Patrick Cher

19 Empowering Library Services: Building a ChatGPT Chatbot

Abstract: In an era characterised by rapid technological advancements, libraries have evolved to meet the everchanging demands of their patrons. Artificial Intelligence (AI) has emerged as a transformative tool. This chapter delves into the multifaceted realm of AI's applications in library and reference services, with a particular focus on ChatGPT-based chatbots enhanced by custom knowledge bases. The chapter contains an introduction to AI technology and its implications in contemporary library services and encapsulates the profound impact of AI on resource accessibility, data accuracy, efficient management, cultural heritage preservation, innovative user engagement, and the crucial ethical considerations that underpin responsible AI deployment. ChatGPT's capabilities are explored with the aim of equipping library professionals with the knowledge and resources required to harness AI through tailored chatbot solutions and outlines the basic competencies necessary for constructing customised chatbots that cater to the unique demands of libraries and their users. A demonstration chatbot was prepared for presentation at the International Federation of Library Associations and Institutions (IFLA) 88th World Library and Information Congress (WLIC) in Rotterdam, The Netherlands, in August 2023. The step-by-step instructions and guidance for building this chatbot are provided. In summary, this chapter embarks on an exploration of AI's transformative potential within libraries, outlining the knowledge and tools required to navigate the evolving landscape of AI-driven library services while adhering to ethical principles, and provides practical guidance to building a chatbot.

Keywords: Artificial intelligence; ChatGPT; Chatbots; Library reference services; Libraries – Information technologies

Introduction

In the ever-evolving landscape of libraries, where information is not just sought but expected at the speed of thought, the integration of Artificial Intelligence (AI) has emerged as a transformative force. Among the diverse AI technologies, generative AI, with its ability to create human-like text and engage in natural language conversations, stands at the forefront of innovation. At the heart of the AI revolution lies ChatGPT, a remarkable example of generative AI, poised to reshape library and reference services in ways previously unimagined.

https://doi.org/10.1515/9783111336435-020

Artificial Intelligence, Machine Learning, Deep Learning and Generative AI

Artificial Intelligence pertains to the theory and advancement of computer systems with the capability to execute tasks typically demanding human-like intelligence (Stripling 2023). AI has transcended the boundaries of science fiction to become an integral facet of contemporary life. It has permeated diverse sectors, optimising processes, providing data-driven insights, and even demonstrating creative capacities.

Within the expansive realm of AI, machine learning (ML) has emerged as a pivotal subfield. Machine learning enables computers to learn from data, recognise patterns, and make intelligent decisions without explicit programming (Stripling 2023). Deep learning is a subset of ML that uses neural networks with many layers, deep neural networks, to model and solve complex problems. Deep learning has gained significant attention and success in recent years due to its ability to automatically learn hierarchical representations from data, making it particularly well-suited for tasks like image and speech recognition. AI, ML and deep learning are integrated (Figure 19.1).

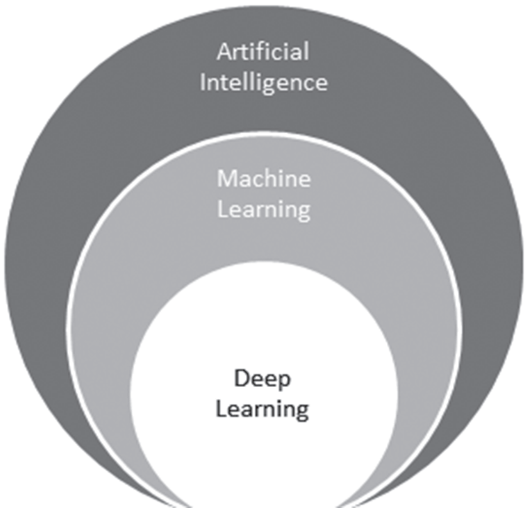


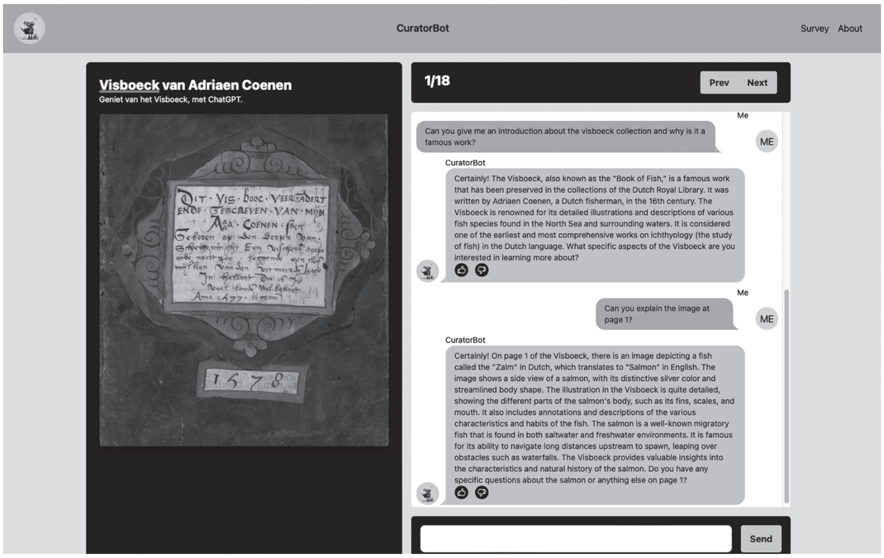
Figure 19.1: Relationship between AI, ML and deep learning

Generative AI is a technique associated with deep learning. Generative AI uses deep neural networks to create new data, such as text, images, music or other forms of content, based on patterns and structures learned from existing data. ChatGPT falls into this category, with its ability to understand and create content through human-like conversations.

Relevance of Artificial Intelligence in Today's Libraries

AI is playing a significant role in libraries today, exemplified by initiatives presented at the International Federation of Library Associations and Institutions (IFLA) World Library and Information Congress (WLIC) in 2023:

– SeTA@OP: Developed by the Publications Office of the European Union (EU), SeTA@OP employs ML and knowledge graphs to semi-automatically index EU publications. It improves information accessibility, accuracy, and ethical management, benefiting users by facilitating faster and more reliable resource discovery (Küster 2023).
– Visboeck CuratorBot: A conversational agent (Gukuma 2023) created by Delft University of Technology (TU Delft) Future Libraries Lab employs OpenAI's ChatGPT application programming interface (API) to engage users with cultural heritage objects, such as digitised drawings from *Het Visboeck/*[Book of Fish], a 16th century publication by Adriaen Coenen (Figure 19.2) It explores the application of natural language technology within cultural heritage, enhancing user education and interaction (Love, Gu, and Vandommele 2023)
– Poem Booth by Vouw: An AI Poetry machine that transforms portrait photos into personalised poems (Figure 19.3), fostering user engagement with literature and encouraging local writers to participate in library activities (Little Robots 2023; Vouw 2023).


CuratorBot
Survey
About
Visboeck van Adriaen Coenen
Geniet van het Visboeck, met ChatGPT.
1/18
Prev
Next
Send

Me

Can you give me an introduction about the visboeck collection and why is it a famous work?

CuratorBot

Certainly! The Visboeck, also known as the "Book of Fish," is a famous work that has been preserved in the collections of the Dutch Royal Library. It was written by Adriaen Coenen, a Dutch fisherman, in the 16th century. The Visboeck is renowned for its detailed illustrations and descriptions of various fish species found in the North Sea and surrounding waters. It is considered one of the earliest and most comprehensive works on ichthyology (the study of fish) in the Dutch language. What specific aspects of the Visboeck are you interested in learning more about?

Me

Can you explain the image at page 1?

CuratorBot

Certainly! On page 1 of the Visboeck, there is an image depicting a fish called the "Zalm" in Dutch, which translates to "Salmon" in English. The image shows a side view of a salmon, with its distinctive silver color and streamlined body shape. The illustration in the Visboeck is quite detailed, showing the different parts of the salmon's body, such as its fins, scales, and mouth. It also includes annotations and descriptions of the various characteristics and habits of the fish. The salmon is a well-known migratory fish that is found in both saltwater and freshwater environments. It is famous for its ability to navigate long distances upstream to spawn, leaping over obstacles such as waterfalls. The Visboeck provides valuable insights into the characteristics and natural history of the salmon. Do you have any specific questions about the salmon or anything else on page 1?

