similarity. Additionally, with the use of a language model, when the user mentioned a case where we could filter results based on metadata, that specific case was extracted from the user's input and incorporated into the database query (self query retriever).

In the chatbot section, the model was instructed to suggest books based on the user input and the first 8 items in the retriever's output (most relevant to the user input), along

with its own knowledge. Interestingly, in many cases where the model did not utilize the retrieved books, it still provided better responses than the model without retrieval.
In our retrieval and generation pipeline, guided by the insights gained from our experiments, we observed that the optimal performance was attained when the outputs of the retrievers were translated into English and presented as context to the language model (LLM). Subsequently, we applied this approach and translated the final generation of the LLM back into Persian.

7 Experiments

Our experiments consisted of two main parts. The first part involved the systematic evaluation of the retrieval module, while the second part focused on the end-to-end evaluation by a human annotator, comparing the results with vanilla ChatGPT3.5.

7.1 Systematic Evaluation of Retrieval Module

In this part, we utilized BM25 as the baseline and considered two random retrievers with different seeds. Additionally, we employed the Self-Query Retriever, as described in the Approach section. Another option explored was an ensemble of BM25 and semantic (Self-Query) with different weights of importance assigned to their scores. As illustrated in Figure **??** and detailed in Table **??**, the hybrid retriever with more importance on Self-Query achieved the highest scores across all metrics. Consequently, this configuration was selected for further evaluation by human annotators.

7.2 Human Annotator Evaluation

In this phase, a human annotator actively collaborated with our system, offering valuable feedback. The annotator was tasked with submitting queries to our system and presenting identical queries to vanilla ChatGPT.

1. Whether our response was better (1), ChatGPT was better (-1), or both were the same (0).

2. Whether the response generated by our approach was available in our retrieved book set or if ChatGPT answered with its own knowledge.

3. The rank of the chosen book by ChatGPT in our RAG pipeline within the retrieved bookset.

Among the 20 comprehensive evaluations gathered, our model was preferred over ChatGPT in 13 instances, ChatGPT was chosen twice, and in the remaining five instances, both models performed equally well or equally poorly. The detailed queries and results are available in the appendix pdf file.

8 Error analysis

Upon reviewing the retrieval results, it became evident that the embeddings do not adequately encapsulate meaning or provide suitable semantic reflections. Considering the potency of the LaBSE model, such an outcome was anticipated.
While the final model performed well in some cases, it was expected to recommend more well-known books. This issue could potentially be improved by influencing the score and the number of users who have rated the book.
In the section related to extracting metadata-related data, as we utilized a language model, in some cases, the language model extracted a feature that did not belong to the metadata. This led to errors when executing the database query. It is advisable to verify and ensure such aspects before executing the database query.
Furthermore, in scenarios like asking for a

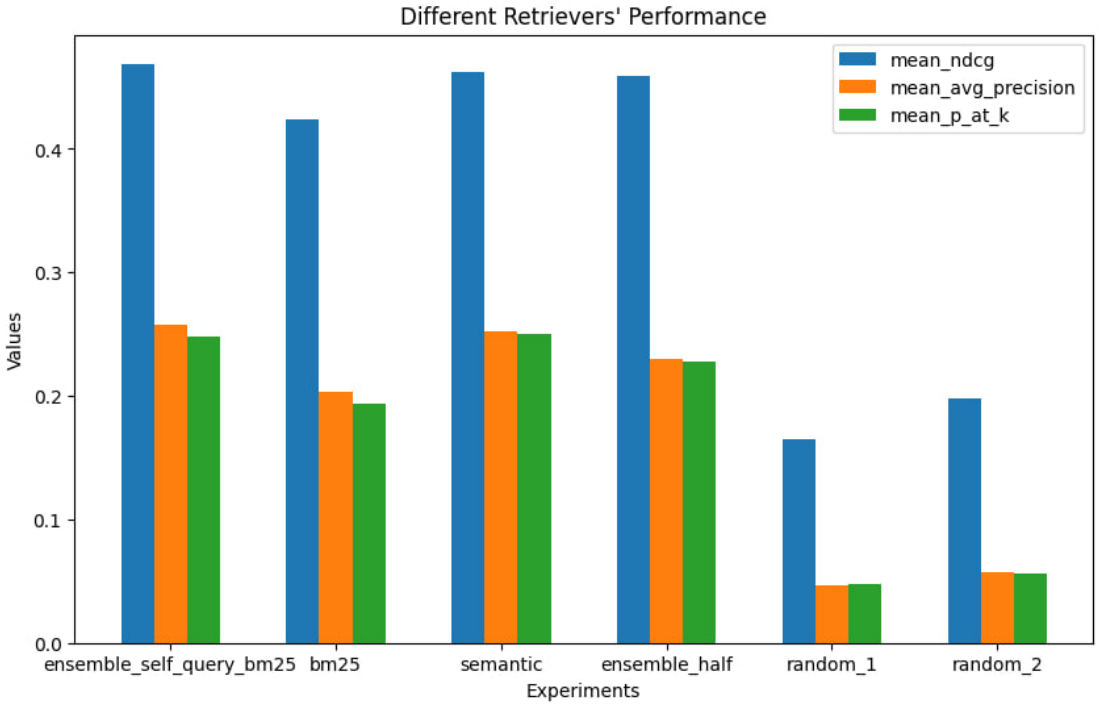

Figure 2: Different Retriever Settings Performance on Query Relevancy Dataset

book similar to another book, the model's output is not suitable. To enhance this aspect, employing a language model trained on the text of books could be beneficial. It would allow generating similar queries or expanding the user's query, and then use them retrieval.

