Deploying and Evaluating a Conversational Agent Using LLMs for Academic Library Reference

DOI: 10.1108/RSR-05-2025-0030

Authors

Megan Fitzgibbons
(Corresponding Author, First Author, Submitting Author)
ORCID: 0000-0003-0409-6321
Concordia University

Francisco Berrizbeitia
ORCID: 0000-0002-1542-8435
Concordia University

Joshua Chalifour
ORCID: 0000-0001-7663-0509
Concordia University

Yara Stouhi
ORCID: 0009-0004-8383-5448
Concordia University

Olivier Charbonneau
ORCID: 0000-0001-7377-7695
Concordia University

Aviva Majerczyk
ORCID: 0009-0006-6058-0408
Concordia University

Funding

- Concordia University Applied AI Institute
- Concordia University (Library Research Grant)

Author accepted manuscript
Accepted for publication in Reference Services Review: 20 December 2025

emerald PUBLISHING

Reference Services R

Deploying and Evaluating a Conversational Agent Using LLMs for Academic Library Reference

Journal:	*Reference Services Review*
Manuscript ID	RSR-05-2025-0030.R2
Manuscript Type:	Original Article
Keywords:	Reference Services, Technological change, Information services, Technological Innovation, Assessment, service delivery

Abstract

Purpose

This study has two aims. First, we sought to implement a RAG-based GenAI system capable of answering reference questions. Second, we aimed to develop an evaluation protocol to assess the chatbot by means of comparing implementations that use three different LLMs. An evaluation rubric was piloted to gauge its viability as an assessment tool.

Approach

The RAG-based chatbot uses a two-step approach. First, in response to a query, the system retrieves relevant documents from a knowledge base. Each document is vectorized and matched by relevance. Second, retrieved data is combined with an LLM's generative capabilities to produce a context-aware response.

Fourteen common questions representing different areas of the knowledge base were tested with the chatbot versions. The research team developed and then used an evaluation rubric to score the chatbots' responses according to: accuracy, groundedness, elicitation, completeness, and further assistance. The rubric was also evaluated by calculating the standard deviation among reviewers' scores.

Findings

The RAG implementations were largely successful in restricting the chatbot's responses to the knowledge base. The evaluation rubric was effective for assessing the models, highlighting each's strengths and weaknesses. Despite the evaluation being subjective, the evaluators gave similar scores, with the greatest variation in the elicitation dimension.

Originality

This study offers a technical description of a practical way to implement a RAG-based chatbot in a library setting as well as a protocol for evaluating such chatbots in multiple dimensions that hasn't been discussed in previous literature.

Introduction

Generative artificial intelligence (GenAI) leveraging Large Language Models (LLMs) has captivated the world's attention as a significantly disruptive technology. GenAI tools have been touted as having the potential to transform how information is provided

through digitally mediated services, ranging from customer service to deeper interactions (Cox, 2023). It's not novel that libraries provide online human or machine-based chat services, but GenAI requires new technical approaches and considerations around the ethics and usefulness of conversational agents. Testing this technology is therefore a burning issue in library reference, instruction, and research support services as it could significantly impact how users discover, access, and use knowledge in the foreseeable short term.

In this study, we developed a chatbot, known as Gaby, configured for delivering academic library information services using retrieval-augmented generation (RAG) and defined a protocol for assessing different versions of the chatbot, using different LLMs, in order to evaluate the tool's usefulness and guide potential implementation decisions. Each chatbot interaction was assessed for alignment with verified library information, engagement with users, comprehensiveness of answers, and guidance to additional resources as appropriate. This assessment approach aims to balance quality in multiple dimensions to, ensure that responses are reliable, relevant, and user-centered. In this article, we present the technical design, the application of an evaluation method as a proof of concept, and our assessment of the approach.

This study has two aims. First, we sought to implement a RAG based GenAI system capable of answering reference questions. Second, we aimed to develop an evaluation instrument and protocol to assess the usefulness of the GenAI chatbot by means of comparing RAG implementations that use three different LLMs. An evaluation rubric was piloted in order to gauge its viability as an assessment tool for decision-making in libraries.

Literature Review

This study is informed by two main categories of literature: 1) the implementation of GenAI-based chatbots in academic libraries and 2) the assessment of GenAI in the context of reference services.

Implementation of generative AI chatbots in academic libraries

As noted by Rodriguez and Mune (2022), libraries have been experimenting with and implementing chatbots using AI and natural language processing (NLP) since the first decade of the 21st century, with a marked uptick immediately preceding and during the COVID-19 pandemic. The chatbots that immediately preceded LLM technology often used NLP and artificial intelligence markup language (AIML) with some type of system for retrieving information from a knowledge base (e.g., Barus & Surijati, 2022;

Ehrenpreis & DeLooper, 2022; Ivanovskaya et al., 2019; Kane, 2019; Panda & Chakravarty, 2022; Rodriguez & Mune, 2022; Thalaya & Puritat, 2022).

The advent of GenAI and LLMs has given rise to a new wave of possibilities for chatbots as well as new considerations for the assessment of this technology. Within the wider field of study on AI-based chatbots in libraries, Guy et al. (2023) argue that, while many of the articles on the topic are in the stage of theorization, it is now time to move to "begin assessing their use and impact" (p. 2). They explain that while there are numerous studies on AI's use in other domains within the library, such as reference and draft-writing, fewer studies have centered on a real-life case study of a created AI-powered chatbot for library settings. However, there are still a notable few that informed this project. Several institutions have implemented chatbots using the Ivy.ai service, including the University of Calgary (Bryant, 2024), University of Texas (University of Texas Libraries, 2024), City University of New York (Ehrenpreis & DeLooper, 2022; 2025), University of Oklahoma (University of Oklahoma Libraries, n.d.), and hundreds of other higher education implementations, according to the product website (https://ivy.ai/higher-education). Although originally available before LLM technology, the product currently uses RAG techniques to confine responses to a defined dataset combined with OpenAI's GPT-4 models (https://ivy.ai/generative-chatbot). Because Ivy.ai is a vendor-supplied product, communications about its implementation do not include much in the way of technical documentation.

The present study built directly on the work of Lappalainen and Narayanan (2023), who document the development of a chatbot powered by the OpenAI API. It was developed by first constructing a knowledge base from a university library's website through automated scraping and manual data entry. Embeddings were created and stored using Chroma, and then LangChain was used to create a script that identifies the context and queries the OpenAI API. The chatbot interface was created using Streamlit. The use of a knowledge base was employed to balance the generalities of ChatGPT with the specific information necessary to students at the university. Overall, the authors considered the chatbot prototype a success, but the issues apparent at the early stages of its implementation included generation of incorrect and broken links (a concern echoed later by our own project), its inability to give time-sensitive information, and the ongoing presence of inaccurate or incorrect information in responses (sometimes known as "hallucinations").

Assessment of generative AI in reference

Hobert (2019)'s literature review suggests a number of dimensions for evaluating chatbots in educational settings: technology acceptance and adoption, learning success, increased motivation, further beneficial effects on learning processes (e.g.,

motivation), usability, algorithmic or technical correctness, and psychological factors (e.g., enjoyment).

These are echoed in the library context in existing research on chatbots in reference contexts, with a heightened focus on the dimensions of correctness/accuracy and usability in the GenAI era. There is also an emphasis on comparison with the standard of human responses to queries when assessing chatbot performance.

For example, Lai (2023) posed questions received through an email chat service to ChatGPT, evaluating the responses using a rubric for completeness of answer, accuracy, and generation of further assistance. Lai concluded that, at the time of the study, ChatGPT was not able to provide satisfactory responses in the studied criteria. ChatGPT was not able to decipher the specificities necessary when dealing with inquiries about the large academic institution.

Yang and Mason (2024) conducted a similar study, entering 30 questions received via email, chat, and at an in-person reference desk into ChatGPT and evaluating the answers for accuracy, relevance, and friendliness. These were compared to librarians' responses to the original queries, which were likewise scored. Librarians were found to outperform ChatGPT in all three dimensions across all 30 queries on average, although not on every dimension on every query.

In terms of comparative studies, Feng, Wang, and Anderson (2024) compared the performance of four chatbots (ChatGPT-3.5, ChatGPT-4, Bard, and Perplexity) in responding to a series of related questions on the topic of information seeking in social work. The responses were assessed in terms of factual accuracy and relevance. It was found that some responses had factually incorrect information including fabricated references. ChatGPT-4 was judged to have the highest quality information, although the article lacks specific detail on the evaluation methods, a decision likely made because the focus is the larger educational and ethical implications of these tools.

There have also been some reports of how the implementations of chatbots previously mentioned have been evaluated, again, with the highest emphasis on factual accuracy in the context of GenAI. Lappalainen and Narayanan (2023) reported testing their chatbot internally amongst library staff and analyzed 500 interactions for accuracy. "Very few" factual errors were identified, with the primary problems found to be non-existent links or lack of capacity to answer questions that require real-time information. At the University of Calgary (Bryant, 2024), the live Ivy.ai-powered chatbot is continually assessed with interactions scored on a 5-point scale for overall quality of response. It is reported that about half of all questions are rated with a score of 4 or 5. Although published after the present study was conducted, it should be noted that Ehrenpreis and DeLooper (2025) updated an earlier publication (2022) to assess the performance of the Ivy.ai service ("IvyQuantum") mentioned above in comparison with the earlier rules-

based version of the chatbot. A rubric was used to assess a random sample of the chatbot's interactions according to accuracy and completeness, with the Reference and User Services Association (American Library Association) guidelines used to interpret the characteristics of "complete" answers. They further broke down the interactions into categories of user queries and identified three primary areas where the chatbot was unable to achieve accurate and complete answers: requests for an agent (i.e., a live staff member), requests for books, and requests for articles and research help.

Overall, there have been several articles that discuss the implementation of chatbots in libraries with earlier AI technology, but few reporting on GenAI chatbots. The current study builds directly on Lappalainen and Narayanan (2023)'s RAG-based approach and addresses the issue of incorrect link generation. In terms of evaluation, previous studies have recognized the importance of assessing multiple dimensions of chatbots' performance, with an emphasis on factual accuracy, and generally used some type of scoring scale for the evaluation.

Approach

Developing the RAG-based implementation

Conversational agent development

In this project, we used an LLM as an intermediary to facilitate interaction between the user and the knowledge or database. Retrieval augmented generation (RAG) is a method that combines information retrieval with LLM generation to produce accurate, context-aware responses to user prompts. RAG integrates two components to enhance the quality and the truthfulness of the responses (Danuarta et al., 2024):

- **Retriever:** responsible for identifying and retrieving the most relevant knowledge from a pre-defined vector database.
- **Generator:** uses retrieved information as context to generate an informed response with the capabilities of an LLM.

Figure 1 depicts our implementation. The retriever portion (top right portion of the scheme) was created by manually curating a set of pages from the library website into a knowledge base. This knowledge base was then coded into word vectors using the freely available word embedding from OpenAI and stored into a ChromaDB vector database.

We developed our workflow using the LangChain Python library, which allowed us to easily try different LLMs, both locally-hosted and cloud-based to run the experiments. In

our workflow, we added an extra step to mitigate the errors the chatbot tends to introduce when providing links to resources such as library databases. Our process, shown on the bottom right of the figure, took a very aggressive approach to ensure no incorrect links were provided to the user. The system first deleted every link on the generated response and ran a matching algorithm with an exhaustive list of URLs to library resources at the database level. When a match was found, the link was inserted in the response. This new corrected response was then passed again to the LLM to rephrase the response. This ensured that no incorrect links were included in the generated responses.