Figure 19.2: Visboeck CuratorBot (CuratorBot 2023)

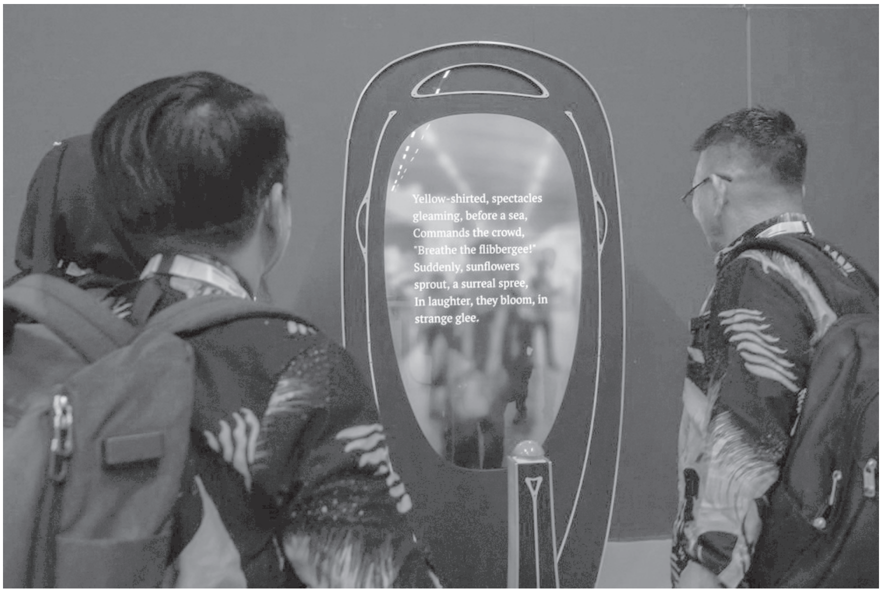


Figure 19.3: Poem Booth (Vouw n.d.)

The advantages of use of the technology include enhanced accessibility, accuracy, efficiency, preservation and innovative engagement.

- Accessibility: AI-powered tools like SeTA@OP significantly enhance the accessibility of information within libraries. They automate the process of indexing and categorising publications, making it easier for library users to search, discover, and access relevant materials quickly. The result is a profound impact on the overall user experience, ensuring that library resources are effectively utilised.
- Accuracy: AI can play a crucial role in maintaining the accuracy and quality of information within library collections. Through the automation of tasks such as indexing, metadata generation, and content validation, AI can minimise susceptibility to human error. By using reliable and high-quality training data, AI can impart information to library users that is not merely accessible but also dependable and of superior quality, thereby enhancing the library's standing as a reliable font of knowledge.
- Efficient management: AI has the capacity to assist libraries in efficiently managing their digital collections. ML algorithms can identify duplicate materials, streamline cataloguing processes, and ensure that digital resources are organised effectively. Valuable time and resources are conserved freeing library personnel for other activities and digital assets deployment is optimised.
- Cultural heritage preservation: Initiatives like the *Visboeck* CuratorBot highlight how AI can contribute to the preservation and exploration of cultural heritage. AI-driven conversational agents engage users in meaningful and educational conversations about historical artefacts and documents, not only bringing cultural heritage to life but also fostering a deeper appreciation and understanding of the past, aligning with the mission of libraries to educate and inform.
- Innovative engagement: AI-powered projects like the Poem Booth add a novel dimension to library services. By using AI to turn portrait photos into personalised poems, libraries can engage users in creative and interactive ways and encourage a deeper connection with literature. Libraries can use such services to promote the participation of local writers in library activities, fostering a sense of community and creativity within the library environment.

AI applications enrich library services by ensuring effective resource utilisation, increasing user satisfaction, and promoting access to digital cultural heritage.

What is ChatGPT

ChatGPT, an AI-driven chatbot created by OpenAI, operates on the advanced foundation of the generative pre-trained transformer (GPT) machine learning architecture. ChatGPT belongs to the class of large language models (LLMs) in the realm of ML for natural language processing (NLP). LLMs process vast volumes of text data, extracting intricate word relationships from the text. The performance of LLMs is related to the scale of their input datasets and parameter space. It is crucial to recognise that ChatGPT, despite its significant promise, is not impervious to substantial limitations. Concerns have arisen regarding instances where it generates responses that deviate from factual accuracy and may inadvertently perpetuate societal biases. The various concerns have prompted rigorous examination and widespread discussions within the international community.

Training Data

The model used for ChatGPT underwent training utilising extensive textual datasets sourced from the internet. The training encompassed an extensive corpus of data amounting to 570 gigabytes derived from various sources such as ebooks, web texts, Wikipedia, online articles, and assorted written materials accessible on the internet (Hughes 2023). To provide precise granularity, the training data comprised a staggering volume of 300 billion words that were input into the system for learning and model development (Hughes 2023).

How Does ChatGPT Work?

At its core, ChatGPT is like a virtual librarian capable of understanding and responding to natural language queries. At first glance, the technological premise of ChatGPT appears straightforward. It receives user requests, questions, or prompts and promptly delivers corresponding answers. However, it is imperative to underscore that the underlying technology facilitating the interactions is considerably more intricate than initial impressions may suggest. Figure 19.4 provides a high-level overview of the processes.

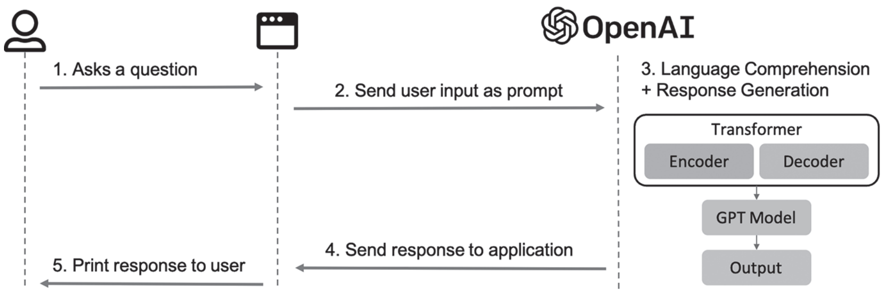

OpenAI
1. Asks a question
2. Send user input as prompt
3. Language Comprehension + Response Generation
Transformer
Encoder
Decoder
GPT Model
Output
4. Send response to application
5. Print response to user

Figure 19.4: High level flow of ChatGPT

As an LLM, ChatGPT operates probabilistically, predicting the subsequent word in a sentence. To attain proficiency, the model underwent a phase of supervised testing as shown in Figure 19.5.

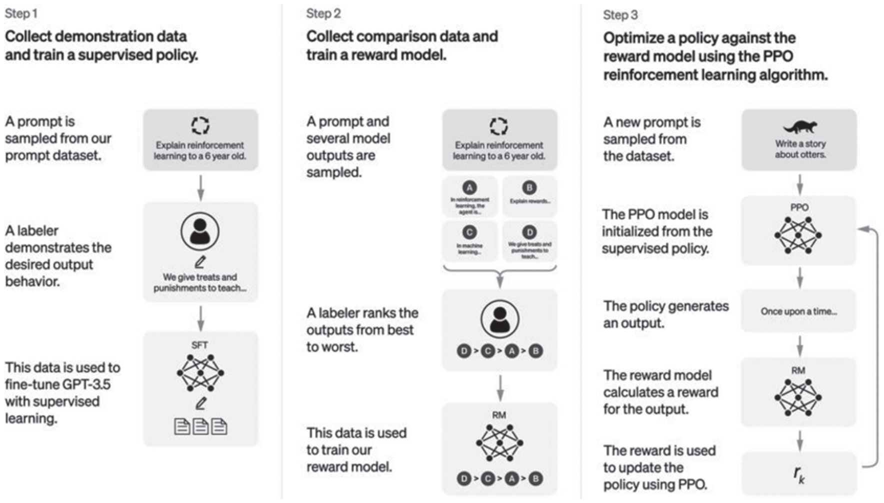

Step 1
Collect demonstration data and train a supervised policy.
A prompt is sampled from our prompt dataset.
Explain reinforcement learning to a 6 year old.
A labeler demonstrates the desired output behavior.
We give treats and punishments to teach...
SFT
This data is used to fine-tune GPT-3.5 with supervised learning.
Step 2
Collect comparison data and train a reward model.
A prompt and several model outputs are sampled.
Explain reinforcement learning to a 6 year old.
A
In reinforcement learning, the agent is...
B
Explain rewards...
C
In machine learning...
D
We give treats and punishments to teach...
A labeler ranks the outputs from best to worst.
D > C > A > B
This data is used to train our reward model.
RM
D > C > A > B
Step 3
Optimize a policy against the reward model using the PPO reinforcement learning algorithm.
A new prompt is sampled from the dataset.
Write a story about otters.
The PPO model is initialized from the supervised policy.
PPO
The policy generates an output.
Once upon a time...
The reward model calculates a reward for the output.
RM
The reward is used to update the policy using PPO.
r_k

