

6-14-2023

Aisha: A Custom AI Library Chatbot Using the ChatGPT API

Yrjo Lappalainen

Zayed University, yrjo.lappalainen@zu.ac.ae

Nikesh Narayanan

Zayed University, nikesh.narayanan@zu.ac.ae

Follow this and additional works at: <https://zuscholars.zu.ac.ae/works>



Part of the [Computer Sciences Commons](#), and the [Library and Information Science Commons](#)

Recommended Citation

Lappalainen, Yrjo and Narayanan, Nikesh, "Aisha: A Custom AI Library Chatbot Using the ChatGPT API" (2023). *All Works*. 5874.

<https://zuscholars.zu.ac.ae/works/5874>

This Article is brought to you for free and open access by ZU Scholars. It has been accepted for inclusion in All Works by an authorized administrator of ZU Scholars. For more information, please contact scholars@zu.ac.ae.

Aisha: A Custom AI Library Chatbot Using the ChatGPT API

Document version: This is an Accepted Author Manuscript of an article published by Taylor & Francis Group in the Journal of Web Librarianship on 14/06/2023, available online:

<https://doi.org/10.1080/19322909.2023.2221477>

Keywords: chatbots, ChatGPT, OpenAI, GPT-3, GPT-3.5, generative pre-trained transformer, academic libraries, artificial intelligence, AI

License: This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-commercial International Licence 4.0 (CC BY-NC 4.0), which permits all non-commercial use, distribution and reproduction in any medium, provided the original work is properly cited. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc/4.0/>.

Aisha: A custom AI library chatbot using the ChatGPT API

Yrjo Lappalainen^{a*} and Nikesh Narayanan^a

^a *Library and Learning Commons, Zayed University, Dubai, United Arab Emirates*

* Corresponding author: Yrjo Lappalainen, yrjo.lappalainen@zu.ac.ae, Zayed University, Library and Learning Commons, P.O. Box 19282, Dubai, United Arab Emirates

Aisha: A custom AI library chatbot using the ChatGPT API

This article focuses on the development of a custom chatbot for Zayed University Library (United Arab Emirates) using Python and the ChatGPT API. The chatbot, named Aisha, was designed to provide quick and efficient reference and support services to students and faculty outside the library's regular operating hours. The article also discusses the benefits of chatbots in academic libraries, and reviews the early literature on ChatGPT's applicability in this field. The article describes the development process, perceived capabilities and limitations of the bot, and plans for further development. This project represents the first fully reported attempt to explore the potential of a ChatGPT-based bot in academic libraries, and provides insights into the future of AI-based chatbot technology in this context.

Keywords: chatbots, ChatGPT, OpenAI, GPT-3, GPT-3.5, generative pre-trained transformer, academic libraries, artificial intelligence, AI

Introduction

Zayed University Library (United Arab Emirates) plays a crucial role in providing access to various resources and information to support its students and faculty's research and academic needs. The library has been providing an online chat service managed by Reference librarians to its users during the working hours of the library. However, extending the chat service when users require assistance outside of regular operating hours is difficult. Chatbots offer a promising solution to address this issue. A chatbot is a computer program simulating communication with human users, typically using a message interface. Chatbots are designed to comprehend user inquiries and provide responses that resemble a human conversation. They can be utilized for many things, such as work automation, information retrieval, and customer service. By leveraging chatbots, libraries can provide quick and easy access to resources, answer research-related questions, and offer Reference help to students and faculty 24/7. However, until recently, chatbots have been limited in their ability to understand and respond to user queries accurately. Traditional chatbot solutions have relied primarily on predetermined rules, such as pattern matching and keyword-based question answering, while lacking any advanced logical reasoning capabilities.

In November 2022, the AI landscape changed significantly when OpenAI introduced ChatGPT, a new type of chatbot that is built upon the highly advanced GPT-3.5 and GPT-4 (Generative Pre-trained Transformer) large language models (LLM) (OpenAI, 2022). Although the GPT-3 API had been in private beta testing since 2020, it was the introduction of ChatGPT that launched a huge public interest in the technology. According to a UBS analyst note, ChatGPT was estimated to have reached 100 million monthly active users in January 2023, just two months after its launch (Hu, 2023). Alongside the launch of ChatGPT, the OpenAI API (<https://platform.openai.com>) was

made available to all developers worldwide, leading to a surge in new projects that utilize the technology. Following the launch of ChatGPT and the OpenAI API, Zayed University Library started a project to build a custom chatbot using Python and the OpenAI API. The bot was subsequently named Aisha, meaning “alive” or “she who lives” in Arabic.

Objectives and scope

This article describes the development process, perceived capabilities and current limitations of the bot, and plans for further development. We also present a brief history of chatbots and review the use of chatbots in the context of academic libraries. Additionally, we review the early literature regarding ChatGPT's applicability in this field. At the time of writing, no custom chatbot utilizing the OpenAI API has been reported in the field of academic libraries. Therefore, this project represents the first fully reported attempt to explore its potential in this area.

Literature review

Alan Turing developed the Turing Test (originally called the “imitation game”) in the 1950s to assess the intelligence of computer programs, and Mauldin (1994) coined the term “chatbot” to characterize systems that can simulate human interaction and attempt to pass the Turing test. Midway through the 1960s, the MIT Artificial Intelligence Laboratory created ELIZA, the first chatbot capable of locating keywords in a given input sentence and matching those keywords against predefined rules to produce appropriate responses (Weizenbaum, 1966). Following ELIZA, the development of increasingly intelligent chatbots advanced, most notably with the creation of PARRY, developed by Kenneth Colby, a psychiatrist, in the early 1970s to simulate a paranoid patient's conversational style for use in therapy and research (Deshpande et al., 2017). When users interact with keyword-based chatbots, the system identifies keywords in their queries and

matches them to pre-programmed responses. While keyword-based chatbots can handle simple queries, they may struggle with complex, nuanced, or context-dependent questions due to their limited scope and less-flexible nature.

During the early 1980s, the creation of the Artificial Linguistic Internet Computer Entity (ALICE) marked a significant milestone in the development of Artificial Intelligence Markup Language (AIML) and became a cornerstone of many chatbot platforms and services in sophisticated chatbot projects (Wallace, 2009). With the development of new technologies, chatbot capabilities grew to include Artificial Intelligence (AI), which has improved machine learning, data analytics, and natural language processing (NLP) skills. The sophistication and effectiveness of chatbots have significantly increased due to this evolution. AI chatbots can discern context, semantics, and language nuances, enabling them to manage more complex and varied queries and continuously learn and improve their performance based on user interactions, resulting in a more human-like and engaging conversational experience (Hussain et al., 2019). Chatbot technology has grown significantly over the years, and several AI-based models have been developed to provide more natural and engaging user interactions. Notable chatbots, such as JABBERWACKY (now known as Cleverbot; 1988), Watson (2006), ALEXA (2015), Cortana (2015), and Tay (2016), have been developed to offer either text-to-speech or speech-to-speech interactions, utilizing machine learning techniques and NLP. These chatbots have been developed for various purposes, ranging from virtual assistants to gaming and entertainment (Ashfaq, 2022).

