# Leveraging Retrieval-Augmented Generation in Local Library Systems: The BiblioGPT Prototype

**Patit Paban Santra  Rupsikha Kashyap  Anima Mahata**
**Durga Sankar Rath  Md. Ajimuddin Sk**

**Abstract**

*In today's digital age, library users often seek concept-oriented information that keyword searching cannot easily pro-vide, particularly when dealing with complex classification schemes such as the Dewey Decimal Classification (DDC) or the Library of Congress Classification (LCC). This work introduces BiblioGPT, a locally deployed conversational search application designed to fill the gap by providing natural language queries over structured library knowledge. The architecture combines the open-source Mistral language model with a Retrieval-Augmented Generation (RAG) pipeline through the WARC-GPT framework, which ingests and semantically processes WARC (Web ARChive) files as searchable content utilizing vector embedding (Chroma) and Groq-based inference. An easy-to-use interface enables smooth interaction, pulling contextually appropriate and accurate responses. Two case studies, a theoretical and a prac-tical one, demonstrate the prototype's capability to correctly interpret and answer questions, including assigning the proper DDC numbers to book titles. Although trained on a smaller dataset than the popular cloud models, BiblioGPT preserved stable performance while protecting user privacy by being deployed locally. The results confirm BiblioGPT's promise as a privacy-protecting, scalable solution that reinvents library system interaction, transforming from inflexi-ble keyword searching to flexible, smarter, and natural language-supported information retrieval. This paper describes a visionary strategy for digital library services, setting BiblioGPT as a model for future domain-specific AI-based li-brary software.*

**Keywords:** BiblioGPT, Dewey Decimal Classification, Generative Pretrained Transformer, Large Language Model, Library of Congress Classification, Mistral, Ollama, Retrieval Augmented Generation, Web ARChive

## 1. Introduction

Libraries in today's digital world are evolving to serve users whose information needs are increasingly concept-oriented and complex (Dicheva & Dichev, 2004). Keyword-based search systems often fail to yield effective results, particularly when users seek professional or subject-oriented materials that require an



**Corresponding Author**
Patit Paban Santra
Email: ppsantra95@gmail.com

understanding of structured classification systems, such as the Dewey Decimal Classification (DDC) (Satija, 2013) or the Library of Congress Classification (LCC) (Chan et al., 2016). These powerful systems, however, can be challenging to use without prior experience with their organization. To address this issue, this work presents BiblioGPT, a locally installed conversational search tool designed to facilitate the use of natural language in searching library classification data. In contrast to cloud-based AI models, BiblioGPT is constructed from a local system. It employs WARC-GPT to semantically index library content archived on the web and utilizes open-source language models, such as Mistral, for secure, intelligent processing on local servers. The system utilizes Retrieval-Augmented Generation (RAG) to retrieve pertinent data from its local archive and generate precision-crafted, context-sensitive responses. By allowing users to pose natural language questions and providing them with relevant answers based on real library knowledge, BiblioGPT redefines conventional searching into a friendlier, smarter, and more privacy-aware experience. This essay elaborates on the implementation of BiblioGPT and its features, providing a step forward in redesigning how library patrons interact with classification systems and search interfaces.

In this paper, we describe the design and implementation of an instance of the BiblioGPT prototype. We discuss how WARC-GPT is used to consolidate disparate content from the Web Archive into a meaningful, searchable knowledge base, and how a locally hosted Mistral LLM integrates with interactive natural language content that searches and accesses points of existing knowledge related to classification schemes, reliability, trust, and more. This encapsulates the essence of the mechanism. RAG will retrieve relevant knowledge from the internal private knowledge base (in this instance, internal WARC files), rather than generating responses solely from its training data, and then use that information to ground its response. This method has been shown to improve accuracy and reduce hallucinations.

BiblioGPT represents the beginning of a new paradigm for the future of library information retrieval, one that is innovative and in line with the current climate of privacy. It presents an idea of allowing access to specialist knowledge via sensitive, complex queries in a way that is more familiar and less technically based, thereby enhancing knowledge access.