Figure 19.5: An Overview of ChatGPT's Learning Process (OpenAI 2022)

Step 0 – Initial Pre-Training

ChatGPT's training commences with the foundational step of pre-training. During this phase, a substantial neural network architecture, denominated as the GPT, is subjected to training on an extensive corpus of publicly accessible textual content derived from the internet. The pre-training serves as the bedrock upon which the model acquires fundamental linguistic attributes, encompassing grammatical constructs, language structure, factual knowledge, and rudimentary reasoning capabilities. This phase does not impart task-specific knowledge to the model.

Step 1 – Supervised Fine-Tuning

Subsequently, ChatGPT progresses to supervised fine-tuning. This phase entails the generation of a dataset where human AI trainers engage in simulated conversations, assuming dual roles as both user and AI assistant. Additionally, the trainers are furnished with model-written suggestions to aid in composing responses. The objective is to imbue ChatGPT with conversational prowess and responsiveness. The model learns from human-generated conversations and is fine-tuned to generate contextually appropriate responses based on the conversations it has seen in the training data.

Step 2 – Reinforcement Learning from Human Feedback

After supervised fine-tuning, ChatGPT proceeds to the reinforcement learning from human feedback (RLHF) phase. AI human trainers evaluate responses generated from diverse models and rank them based on perceived quality. The evaluative feedback serves as the foundation for the creation of reward models that quantitatively gauge the superiority of various model responses. The reward models crystallise the notion of what constitutes a better or more contextually relevant response.

Step 3 – Fine-Tuning with Proximal Policy Optimisation

The next step is proximal policy optimisation PPO) which is employed to fine-tune ChatGPT's response generation policy. The reward models informed by human evaluator feedback guide the model's behaviour. Specifically, PPO orchestrates refinements in the model's response generation strategy, fostering the production of responses that closely align with those receiving higher rankings from human evaluators. The RLHF phase, complemented by PPO fine-tuning, is often an iterative journey. Multiple rounds of fine-tuning occur to continually improve the model's performance. The ultimate goal is to enable ChatGPT to generate responses characterised not only by grammatical accuracy but also by contextual relevance, thereby conforming to the criteria of high quality set forth by human standards.

The Custom Knowledge Base

This section explains LLMs and the concept of a custom knowledge base through an illustrative analogy. Imagine a scenario where the objective is to impart specific skills to a canine for the purpose of executing diverse tasks. In this metaphorical framework, the dog serves as a representative metaphor for an AI model, and the competencies it acquires correspond to the proficiencies inherent in the AI model.

LLMs (Large Language Models)

Using the dog analogy, an LLM may be conceptualised as akin to the innate attributes and abilities of a dog. The training of a dog relates to basic skill: sit, come, down and stay (Figure 19.6). Certain dog breeds are renowned for distinct traits, such as agility or intelligence. Similarly, LLMs, exemplified by models like GPT-3, exhibit inherent linguistic capabilities. Through exhaustive training procedures, the LLMs attain an expansive comprehension of language, akin to how specific dog breeds naturally incline toward particular aptitudes.

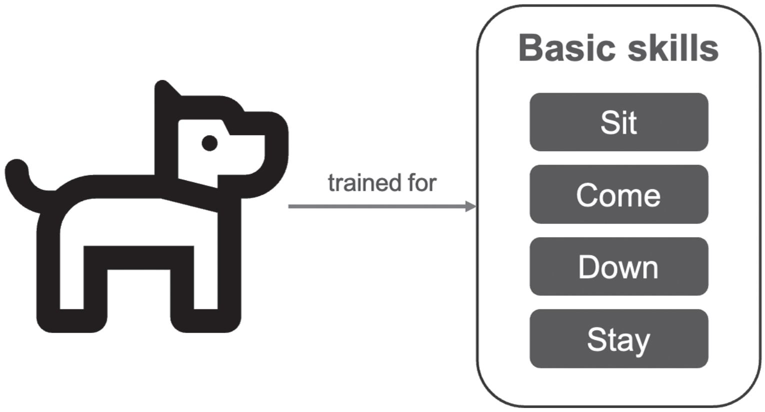


Figure 19.6: Dog’s innate abilities

Custom Knowledge Base

Extending the dog analogy, the custom knowledge base can be compared with the supplementary training and information administered to the canine. In practice, dogs can be meticulously trained to execute particular tasks with special services skills as a police, guide or hunting dog, and provide specialised assistance to individuals, for example, people with disabilities (Figure 19.7). In a parallel vein, the custom knowledge base embodies the specialised tutelage imparted to an LLM, thereby enabling it to demonstrate proficiency within a designated domain. The reservoir of knowledge comprises curated datasets, factual information, and contexts specific to a chosen field, mirroring the tailored training received by a dog to excel in precise roles and functions.

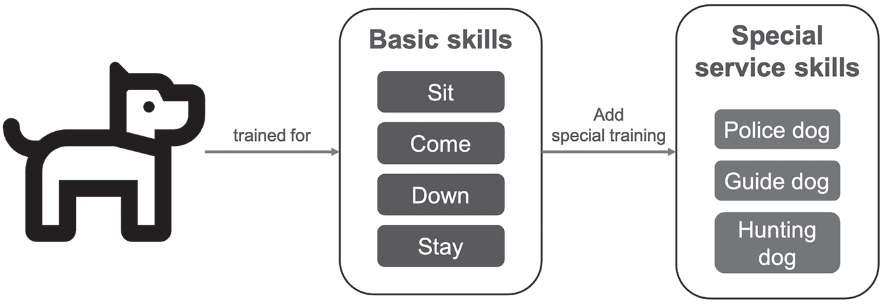


Figure 19.7: Special service skills training for dogs

In ways similar to how canine performance is augmented through rigorous training, an LLM's competencies undergo refinement through the process of fine-tuning with a custom knowledge base. The iterative procedure empowers the LLM to furnish responses that are not only more precise but also contextually pertinent within a specific domain, analogous to the manner in which a well-trained dog excels in its designated tasks. Through the allegorical lens of canine training, the interplay between LLMs and custom knowledge bases becomes more readily comprehensible to readers seeking a nuanced understanding of this sophisticated AI technology.

Setting Up the Environment

A demonstration chatbot was prepared for presentation at the International Federation of Library Associations and Institutions (IFLA) 88th World Library and Information Congress (WLIC) in Rotterdam, the Netherlands, in August 2023. An overview of the work undertaken and guidelines on how to construct such a chatbot are provided in this chapter. Included are details on setting up the environment and building the chatbot.

Setting up the environment addresses the technology stack, a combination of software tools and technologies, used for the demonstration chatbot project as well as the specifics and explanations of their development. The technology stack used for this demonstration comprised a meticulously selected ensemble of tools and libraries, each playing a distinctive role in facilitating the development and deployment of a sophisticated ML model. The technology inclusions are delineated in Figure 19.8 and comprise Python as the programming language, gpt-3.5-turbo as the GPT model, Gradio as the ML model and Google Drive for the storage.