AI chatbots in libraries

The evolution of AI-based chatbots in libraries has seen significant advancements in the last decade, with a growing number of libraries around the globe embracing AI

technology to augment their services and provide support to users. In 2010, Kornelia, the inaugural public library chatbot, was introduced in Bern, Switzerland, marking a pioneering step in the field (McNeal & Newyear, 2013). European libraries played a vital role in the nascent stages of chatbot development. The Stella experiment, an early chatbot, was implemented at Hamburg University, representing Europe's first academic library implementation (Allison, 2012). This innovation was closely followed by the deployment of Chatbot Charlie at the Delft University of Technology in the Netherlands (Ehrenpreis & DeLooper, 2022). In February 2011, the University of Nebraska-Lincoln Libraries in the United States launched the Pixel project, an AI-based chatbot constructed using PHP code and an SQL server for its database to deliver prompt answers to questions concerning library services and resources (Allison, 2012; McNeal & Newyear, 2013). Mentor Public Library (MPL) and Akron-Summit County Public Library (ASCPL) in the United States were among the early adopters of chatbots, both operational by 2012 (Allison, 2012). In Australia, the University of Technology Sydney developed a chatbot prototype (Mckie & Narayan, 2019). In 2013, the University of California, Irvine (UCI) initiated the development of its chatbot, ANTsvers, constructed on an open-source platform, envisioned as a point-of-need reference tool that would complement existing online reference services without necessitating live staffing (Kane, 2019). Recent chatbot implementations encompass San Jose State University's Kingbot and the University of Oklahoma's Bizzy, both introduced in 2020. Kingbot was developed using Kommunicate, a proprietary software that leverages Google's Dialogflow tool (Rodriguez & Mune, 2021). The University of Oklahoma's Bizzy chatbot employs Ivy machine learning software to answer routine questions (University of Oklahoma, 2020).

Conversational AI with large language and generative AI models

The last two years have marked a significant breakthrough in generative AI, with the introduction of ChatGPT and many other LLM such as [LaMDA](https://blog.google/technology/ai/lamda) (<https://blog.google/technology/ai/lamda>), [AlexaTM](#) (Soltan et al., 2022), [Chinchilla](#) (Hoffmann et al., 2022), [PaLM](#) (Chowdhery et al., 2022), [PaLM 2](#) (<https://ai.google/discover/palm2>), [Falcon](#) (Technology Innovation Institute, 2023), [BloombergGPT](#) (Bloomberg, 2023), [PanGu-Sigma](#) (Ren et al., 2023), [GPT-NeoX](#) (Black et al., 2022), [LLaMA](#) (Meta AI, 2023), [Alpaca](#) (Taori et al., 2023), [Cerebras-GPT](#) (Dey et al., 2023), [GPT-J](#) (<https://huggingface.co/EleutherAI/gpt-j-6b>), [Vicuna](#) (<https://vicuna.lmsys.org>), [Koala](#) (Geng et al., 2023) and [StableLM](#) (<https://github.com/Stability-AI/StableLM>). (For an up-to-date list of over 100 LLM, see <https://lifearchitect.ai/models-table>). Simultaneously, there has been another major development in the field of AI image generation, with ground-breaking large-scale text-to-image models such as [DALL-E 2](#) (<https://openai.com/product/dall-e-2>), [Midjourney](#) (<https://www.midjourney.com>) and [Stable Diffusion](#) (<https://github.com/CompVis/stable-diffusion>) released in the past two years.

LLM comprehend and produce natural language responses to various queries using deep learning approaches like transformers. Generative models can produce completely original content that can be applied in various ways. A generative chatbot, for instance, can be taught to generate news articles, poems, and even movie or television screenplays. These models are more accurate and efficient than traditional rule-based or retrieval-based chatbots, providing more personalized and engaging interactions with users. Google Bard is an experimental conversational AI tool that originally used the Language Model for Dialogue Applications (LaMDA) to generate responses to user inputs. In May 2023, Google announced that Bard had moved from LaMDA to PaLM2,

a more advanced language model. Bard initially had a limited availability through a waitlist, but in May 2023 it was made publicly available in 180 countries and territories. (Hsiao, 2023). By utilizing data from the internet, Bard can offer up-to-date and high-quality responses to user queries (Pichai, 2023).

ChatGPT, developed by OpenAI, is a conversational AI model that has gained widespread popularity for its ability to engage in natural language conversations with humans. It is based on the GPT-3.5 language model and has been trained on a large corpus of text data to understand the nuances of language and generate contextually relevant responses in real time. ChatGPT can answer a wide range of questions and handle complex and context-dependent queries. One of the unique features of ChatGPT is that it can answer follow-up questions, challenge incorrect premises, and reject inappropriate requests (OpenAI, 2022). The model is based on a transformer architecture and is trained using Reinforcement Learning from Human Feedback (Gozalo-Brizuela & Garrido-Merchan, 2023). Developers can also create custom GPT-based chatbots and virtual assistants that can interact with users in a more natural and intuitive way, providing them with personalized responses and assistance. A custom chatbot developed using the ChatGPT API can be trained to understand and respond to user queries related to the custom environment. In March 2023, OpenAI announced that it is rolling out a web browsing feature and other ChatGPT plugins that can extend the language model's functionality by accessing external data sources and services (OpenAI, 2023b). However, these plugins are still in an early testing phase and are only available to ChatGPT Plus subscribers at the time of writing (OpenAI, 2023c).

ChatGPT in the context of academic libraries

Since its public launch in November 2022, ChatGPT has received significant attention with numerous articles and preprints already published about its potential impact on various fields. The impact of ChatGPT on academia and education, particularly in regards to academic integrity, has been an area of major interest (see e.g. King & ChatGPT, 2023; Cotton et al., 2023; Lim et al., 2023). Some articles have also been published about the role of ChatGPT in the context of academic libraries. Lund and Wang (2023) examined the potential impact of ChatGPT on academia and libraries by interviewing ChatGPT itself. Based on its responses, they identified that ChatGPT has the capability to improve several library services such as search and discovery, reference and information services, cataloging and metadata generation and content creation. However, they also emphasized that the technology needs to be used responsibly and that ethical considerations such as privacy issues and bias need to be taken into account.

Cox and Tzoc (2023) discussed the potential implications of ChatGPT for academic libraries from a wide perspective. They suggested that ChatGPT could complement or even replace existing search methods. They also commented that ChatGPT can be integrated into library discovery tools, which may lead into an “arms race” between providers as they contend to add this functionality into their products. The authors also highlighted the role of ChatGPT in research, where it could be used for brainstorming and finding relevant literature. The authors suggested that as the technology develops, AI tools could function as intelligent research assistants that conduct virtual experiments, analyze data, do copywriting, edit texts, and generate citations. In terms of library reference services, the authors discussed the increasing use of AI chatbots to answer basic reference questions, which can free up librarian time for more complex research queries or tasks. The authors also noted that AI tools will make

information literacy and digital literacy more important than ever and that librarians need to teach critical thinking skills to validate facts and evaluate the quality of the answers provided by ChatGPT. They concluded that the introduction of ChatGPT seems similar to other innovative developments such as the introduction of calculators, cell phones, the World Wide Web, and Wikipedia, and that libraries should evaluate these new tools and develop services to support their use.