### 1.1 History of RAG

Retrieval Augmented Generation (RAG) is an important new development in the field of Large Language Models (LLMs) to address some of the limitations of purely generative models, particularly “hallucinations” (generation of incorrect or nonsensical information) and their failure to generate information that is outside of what was captured when they were last trained.

Before RAG, LLMs were expected to create messages based on the knowledge they had memorized during their last periodic “knowledge dump.” Purely generative LLMs could not interact with any more recent, domain-specific, real-time, or proprietary information relevant to a business. The premise of RAG, established by Lewis et al. (2021), and their introduction of the “Retrieval-Augmented Generation” (RAG) is to have LLMs with IR capabilities to dynamically retrieve information from either an internal-qualified IR source (or external, using the Internet) and leverage the output from the retrieval aspect to generate helpful messages.

**Table 1: The historical trajectory of RAG (Gupta et al., 2024)**

| Phase | Period | Key Characteristics | Limitations/Challenges |
|---|---|---|---|
| **Traditional Information Retrieval** | Pre-LLM Era | Keyword-based search- Use of indexing (e.g., TF-IDF, BM25) focused on document retrieval rather than generation | Lacked semantic understanding- No generative capability Ineffective for ambiguous or complex queries |
| **Early Generative LLMs** | 2018–2020 (e.g., GPT-2, early GPT-3) | Capable of fluent, human-like text generation<br>Useful in open-ended tasks | Poor factual accuracy - no citation or source tracing provided. Prone to hallucinations |
| **Rise of Embeddings & Vector Databases** | Around 2019–2021 | Semantic representation of text using embeddings- Use of vector databases for semantic search (e.g., FAISS, Pinecone). Enabled similarity-based information retrieval | Still disconnected from the generation process. Retrieval and generation are not integrated |
| **Formalization of RAG** | 2020 onwards | Combined retrieval with LLM generation- Contextual augmentation using external documents. Improved factual accuracy and traceability | Computational complexity- Still evolving in terms of grounding and real-time integration |

RAG is an especially apt implementation for web archives. WARC-GPT (Botello et al., 2024) is purpose-built to address the unique challenges of web archives. Web archives in WARC (Web ARChive) file format are crucial for digital preservation, historical investigations of the digital past, and studies of the online information landscape. However, web archives are typically very difficult to search and analyze using conventional approaches, as they are raw files stored in complex formats that contain HTTP responses, HTML, CSS, JavaScript, and other files.

### 1.2 RAG in Libraries

Libraries, guardians of enormous amounts of information, can benefit immensely from (Yang & Zhang, 2025) Retrieval-Augmented Generation (RAG). Library searches are traditionally done with a search engine based on metadata and keyword matching. This method and process can fail by not executing in response to complex and natural language queries or synthesizing semantic relationship quintessence that leaves gaps in their collections. RAG can shift this dynamic by enabling semantic search, allowing users to pose natural language questions with less emphasis on using the exact keyword(s) used for the request. RAG can provide contextualized information retrieval with synthesized content from various sources to produce directly cited answers. When searching specialized domains like legal harvesting, medical literature, and library classification systems, RAG recognizably navigates complex information with great clarity. RAG can also enhance access to born-digital and archived content by making unstructured materials more easily searchable. Also, RAG can provide external and internal knowledge management (useful for librarians carrying out different tasks) by employing meaning-based observations, building collections, finding institutional documents, or interpreting policy documents.

## 2. Review of Literature

Some important literature on this domain is reviewed for further study and to gain a better understanding of this topic.

**Table 2: Some Important Literature Reviews on the Topic**

| Research Highlights | Author (s) |
|---|---|
| Reviewed the application of RAG, covering studies that were particular to a given work and discipline. The authors aim to stimulate more research and advancement in this continually developing sector. | (Arslan et al., 2024) |
| Presented on the contribution of AI, LLM and RAG in Library & Information Science. Highlighted the role that LLM and RAG would play for future knowledge description and cataloging, highlighting the improved accessibility to information resources made possible by their use in digital spaces intended for cultural heritage explorations. | (Di Marcantonio, 2024) |
| Explored RAG architecture and applications, combining generator and retriever models to access and utilize vast external data repositories. | (Gummadi et al., 2024) |
| Examines the use of the WARC GPT tool to enhance meta-literacy skills among library professionals. | (Jana & Rout, 2024) |
| Investigate the implementation of Large Language Models (LLMs) in Indian libraries to deal with linguistic diversity and enhance information distribution. | (Kumar et al., 2024) |
| Presented the main features of open source LLMs, including model design, training, and pre-training, including data gathering and pre-processing. | (Kukreja et al., 2024) |
| Presented a prototype and demonstrated that libraries can develop a low-cost conversational search system using open-source software tools and large language models (LLMs) through a Retrieval-Augmented Generation (RAG) framework. | (Mazumder & Mukhopadhyay, 2024) |