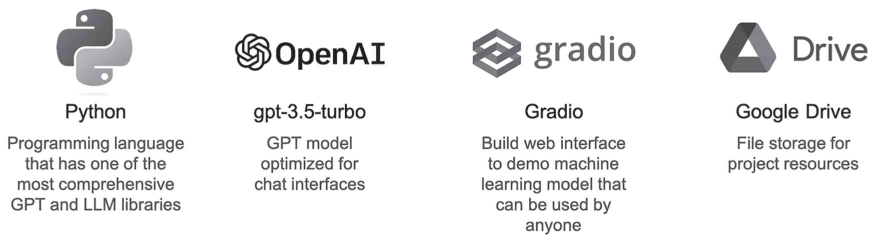

OpenAI
gradio
Drive
Python
Programming language that has one of the most comprehensive GPT and LLM libraries
gpt-3.5-turbo
GPT model optimized for chat interfaces
Gradio
Build web interface to demo machine learning model that can be used by anyone
Google Drive
File storage for project resources

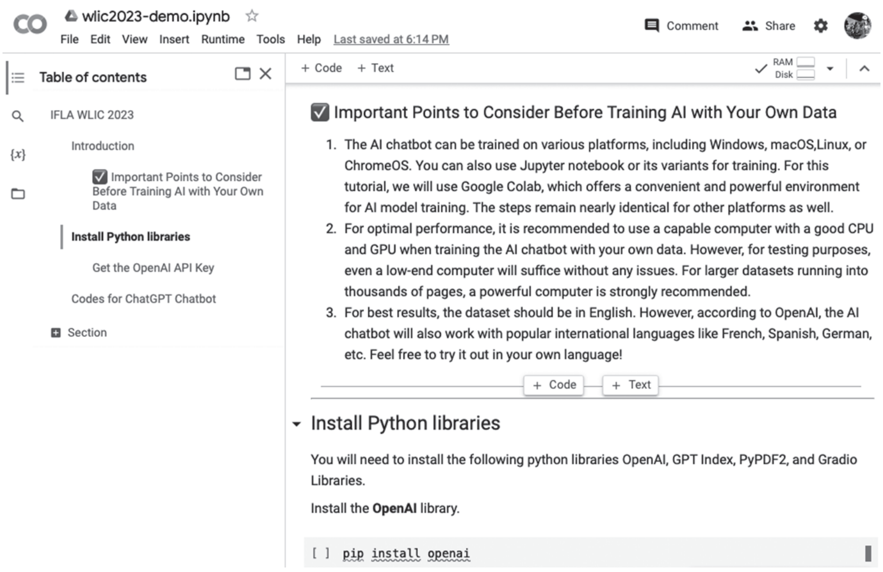
Figure 19.8: The technology stack

Prior to delving into the intricacies of configuring and executing Python source code for a ChatGPT chatbot, it is essential to recognise the two environments available: web-based notebook platform and the integrated development environment (IDEs) (Nehme 2023). Executing Python code within a web-based notebook platform like Google Colaboratory (Colab) through a web browser affords users several advantages. It encompasses convenience and universal accessibility across devices, streamlined access to pre-installed libraries for efficient code development, and the availability of powerful graphics processing units (GPUs) and tensor processing units (TPUs) that can substantially augment computational performance (Nehme 2023). Conversely, an IDE like Microsoft Visual Studio provides greater privacy and security, as data and code reside locally, mitigating exposure to potential online vulnerabilities. A local development setup empowers users to harness the full potential of their hardware resources without constraints. It is pertinent to mention that setting up and configuring an IDE adds an additional layer of complexity (Nehme 2023).

The choice between the two environments depends on factors like convenience, resource requirements, collaboration needs, and data security considerations, with each approach having its own strengths and trade-offs. To facilitate the exploration of building ChatGPT chatbots within the context of this chapter, the web-based notebook approach has been adopted.

The Google Colab Environment

Google Colab is a cloud-based platform tailored to facilitate the creation and execution of Python code with minimal setup (Google n.d.). Access to the platform is straightforward, requiring a Google account, and it offers the flexibility to create new Python notebooks or access existing ones (Google n.d.b). Figure 19.9 shows the user interface.


wlic2023-demo.ipynb
File Edit View Insert Runtime Tools Help Last saved at 6:14 PM
Comment
Share
Table of contents
IFLA WLIC 2023
Introduction
Important Points to Consider Before Training AI with Your Own Data
Install Python libraries
Get the OpenAI API Key
Codes for ChatGPT Chatbot
Section
+ Code + Text
RAM
Disk

Important Points to Consider Before Training AI with Your Own Data

1. The AI chatbot can be trained on various platforms, including Windows, macOS,Linux, or ChromeOS. You can also use Jupyter notebook or its variants for training. For this tutorial, we will use Google Colab, which offers a convenient and powerful environment for AI model training. The steps remain nearly identical for other platforms as well.
2. For optimal performance, it is recommended to use a capable computer with a good CPU and GPU when training the AI chatbot with your own data. However, for testing purposes, even a low-end computer will suffice without any issues. For larger datasets running into thousands of pages, a powerful computer is strongly recommended.
3. For best results, the dataset should be in English. However, according to OpenAI, the AI chatbot will also work with popular international languages like French, Spanish, German, etc. Feel free to try it out in your own language!

Install Python libraries

You will need to install the following python libraries OpenAI, GPT Index, PyPDF2, and Gradio Libraries.

Install the **OpenAI** library.

```
[ ] pip install openai
```

Figure 19.9: Screenshot of Google Colab's interface

Google Colab distinguishes itself by providing pre-configured computational environments and complimentary access to GPUs and TPUs which is particularly advantageous for computationally intensive tasks like training MLMs. It seamlessly integrates with Google Drive for data storage and collaborative sharing, executing code at the cell level while automatically preserving all work within Google Drive which makes Google Colab an appealing choice for individuals seeking a Python development environment without the need for local installations, while also benefiting from the capabilities of robust cloud-based computing resources.

Setting Up Google Colab Notebook

Creating a Google Colab notebook is a straightforward process which is described in the following five steps:

6. Access Google Colab. Firstly, open a web browser and navigate to the Google Colab website
7. Sign in to your Google Account if not already signed in and proceed to a Google account so that notebooks can be accessed and saved on Google Drive
8. Create a notebook. On the Google Colab homepage, click on the New Notebook button under File. This action will initiate the creation of a new Colab notebook.
9. Name the Notebook (Optional). Naming the notebook can be helpful for organisation. Click on "Untitled" at the top left of the notebook interface and provide a meaningful name.
10. Setup the file storage and structure. Click on Mount Drive, the third button underneath the Files title which will seek permission to store and retrieve files in Google Drive.

Once permission has been granted, navigate to drive >MyDrive > Colab Notebooks, and create a folder to store all files required for the chatbot project. Ensure that the newly created notebook is stored in the folder (Figure 19.10). In the same folder, create a docs folder and upload the documents for the custom knowledge base as indicated in Figure 19.10. Take note of the file path for the construct_index function which will be covered in this chapter.

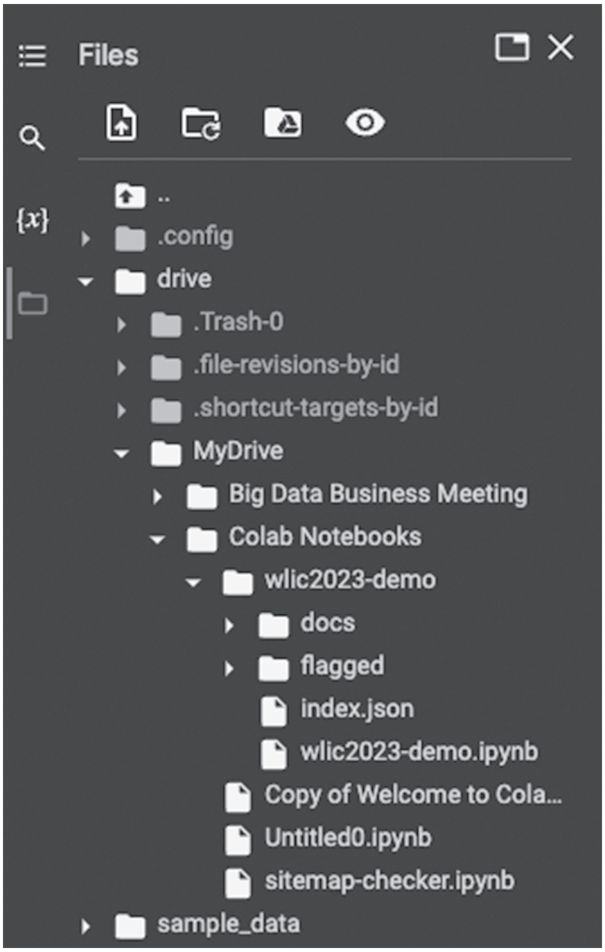
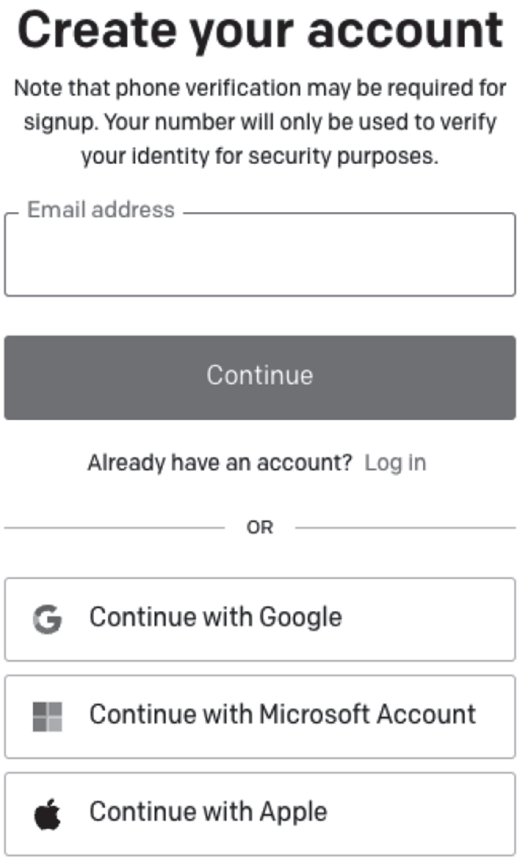