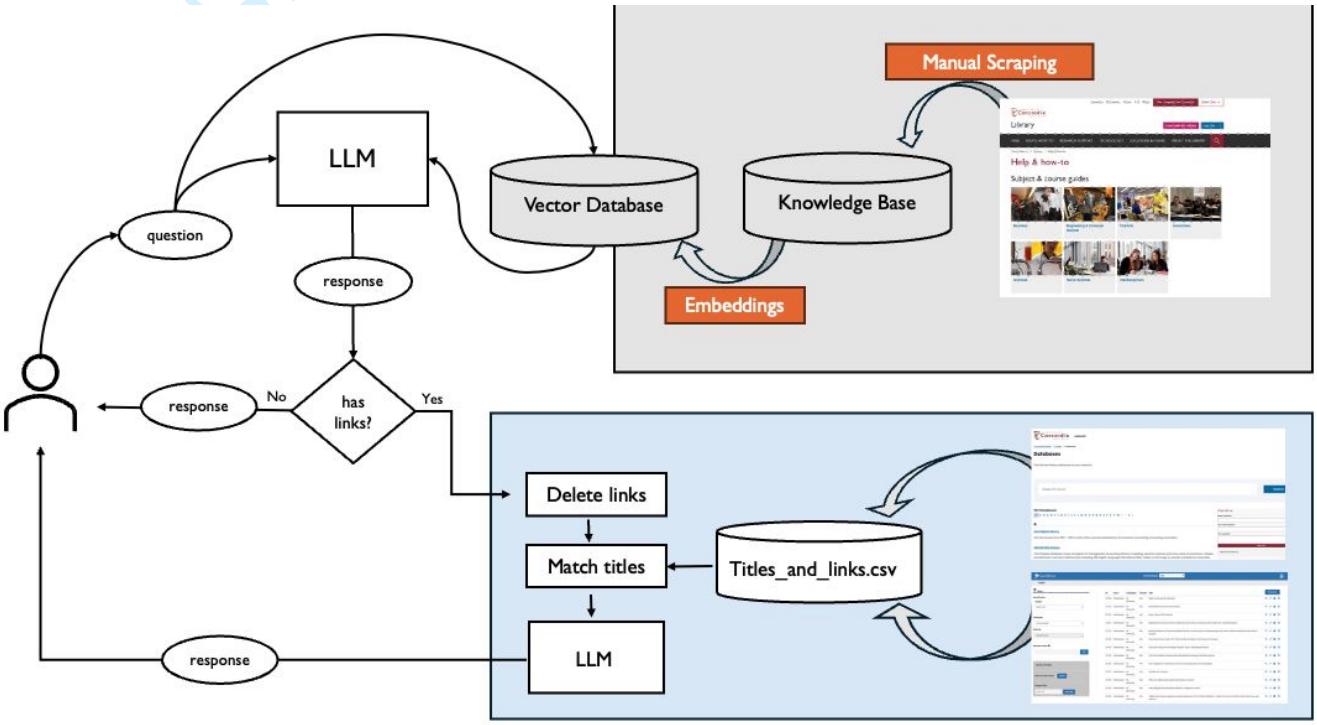

Figure 1. RAG implementation

A RAG process begins when a user submits a query, such as a library-related question: “How do I access eBooks?” The “retriever” component of the system identifies the most relevant information from the knowledge base by converting both queries and documents into vector representations using an embedding model. It then performs a similarity search, ranking the documents by relevance. The retriever selects the top-ranked documents (e.g. the top 5) and passes them to the generator. These documents serve as contextual inputs.

After that, the query is combined with the retrieved documents to generate an augmented query. This augmentation provides the generator with the necessary context to produce context-aware responses. The generator processes this augmented query to deliver a natural-language response that is both coherent and grounded in the retrieved

context. Finally, the system delivers the response to the user through Gaby's interface, ensuring clarity and relevance.

The following sections explain in more detail the different parts of the implementation.

Building the knowledge base

The chatbot Gaby's "knowledge" was built by manually scraping portions of the Concordia Library website and storing the information as articles in a knowledge base. We used Swallow, an in-house open-source metadata management system, for the knowledge base. This provided an interface with the system's contextual information, enabling us to verify whether the chatbot had access to the necessary background information to accurately answer specific questions during the performance evaluations.

Public-facing web pages were selected for inclusion in the knowledge base in order to provide a scoped, workable sample for the project. For the purpose of testing the chatbot's capabilities to respond to concrete queries that are frequently asked during reference interactions, selection was made with a focus on library information and pages that cover "how to" information. More specifically, the knowledge base included pages that cover information about borrowing materials (including requesting materials from other libraries and accessing ebooks), introductions to library services, several "how to find" pages (how to find articles, newspapers, data, government information, etc.), research data management guide, copyright guide, citation guides, and guidance on evaluating resources. These pages were prioritized as they contain institution-specific information that is less likely to have been ingested and "learned" by general-purpose LLMs. Pages excluded from the sample included subject guides, pages about research support services and open educational resources, and pages about Special Collections & Archives.

System prompt

The system configuration prompt is a crucial part of the RAG pipeline as it gives the chatbot instructions and personality. System prompts generally serve as instructions or templates that set the context for how the model interprets the augmented query and retrieved documents (LangChain, n.d.; Kansal, 2024). This is different from the prompt that end users input; rather, it directs the system's behavior (see Appendix D). Our system prompt was modeled after Lappalainen & Narayanan (2023) and included instructions for what the chatbot should do when questions could not be answered by its "knowledge," namely acknowledge that the question was not in scope, and also included instructions for referring users to library services. (See also similar examples in Olawore et al., 2025.)

Embeddings and vector database

Word embeddings convert text into vector representations that, in a sense, capture semantic meaning, enabling efficient retrieval and ranking of relevant information. The process of transforming the textual data from the knowledge base to word embeddings requires the use of an embedding model, which are precomputed models derived from massive amounts of texts that capture the relationship between words while representing documents in such a way that enables mathematical operations, including comparison such as cosine similarity (Olawore et al., 2025). In our implementation, we used the freely available OpenAI embeddings to vectorize our knowledge base.

These vectors are then stored in a vector database, namely ChromaDB in our particular implementation. This database engine allows for efficient similarity searches on the knowledge base during run time. This is what enables the generator to provide contextually relevant information, forming the backbone of the RAG pipeline.

Large Language Models

For the development of Gaby, we selected three different LLMs for the RAG implementation: OpenAI's ChatGPT Turbo 3.5, Google's Gemini, and Microsoft's Phi-3.

The models were selected for their popularity and wide availability, as well as their differences in size and features, as known prior to our testing, as summarized below. We chose the models despite some known limitations because of the potential of other benefits that would outweigh drawbacks.

1. OpenAI's ChatGPT Turbo 3.5

ChatGPT Turbo 3.5 is known as a reliable model. It offers a fast response time via API and does not require any specific hardware to run. It seamlessly integrates into the RAG implementation, facilitating easy experimentation.

Limitations: Proprietary, ongoing costs.

2. Google's Gemini 1.5 Pro

The Gemini model produces relatively good quality responses. It offers a fast response time via API. It is easy to integrate into the RAG pipeline but involves a monthly cost. Pre-implementation showed that the Gemini model had a tendency to produce falsehoods, at least in our particular setting.

Limitations: Proprietary, ongoing costs, less reliable.

3. Microsoft's Phi-3 Small Language Model

Phi-3 is a 3.8B parameter compact LLM designed for lightweight applications and enhanced groundedness in responses. It performs efficiently in resource-constrained

environments. It can be easily integrated into the RAG pipeline and does not require advanced hardware or RAM to run. Unlike ChatGPT and Gemini, it is run locally.

Limitations: Less capable due to its smaller size.

Creating the Interface

Finally, after developing the knowledge base of our chatbot and completing the RAG pipeline, we focused on adding a user-friendly interface. Like Lappalainen and Narayanan (Lappalainen & Narayanan, 2023), we used Streamlit, an open-source framework that streamlines the development of web applications. We were able to integrate our RAG application with it seamlessly to create an intuitive interface.

Evaluating the chatbot implementations

Questionnaire

In order to test the three versions of the chatbot, we created a questionnaire consisting of commonly-asked reference questions, per categories proposed by Arce & Ehrenpreis, 2023 and Reinsfelder & O'Hara-Krebs, 2023, based on their analyses of reference transaction logs, namely: directional, ready-reference, specific search, in-depth research, requests for information on a specific topic, course reserves/textbook access/streaming video, circulation (holds, borrowing policies, fines/fees), citation help (APA/MLA), and technical problems (for this study, we excluded known item queries as our chatbot was not configured to search catalogues or databases). We made an effort to word questions in a way that was natural to how they might be posed by university students and that were not necessarily explicit in what is being asked. Prior to developing Gaby, the project team discussed ways that the chatbot might be used to add value to services the library already does or could offer. For example, if a chatbot is merely repeating information directly from an FAQ, it's not really serving a value-added purpose. To that end, all the questions were answerable based on content that existed in the knowledge base but were not direct repetitions of the content. Some questions were fairly straightforward, while others would benefit from a more interactive process between the user and the respondent (whether human or AI). 14 common questions were chosen to represent different areas of information that were included in the knowledge base (which is only partial data from the library website as previously described) based on the teams' librarians' professional experience and expertise across the range of the categories prescribed by Arce & Ehrenpreis (2023) and Reinsfelder & O'Hara-Krebs (2023). Most are not in-depth reference questions but represent a variety of topics that might be addressed through a virtual reference interaction. Crucially, most questions were selected that would have the capacity to reveal whether the chatbot was

drawing specifically from the institutionally specific web pages rather than the LLMs' general "knowledge."

The 14-question questionnaire (see Appendix A) was run on the three versions of Gaby. A research assistant (RA) ran all interactions to ensure consistency. The RA saved all the interactions in a spreadsheet for the research team to view.

Evaluation rubric

The responses generated by each model were then scored according to the evaluation rubric (see Appendix B) by research team members, comprised of three librarians with extensive experience in reference/instruction and one research assistant who has a background as a researcher and teacher in the social sciences but no specific library training. The three librarians drew from their different perspectives to reflect on personal experience answering such questions, and the research assistant brought her student perspective. Had we wanted to focus more on the output quality, we might include more evaluators but our goal here was to assess how the process worked for doing such an evaluation as a proof of concept.

As a starting point, we considered the rubric defined by Lai (2023), who focused on evaluating three aspects: "completeness, accuracy, and the provision of further assistance" (977). While Lai sought to evaluate how well ChatGPT handled questions, we wanted to produce a protocol for testing different chat systems more comprehensively. To that end, we added the dimensions of "groundedness" and "elicitation" to the rubric.

In our final rubric, the chatbot's responses were evaluated according to the following categories:

- **Accuracy**: factual correctness, lack of errors, lack of falsehoods, use of terminology specific to the institution.
- **Groundedness**: provision of information derived from the knowledgebase.
- **Elicitation**: indication that further interaction with the system was possible, requested clarification of the inquiry when appropriate.
- **Completeness**: addressed the question fully.
- **Further Assistance**: referred the user to other relevant sources of help when appropriate.

Groundedness was an essential item to evaluate in our study, as it is the dimension in which we could assess whether the chatbot appeared to be drawing information accurately and as intended from the knowledge base rather than from the LLM's "knowledge." In other words, it is the dimension through which we could evaluate whether the RAG implementation was effective.

Elicitation was also an important item to add for our study. Librarians elicit information from users during a reference interview, and we wanted to see whether the chatbot could mimic a useful form of similar behaviour. Our chatbot implementation involved configuration of Gaby's behavior during user interactions, and adding this dimension allowed us to rate the chatbot's elicitation behavior, which was partially controlled by our configuration and partially by the LLM's inherent behavior.