Chen (2023) conducted a simple test where ChatGPT was asked questions about library services, and its responses were compared with those provided by conventional library chatbots. As a result, ChatGPT was able to suggest specific databases, whereas the conventional chatbots did not understand the question or only suggested visiting the library's A-Z database page or general instructions. The author also noted that a customized ChatGPT might better answer local questions, such as library hours and local resources. The author also suggested that past lessons from the adoption of Google and Web 2.0 can guide how to approach ChatGPT. According to the author, the library community failed to fully recognize and utilize the potential of Google when it was first introduced and also failed to anticipate the impact of social media in spreading misinformation. The author concluded that the library community should avoid underestimating or underutilizing ChatGPT's potential to enhance library services but also acknowledge and address its potential weaknesses and pitfalls, such as plagiarism and the possibility of erroneous output due to poor data quality.

Panda and Kaur (2023) examined the viability of ChatGPT as an alternative to traditional chatbot systems in library and information centers. According to the authors, ChatGPT represents a significant advancement over traditional chatbots because it enables more flexible and natural language conversation. Traditional chatbots rely on predefined rules and responses to generate answers to user queries, which can limit their

flexibility, scalability, and natural language capabilities. In contrast, ChatGPT is trained on a large corpus of data and specifically designed to generate natural language responses, making it more flexible and adaptable to various user needs. Traditional chatbots also often struggle with unexpected and non-standard queries, while ChatGPT can use contextual clues to generate responses even if the question is not phrased in a typical way. Furthermore, ChatGPT can learn from new data, while traditional chatbots require regular maintenance to ensure that the questions and answers stay up to date. Finally, the cost of developing and maintaining traditional chatbots can be significant, while ChatGPT can be trained on existing data and fine-tuned for specific tasks, potentially reducing costs.

Adetayo (2023) explored the potential of AI chatbots, particularly ChatGPT, in academic libraries. According to the author, chatbots can assist library patrons in accessing materials and completing tasks without human assistance, freeing up librarians' time for more in-depth assistance. ChatGPT has unique features, such as generating diverse and lifelike responses, and recognizing user intent. It can also be used for language translation purposes. However, there are potential risks and challenges associated with ChatGPT's use in academic libraries, including the risk of job loss and the possibility of misuse. Additionally, ChatGPT may produce inaccurate query responses during reference transactions and lacks the ability to comprehend reference queries like a human librarian. The author concludes that while ChatGPT has the potential to benefit academic libraries, it is essential to carefully assess and address the possible risks and challenges associated with its use. Libraries must develop clear standards and guidelines, monitor their performance regularly, and use them ethically and effectively to provide the best possible user experience.

Developing the bot

Interacting with the OpenAI API

The [OpenAI API](#) is an application programming interface that allows developers to use the GPT large language models in their applications. The API provides a way to interact with the GPT models and generates natural language responses to queries. OpenAI's chat-optimized models, gpt-3.5-turbo and gpt-4, can discuss almost any topic and perform various tasks without any additional training. The API's [pricing model](#) (<https://openai.com/pricing>) is based on the number of used "tokens", which can be considered as pieces of words. The API breaks all inputs into tokens, which are then converted into numerical representations (vectors) that can be processed by the language model. At the time of writing, the usage cost of the gpt-3.5-turbo model (the same as the default publicly-available ChatGPT model) is USD \$0.002 per 1,000 tokens. A single question and answer typically require fewer than 1,000 tokens, which means that the model can answer at least 1,000 questions for approximately USD \$2. This pricing model offers a very low entry barrier and makes the API accessible for a wide range of users.

For a custom chatbot, incorporating domain-specific data is crucial. Currently, there are two ways to use custom data with the GPT models:

- (1) providing context in the prompt and instructing the model to base its response on the context; or
- (2) fine-tuning the model on a custom dataset.

Modifying the prompt (also known as the emerging art of "prompt engineering") can be used to limit the responses into the specified context, making it a good starting-point for a tailored chatbot. Fine-tuning is another approach that updates the model's weights

towards the custom dataset. According to OpenAI's documentation (2023a), the benefit of fine-tuning over prompt engineering is that once the training is complete, the model can produce higher-quality results without the need to provide the context every time in the prompt. In this approach, the responses are not limited to any context, which potentially makes fine-tuning a more suitable approach for general-purpose applications. Fine-tuning can also enable lower-latency requests (OpenAI, 2023a). However, fine-tuning is priced separately and the price depends on the selected base model.

OpenAI maintains a [cookbook](https://github.com/openai/openai-cookbook) (<https://github.com/openai/openai-cookbook>) with example scripts for accomplishing common tasks using the OpenAI API. One of the examples, [web-crawl-q-and-a](https://github.com/openai/openai-cookbook/tree/main/apps/web-crawl-q-and-a) (<https://github.com/openai/openai-cookbook/tree/main/apps/web-crawl-q-and-a>), was an important starting-point for our project. The code shows how to use web crawling and OpenAI's API to answer questions based on the crawled data. Crawling is a technique that extracts data from a website and then stores it in a structured format for further use. The example code first extracts text from a webpage, saves it into a csv file, and then generates embeddings using the OpenAI API. Embedding is the process of converting text strings into vectors (lists of floating-point numbers). These vectors can then be used to calculate the distance between items to measure their relatedness. If the distance is small, they are closely related, while large distances suggest low relatedness. Embeddings are widely used in natural language processing and the OpenAI API has a dedicated endpoint for generating embeddings. (OpenAI, 2023a). Using embeddings is a critical part of the process because it allows the most relevant context to be identified and added into the prompt. Generating embeddings with the OpenAI API is also priced separately, and the price is USD \$0.0004 per 1,000 tokens at the time of writing. Embeddings can also be created using free alternatives such as the [SentenceTransformers](https://www.sbert.net) framework (<https://www.sbert.net>).

Collecting and preparing the data

Library websites and guides often contain a wealth of information about the library's services, so these served as a natural starting-point for our data collection. Zayed University Library maintains over 100 guides and a large number of webpages, so collecting this data manually would be a tedious and time-consuming project. To speed things up, we created a custom script to crawl the guides automatically. We also used ChatGPT as a coding assistant during the project and found it extremely useful due to its capability to generate, debug, and comment code. This speeded up the development process significantly.

The scraping script collects all the links from the list of guides and then follows every link under the domain <https://zu.libguides.com>. The list of LibGuides contains links that are created dynamically by JavaScript, so the Selenium library was used to capture all the data. Selenium launches a browser window and finds all the links on the page. The script then extracts the headings and contents from every page, and stores them in a dataframe. Zayed University Library's guides have a fairly consistent structure so we were able to collect all the content. The script also collected in-text links and embedded content (e.g., videos and other pages).

Web crawling can be challenging due to the variability of data and lack of standardization across webpages. Information can be presented in various styles and formats, making it more difficult to extract automatically. Initially, we also planned to crawl the library's main website, but since it has a less-structured format than our LibGuides, we decided to collect the information manually instead. This included basic information such as library hours, available services, and contact details. In addition, we used ChatGPT to generate a list of 100 typical questions and answers regarding academic

libraries. We then revised the list and updated it with Zayed University-specific information. Furthermore, we used a set of previously asked questions and answers from LibAnswers. The last step was to review all data and to remove all duplicates and inconsistencies manually. The final dataset consisted of a total number of 2,200 rows.