Figure 19.10: File interface at Google Colab

Obtaining an OpenAI API

OpenAI's API enables users to integrate ChatGPT-style AI capabilities into their own applications and tools. OpenAI offers users a seamless process for obtaining an API key, complete with a US$5 free credit upon registration (OpenAI 2023). To initiate the integration of the OpenAI capabilities, an API key can be obtained from OpenAI by following the steps outlined:

1. Create account and sign-in. To embark on the journey of obtaining an OpenAI API key, access the official OpenAI developer platform website (OpenAI n.d.). In the absence of an existing account, prospective users are required to initiate the account creation process by adhering to the straightforward registration protocol detailed on the website. For pre-existing users, a conventional sign-in mechanism is available through registered email credentials and associated passwords. Alternatively, users may opt to employ Google or Microsoft accounts for authentication (Figure 19.11).
2. Access the user account. Upon successful authentication, the user's name and profile icon will be visible at the top-right corner of OpenAI's platform homepage. Click on the user's name, and a dropdown menu containing various options is revealed (Figure 19.12).


Create your account

Note that phone verification may be required for signup. Your number will only be used to verify your identity for security purposes.

Email address
Continue
Already have an account? Log in
OR
Continue with Google
Continue with Microsoft Account
Continue with Apple

Figure 19.11: Account creation page on OpenAI website

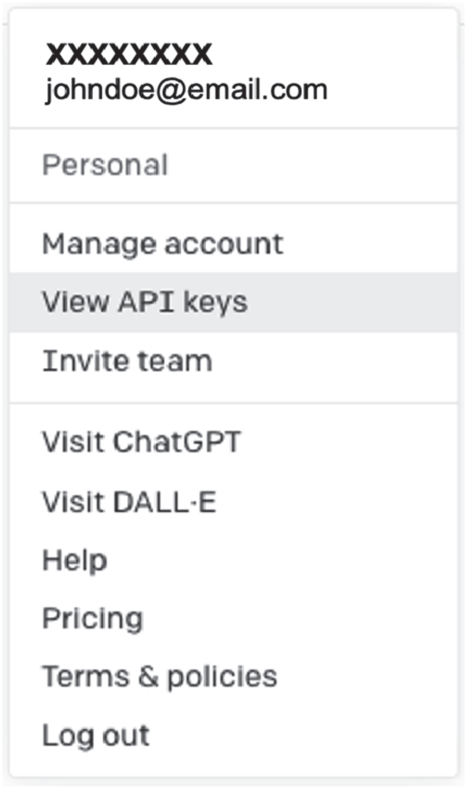
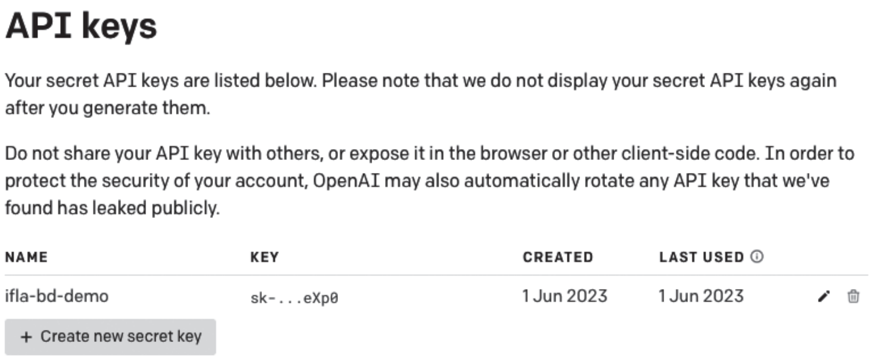


Figure 19.12: Account dropdown menu

3. Within the dropdown menu, select the View API keys option (Figure 19.12). API keys listing in OpenAI platform which will redirect to a page for the management of API keys. Should a new API key need to be generated, a conspicuous button named “Create new secret key” situated centrally on the page will initiate the procedure for the generation of an API key. It is imperative to promptly save the API key once it is generated and essential to acknowledge that closing the window displaying the API key will irreversibly lead to an inability to retrieve the key thereafter. Consequently, it is of utmost importance that the API key is securely stored (Figure 19.13) to ensure uninterrupted and continuous access to OpenAI’s comprehensive suite of services.


API keys

Your secret API keys are listed below. Please note that we do not display your secret API keys again after you generate them.

Do not share your API key with others, or expose it in the browser or other client-side code. In order to protect the security of your account, OpenAI may also automatically rotate any API key that we've found has leaked publicly.

NAME	KEY	CREATED	LAST USED
ifla-bd-demo	sk-...eXp0	1 Jun 2023	1 Jun 2023

Figure 19.13: API keys listing in OpenAI platform

Installation of Python Libraries

In the code cell of the Google Colab Notebook, run the code snippet:

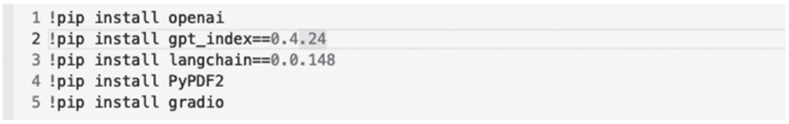

```
!pip install openai
!pip install gpt_index==0.4.24
!pip install langchain==0.0.148
!pip install PyPDF2
!pip install gradio
```

Figure 19.14: Install the Python libraries required for the demonstration chatbot

Pip is the package installer for Python (Figure 19.14). You can use pip to install packages from the Python package index and other indexes. The code snippet can be likened to assembling the necessary ingredients and equipment in a kitchen before cooking a meal. The code snippet guides the computer to obtain essential software tools like Openai, Gpt_index, LangChain, PyPDF2, and Gradio from the Internet and ensures that the computer has all the required tools readily available for use when executing the subsequent code snippets, much the same as having all the ingredients and utensils at hand before starting the cooking process.

Prerequisite Python Libraries

– ***Openai*** is a software package that facilitates interaction with OpenAI's services and APIs. It empowers developers to integrate OpenAI's AI capabilities, including NLP and text generation, into their applications.

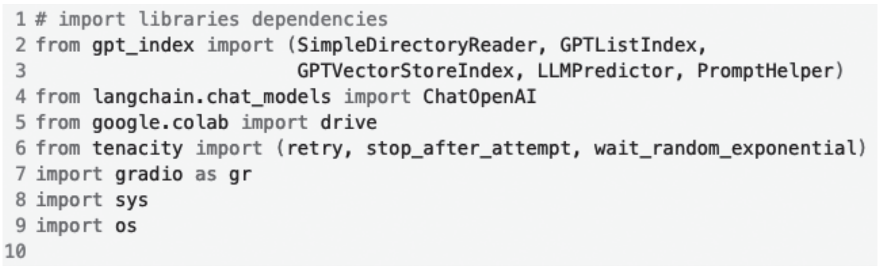
- *Gpt_index* is a framework designed to facilitate the development of LLM applications. It offers essential tools, including data connectors for ingesting data sources, data structuring capabilities, and an advanced retrieval and query interface.
- *LangChain* is a framework designed for building applications that harness language models. It offers customisable components, facilitating the creation of structured chains for specific tasks and includes interfaces for various modules.
- *PyPDF2* is a software package that provides developers with tools for working with Portable Document Format (PDF) files programmatically. It offers functionalities for reading, manipulating, and extracting data from PDF documents.
- *Gradio* is a software package that provides developers with tools and functionalities to quickly build and deploy interactive web applications that display ML models.