We developed a 5-point scale within each of the dimensions of the rubric to allow for more nuance in scoring, in contrast with Lai's 3-point scale.

Each evaluator read and scored the responses to a given inquiry returned by each version of the chatbot before moving to the next interaction to repeat the scoring process.

Assessing the rubric

In order to assess the rubric's fitness for purpose in evaluating the performance of the chatbot versions, we calculated the standard deviation amongst the scores in each dimension for each model. We posited that where there is a low variation among scores, this could be an indication that rubric was sufficiently clear to evaluators and that the categories were a valid aspect of the chat interaction that could be evaluated.

Because there was no data collection from human research participants in this study, and publicly available information was used to populate the chatbot's knowledge base, ethics review was not required according to our institution's policies, in keeping with the Canadian national framework (Tri-Council Policy Statement: Ethical Conduct for Research Involving Humans – TCPS 2 (2022)).

Findings

Evaluating the chatbot implementations

Table 1 shows how the chatbot implementations scored in the given dimensions (see also average scores per question in Appendix C, Figures 4-8). The score was calculated as the mean of the ratings given by the research team members.

	OpenAI	Gemini	PHI-3	Average
Accuracy	4.04	4.14	3.67	3.95
Groundedness	4.33	4.41	4.20	4.31
Elicitation	3.42	1.75	2.50	2.56
Completeness	3.75	3.27	3.34	3.45
Further Assistance	3.48	2.81	4.20	3.50

Table 1. Model Comparison: mean score for all raters

Accuracy

The models' mean scores (calculated as a mean of all raters' scores) ranged between 3.67 and 4.14 points on our rubric scale for accuracy, meaning that they were generally factually accurate and used the institution's terminology in responses—but not always. An example of a question that reviewers deducted for accuracy was an inquiry about copyright. The chatbot referred to the concept of "fair use," which is an American legal concept, instead of "fair dealing" that should have been used in our particular context of Canadian copyright law.

Groundedness

All models averaged a mean score higher than 4 in the area of groundedness (ranging from 4.20 to 4.41), meaning that reviewers perceived that they derived information directly from the knowledge base, i.e., information from the institution's website. Where possible, we selected items for the questionnaire that made it possible to discern whether the chatbot was drawing information from the institution's website. This was confirmed during the scoring by reviewing the website against the chatbots' responses as well as inclusion of institution-specific terminology, procedures, and other details.

Elicitation

The mean scores were more variable for the elicitation dimension, ranging from 1.75 to 3.42. An example of successful elicitation was when the chatbot indicated that a further interaction was possible specifically in the context of the preceding information exchanged, such as concluding a response with "Is there any specific resource or assistance you require for your online class?" Reviewers gave a score of "1" when the chatbot provided an answer but did not indicate that further interaction was possible (it did not attempt to continue the conversation).

Completeness

The models scored on average between 3.27 and 3.75 for completeness. Reviewers based the evaluation of this dimension on whether the enquiry was fully answered and whether information was provided that a human reasonably would in the same circumstances. An example scored as lacking in completeness (usually scored as "3") was the response generated about downloading an e-book that referred to instructions focused on a summary of instructions for one type of e-book platform but didn't include information about other types of e-books available through the library.

Further assistance

The models had a wider range on the scores for further assistance, with their mean ranging from 2.81 to 4.20. Examples of high scoring responses in the dimension of further assistance included links to web pages with further information about the topic at hand or suggestions to consult a librarian for assistance.

Figure 2 below depicts the data from Table 1 as a visualization of the mean scores in each dimension per model.

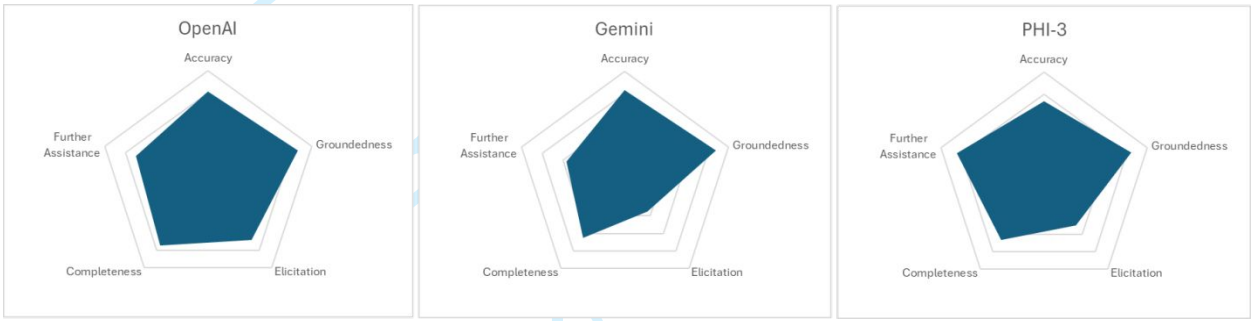

Figure 2. Model Comparison: mean score for all raters

Evaluating the rubric

As mentioned, our central objective, in addition to developing an understanding of the RAG technology, was assessing the evaluation protocol. As shown in Figure 3, we calculated the standard deviation amongst scores within each dimension to provide an indication of the rubrics' reliability across multiple evaluators.

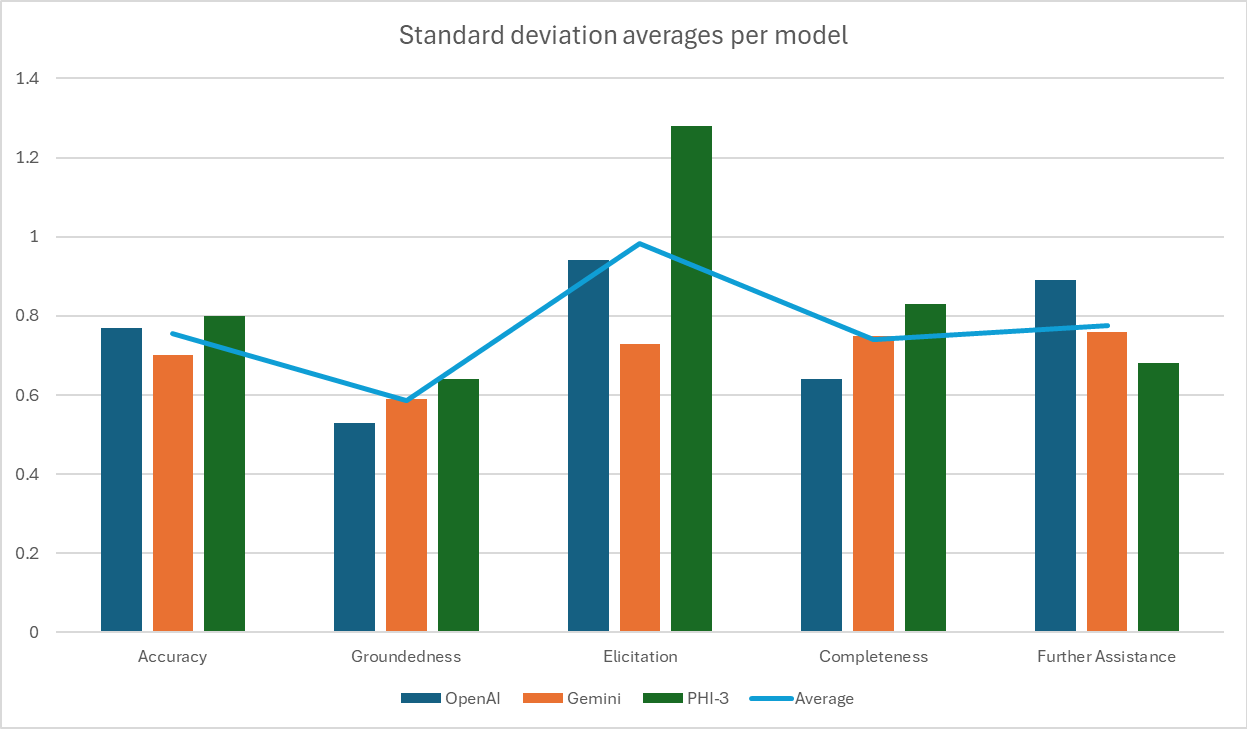

Figure 3. Standard deviation averages per model

The elicitation dimension had the greatest variance across evaluators with an average of 0.98 for all models, while the groundedness dimension had the most uniform scores (lowest deviation) with an average of 0.59 for all models.

Discussion

Chatbot implementation evaluation

We found the most well-rounded model to be OpenAI, with the highest scores in each dimension except further assistance. Phi-3 was a close second, however, with higher scores in further assistance and lower scores in elicitation and completeness.

In the accuracy, it is worth noting that GenAI is a probability machine and as such is not reliable to provide an ideal response every time. Even when used in the context of a RAG system, our tests did not return a 100% accuracy level with any model. A larger scale evaluation over a wider range of questions with more evaluators would be necessary to determine whether there were patterns in the types of accuracy challenges encountered by the chatbot and how minor they were. Previous studies indicate that in the context of library reference, accuracy gaps are most likely to be in the area of complex queries that require subject knowledge (Lai, 2023; Yang & Mason, 2024) as well as local, real-time information that is outside the scope of the knowledge base (Lappalainen & Narayanan, 2023).

That said, the relatively high scores in the area of groundedness confirmed that the RAG approach was generally, although not wholly, successful in restricting the chatbot's responses to the information in the knowledge base. It demonstrably drew information from the knowledge base, frequently using library terminology such as the library's building names and discovery tool. However, in some cases, answers were discernibly generated that did not correspond to the knowledge base. For example, Phi-3 generated a response that read "For additional scholarly resources, consider searching via Digital Object Identifier (DOI) systems. While I cannot provide specific links here, you can access these databases through Concordia's online portal." This response doesn't make logical sense, and the library doesn't use the terminology "portal" to refer to online resources.

In the area of completeness, the testing conditions were somewhat artificial as the chat interactions were ended arbitrarily, but potentially a real interaction could continue longer to increase the completeness scores. Models that tended to be more verbose in the initial response may therefore have received higher completeness scores than those that are tuned for shorter, more iterative interactions that may have been cut off prematurely. However, the rubric did prove to be of practical use in evaluating the completeness of a response in relation to the standard of a human response.

Further assistance and elicitation are both areas that are the most "controllable" on the development side of the system configuration and less inherent to the LLMs. As previously mentioned, Gaby's configuration prompt included the instruction to ask follow-up questions and to suggest speaking to a human librarian. The prompt could potentially be improved or better refined through trial and error to produce better "elicitation" and "further assistance" scores in each of the models. With the configuration prompts we used, we found that OpenAI performed much better than Phi-3 and Gemini in indicating that ongoing interaction was possible by including questions that allowed the user to clarify the need or area of interest or instructing the user to specify what information would be useful next to continue the interaction in context. In terms of further assistance, Phi-3 was consistent in tacking on the suggestion from the configuration prompt to seek help from a "human librarian," while Gemini rarely followed the instruction. OpenAI was mixed in including a suggestion at the end to seek further assistance in the library and sometimes suggested further assistance that was more in the context of the interaction.

Rubric evaluation

One aspect of the evaluation was calculating the deviation among scores awarded to the models' responses. The highest deviation among scores awarded to the models' responses was in the elicitation dimension. This was likely due to the subjectivity of interpreting what further interaction may look like. The rubric could be improved to

provide examples of what might constitute elicitation in different types of models so that it could be better applied to evaluate ideal interactions in a reference setting where users are invited to provide input that shapes the interaction. As mentioned, elicitation is also something that is more controllable in system development than inherent to the LLM, so it may be configured to optimize to the behavior of particular models.