The Q&A script provided by OpenAI is a simple solution that creates the embeddings, saves them into a csv file, loads them into a dataframe, and then calculates the distances between the question and the custom dataset. This is fine for smaller-scale implementations or testing purposes but it can also lead to slow performance when dealing with larger datasets. This is why OpenAI recommends using a vector database for searching over many vectors quickly (OpenAI, 2023a). Based on this recommendation and extremely helpful instructions published by Kim (2023), Yang (2023), and Chase (2023), we decided to set up a vector database using [Chroma](https://www.trychroma.com) (<https://www.trychroma.com>), a toolkit designed for building AI applications with embeddings. It uses an in-process (serverless) DuckDB database, allowing the storage and querying of embeddings and their metadata without having to set up a dedicated server environment. At the same time, we started using [LangChain](https://python.langchain.com) (<https://python.langchain.com>), a framework for interfacing and working with various LLM. It facilitates data ingestion, prompt management, embedding creation, and output parsing. Above all, it can also be used to create chains, i.e., sequences of multiple LLM calls, and advanced agents that use LLMs to interact with other systems and tools. Chroma has a LangChain integration which makes it possible to create a chain that queries the vector database first, before passing the data to the OpenAI API. LangChain can also be used with other LLM, which gives us more possibilities for further development.

Instructing the bot

The Q&A script in OpenAI's cookbook is designed to answer a single question based on the context, so it does not save the conversation history. Since our main goal was to build a conversational bot for customer service purposes, it was essential that the bot can keep up the conversation and ask follow-up questions. To achieve this, we created a conversational memory that adds one previous question and response as part of the prompt. Later, we also added a setting where the number of previous conversations used in the prompt can be selected.

Our source data is in English, and initially the bot only responded in English. Since ChatGPT can also do translations and have conversations in different languages, we wanted to take advantage of this feature in our bot as well. We also wanted the bot to sound natural and ask follow-up questions. The GPT models can be instructed by using natural language, so the bot was given the following instructions:

- *You are Aisha, a friendly and helpful library assistant at Zayed University Library. Provide clickable links for any URLs. Answer the questions from the perspective of Zayed University Library. Translate responses to the language of the question. Ask follow up questions. If you don't know the answer, say that you don't know. Ask for clarifications if you don't understand the question. Provide direct links to the mentioned library databases and services. Remind users that all databases can also be accessed from <https://zu.libguides.com/az.php>. Don't respond if the question is not related to Zayed University Library or its resources and services. If you cannot answer the question, recommend contacting a relevant subject librarian.*

Finding the right balance between a ChatGPT-style general chatbot and a library-specific bot has been a challenge throughout the project. We wanted to limit the bot to answer questions about our library, but also give it some room for improvisation. OpenAI's example Q&A script limits the response to the given context using the following wording: *Answer the question based on the context below, and if the question can't be answered based on the context, say "I don't know"*. However, after initial testing, we found this approach too restrictive because it often resulted in the bot providing "I don't know" responses. The bot also kept mentioning the word "context" in its responses. To fix this, we changed the instruction to: *Answer the question using only the information below*". In addition, we explicitly instructed the bot to *"act as a library assistant and remain in this role throughout the conversation"*. These instructions, together with the conversational memory, significantly improved the quality of the output and overall user experience (see 1). This resulted in conversations that were more natural, and the bot was able to remember the previous conversations and ask follow-up questions (see our chats in Figure 1 and the Appendix). After including the instructions to translate responses, the bot was also able to have conversations in different languages based on the English source material.

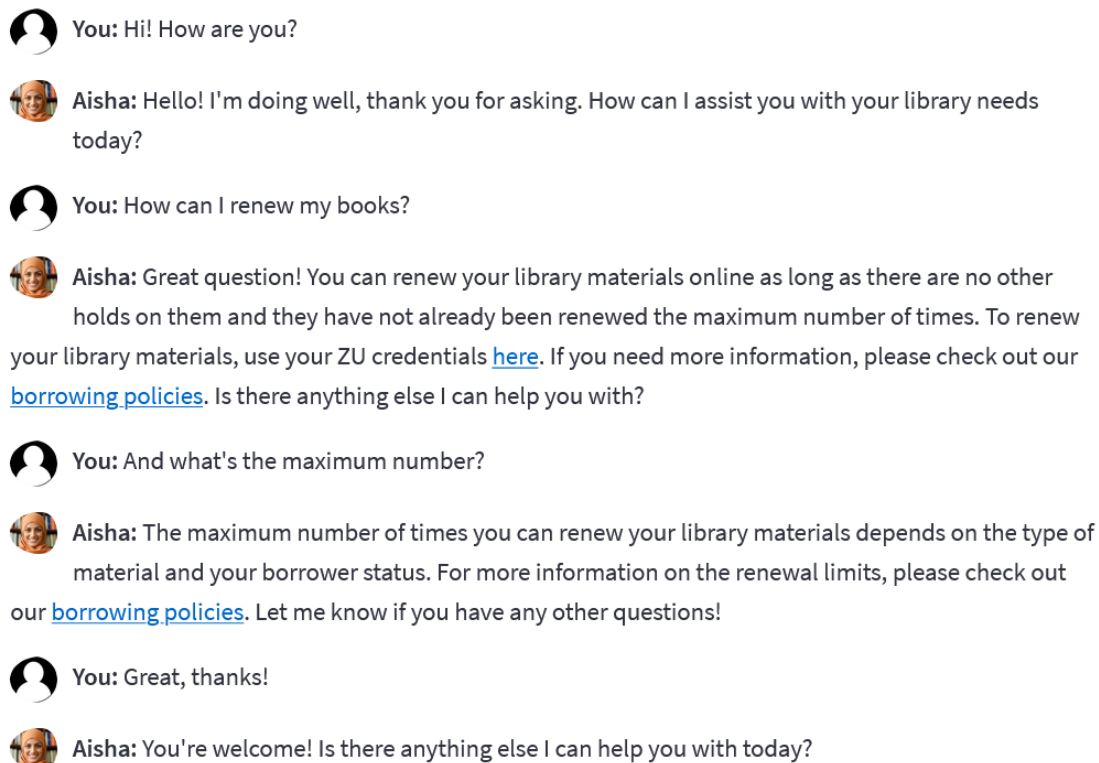


Figure 1. A conversation with Aisha

The GPT models are susceptible to a phenomenon known as “hallucination”, where the model makes up content when it encounters gaps in its knowledge (Alkaissi & McFarlane, 2023). Unfortunately, this has been an issue in our project as well, with the bot occasionally promoting non-existent links and library services. We tried to reduce this by providing the following additional instructions after the context: *“When providing links, prefer those that start with <https://zu.libguides.com> or <https://zulib.idm.oclc.org>. Do not invent non-existent links or services that are not listed in the context”*. The OpenAI API also has a “temperature” setting that controls the randomness in the generated text. Higher values will make the output more random and lower values more predictable, closer to the training data. Another approach was to provide one example question and the correct answer to the bot before the actual prompt (also known as “one-shot learning”). Revising the instructions, setting the temperature to 0 and incorporating one-shot learning slightly reduced the frequency of the hallucinations, but Aisha continues to occasionally generate

non-existent links and other minor hallucinations (e.g., personal names that do not appear anywhere in the embedded source data).