Building the Chatbot

Once the environment has been set up, the groundwork for building the chatbot is laid. In the following step, the newly installed libraries would be imported into the Google Colab Notebook, OpenAI API key is integrated to unlock ChatGPT capabilities, Google Drive is linked to store the generated index for knowledge retrieval, and a user interface is built to enable interaction with the demonstration chatbot.

Importing Prerequisite Library Packages

In the code cell of the Google Colab Notebook, run the code snippet:



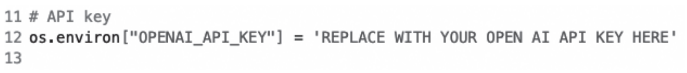
```
# import libraries dependencies
from gpt_index import (SimpleDirectoryReader, GPTListIndex,
                       GPTVectorStoreIndex, LLMPredictor, PromptHelper)
from langchain.chat_models import ChatOpenAI
from google.colab import drive
from tenacity import (retry, stop_after_attempt, wait_random_exponential)
import gradio as gr
import sys
import os

```

Figure 19.15: Import the dependent libraries in Google Colab Notebook

The code snippet (Figure 19.15) ensures that everything essential for the functioning of the chatbot application is readily available. The tools are used for file reading, working with specialised models, and interfacing with Google Drive. Additionally, the code includes a tool for error handling, and another tool for creating a web-based interface for the demonstration chatbot to function.

Specifying OpenAI API Key



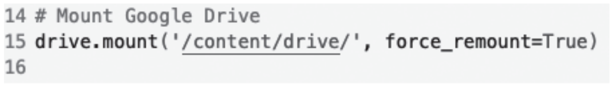
```
# API key
os.environ["OPENAI_API_KEY"] = 'REPLACE WITH YOUR OPEN AI API KEY HERE'

```

Figure 19.16: Specify OpenAI API key in Google Colab Notebook

The line of code configures the storage of the OpenAI API key (Figure 19.16) in the computer's memory, reserving a designated location where the computer securely retains a confidential code, known as the API key. The key is essential for executing tasks requiring access to OpenAI's AI capabilities, including responding to prompts and training a customised knowledge base. Substitute the text PLACEHOLDER FOR API KEY with the previously acquired API key enclosed within the single quotation marks.

Connecting Google Drive



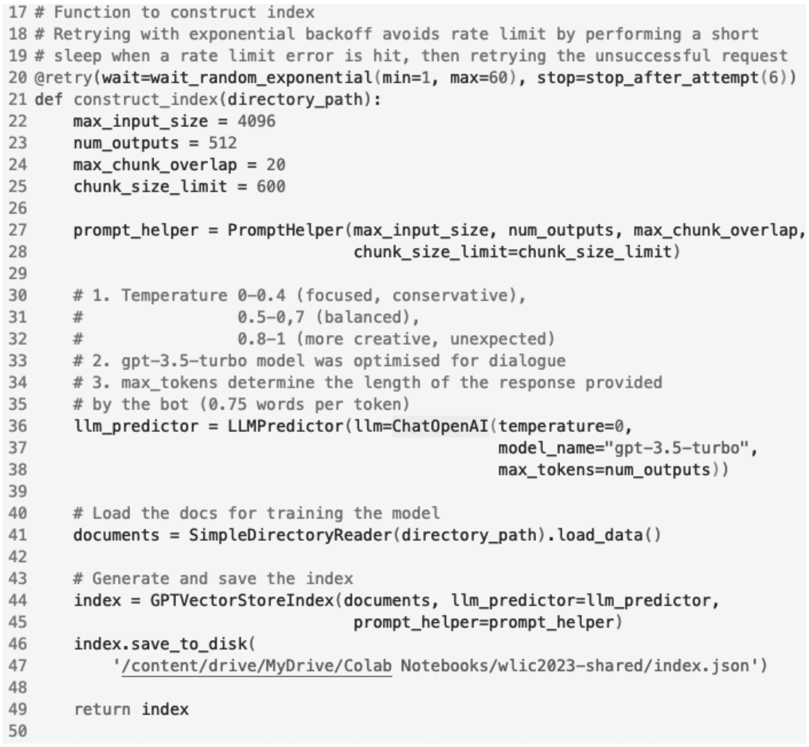
```
# Mount Google Drive
drive.mount('/content/drive/', force_remount=True)

```

Figure 19.17: Connect Google API for accessing and storing files in Google Colab Notebook

The line of code directs the computer to use a file path to connect with a place called /content/drive/ within Google Drive (Figure 19.17), and makes sure the connection is established even if previously disconnected so that the computer can access files and data stored.

Ingesting Documents and Creating an Index for Knowledge Retrieval



```
# Function to construct index
# Retrying with exponential backoff avoids rate limit by performing a short
# sleep when a rate limit error is hit, then retrying the unsuccessful request
@retry(wait=wait_random_exponential(min=1, max=60), stop=stop_after_attempt(6))
def construct_index(directory_path):
    max_input_size = 4096
    num_outputs = 512
    max_chunk_overlap = 20
    chunk_size_limit = 600

    prompt_helper = PromptHelper(max_input_size, num_outputs, max_chunk_overlap,
                                 chunk_size_limit=chunk_size_limit)

    # 1. Temperature 0-0.4 (focused, conservative),
    #                0.5-0,7 (balanced),
    #                0.8-1 (more creative, unexpected)
    # 2. gpt-3.5-turbo model was optimised for dialogue
    # 3. max_tokens determine the length of the response provided
    # by the bot (0.75 words per token)
    llm_predictor = LLMPredictor(llm=ChatOpenAI(temperature=0,
                                                model_name="gpt-3.5-turbo",
                                                max_tokens=num_outputs))

    # Load the docs for training the model
    documents = SimpleDirectoryReader(directory_path).load_data()

    # Generate and save the index
    index = GPTVectorStoreIndex(documents, llm_predictor=llm_predictor,
                                 prompt_helper=prompt_helper)
    index.save_to_disk(
        '/content/drive/MyDrive/Colab Notebooks/wlic2023-shared/index.json')

    return index
```

Figure 19.18: Create the function to ingest documents and create index in Google Colab Notebook

The code defines a function named construct_index. The function is designed to create an index, which is like a digital library catalogue, by processing the collection of documents located in the specified folder on Google Drive. The created index helps the chatbot quickly find and retrieve information from the documents when needed. It is similar to how a library keeps track of its books so that users can easily find the one to read. The code is preparing the index to answer questions based on the contents of specified documents.

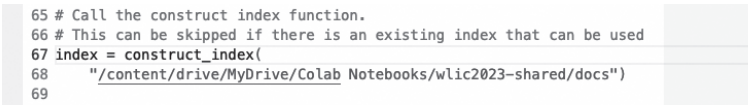
The process to generate an index involves several tasks:

- Setting parameters like input size, the number of desired outputs, and other factors that influence the indexing process
- Preparing a helper tool called Prompt_Helper to assist with generating prompts for the language model

– Configuring a language model known as gpt-3.5-turbo to be used for generating responses with the predefined temperature and response length within the index. Temperature is a parameter that controls the creativity or randomness of the text. A higher temperature results in a diverse output, while a lower temperature makes the output more focused
– Loading the documents from the specified directory for training the model, and
– Creating and saving the index, which helps in organising and retrieving information from the documents.

The construct_index function (Figure 19.18) is designed to handle errors effectively, as indicated by the @retry decorator. When the function encounters a problem during the indexing process, it does not give up right away but instead, tries repeatedly, up to six times, with a random pause of one to 60 seconds between each try, to ensure that the process eventually succeeds, even if there are temporary issues.