Across the other dimensions, there was less variation numerically (0.59 to 0.78 on average), but in debriefing discussions, we found variations in how we interpreted the models' responses in relation to the criteria of our rubric. This suggests that making the rubric definitions more precise would lead to a more accurate and granular comparison of the models. Some rubric definitions, such as accuracy, elicitation, and further assistance, included multiple indicators within each level on the scale, and splitting these into subcategories would also improve consistency in evaluation.

Debriefing discussions also revealed variations in opinions about what constituted acceptable responses to questions and what thresholds each evaluator had for an acceptable output from an AI tool, which sometimes depended on the nature of the query. Using the rubric also raised questions about the proper placement of a GenAI chat tool on a library website or alongside existing reference services. Is a low-level of error enough to be useful for simple queries? Is a lack of elicitation an unacceptable flaw for a tool to augment reference help? One team member raised the possibility that even if the tools aren't good enough for more involved inquiries, providing a small bit of utility could be useful toward helping someone who would normally be reticent to contact the library at all to start interacting and eventually maybe seek more help.

In the end, we determined that the rubric was fit for purpose in helping us determine whether the models being tested achieved the RAG technique, compare their performance in accuracy and completeness, and identify aspects of desired interactive behaviors for eliciting interaction and suggesting further assistance. Further tweaks to the rubric are necessary to improve consistency among evaluators.

Limitations

One potential limitation of the RAG approach lies in the challenge of ensuring the accuracy of the URLs included in the responses generated. LLMs sometimes generate non-existent or incorrect links. To address this, we designed a link correction algorithm that leverages validation techniques alongside LLMs' NLP capabilities. The process is as follows:

1. Initial Link Removal: The system scans the generated response for any links. If any links are detected, they are removed to eliminate potential inaccuracies.
2. Matching Titles with Links: A CSV file containing verified titles and their corresponding URLs (e.g., "Finding Ebooks" and

"library.concordia.ca/finding/ebooks") is used. The system scans the response for matching titles from the CSV file.

3. Link Insertion: If a title is found in a response, it is replaced with a formatted reference, including the title and the verified URL. For example, "Finding eBooks" in the response becomes "Finding eBooks: library.concordia.ca /finding/eBooks."
4. Inconsistency Fixing: To address potential inconsistencies (e.g., "I do not have the link for the guide on finding eBooks: library.concordia.ca /finding/ebooks"), the updated response is passed back to the LLM with a prompt specifically designed to correct inconsistencies in the text.

While this approach greatly improves the reliability of the responses, there remains a slim chance of the LLM introducing new links that are not present in the database during the second pass through the LLM.

Another potential limitation of the study was that the configuration prompts were developed first for the OpenAI API and then used for the other models. This may have biased the results, especially in the areas of elicitation and further assistance to the OpenAI model. The other models may have performed differently or more effectively if they were tuned individually. However, since our primary goal was to test the RAG technology and to develop an evaluation method, we were not as concerned with the raw score of each model in the context of this study.

In addition, this study used LLMs that were available at the time of testing. GenAI technology continues to evolve, and the capabilities of more recent models may provide different and more contextually relevant results than the models available at the time of this study. It should also be noted that in attempting to constrain the chatbot's responses to library-specific (knowledgebase) information, the user experience could potentially be limited from the benefits of the full utility of the LLM. The pros and cons of a RAG-based but manually implemented chatbot versus an unconstrained but not contextually-specific LLM could be explored in future research.

This study and the criteria in the evaluation rubric were designed to help us evaluate the efficacy of the RAG technique in the context of delivering information services. It's worth noting that this should not be the only set of criteria considered before choosing to implement such a service. Other factors are also extremely important to evaluate, including (but not limited to):

- accessibility
- resource consumption
- jurisdiction
- content ownership
- license requirements

- privacy
- security.

Conclusion

Overall, this study found that the RAG implementation with a local, static knowledgebase was generally successful in constraining the LLMs to generate contextual and accurate responses with library information, but there are limitations to the approach. These included a less-than-100% accuracy as well as the need to populate and update the knowledgebase manually.

Therefore, an institution considering an approach following the steps described here would need to weigh the need for in-house technical capacity and time required for manual knowledgebase updates with potential benefits like chatbot responses that are contextually relevant to local users and potentially lower resource consumption and subscription costs if a smaller LLM is chosen or if the LLM is run locally, compared with the use of out-of-the box general GenAI chatbots.

The testing protocol and rubric allowed us to differentiate between models and could be used for decision-making with some improvements. We found that the protocol for testing requires iterations, primarily to fine-tune how we perceive what is most pertinent and essential in determining an acceptable response to a library user's query. It should be noted that evaluation of performance is inherently subjective, in some dimensions more than others. In addition to developing technical knowledge, the experience of this study led to fruitful discussions of the value of GenAI technology, where it is appropriate, and how it may fit into reference processes (if at all), which are essential questions to be answered before adopting the technology.

Potential next steps

Given that there was not much difference found in the performance of the OpenAI model and the much smaller Phi-3, a potential next step of this project is to fine tune the configuration for Phi-3 and re-evaluate the performance. The conclusion that smaller language models may perform as well as larger ones in a RAG context is an important possible finding from a resource conservation perspective. Testing with a wider array of questions, including authentic user questions, with a revised rubric would also further indicate the viability and utility of the testing protocol, which could then lead to end-user testing of a chatbot tool, potentially also with newer versions of LLMs.

References

Arce, V., & Ehrenpreis, M. (2023). Improving a library FAQ: Assessment and reflection of the first year's use. *The Reference Librarian*, *64*(1), 35–50. https://doi.org/10.1080/02763877.2023.2167898

Barus, S. P., & Surijati, E. (2022). Chatbot with Dialogflow for FAQ services in Matana University Library. *International Journal of Informatics and Computation*, *3*(2). https://doi.org/10.35842/ijicom.v3i2.43

Bryant, R. (2024, December 12). Implementing an AI reference chatbot at the University of Calgary Library. *Hanging Together*. https://hangingtogether.org/implementing-an-ai-reference-chatbot-at-the-university-of-calgary-library/

Cox, A. (2023). How artificial intelligence might change academic library work: Applying the competencies literature and the theory of the professions. *Journal of the Association for Information Science and Technology*, *74*(3), 367–380. https://doi.org/10.1002/asi.24635

Danuarta, L., Mawardi, V. C., & Lee, V. (2024). Retrieval-Augmented Generation (RAG) Large Language Model for educational chatbot. *2024 Ninth International Conference on Informatics and Computing (ICIC)*, 1–6. https://doi.org/10.1109/ICIC64337.2024.10957676

Ehrenpreis, M., & DeLooper, J. (2025). Chatbot assessment: Best practices for artificial intelligence in the library. *Portal: Libraries and the Academy*, *25*(4), 671–702.

Ehrenpreis, M., & DeLooper, J. (2022). Implementing a chatbot on a library website. *Journal of Web Librarianship*, *16*(2), 120–142. https://doi.org/10.1080/19322909.2022.2060893

Feng, Y., Wang, J., & Anderson, S. G. (2024). Ethical considerations in integrating AI in research consultations: Assessing the possibilities and limits of GPT-based chatbots. *Journal of eScience Librarianship*, *13*(1), e846. https://doi.org/10.7191/jeslib.846

Guy, J., Pival, P. R., Lewis, C. J., & Groome, K. (2023). Reference Chatbots in Canadian Academic Libraries. *Information Technology and Libraries*, *42*(4). https://doi.org/10.5860/ital.v42i4.16511

Ivanovskaya, A., Aksyonov, K., Kalinin, I., Chiryshev, Y., & Aksyonova, O. (2019). Development of the text analysis software agent (chat bot) for the library based on the question and answer system TWIN. *ITM Web of Conferences*, *30*, 04006. https://doi.org/10.1051/itmconf/20193004006

Kane, D. (2019, April 10). *Analyzing an interactive chatbot and its impact on academic reference services*. ACRL 19th National Conference, Cleveland, Ohio. http://hdl.handle.net/11213/17624

Kansal, A. (2024). *Building generative AI-powered apps: A hands-on guide for developers*. Apress. https://doi.org/10.1007/979-8-8688-0205-8

Lai, K. (2023). How well does ChatGPT handle reference inquiries? An analysis based on question types and question complexities. *College & Research Libraries*, *84*(6). https://doi.org/10.5860/crl.84.6.974

LangChain. (n.d.). Overview. https://docs.langchain.com/oss/python/langchain/overview

Lappalainen, Y., & Narayanan, N. (2023). Aisha: a custom AI library chatbot using the ChatGPT API. *Journal of Web Librarianship*, *17*(3), 37–58. https://doi.org/10.1080/19322909.2023.2221477

Olawore, K., McTear, M., & Bi, Y. (2025). Development and evaluation of a university chatbot using deep learning: A RAG-based approach. In A. Følstad, S. Papadopoulos, T. Araujo, E. L.-C. Law, E. Luger, S. Hobert, & P. B. Brandtzaeg (Eds.), *Chatbots and Human-Centered AI* (pp. 96–111). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-88045-2_7

Panda, S., & Chakravarty, R. (2022). Adapting intelligent information services in libraries: A case of smart AI chatbots. *Library Hi Tech News*, *39*(1), 12–15. https://doi.org/10.1108/LHTN-11-2021-0081

Reinsfelder, T. L., & O'Hara-Krebs, K. (2023). Implementing a rules-based chatbot for reference service at a large university library. *Journal of Web Librarianship*, *17*(4), 95-109. https://doi.org/10.1080/19322909.2023.2268832

Rodriguez, S., & Mune, C. (2022). Uncoding library chatbots: Deploying a new virtual reference tool at the San Jose State University library. *Reference Services Review*, *50*(3/4), 392–405. https://doi.org/10.1108/RSR-05-2022-0020

Thalaya, N., & Puritat, K. (2022). BCNPYLIB CHAT BOT: The artificial intelligence Chatbot for library services in college of nursing. *2022 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT & NCON)*, 247–251. https://ieeexplore.ieee.org/abstract/document/9720367/

University of Oklahoma Libraries. (n.d.). *Project Highlight: Bizzy Chat Bot | OU Libraries*. Retrieved March 13, 2025, from https://libraries.ou.edu/content/project-highlight-bizzy-chat-bot

University of Texas Libraries. (2024, November 30). Building a bot: An exploration of AI to assist librarians. *TexLibris*. https://texlibris.lib.utexas.edu/2024/11/building-a-bot-an-exploration-of-ai-to-assist-librarians/

Yang, S. Q., & Mason, S. (2024). Beyond the algorithm: understanding how ChatGPT handles complex library queries. *Internet Reference Services Quarterly*. https://doi.org/10.1080/10875301.2023.2291441

Appendices

Appendix A: Test questionnaire

1. What should I do if I have a link and it's broken
2. Can I do an online class at the library?
3. How do I know if an article is peer-reviewed?
4. Can I rent textbooks?
5. How can I find primary sources?
6. Can I show a film in my class
7. Can I include an image from a website in my thesis
8. I have a research essay and don't know where to start
9. How do I request a book?
10. What if I need a book that Concordia doesn't have?
11. How can I download an eBook?
12. How can I find articles about social media methodology
13. How do I cite a source that I found referenced in another work?
14. Can you give me a link to a database for articles on the effects of climate change?