9 Contributions of group members

List what each member of the group contributed to this project here. For example:

- member 1: did books data collections, keyword extraction and choosing using LLMs, generating queries for documents using LLMs, prompt engineerings needed and creating the book query relevancy dataset, wrote the code for retriever evaluation and visualizations needed, wrote the following sections in report:
  1. Section 2
  2. Section 4.2
  3. Figure 1, Figure 2, Table 1
  4. Section 5
  5. Section 7
  6. Section 9
- member 2: Wrote the core logic for RAG pipeline and LLM integration with retrievers, did data cleaning on gathered book dataset, Wrote the below Sections in report:
  1. Section 1
  2. Section 3
  3. Section 4.1
  4. Section 6
  5. Section 8

Table 1: Mean NDCG, Mean P at K, and Mean Average Precision

Method	Mean NDCG	Mean P at K	Mean Avg. Precision
Ensemble (0.75*Self Query + 0.25*BM25)	**0.47**	**0.25**	**0.26**
BM25	0.42	0.19	0.20
Self Query	0.46	**0.25**	0.25
Ensemble (0.5*Self Query + 0.5*BM25)	0.46	0.23	0.23
Random 1	0.17	0.05	0.05
Random 2	0.20	0.06	0.06

10 Conclusion

In conclusion, this project has been a valuable and enriching experience for us. We embraced challenges by transitioning to a Persian dataset, curating extensive datasets for both books and query document relevancy, and incorporating language models for query generation. Our journey involved substantial prompt engineering, utilization of various retrievers, and exploration of diverse evaluation metrics.

The noteworthy finding was that the optimal retrieval performance was achieved through Semantic+Lexical retrieval, aligning with our expectations. We leveraged one of the most promising Persian text embedders, LaBSE, and observed that the performance ceiling for Persian text embedding was considerably lower than that for English language text embedding. Intriguingly, our model consistently outperformed vanilla GPT3.5, demonstrating its ability to provide accurate responses even when the requested book was not present in the retrieved bookset. The contextual information from our retrieved bookset proved instrumental in enhancing ChatGPT's performance.

Looking ahead, our future endeavors include exploring additional retrieval algorithms, such as FAISS, expanding and diversifying our book dataset to encompass a broader range of categories, testing alternative embeddings like OpenAI embeddings, conducting further prompt engineering, and refining our prompts. Detailed insights and avenues for future research are outlined in the error analysis section.

References

[1] Borgeaud, S., Mensch, A., Hoffmann, J., Cai, T., Rutherford, E., Millican, K., van den Driessche, G., Lespiau, J.-B., Damoc, B., Clark, A., de Las Casas, D., Guy, A., Menick, J., Ring, R., Hennigan, T., Huang, S., Maggiore, L., Jones, C., Cassirer, A., Brock, A., Paganini, M., Irving, G., Vinyals, O., Osindero, S., Simonyan, K., Rae, J. W., Elsen, E., and Sifre, L. (2022). Improving language models by retrieving from trillions of tokens.

[2] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

[Carnevali] Carnevali, L. Evaluation measures in information retrieval. Accessed: 12/30/2023.

[4] Gao, J., Xiong, C., Bennett, P., and Craswell, N. (2023a). *Conversational Search*, pages 39–69. Springer International Publishing, Cham.

[5] Gao, J., Xiong, C., Bennett, P., and Craswell, N. (2023b). *Evaluating Conversational Information Retrieval*, pages 23–38. Springer International Publishing, Cham.

[6] Hashemi, H., Zamani, H., and Croft, W. B. (2020). Guided transformer: Leveraging multiple external sources for representation learning in conversational search.

[7] Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., and tau Yih, W. (2020). Dense

passage retrieval for open-domain question answering.

[Ketabrah] Ketabrah. Ketabrah — online persian book platform. Accessed: 01/30/2024.

[9] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., tau Yih, W., Rocktäschel, T., Riedel, S., and Kiela, D. (2021). Retrieval-augmented generation for knowledge-intensive nlp tasks.

[10] Lin, S.-C., Yang, J.-H., Nogueira, R., Tsai, M.-F., Wang, C.-J., and Lin, J. (2020). Query reformulation using query history for passage retrieval in conversational search. *arXiv preprint arXiv:2005.02230*.

[11] Reynolds, L. and McDonell, K. (2021). Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–7.

[12] Rosset, C., Xiong, C., Song, X., Campos, D., Craswell, N., Tiwary, S., and Bennett, P. (2020). Leading conversational search by suggesting useful questions. In *Proceedings of The Web Conference 2020*, WWW '20, page 1160–1170, New York, NY, USA. Association for Computing Machinery.

[13] Sekulić, I., Aliannejadi, M., and Crestani, F. (2021). Towards facet-driven generation of clarifying questions for conversational search. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*, ICTIR '21, page 167–175, New York, NY, USA. Association for Computing Machinery.

[14] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

[15] Yu, S., Liu, Z., Xiong, C., Feng, T., and Liu, Z. (2021). Few-shot conversational dense retrieval. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21. ACM.

[16] Zou, J., Kanoulas, E., and Liu, Y. (2020). An empirical study of clarifying question-based systems.