Execute the following code snippet to invoke the construct_index function:



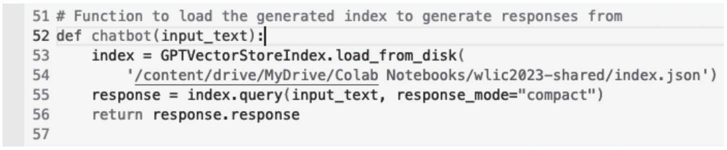
```
# Call the construct index function.
# This can be skipped if there is an existing index that can be used
index = construct_index(
    "/content/drive/MyDrive/Colab Notebooks/wlic2023-shared/docs")

```

Figure 19.19: Invoke the construct_index function in Google Colab Notebook

The code should be executed whenever there is a need to update the model with changes to the custom knowledge base, such as adding, removing, or updating documents, or performing the initial model training. It is essential to run this code snippet (Figure 19.19) to ensure that any modifications made to the knowledge base are properly integrated into the model's index. The execution of the construct_index function may require some time to complete. When dealing with a relatively small custom knowledge base consisting of approximately 2,000 textual PDF articles, ranging from 1,000 to 2,000 words in length, the process may take around 45 minutes. Users should be prepared for a potentially lengthy execution time, especially when working with larger or more extensive knowledge bases.

Providing Responses to Users' Prompts



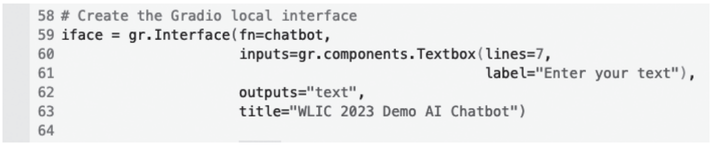
```
# Function to load the generated index to generate responses from
def chatbot(input_text):
    index = GPTVectorStoreIndex.load_from_disk(
        '/content/drive/MyDrive/Colab Notebooks/wlic2023-shared/index.json')
    response = index.query(input_text, response_mode="compact")
    return response.response

```

Figure 19.20: Create the chatbot function in Google Colab Notebook to load the generated index file to generate responses to users' questions submitted in the demonstration chatbot

The code defines a function called chatbot that takes an input text as an argument. Inside the function, a pre-existing index consisting of a structured collection of data from a specific location on Google Drive is loaded and used to process the input text, generate a response. and return the response (Figure 19.20). The code creates a chatbot that can understand and respond to user input based on the data stored in the index.

Creating the User Interface with Gradio

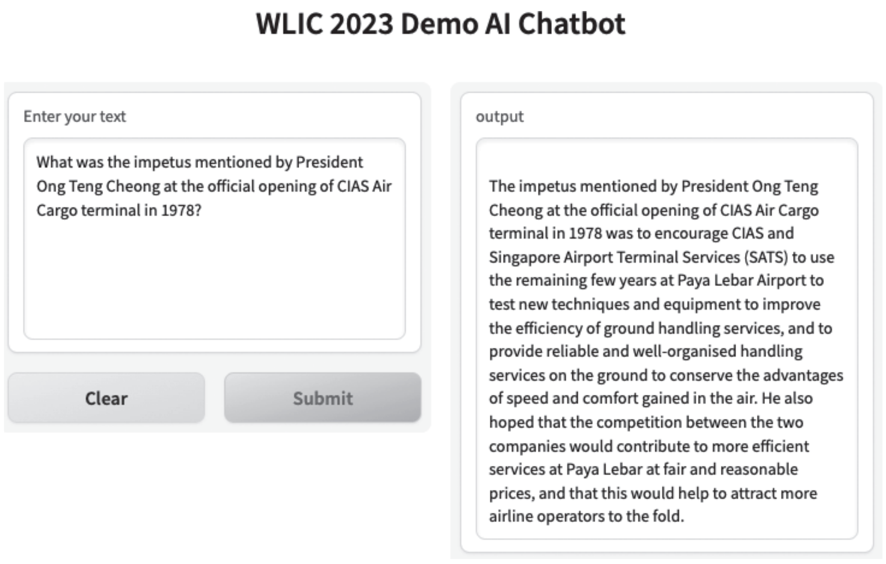


```
# Create the Gradio local interface
iface = gr.Interface(fn=chatbot,
                     inputs=gr.components.Textbox(lines=7,
                                                  label="Enter your text"),
                     outputs="text",
                     title="WLIC 2023 Demo AI Chatbot")

```

Figure 19.21: Create the User Interface with Gradio in Google Colab Notebook

The code sets up the visual and interactive components for engaging with the AI chatbot in a formal and user-friendly manner. Users can input up to seven lines of text in the text box, and the chatbot responds with text. Access to the user interface is limited to the web-based notebook. After successfully executing the provided code (Figure 19.21), users will have access to a web-based user interface. Figure 19.22 provides a screenshot illustrating an interaction between a user and the chatbot, showcasing the seamless engagement facilitated by the interface.

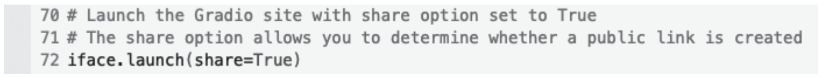


What was the impetus mentioned by President Ong Teng Cheong at the official opening of CIAS Air Cargo terminal in 1978?

The impetus mentioned by President Ong Teng Cheong at the official opening of CIAS Air Cargo terminal in 1978 was to encourage CIAS and Singapore Airport Terminal Services (SATS) to use the remaining few years at Paya Lebar Airport to test new techniques and equipment to improve the efficiency of ground handling services, and to provide reliable and well-organised handling services on the ground to conserve the advantages of speed and comfort gained in the air. He also hoped that the competition between the two companies would contribute to more efficient services at Paya Lebar at fair and reasonable prices, and that this would help to attract more airline operators to the fold.

Figure 19.22: User interface of the chatbot

Run this code snippet to create a public link:



```
# Launch the Gradio site with share option set to True
# The share option allows you to determine whether a public link is created
iface.launch(share=True)
```

Figure 19.23: Optional step to create a public link for the chatbot user interface in Google Colab Notebook

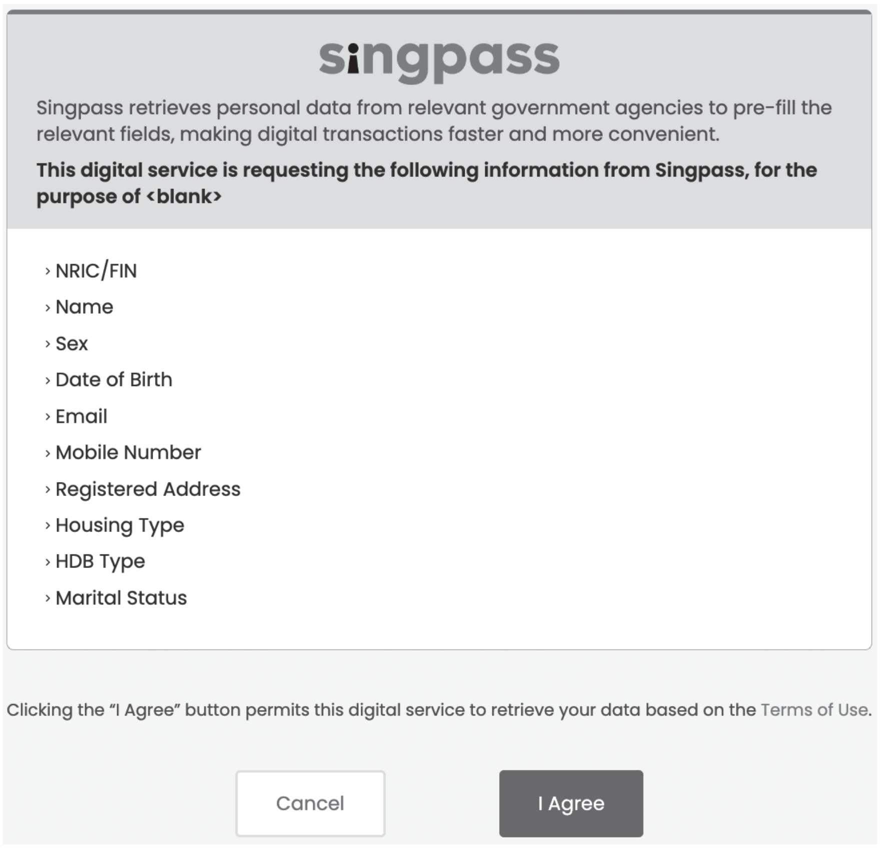
The code snippet (Figure 19.23) is optional and can be invoked when needed. The share=True part means that the chatbot user interface can be shared via a public Uniform Resource Locator (URL) accessible by multiple users over the Internet.