Appendix B: Evaluation rubric

	1	2	3	4	5

Accuracy	The information provided had factual inaccuracies. Included hallucinations. Did not use Concordia Library terminology.	Some of the information provided was correct while some was inaccurate. May have included hallucinations. Did not use Concordia Library terminology.	Most of the information provided was factually correct but included some errors. May have included hallucinations. Sometimes, but not always, used Concordia Library terminology.	Most of the information provided was factually correct but may have been misleading I some way. Did not include hallucinations. Used Concordia Library terminology.	All information provided was factually correct. Used Concordia Library terminology.
Groundedness	None of the information provided appeared to be derived from the knowledgebase.	Little of the information provided appeared to be derived from the knowledgebase.	Around half of the information appeared to be derived from the knowledgebase.	Most of the information appeared to be derived from the knowledgebase.	All of the information appeared to be derived from the knowledgebase.
Elicitation	The system did not elicit any information or precision from the user, nor did it indicate that further interaction was possible	The system provided a generalized indication that further interaction was possible.	The system indicated that a specific type of ongoing interaction was possible.	The system requested that the user clarify the question or provide additional information in order to properly answer.	The system requested that the user clarify the question or provide additional information and indicated lateral avenues of inquiry for the user to explore.
Completeness	Did not address any aspect of the question.	Only partially addressed the question.	Addressed the question but more information could reasonably be expected to be provided.	Addressed the question adequately.	Completely addressed all the question by offering relevant information beyond what was immediately asked to the level that a

					human reasonably would.
Further assistance	Did not do any of the following: Referred to other relevant sources/help when not able to fully answer question, or provided accurate additional information beyond initial inquiry; Invited user to contact a librarian.	Did not do any of the following but it did not impede the interaction: Referred to other relevant sources/help when not able to fully answer question, or provided accurate additional information beyond initial inquiry; Invited user to contact a librarian.	Did one of the following but in a way that didn't appear to be immediately useful: Referred to other relevant sources/help when not able to fully answer question, or provided accurate additional information beyond initial inquiry; Invited user to contact a librarian.	Did one of the following: Referred to other relevant sources/help when not able to fully answer question, or provided accurate additional information beyond initial inquiry; Invited user to contact a librarian.	Did one or more of the following in a helpful and natural manner in the context of the interaction: Referred to other relevant sources/help when not able to fully answer question, or provided accurate additional information beyond initial inquiry; Invited user to contact library staff.

Appendix C: Ratings per question by model

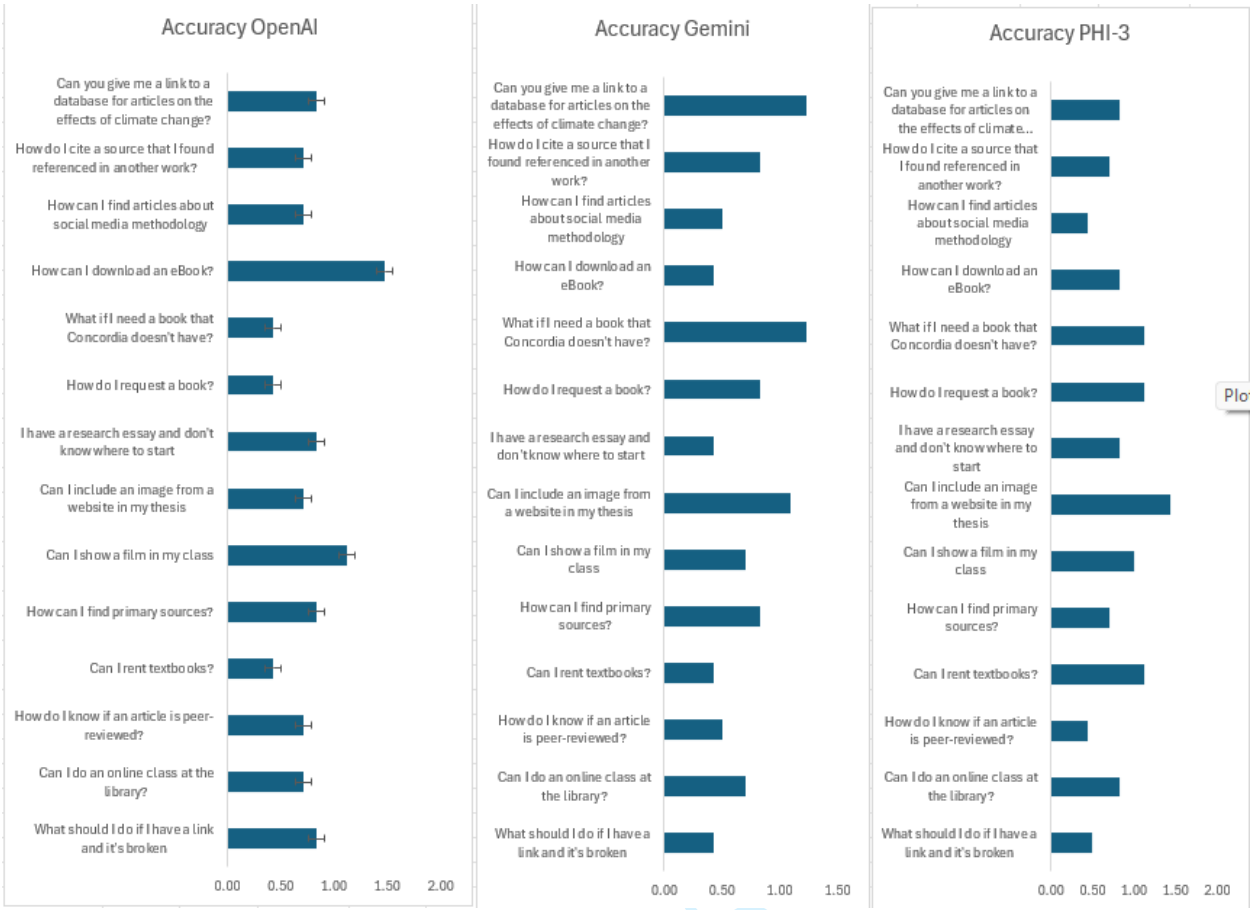

Figure 4. Average scores per question for each model in the accuracy dimension

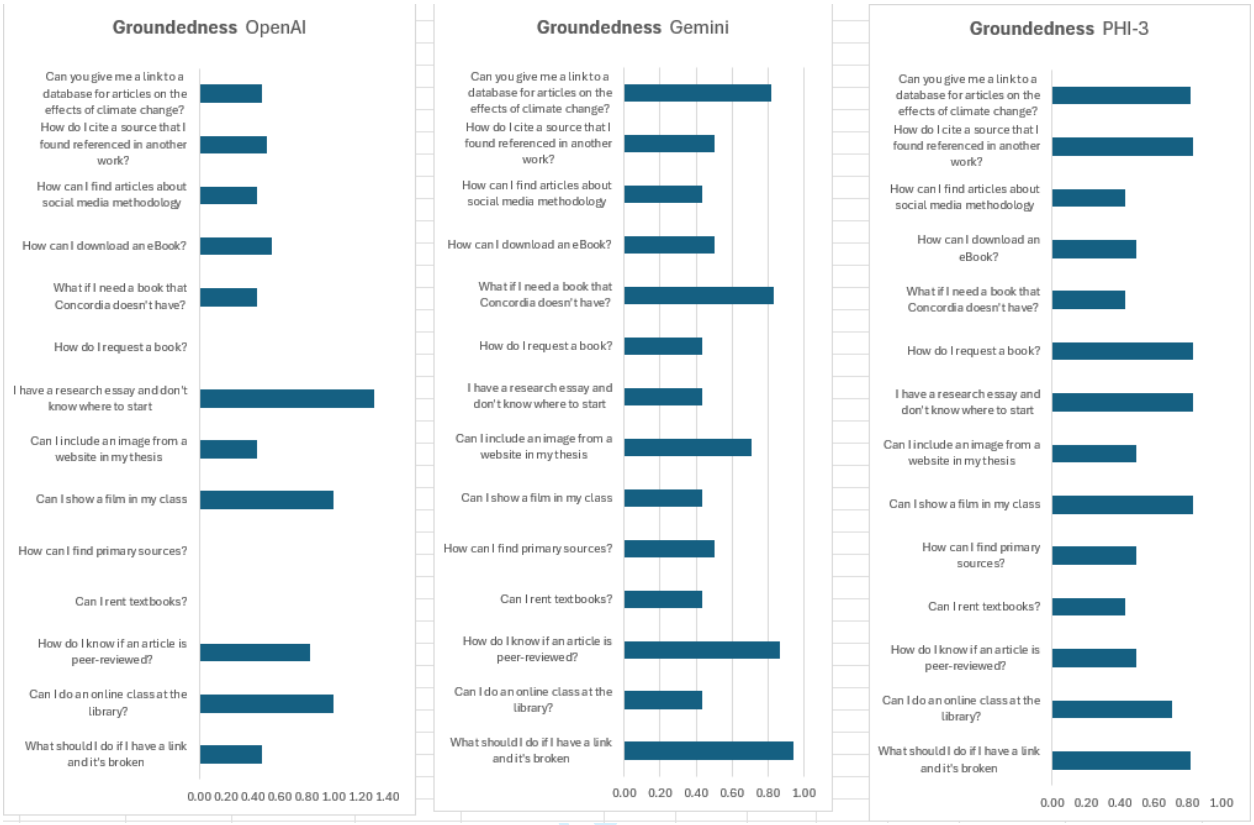

Figure 5. Average scores per question for each model in the groundedness dimension

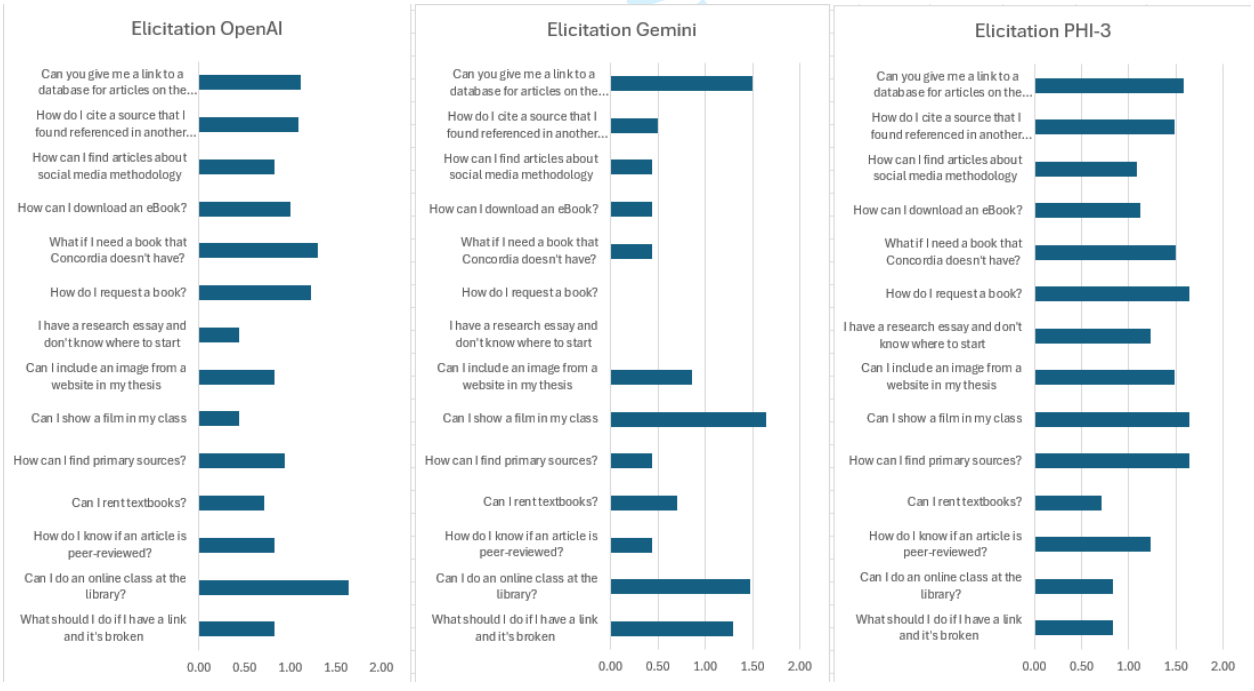

Figure 6. Average scores per question for each model in the elicitation dimension