Creating the interface

There are many options for deploying Python apps online. Based on the instructions published by Biswas (2023), we chose [Streamlit \(https://streamlit.io\)](https://streamlit.io), which is an open-source Python library for creating interactive web applications. In Streamlit, a chatbot interface can be created with just a couple of lines of code using the [streamlit-chat](https://pypi.org/project/streamlit-chat) component (<https://pypi.org/project/streamlit-chat>). However, we decided to print the outputs using Streamlit's "st.markdown" function instead since this gave us more options for customizing the look and feel of the chat. We added a custom avatar generated by another AI model, Stable Diffusion. After initial testing, we also added a debug mode that prints the prompt history, number of tokens, and costs of usage. Based on Streamlit's documentation, we also created a Google Drive integration to record all questions and answers in a spreadsheet that is only accessible to a few selected developers (Streamlit, 2023). This allows us to monitor the bot's performance and to modify the settings based on the outputs. While Google Drive is convenient for logging the conversations, we only intend to use it as a temporary solution during testing as it may not meet the necessary data privacy standards in a production environment.

Summary

To summarize, the following steps were required to set up and customize the bot:

1. Crawling content from LibGuides and website.
2. Adding more content manually.
3. Checking and cleaning the dataset.
4. Creating and storing the embeddings (using Chroma).

5. Creating a script that identifies the correct context and queries the OpenAI API (using LangChain).
6. Instructing the bot.
7. Creating the chat interface.
8. Deploying the application (using Streamlit).

Each chat query was then processed using the following steps (Figure 2):

1. Creating an embedding for the question.
2. Identifying the nearest context from the vector database and adding it into the prompt.
3. Querying the OpenAI API with both the question and the context.

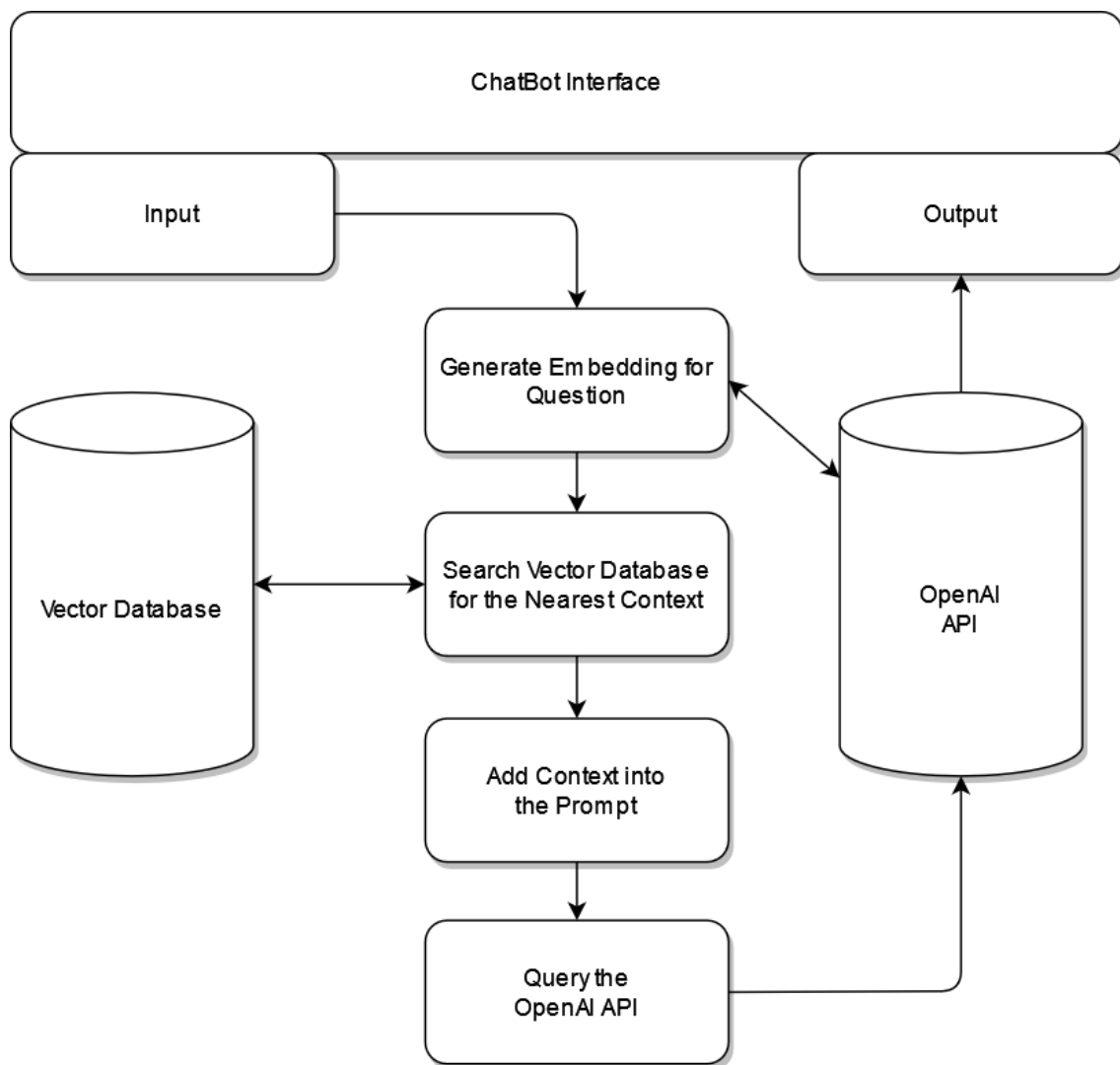


Figure 2. Processing chat queries

Discussion and future development

Perceived capabilities

Zayed University Library is dedicated to offering exceptional information and support services to its students and faculty members. As digital technologies become more prevalent, libraries must seek out innovative solutions to improve their services. Our initial experience has shown that chatbots hold great potential for providing personalized, accessible, and cost-effective support to library users. Based on initial testing, the chatbot can provide very realistic and human-like responses, keep up the conversation by asking follow-up questions, and respond in different languages based on English source data. Since the bot is using the gpt3.5-turbo model, it can improvise its responses, write poems, tell jokes, etc., just like its “bigger cousin” ChatGPT, although the number of tokens and the context window size are smaller in our solution.

The chatbot has tremendous potential to speed up reference services and make them more accessible, providing library users with round-the-clock assistance. The chatbot could also be taught to recognize specific users and tailor its responses to them based on their previous experiences with the system by utilizing machine learning methods. This could produce a smoother and more exciting user experience, increasing user satisfaction and library service use. Additionally, by including features like speech-to-text or text-to-speech conversion or alternative methods to engage with the system, the bot could be made to deliver accessible responses for people with impairments. This can ensure that everyone who uses the library can get assistance and knowledge from it, regardless of their abilities. The bot could also be integrated into other library systems, which could increase the visibility and accessibility of all library services.