Ethical Considerations

In the development and deployment of AI technologies such as ChatGPT chatbots, adherence to ethical principles is of utmost importance in library services. Libraries must prioritise ethical guidelines to establish trust and uphold responsible AI practices throughout the system design and development phases. Google's 2018 guidelines for AI development underscore societal benefit, fairness, safety, accountability, privacy, scientific excellence, and responsible use and are in their fifth revised edition (Google 2023). The commitment to avoiding harmful and unethical uses of AI demonstrates a responsible approach. It also reflects an awareness of the potential risks and consequences associated with AI technologies.

The IFLA Statement on Libraries and Artificial Intelligence recognised the potential for libraries to encourage the use of responsible AI by others in 2020, as well as adopting principles for use within libraries and emphasised two key ethical considerations: privacy and bias (IFLA 2020). Privacy concerns arise from the extensive use of data for AI training and decision-making, encompassing both non-personal and personally identifiable information. AI-driven analysis outputs may also expose sensitive data, raising questions about individual privacy and data protection rights (Cox, Pinfield, and Rutter 2019). Libraries employing AI solutions, such as ChatGPT, must prioritise data privacy, obtain informed consent for personal data processing, enhance transparency, and implement flexible consent mechanisms.

An example of a consent mechanism can be found in Singpass, the Singapore government's digital identity service, which facilitates the secure sharing of government-verified personal data with government agencies and businesses for transactions (Singpass 2023a). It exemplifies a robust approach to respecting individuals' data privacy rights while ensuring the responsible use of AI-driven services (Figure 19.24).


singpass

Singpass retrieves personal data from relevant government agencies to pre-fill the relevant fields, making digital transactions faster and more convenient.

This digital service is requesting the following information from Singpass, for the purpose of <blank>

› NRIC/FIN
› Name
› Sex
› Date of Birth
› Email
› Mobile Number
› Registered Address
› Housing Type
› HDB Type
› Marital Status

Clicking the "I Agree" button permits this digital service to retrieve your data based on the Terms of Use.

Figure 19.24: Consent request in Singpass (Singpass 2023b)

Bias is another significant ethical concern. Unfair outcomes must be avoided (Latonero 2018). Unintentional biases may infiltrate AI development during problem framing, data labelling, or attribute selection, often perpetuating societal inequalities (Borgesius 2018). Addressing biases involves proactive measures such as conducting practice exchanges, organising bias management symposia, and establishing committees to guide responsible AI engagement (Padilla 2019). It is imperative to scrutinise the AI model training processes of external vendors and raise concerns related to ethics, privacy, or bias. The effectiveness of ethical principles and measures hinges on their diligent implementation and enforcement. Ensuring transparency and accountability in adhering to standards and measures is crucial for their practical application.

Conclusion

Artificial intelligence developments are growing apace. Libraries must adapt and adopt new ways of service provision. Library staff must gain competencies in developing and applying various approaches to the use of AI. The use of chatbots has particular usefulness in libraries and simple strategies can maximise their use. Libraries also have a particular responsibility to ensure that continuous observance of ethical principles are in place with regard to evolving AI technologies to ensure their sustainability, relevance and effectiveness over the long term. Libraries deploying AI-driven service must prioritise ethical principles, uphold trustworthiness, and promote transparency, aligning their AI initiatives with societal objectives and fostering public welfare and innovation in library services.

References

Borgesius, Frederick Zuiderveen. 2018. “Discrimination, Artificial Intelligence, and Algorithmic Decision-making.” Strasbourg: Council of Europe. https://rm.coe.int/discrimination-artificial-intelligence-and-algorithmic-decision-making/1680925d73.

Cox, Andrew M., Stephen Pinfield, and Sophie Rutter. 2019. “The Intelligent Library: Thought Leaders’ Views on the Likely Impact of Artificial Intelligence on Academic Libraries.” *Library Hi Tech* 37, no. 3: 418–435. https://doi.org/10.1108/LHT-08-2018-0105. Available at https://eprints.whiterose.ac.uk/136552/1/file.PDF/1000.

CuratorBot. 2024. “Visboeck van Adriaen Coenen: Geniet van het Visboeck, met ChatGPT.” https://visboeck.vercel.app/.

Google. n.d.a. “Google Colab - Frequently Asked Questions.” https://research.google.com/colaboratory/faq.html.

Google. n.d.b. “Google Colaboratory.” https://colab.google/.

Google. 2023. “AI Principles Progress Update 2023.” https://ai.google/static/documents/ai-principles-2023-progress-update.pdf.

Gukuma, Eric. 2023. “Visboeck: CuratorBot (CB): Specifications and Details.” [Developed by H. Gu and J.S. Love at TU Delft]. https://github.com/gukuma/VisboeckGPT.

Hughes, Alex. 2023. “ChatGPT: Everything You Need to Know About OpenAI’s GPT-4 Tool.” *BBC Science Focus*, September 26, 2023. https://www.sciencefocus.com/future-technology/gpt-3.

International Federation of Library Associations and Institutions (IFLA). 2020. “IFLA Statement on Libraries and Artificial Intelligence.” https://repository.ifla.org/bitstream/123456789/1646/1/ifla_statement_on_libraries_and_artificial_intelligence-full-text.pdf.

Küster, Marc W. 2023. “SeTA@OP - Semi-automatic Indexing of EU publications; Combining the Power of Machine Learning, Knowledge Graphs, and Human Intelligence.” Presentation at the 88th IFLA World Library and Information Congress, 21-25 August 2023, Rotterdam, The Netherlands. https://iflawlic2023.abstractserver.com/program/#/details/presentations/404.

Latonero, Mark. 2018. “Governing Artificial Intelligence: Upholding Human Rights & Dignity.” Data & Society report. https://datasociety.net/wp-content/uploads/2018/10/DataSociety_Governing_Artificial_Intelligence_Upholding_Human_Rights.pdf.

Little Robots. 2023. “The Tech Behind the Poem Booth.” [Blog]. Posted by Hugo September 7, 2023. https://www.littlerobots.nl/blog/the-tech-behind-the-poembooth/.

Love, Jeff, Heng Gu, and Jeroen Vandommele. 2023. “The Visboeck Curatorbot: Discussing Heritage Collections with Machines.” Poster presented at 88th IFLA World Library and Information Congress (WLIC), 2023 Rotterdam. https://repository.ifla.org/handle/123456789/3058.

Nehme, Adel. 2023. “6 Best Python IDEs for Data Science in 2023.” *Datacamp. Tutorials.* Updated February 2023. https://www.datacamp.com/tutorial/data-science-python-ide.

OpenAI. n.d. “Welcome to the OpenAI Developer Platform.” https://platform.openai.com/.

OpenAI. 2022. “Aligning Language Models to Follow Instructions.” https://openai.com/research/instruction-following.

OpenAI. 2023. “Usage Openai API, 5 Dollar Free Gift.” *OpenAI. Documentation. API Reference.* OpenAI Developer Forum. November 2023. https://community.openai.com/t/usage-openai-api-5-dollar-free-gift/512163.

Padilla, Thomas. 2019. “Responsible Operations: Data Science, Machine Learning, and AI in Libraries.” Dublin, OH: OCLC. https://www.oclc.org/content/dam/research/publications/2019/oclcresearch-responsible-operations-data-science-machine-learning-ai.pdf.

Singpass. 2023a. “Discover What You Can Do With Singpass.” https://www.singpass.gov.sg/main/individuals/.

Singpass. 2023b. “User Journey Template for Integrating with MyInfo.” https://public.cloud.myinfo.gov.sg/dpp/frontend/assets/api-lib/myinfo/downloads/user-journey-template.pptx.

Stripling, Gwendolyn. 2023. “Introduction to Generative AI.” *Google Cloud Tech*. AI and Machine Learning with Google Cloud. Video. 22.07. https://www.youtube.com/watch?v=G2fqAlgmoPo.

Vuow. n.d. “Poem Booth: An AI Poetry Machine That Turns Portraits into Poems.” https://www.vouw.com/poem-booth.