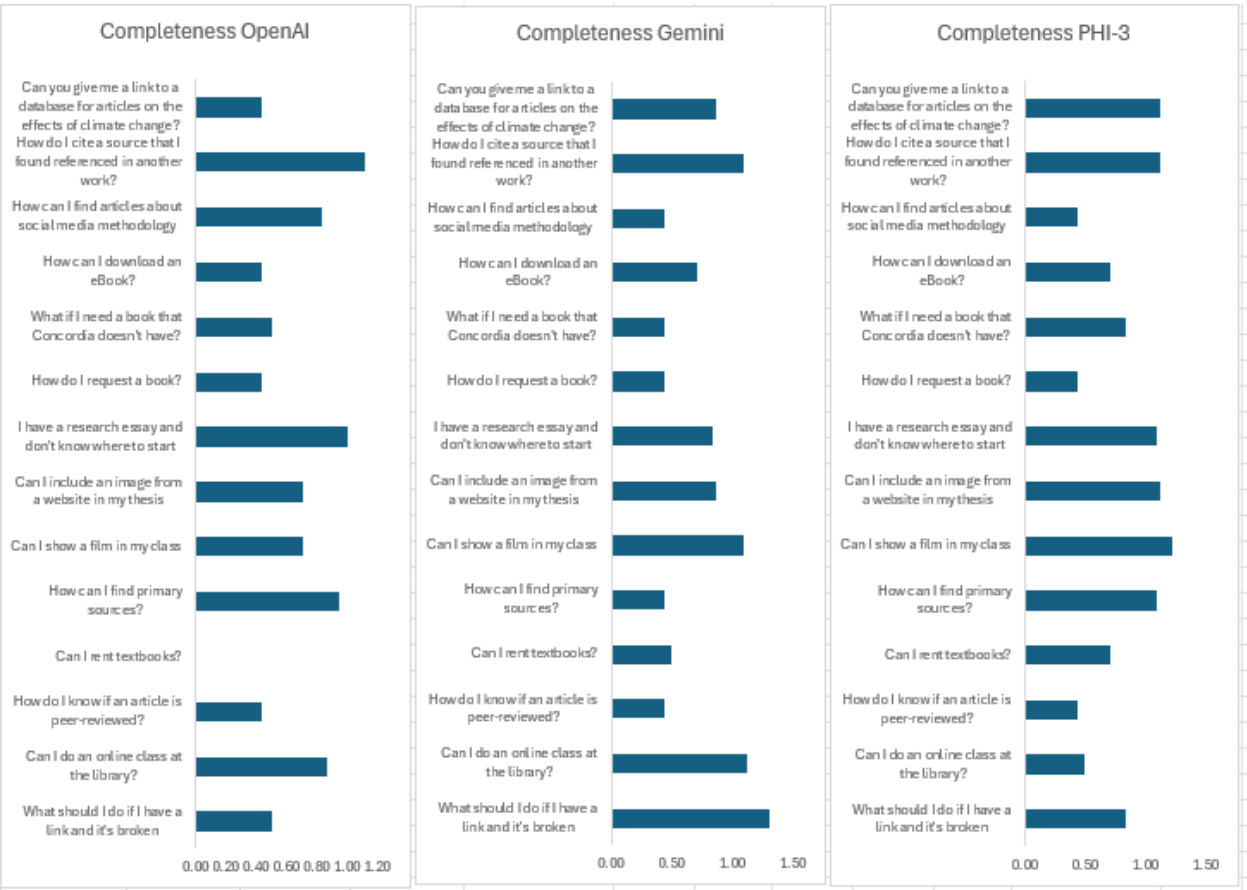

Figure 7. Average scores per question for each model in the completeness dimension

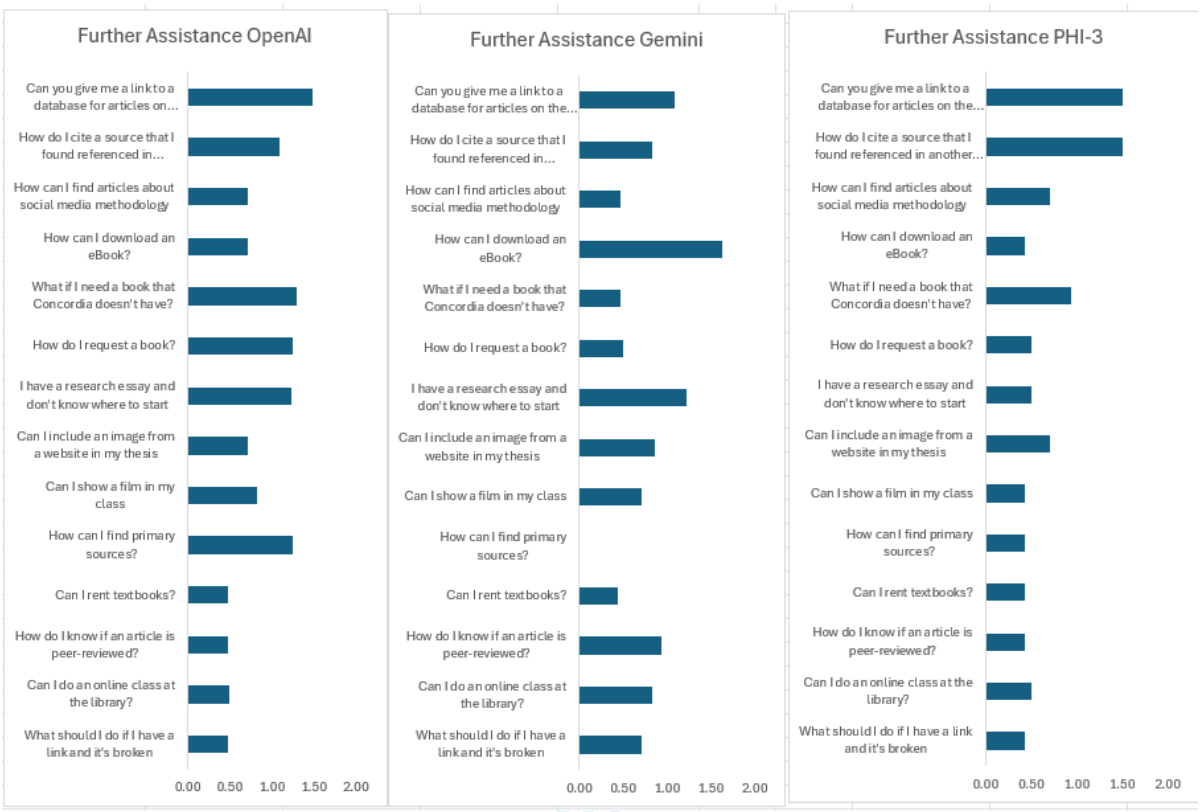

Figure 8. Average scores per question for each model in the further assistance dimension

Appendix D: Technical documentation

Configuration prompt

We used two configuration prompts to direct system behavior:

1. Behavior-specific prompt: a custom prompt was designed to adapt the tone and the style of the responses. The first prompt was:

> `You are Gaby, a helpful and resourceful AI library assistant at Concordia Library. Answer the questions from the perspective of Concordia Library. Ask follow-up questions for clarification if needed. If you don't know the answer, say that you don't know and suggest speaking to a human librarian. Only provide links that are available in the context. If asked about recommendations for books or articles always provide the link to the Sofia Discovery Tool and never recommend books`

2. Correction prompt: During the link correction process, a prompt was designed to address the inconsistencies in the first generated response. The second prompt was:

```
Rewrite this to be more grammatically correct. Use clearer language.
```

Hardware and Software Requirements

- **Local (Ollama: Llama 2, Llama 3, Phi-3):**

To run Ollama, you would need a Linux OS preferably. A windows version is available for preview only for Windows 10 or 11, and a version of macOS is available for **macOS 11 Big Sur or later.**

Command

Install Ollama (Linux Ubuntu)

curl -fsSL https://ollama.com/install.sh | sh

COPYDOWNLOAD

The instructions that were followed to install Ollama are available here: https://github.com/ollama/ollama

The RAM requirements as provided by Ollama.

1. Llama 2 can be run with 8GBs of RAM
2. Llama 3 requires more RAM depending on the number of parameters you choose
3. Phi-3 Mini can run easily with 8GBs of RAM

Software needed:

1. Visual Studio Code
2. Python

Installation and Setup

Install Dependencies: Make sure you have Python installed. Then, install the required Python packages:

pip install -r requirements.txt

To set up the project locally, follow these steps:

1. **Clone the Repository**:

git clone [Will insert URL but identifies author and institution]

cd gaby

2. **Set Up OpenAI API Key**

Create a credentials.json file in the directory with your OpenAI API key:

```
[
    {
        "service_provider": "openai",
        "key": "your-openai-api-key"
    }
]
```

3. **Prepare the CSV File** Add your CSV file named titles_and_links.csv in the directory. The titles_and_links.csv file should contain two columns:

a. **Title**: This represents the name or topic that the chatbot might refer to in its responses.

b. **Link**: This is the URL that corresponds to the title, which will be inserted into the chatbot's response when the title is mentioned.

4. **Download Ollama**

To run an open source model like Llama or Phi3 locally, you first need to download Ollama:

https://ollama.com/download

After downloading Ollama, choose which model you want to use from the models table: https://github.com/ollama/ollama?tab=readme-ov-file#model-library and run:

ollama pull llama3.1

You will then be able to use the model of your choice in your code.

Customization

You can customize the behavior and responses of the chatbot by adjusting the prompt templates or changing the temperature settings of the language model. These customizations allow you to fine-tune the chatbot's tone, formality, and creativity.

1. Prompt Customization

The chatbot's responses are influenced by the system prompts and user prompts defined in the code. You can modify these prompts to adjust how the chatbot behaves.

System Prompt

The system prompt defines the general behavior and constraints of the chatbot. It's set up to make the chatbot respond in the context of Concordia Library. You can find and modify this prompt in the system_prompt variable within the code.

Example:

```
system_prompt = (
    """You are Gaby, a helpful AI library assistant at Concordia Library.
    Answer the questions from the perspective of Concordia Library.
    Ask follow-up questions for clarification if needed. If you don't know the answer, say that you don't know
    and suggest speaking to a human librarian. Only provide links that are available in the context.
    If asked about recommendations for books or articles always provide the link to the Sofia Discovery Tool and never recommend books."""
    "\n\n"
    "{context}"
)
```

To Customize: You can adjust the text within the triple quotes to change how the chatbot interacts with users. For example, you can make the chatbot more formal or casual, or you can focus on different aspects of library services.

2. Temperature Setting

The temperature setting controls the creativity and variability of the chatbot's responses. A higher temperature will make the responses more creative and diverse, while a lower temperature will make them more deterministic and focused.

To Customize: Change the temperature parameter to a value between 0 and 1:

-Lower Temperature (e.g., 0.2): The chatbot will provide more precise and consistent answers, suitable for technical or formal contexts.