Current limitations

Although the bot has performed well in initial testing, the implementation still has certain limitations. First of all, the OpenAI API has token restrictions based on the selected model. OpenAI's gpt-3.5-turbo model currently has a limit of 4,096 tokens, including both the input and the output. This is not a major issue when it comes to chats, because the questions are typically brief and require a total of 500-1,500 tokens per question, including the question, context and the output. However, the full chat history cannot be preserved for long since the token limit would be reached quickly.

Another issue is that the bot could be “tricked” by providing additional data in the prompt (also known as “prompt injection”). This can result in unreliable or questionable responses and the bot could start performing tasks beyond its intended scope. Privacy issues may also arise if personal data are passed to the OpenAI API. However, these issues could be prevented by adding measures for detecting and filtering unwanted prompts before even passing them to the OpenAI API.

One major downside of the implementation is that the bot currently has no real-time access to information online, for example the library website. However, this could be solved by loading and embedding specific information automatically (such as opening hours and library events) on a regular basis. We also expect that the ChatGPT plugins, which are currently under development, will be incorporated into the OpenAI APIs in the coming months, making it easier to retrieve real-time information from websites and other sources.

Librarians often receive complex Reference questions about specific papers and topics. Since the bot currently has no real-time access to online information, it cannot answer specific questions about individual papers. It is possible to build functionality that

allows users to upload their own documents, embed them, and ask questions about their contents (e.g., [Dara, https://www.dara.chat](https://www.dara.chat)). However, copyright and privacy issues could arise if all data are processed on OpenAI's servers. One possible solution would be to embed the paper locally and then query the OpenAI API with the relevant question and context only, instead of passing the full text to the API. When it comes to interpreting a paper's content, a ChatGPT-based bot could easily outperform human librarians, because it can process an entire scientific paper in a matter of seconds. A ChatGPT-based bot could also provide general guidance in research methods and citations, especially if such materials are available in the source data. However, one of the most important aspects of human librarians is the ability to provide search assistance and recommend specific library resources, which requires a thorough and up-to-date understanding of the subject area and the library's collections. Achieving human-level recommendations with a chatbot would require at least full access to the library discovery service and possibly a memory function that keeps track of recommended resources, latest acquisitions and perhaps even the latest trends in different fields. This is an interesting area that calls for further research.

So far, we have only tested the bot informally among library staff - about 15 people representing all library teams. The bot saves all questions and answers in an access-restricted Google spreadsheet, and we have used this data to refine the bot's instructions and source materials. After reviewing around 500 unique questions and answers, we identified three main issues: 1) the bot often generates non-existent links, as described earlier; 2) the bot may mistake a link from the source data as a (subscribed) library service, although it is only mentioned as an additional resource, and; 3) the bot cannot answer questions that require real-time data or access to a specific resource (for example "When is the next library workshop?" or "Can you recommend a good research

article about AI?"). Despite these issues, we were pleased to notice that there were very few factual errors, and most of them were due to outdated or erroneous source data. The generation of non-existent links remains a challenge at the time of writing, but the latter two issues can be corrected by revising the source materials and ingesting certain content (e.g., library event calendar) on a regular basis.

Future development

In the next phase of our project, we plan to move forward with more formal testing by engaging Zayed University students and faculty. We are excited to study how the bot performs in larger-scale testing and to hear feedback and development ideas from library users. As a follow-up study, we intend to do a more-formal analysis of the bot's outputs and compare it with a keyword-based chatbot solution.

During testing, we noticed that managing the embeddings can be challenging in the long run, and another interface is needed for creating and updating embedded contents. By developing an interface, a larger number of library staff will be able to manage the embedded contents. We are also planning to create a feedback mechanism in the bot that allows users to indicate their opinions about the bot's performance (for example, upvoting or downvoting the response). In case of downvoting, the user could be instructed to provide textual feedback or contact a liaison librarian for further questions.

The bot is currently instructed to provide contact details when it cannot answer a question. This could be developed further, for example by connecting the user to a live chat with a librarian automatically during library hours or by providing a form to ask further questions or report the issue. This feedback would be valuable for further development.

Another development idea is adding a cache to reduce the number of LLM calls and increase the bot's performance. A solution called [GPTCache \(https://gptcache.readthedocs.io\)](https://gptcache.readthedocs.io) has already been developed for this purpose. Another interesting direction to explore is the implementation of AI agents which have the potential to facilitate interactions with other library systems. With the use of agents, users could potentially carry out various tasks such as searching library databases, renewing loans, or making other requests, directly through the chatbot. This could greatly enhance the user experience and streamline the overall process of accessing library resources. Projects such as [BabyAGI \(https://github.com/oliveirabruno01/babyagi-asi\)](https://github.com/oliveirabruno01/babyagi-asi), [Auto-GPT \(https://github.com/Significant-Gravitas/Auto-GPT\)](https://github.com/Significant-Gravitas/Auto-GPT), and [AgentGPT \(https://github.com/reworkd/AgentGPT\)](https://github.com/reworkd/AgentGPT) are already available to help with the development of intelligent agents. One particularly interesting possibility would be to connect the chatbot to the library discovery service, allowing it to query and recommend specific library materials. We have also begun testing speech-to-text and text-to-speech capabilities to improve accessibility and to provide an alternative way to interact with the bot.

Assessment of the method

The field of generative AI and LLM has evolved rapidly in early 2023. The emergence of new tools and frameworks (e.g., LangChain and various vector databases) has facilitated this growth, but the fast pace of updates has also made it challenging to keep up with the latest developments. Documentation is still scarce and developers seem to rely heavily on experimenting and scattered instructions posted on discussion forums and other platforms. As a result, much of our development process has been driven by trial and error. Finding the right system messages and prompts for OpenAI's API has been particularly challenging due to the lack of clear instructions. Despite trying different

prompting options, we have not been able to completely eliminate the issue of hallucination, which can lead to inaccurate or confusing bot responses.

Remarkably, if we do not put a price on our own work, the implementation has not cost us anything yet. The OpenAI API provides free tokens worth \$18.00 USD and we have not used all of them at the time of writing. At current API prices, answering questions using the ChatGPT API (gpt3.5-turbo) costs about \$0.002 USD per question. As a result, a GPT-based chatbot could provide 24/7 support at a significantly lower cost than hiring additional staff or implementing other support systems. Nevertheless, we do not consider chatbots as a threat to library jobs. On the contrary, we believe that chatbots have the potential to free up staff time and resources, allowing library personnel to focus on more complex and specialized Reference queries, as well as other critical duties within the library. This shift in responsibilities can ultimately lead to a more productive and efficient library.

Using ChatGPT as a coding assistant during the project significantly helped the coding process. The fact that we can now use ChatGPT for writing and reviewing code and to give us development ideas shows that we have entered a fascinating new era in software development and human-computer interaction in general. We are expecting a wave of new GPT-based solutions in the upcoming months. However, as the field of generative AI continues to evolve, user-friendly tools, accessible documentation and standardized practices are required to help the creation of LLM-based applications.