## 3. Objectives

The primary objectives of this paper are discussed below:

- To design a conversational search system, called "BiblioGPT," for a library using an open-source Large Language Model (LLM) and a Retrieval-Augmented Generation (RAG) pipeline.
- To design and build a conversational search system by selecting and putting together open-source LLM (Mistral) and RAG framework (WARC-GPT) that are appropriate for the application.
- To collect, curate, and ingest WARC (Web ARChive) files into the RAG framework, which will convert them into LLM-digestible chunks, and to facilitate the evaluation of the performance of building a conversational search system.

## 4. Methodology

This study aims to design and develop a private GPT-powered search interface tailored for the library classification domain, utilizing open-source large language models (LLMs) and a Retrieval-Augmented Generation (RAG) framework. To ensure the effective execution of this research, a two-part methodological approach has been adopted.

### 4.1 System Configuration

**Table 3: System Configuration for the Local GPT Model (Mukhopadhyay, 2025)**

| Component | Tool/Framework | Purpose |
|---|---|---|
| **Operating System** | Ubuntu 22.04 LTS | Stable Linux environment for deployment |
| **Python** | Python 3.11.9 | Language for backend scripting and tool integration |
| **LLM Deployment** | Ollama | Local hosting of Mistral and other LLMs |
| **LLM in Use** | Mistral (latest) | Language model for semantic search and question-answering |
| **Search Method** | RAG (Retrieval Augmented Generation) | Combines vector-based retrieval and generation |
| **Data Format** | WARC-GPT | Enables training and search from WARC (Web ARChive) files |
| **Vector Store** | Chroma | Store and retrieve document embeddings efficiently |
| **Hardware Acceleration** | Groq (LPU) | Ultra-fast inference for LLMs |

### 4.2 Data Ingestion and Retrieval

The second part of this methodology outlines the steps to ingest and process WARC data into a conversational AI system, biblioGPT, built on top of WARC-GPT using the Mistral LLM. The purpose is to create an intelligent search assistant that understands and responds to user queries based on the library classification domain.