-Higher Temperature (e.g., 0.9): The chatbot will generate more varied and creative responses, which can be useful in brainstorming sessions or less formal contexts.

There are two ways to set or change the temperature.

Method 1: Changing Temperature Through ChatOpenAI Object

You can set the temperature directly when initializing the ChatOpenAI object in your code.

llm = ChatOpenAI(model="gpt-3.5-turbo", temperature=0.7)

Method 2: Changing Temperature Through Credentials File

Alternatively, you can adjust the temperature setting in the credentials file used to authenticate and configure the language model. This method is particularly useful if you want to centralize your model configuration or if you're deploying the bot in different environments.

Example Credentials File:

```
[
   {
      "service_provider": "openai",
      "key": "your_key",
      "model": "gpt-3.5-turbo",
      "temperature": "0.7"
   },
   {
      "service_provider": "google",
      "key": "your_key",
      "model": "gemini-pro",
      "temperature": "0.6"
   }
]
```

Running the Chatbot

After customizing the chatbot, whether by adjusting prompts, temperature settings, or other parameters, you need to generate the embeddings and set up the vector database to reflect these changes.

Step 1: Generate Embeddings

Once you've made your customizations, run the 01_generate_embeddings.py script to generate the necessary embeddings based on your updated settings. These embeddings are essential for creating a vector database that the chatbot will use to provide contextually relevant responses.

python 01_generate_embeddings.py

After running this script, a .chroma directory will be created in your project folder. This directory contains the vector database, which stores the embeddings generated from your documents or data sources.

Step 2: Run the Application

With the embeddings generated and the vector database in place, you can now run the application using Streamlit.

streamlit run 02x_gaby_version.py

Running the App: This command will launch the Streamlit application, allowing you to interact with your customized chatbot. Make sure that the .chroma directory and the necessary configuration files are present in your working directory, as they are required for the chatbot to function correctly.

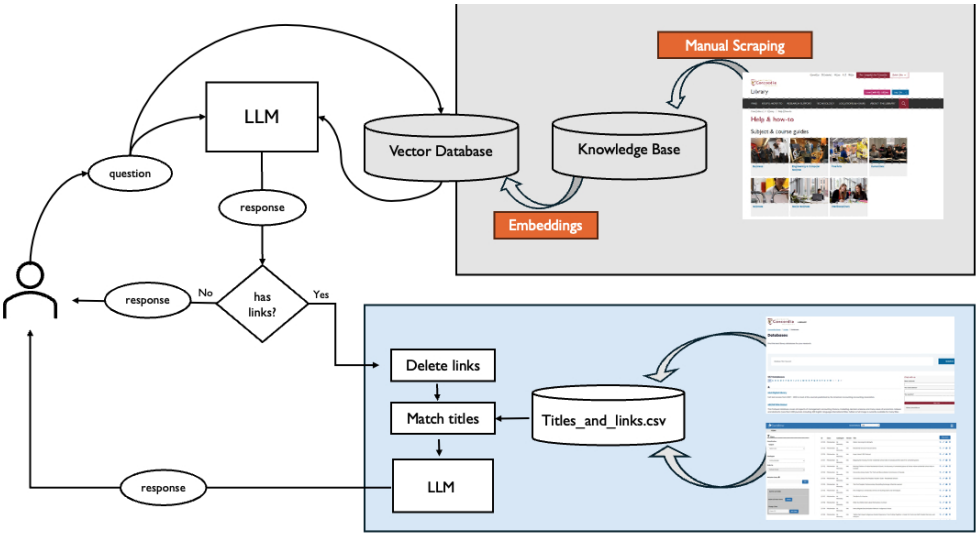

Figure 1. RAG implementation. Source: Authors' own work

384x206mm (72 x 72 DPI)

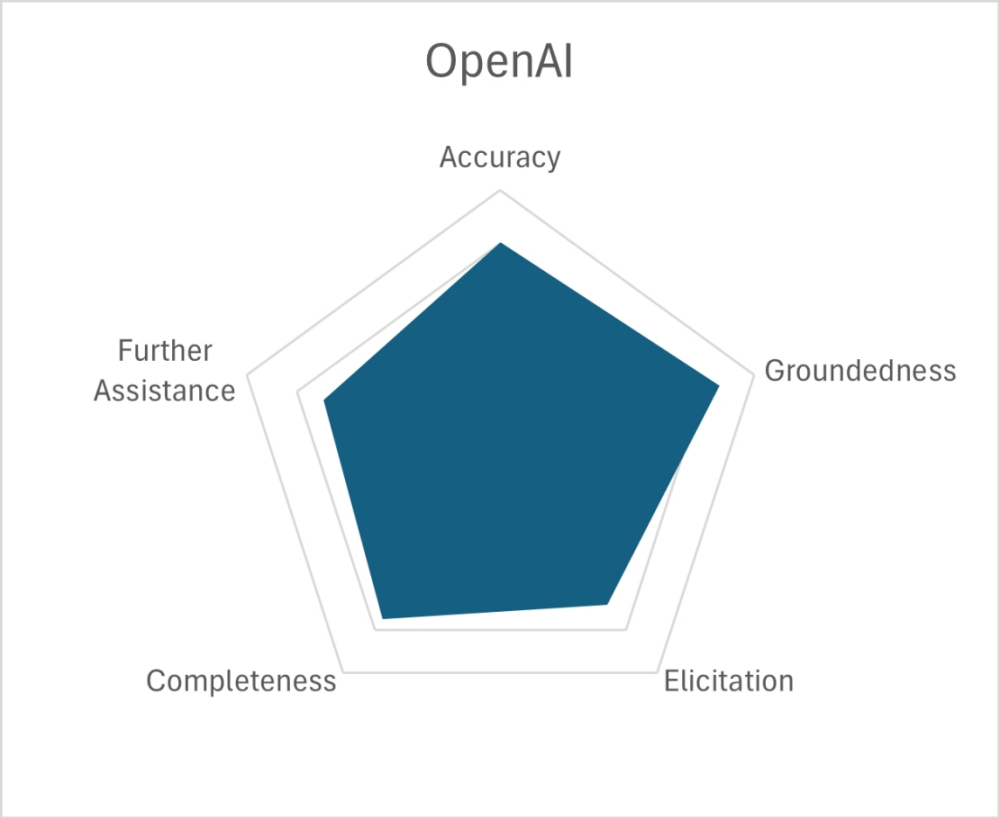

Figure 2. Model comparison. Source: Authors' own work

245x200mm (130 x 130 DPI)

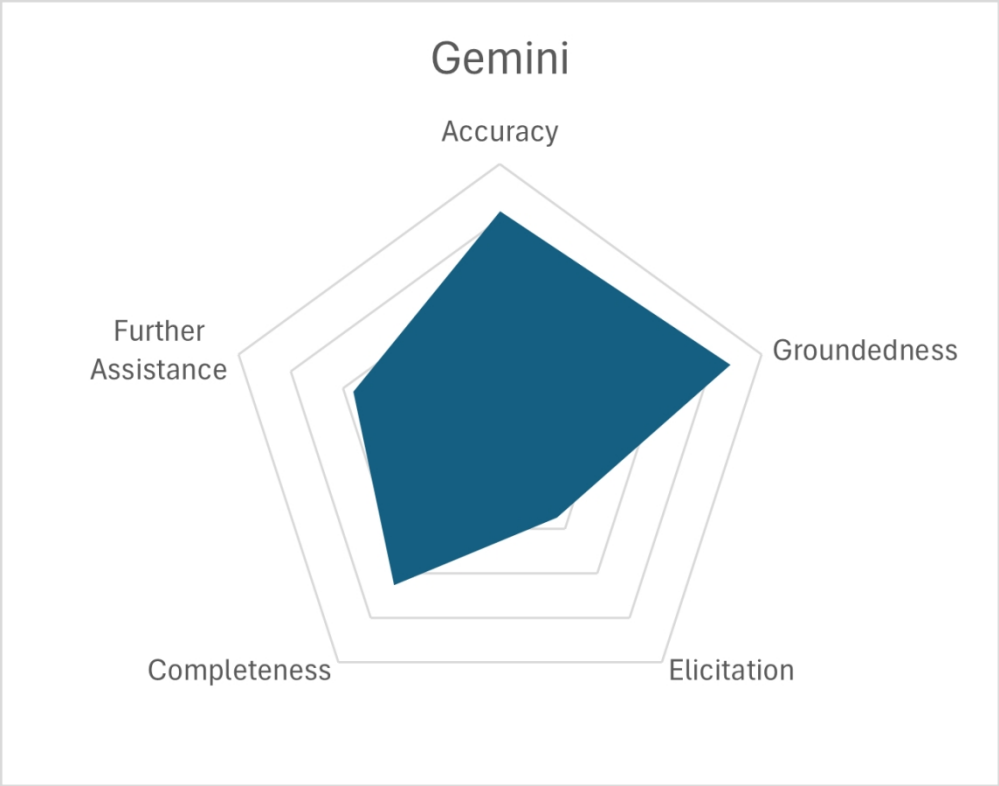

Figure 2. Model Comparison. Source: Authors' own work

254x200mm (130 x 130 DPI)

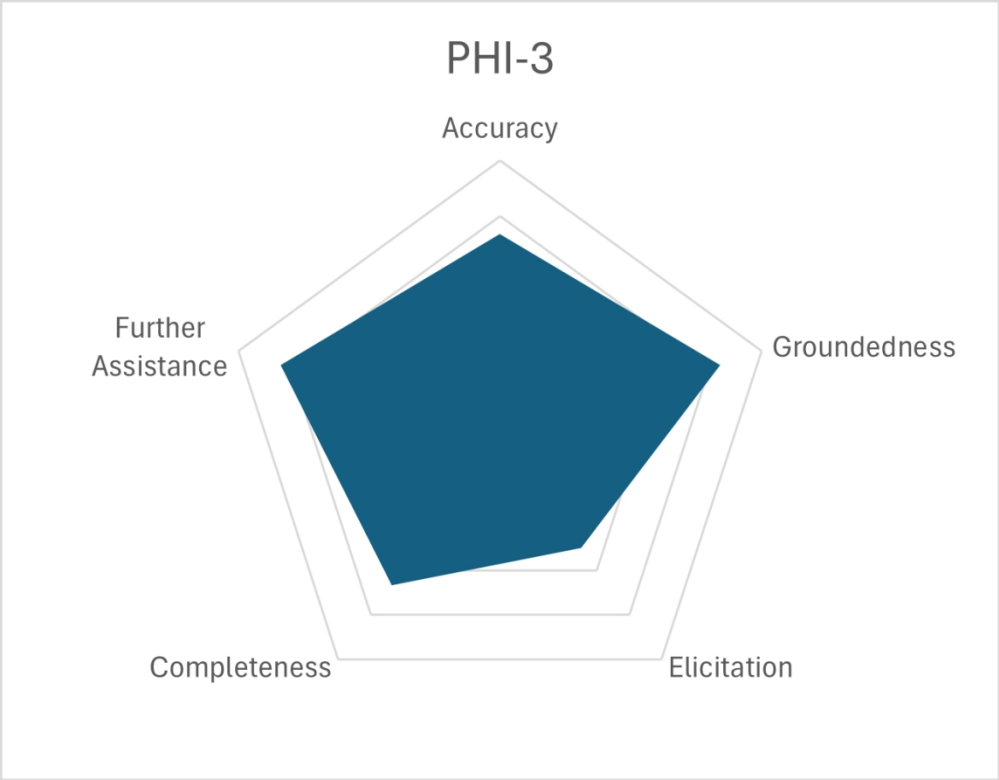

Figure 2. Model Comparison. Source: Authors' own work

256x200mm (130 x 130 DPI)