Conclusion

In this article, we described the development of Aisha, a custom ChatGPT-powered chatbot at Zayed University Library. We also reviewed the history of chatbots and early literature on ChatGPT in the context of academic libraries. In conclusion, we believe that

chatbots based on ChatGPT and other LLM have great potential to transform library Reference services by offering personalized, accessible, and cost-effective support to users.

While chatbots are one potential use case for LLM, we also acknowledge that ChatGPT and other LLM have a lot more to offer for libraries and many other fields. In the context of academic libraries, some potential use cases include personalized search assistance, research analysis, supporting cataloging and metadata generation, analyzing customer feedback, and generating ideas and texts for library's outreach, marketing, instruction and support materials. LLM-based AI research assistants could be particularly helpful for students and researchers, since they can summarize academic texts in an instant and enable users to ask focused questions about the texts.

The last two years have marked a significant breakthrough in the field of generative AI. Several new LLM and other generative AI models have been introduced, resulting in hundreds, if not thousands, of new AI-based services, tools, frameworks and projects in a relatively short period of time. While there are still many technical, ethical, legal and other challenges to overcome, it is clear that a major transformative change is under way and there is no turning back. As academic librarians, we are excited to explore the ways in which AI can improve research, enable better access to information and advance scholarly communication, while also being mindful of the ethical implications and challenges that come with this new technology.

References

- Adetayo, A. J. (2023). Artificial intelligence chatbots in academic libraries: The rise of ChatGPT. *Library Hi Tech News*. <https://doi.org/10.1108/LHTN-01-2023-0007>
- Alkaissi, H., & McFarlane, S. I. (2023). Artificial hallucinations in ChatGPT: Implications in scientific writing. *Cureus*, 15(2). <https://doi.org/10.7759/cureus.35179>
- Allison, D. (2012). Chatbots in the library: Is it time? *Library Hi Tech*, 30(1), 95-107. <https://doi.org/10.1108/07378831211213238>
- Ashfaque, M. W. (2022). Analysis of different trends in chatbot designing and development: A review. *ECS Transactions*, 107(1), 7215-7227. <https://doi.org/10.1149/10701.7215ecst>
- Biswas, A. (2023, March 19). How to build a chatbot with ChatGPT API and a conversational memory in Python. *Medium*. <https://medium.com/@avra42/how-to-build-a-chatbot-with-chatgpt-api-and-a-conversational-memory-in-python-8d856cda4542>
- Black, S., Biderman, S., Hallahan, E., Anthony, Q., Gao, L., Golding, L., He, H., Leahy, C., McDonnell, K., Phang, J., Pieler, M., Prashanth, U. S., Purohit, S., Reynolds, L., Tow, J., Wang, B., & Weinbach, S. (2022). *GPT-NeoX-20B: An open-source autoregressive language model*. ArXiv:2204.06745v1. <https://arxiv.org/abs/2204.06745>
- Bloomberg. (2023, March 30). *Introducing BloombergGPT, Bloomberg's 50-billion parameter large language model, purpose-built from scratch for finance*. <https://www.bloomberg.com/company/press/bloomberggpt-50-billion-parameter-llm-tuned-finance/>
- Chase, H. (2023). *Chat over documents with chat history*. LangChain documentation. Retrieved April 6, 2023, from https://python.langchain.com/en/latest/modules/chains/index_examples/chat_vector_db.html
- Chen, X. (2023). ChatGPT and its possible impact on library reference services. *Internet Reference Services Quarterly*. <https://doi.org/10.1080/10875301.2023.2181262>
- Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., Schuh, P., Shi, K., Tsvyashchenko, S., Maynez, J., Rao, A., Barnes, P., Tay, Y., Shazeer, N., Prabhakaran, V., ... Fiedel, N. (2022). *PaLM: Scaling language modeling with pathways*. ArXiv:2204.02311v5. <https://arxiv.org/abs/2204.02311>
- Cotton, D. R., Cotton, P. A., & Shipway, J. R. (2023). Chatting and cheating: Ensuring academic integrity in the era of ChatGPT. *Innovations in Education and Teaching International*, 1-12. <https://doi.org/10.1080/14703297.2023.2190148>

- Cox, C., & Tzoc, E. (2023). ChatGPT: Implications for academic libraries. *College & Research Libraries News*, 84(3).
<https://crln.acrl.org/index.php/crlnews/article/view/25821>
- Deshpande, A., Shahane, A., Gadre, D., Deshpande, M., & Joshi, P. M. (2017). A survey of various chatbot implementation techniques. *International Journal of Computer Engineering and Applications*, 11(7). <https://www.ijcea.com/survey-various-chatbot-implementation-techniques>
- Dey, N., Gosal, G., Zhiming, Chen, Khachane, H., Marshall, W., Pathria, R., Tom, M., & Hestness, J. (2023). *Cerebras-GPT: Open compute-optimal language models trained on the Cerebras wafer-scale cluster*. ArXiv:2304.03208v1.
<https://arxiv.org/abs/2304.03208>
- 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>
- Geng, X., Gudibande, A., Liu, H., Wallace, E., Abbeel, P., Levine, S., & Song, D. (2023, April 3). Koala: A Dialogue Model for Academic Research. *The Berkeley Artificial Intelligence Research Blog*.
<http://bair.berkeley.edu/blog/2023/04/03/koala/>
- Gozalo-Brizuela, R., & Garrido-Merchan, E. C. (2023). *ChatGPT is not all you need. A state of the art review of large generative AI models*. ArXiv:2301.04655v1.
<https://arxiv.org/abs/2301.04655>
- Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Casas, D. de L., Hendricks, L. A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., Driessche, G. van den, Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., ... Sifre, L. (2022). *Training compute-optimal large language models*. ArXiv:2203.15556v1. <https://arxiv.org/abs/2203.15556>
- Hsiao, S. (2023, May 10). What's ahead for Bard: More global, more visual, more integrated. *Google Blog*. <https://blog.google/technology/ai/google-bard-updates-io-2023>
- Hu, K. (2023, February 2). ChatGPT sets record for fastest-growing user base - Analyst note. *Reuters*. <https://www.reuters.com/technology/chatgpt-sets-record-fastest-growing-user-base-analyst-note-2023-02-01>
- Hussain, S., Ameri Sianaki, O., & Ababneh, N. (2019). A survey on conversational agents/chatbots classification and design techniques. *Advances in Intelligent Systems and Computing*, 927, 946-956. https://doi.org/10.1007/978-3-030-15035-8_93
- Kane, D. (2019). Creating, managing and analyzing an academic library chatbot. *BiD*, 43. <https://doi.org/10.1344/BID2019.43.22>
- Kim, S. (2023, March 20). How to ensure OpenAI's GPT-3 provides an accurate answer using embedding and semantic search. *Dev Genius*. <https://blog.devgenius.io/how->

[to-ensure-openai-gpt-3-provides-an-accurate-answer-and-stays-on-topic-af5da300ba81](https://doi.org/10.1007/s12195-022-00754-8)

King, M. R. & ChatGPT. (2023). A conversation on artificial intelligence, chatbots, and plagiarism in higher education. *Cellular and Molecular Bioengineering*, 16(1), 1-2). <https://doi.org/10.1007/s12195-022-00754-8>