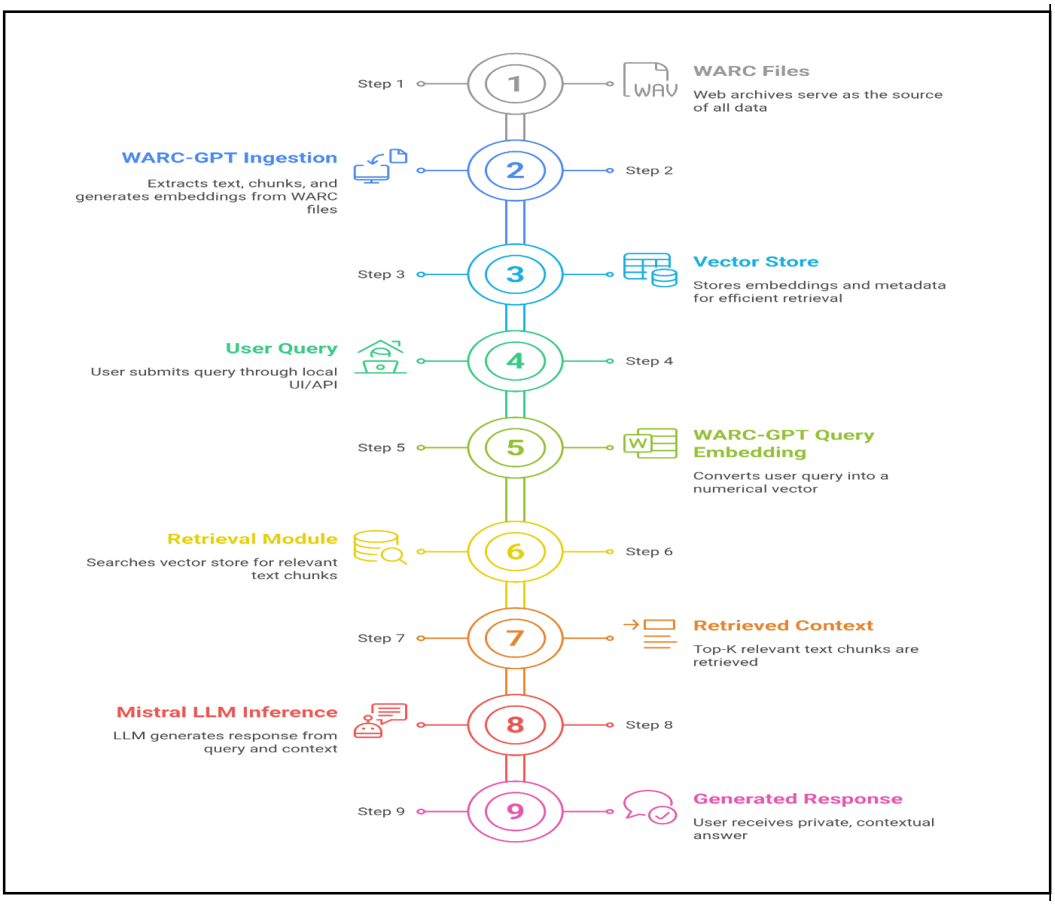


**Figure 1: WARC-GPT Workflow**

### 4.2.1 Step 1, Loading WARC Files

The journey begins with Web ARChive (WARC) files, a common method for storing web content that has been captured, including HTML, CSS, images, and JavaScript. It is the complete source of private knowledge. Because the system, called PrivateGPT, processes the data locally, files never leave your computer, ensuring complete control over the data.

### 4.2.2 Step 2, Processing and Embedding

This is the heart of the data processing. The system first extracts clean, coherent text out of the captured WARC files. Then, the text is partitioned into smaller, single-context segments, with some overlap, to provide context. Then, the original text segments are processed one-at-a-time by an embedding model (intfloat/e5-large-v2), which converts each text segment into a numeric vector/embedding. Embedding is important because it conveys the semantic meaning of the text, thereby preparing it for the next step.

#### 4.2.3 Step 3, Data Storage in a Vector Database

The embeddings are placed in a vector database (Chroma). The database is specifically designed to store embeddings and search for them in high-dimensional space. Within the database, the embedding is stored along with a pointer to the text chunk from which it originated. The primary function of the database is to perform a similarity search, enabling it to quickly retrieve from its indexed knowledge base the embeddings that are semantically related to the input query.

#### 4.2.4 Step 4, Accepting the User Query

The user interacts with the system through either a command-line interface or a custom application built using a Groq API. It serves as the gateway for all user inquiries, as all questions are processed entirely locally and not routed to any external services.

#### 4.2.5 Step 5, Development of a Query Embedding

Once the model receives a user's inquiry, WARC-GPT will transform the inquiry into a numerical vector using the same embedding model that has processed the source data. It will convert the natural language inquiry into an embedding to compare it with the text chunks stored in the vector database.

#### 4.2.6 Step 6, Retrieving Relevant Chunks

This module serves as a crucial bridge between the user's query and the knowledge base stored in the private web archive. This module serves as a bridge that utilizes query embedding to conduct a semantic similarity search within the vector database, finding the top-K most relevant text chunks in the private web archive based on their semantic similarity with the original query embedding, rather than just keywords.

#### 4.2.7 Step 7, Providing context to the LLM

The chunks retrieved from the vector database that are the most similar will now be used to provide factual context for the Large Language Model. By supplying this factual and relevant information, it also prevents the LLM from generating irrelevant and incorrect ("hallucinated") information, resulting in a more accurate and substantiated overall output.