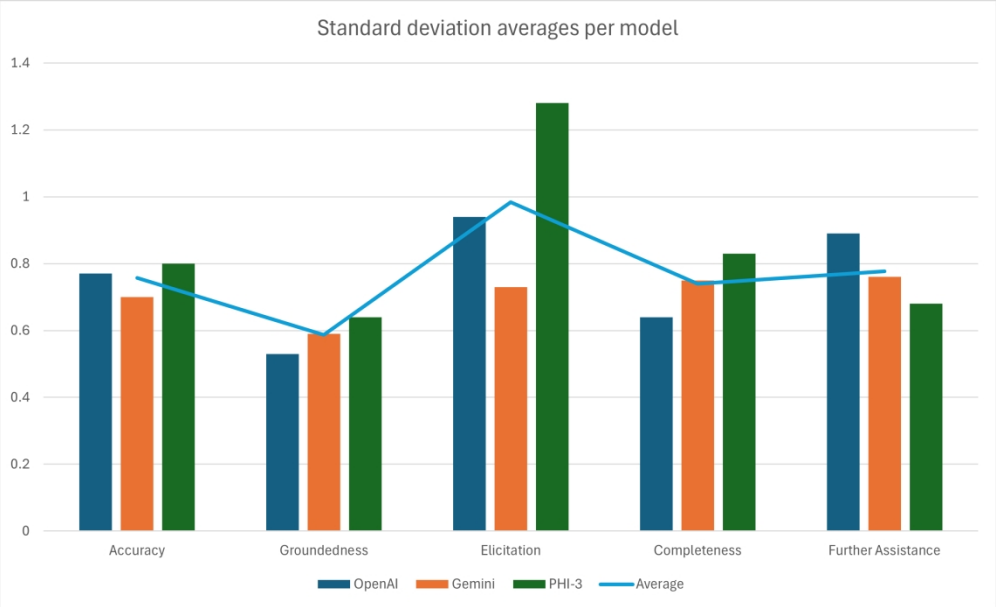

Figure 3. Standard deviation averages per model. Source: Authors' own work

532x323mm (130 x 130 DPI)

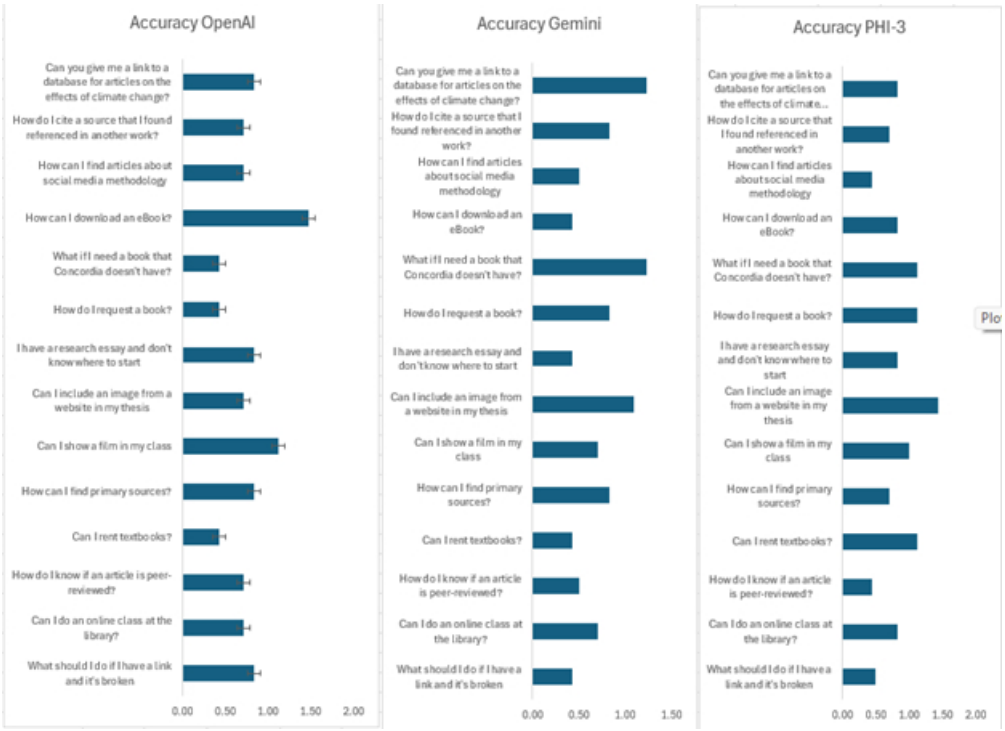

Figure 4. Average scores per question for each model in the accuracy dimension. Source: Authors' own work

417x302mm (38 x 38 DPI)

Figure 5. Average scores per question for each model in the groundedness dimension. Source: Authors' own work

417x273mm (38 x 38 DPI)

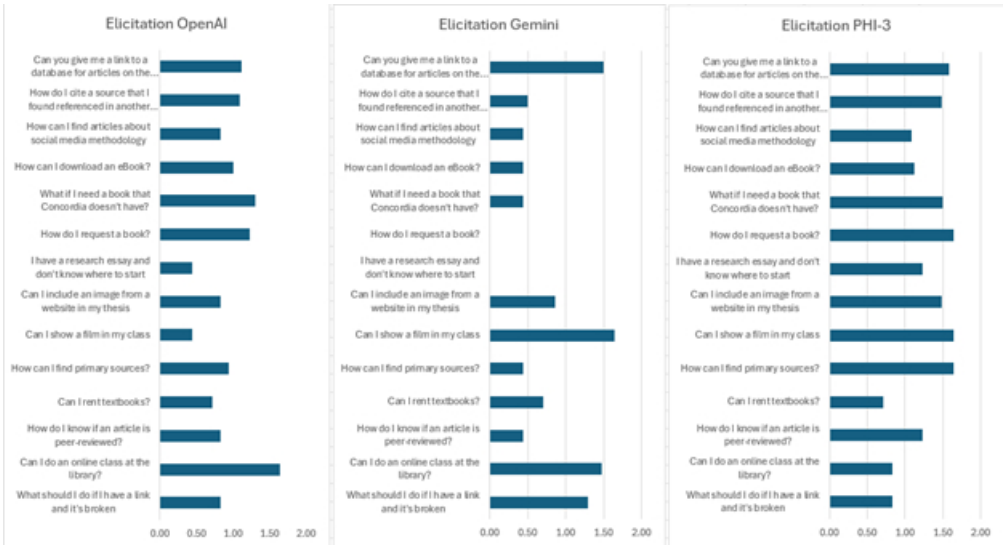

Figure 6. Average scores per question for each model in the elicitation dimension. Source: Authors' own work

417x225mm (38 x 38 DPI)

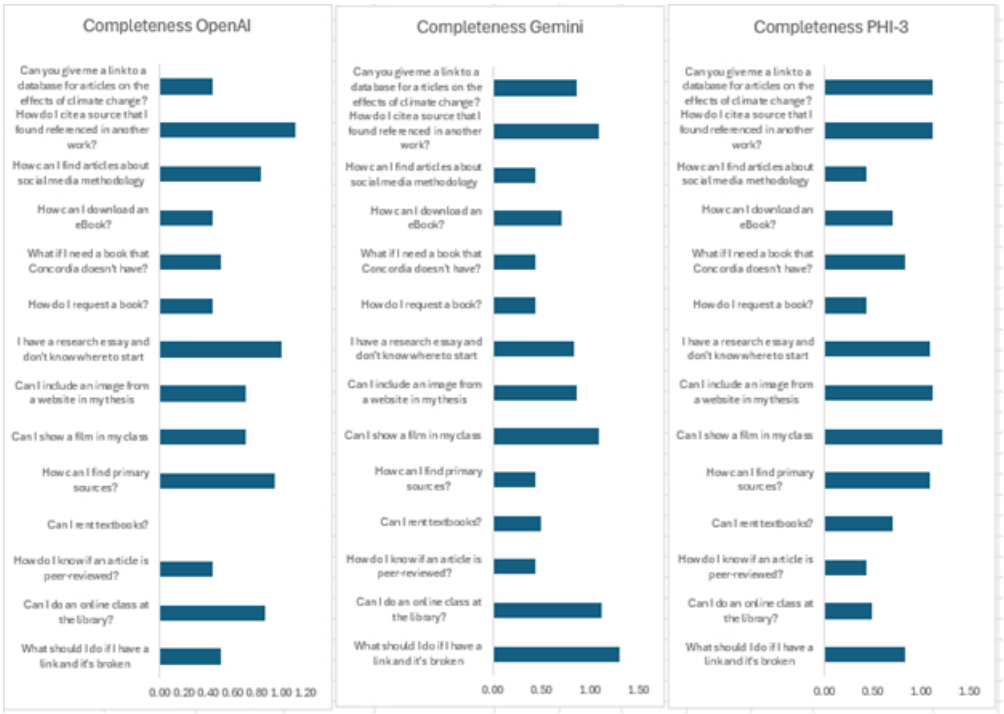

Figure 7. Average scores per question for each model in the completeness dimension. Source: Authors' own work

417x296mm (38 x 38 DPI)

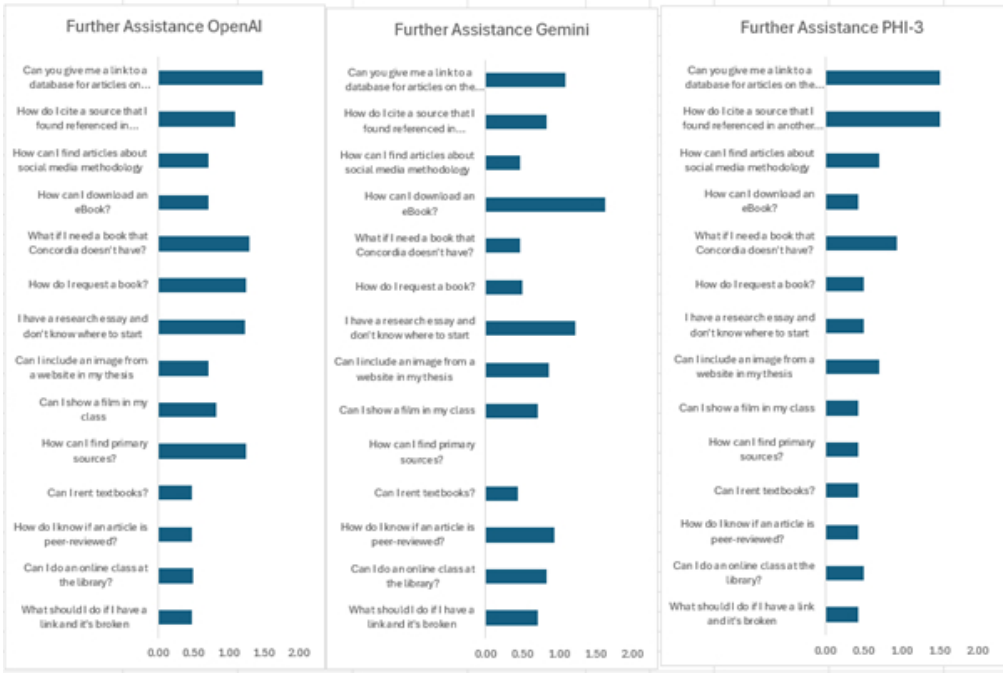

Figure 8. Average scores per question for each model in the further assistance dimension. Source: Authors' own work

399x269mm (38 x 38 DPI)