Lim, W. M., Gunasekara, A., Pallant, J. L., Pallant, J. I., & Pechenkina, E. (2023). Generative AI and the future of education: Ragnarök or reformation? A paradoxical perspective from management educators. *The International Journal of Management Education*, 21(2), 100790. <https://doi.org/10.1016/j.ijme.2023.100790>

Lund, B. D., & Wang, T. (2023). Chatting about ChatGPT: How may AI and GPT impact academia and libraries? *Library Hi Tech News*. <https://doi.org/10.1108/lhtn-01-2023-0009>

Mauldin, M. L. (1994). Chatterbots, TinyMUDs, and the Turing Test: Entering the Loebner Prize competition. In Hayes-Roth, B. & Korf, R. E. (Eds.), *AAAI-94: Proceedings of the 12th National Conference on Artificial Intelligence* (pp. 16-21). AAAI Press. <https://aaai.org/papers/00016-chatterbots-tinymuds-and-the-turing-test-entering-the-loebner-prize-competition/>

Mckie, I. A. S., & Narayan, B. (2019). Enhancing the academic library experience with chatbots: An exploration of research and implications for practice. *Journal of the Australian Library and Information Association*, 68(3), 268-277. <https://doi.org/10.1080/24750158.2019.1611694>

McNeal, M. L., & Newyear, D. (2013). Introducing chatbots in libraries. *Library Technology Reports*, 49(8), 5-10. <https://www.journals.ala.org/index.php/ltr/article/view/4504/5281>

Meta AI. (2023, February 24). *Introducing LLaMA: A foundational, 65-billion-parameter language model*. <https://ai.facebook.com/blog/large-language-model-llama-meta-ai/>

OpenAI. (2022, November 30). Introducing ChatGPT. *OpenAI Blog* <https://openai.com/blog/chatgpt>

OpenAI. (2023a). *OpenAI Documentation*. Retrieved April 6, 2023, from <https://platform.openai.com/docs>

OpenAI. (2023b, March 23). ChatGPT plugins. *OpenAI Blog* <https://openai.com/blog/chatgpt-plugins>

OpenAI. (2023c, May 12). Web browsing and Plugins are now rolling out in beta. *ChatGPT Release Notes* https://help.openai.com/en/articles/6825453-chatgpt-release-notes#h_9894d7b0a4

Panda, S., & Kaur, N. (2023). Exploring the viability of ChatGPT as an alternative to traditional chatbot systems in library and information centers. *Library Hi Tech News*. <https://doi.org/10.1108/LHTN-02-2023-0032>

- Pichai, S. (2023, February 6). Google AI updates: Bard and new AI features in Search. *Google Blog*. <https://blog.google/technology/ai/bard-google-ai-search-updates>
- Ren, X., Zhou, P., Meng, X., Huang, X., Wang, Y., Wang, W., Li, P., Zhang, X., Podolskiy, A., Arshinov, G., Bout, A., Piontkovskaya, I., Wei, J., Jiang, X., Su, T., Liu, Q., & Yao, J. (2023). *Pangu-Sigma: Towards trillion parameter language model with sparse heterogeneous computing*. ArXiv:2303.10845v1. <https://arxiv.org/abs/2303.10845>
- Rodriguez, S., & Mune, C. (2021). Library chatbots: Easier than you think. *Computers in Libraries*, 41(8), 29-32. <https://www.infotoday.com/cilmag/oct21/Rodriguez-Mune--Library-Chatbots-Easier-Than-You-Think.shtml>
- Soltan, S., Ananthakrishnan, S., FitzGerald, J., Gupta, R., Hamza, W., Khan, H., Peris, C., Rawls, S., Rosenbaum, A., Rumshisky, A., Prakash, C. S., Sridhar, M., Triefenbach, F., Verma, A., Tur, G., & Natarajan, P. (2022). *AlexaTM 20B: Few-shot learning using a large-scale multilingual seq2seq model*. ArXiv:2208.01448v2. <https://arxiv.org/abs/2208.01448>
- Streamlit. (2023). *Connect Streamlit to a private Google Sheet*. Streamlit Documentation. <https://docs.streamlit.io/knowledge-base/tutorials/databases/private-gsheet>
- Taori, R., Gulrajani, I., Zhang, T., DuBois, Y., Li, X., Guestrin, C., Liang, P., & Hashimoto, T. B. (2023, March 13). Alpaca: A Strong, Replicable Instruction-Following Model. *Stanford University Center for Research on Foundation Models*. <https://crfm.stanford.edu/2023/03/13/alpaca.html>
- Technology Innovation Institute. (2023, March 15). *Abu Dhabi-based Technology Innovation Institute Introduces Falcon LLM: Foundational Large Language Model (LLM) outperforms GPT-3 with 40 Billion Parameters*. <https://www.tii.ae/news/abu-dhabi-based-technology-innovation-institute-introduces-falcon-llm-foundational-large>
- University of Oklahoma. (2020). *Introducing "Bizzy"*. University of Oklahoma Libraries. <https://libraries.ou.edu/content/introducing-bizzy>
- Wallace, R. S. (2009). The anatomy of A.L.I.C.E. *Parsing the Turing Test: Philosophical and methodological issues in the quest for the thinking computer*, 181-210. https://doi.org/10.1007/978-1-4020-6710-5_13
- Weizenbaum, J. (1966). ELIZA - A computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1), 36-45. <https://doi.org/10.1145/365153.365168>
- Yang, S. (2023, March 27). The easiest way to work with large language models? *Towards Data Science*. <https://towardsdatascience.com/the-easiest-way-to-interact-with-language-models-4da158cfb5c5>

Appendix: An example conversation with Aisha

Authors: How are you doing, Aisha?

Aisha: I'm doing great, thank you for asking! How can I assist you with your library needs today?

Authors: I have a question about Scival

Aisha: Of course, I'd be happy to help! What would you like to know about Scival?

Authors: What can I do with it?

Aisha: Scival is a research analytics tool that can help you analyze research performance, identify research trends, and find potential collaborators. You can use it to search for publications, authors, institutions, and funding opportunities. Is there anything specific you would like to know about Scival?

Authors: How can I access it?

Aisha: You can access Scival by visiting the [Scival login page](#). Please note that each researcher needs to register for an account before using it. We also have a guide available that provides guidance on how to use Scival effectively.

Let me know if you have any other questions about Scival.

Authors: So I need to register?

Aisha: Yes, you need to register for an account before using Scival. Once you are on the login page, click on "Register Now" and follow the instructions to create your account. If you need any further assistance, please let me know.

Authors: Can you give me step by step instructions?

Aisha: Certainly! To register for an account on Scival, please follow these steps:

1. Go to the [Scival login page](#).
2. Click on "Register Now" located below the login form.
3. Fill in the required information such as your name, email address, and password.
4. Click on "Create Account" to complete the registration process.
5. Once you have registered, you can log in to Scival using your email address and password.

I hope this helps! Let me know if you have any further questions or if there's anything else I can assist you with.

Authors: Thank you so much!

Aisha: Sure, I'm glad I could help! If you have any other questions or need further assistance, feel free to ask.