#### 4.2.8 Step 8, Producing the Final Answer

A local Mistral LLM (Mistral: latest) combines the user's original question with the retrieved context into a single, unified prompt. As this LLM is running locally, it maintains the privacy of the process. This LLM simply processes the integrated data and generates a final answer that is coherent and potentially accurate.

#### 4.2.9 Step 9, Furnishing the Answer

This final step represents the end of the entire process, also known as the Retrieval-Augmented Generation (RAG) pipeline. With the refined, private, and contextualized information, the Mistral LLM produces a final,

factually accurate answer to the original user question, representative of the system's ability to efficiently retrieve information and generate an accurate response.

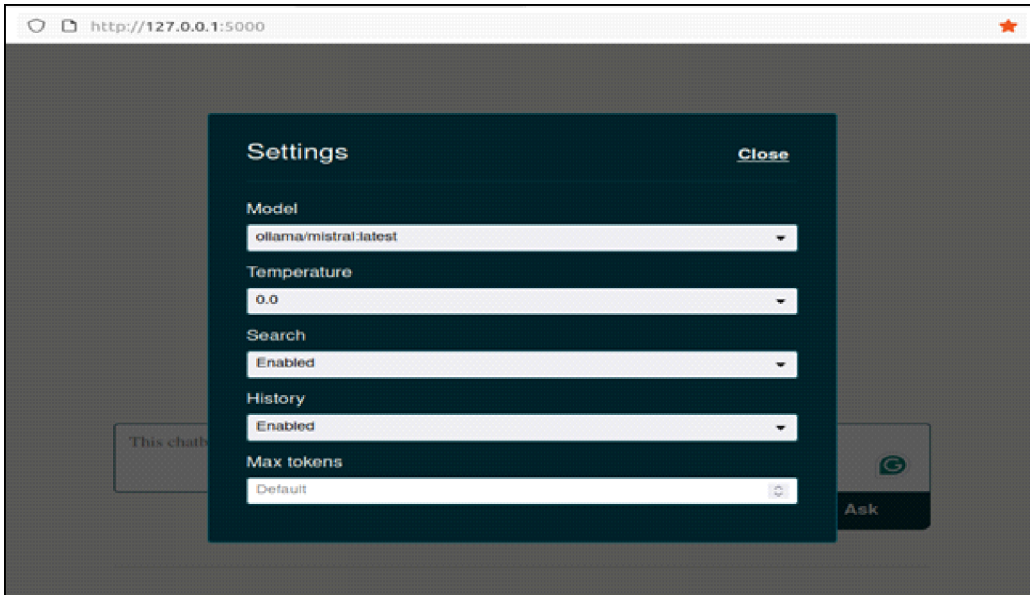


**Figure 2: LLM Model Setting**

In this WARC-GPT setting window, we can change our background LLMs. In this study, we have deployed Ollama for local hosting of LLMs and used Mistral (latest) language model for semantic search and question answering.

## 5. Results/Findings

After successfully setting up the required tools and ingesting WARC data into the RAG framework, it was operating smoothly. For a practical demonstration, two case studies are presented below.

**5.1 CS1**, Theoretical Question: To the question "What is a book classification?", this locally utilized GPT model has provided the following response, which is given below in the form of Case Study 1. Although the response is quite similar to those given by other popular GPT models, such as ChatGPT or Gemini, differences can still be identified. These differences are caused by the relatively small amount of WARC data employed to train the local model as opposed to the large sets utilized in publicly available models. Consequently, differences can be observed in areas such as response time, response structure, and the depth of information provided.


**Figure 3: Response based on General Query**

**5.2 CS2**, Practical Question: To the question "Make Dewey Decimal Classification (DDC) number of the book titled 'Foundation of Library and Information Science'", the locally deployed GPT model replied as follows, and we are utilizing the same as Case Study 2. With the help of the provided WARC data, the model accurately responded to the question, as with common GPT models. It first interpreted the question and then categorized the book title following the DDC system. According to the third summary of DDC, the book falls within the range 025–029, and is particularly recommended for classification under 025.3, which applies to library organization and administration. This demonstrates that the model can provide accurate responses. However, if additional freely available and pertinent data from the Internet are incorporated into the local model, the responses can be even more precise and comprehensive.

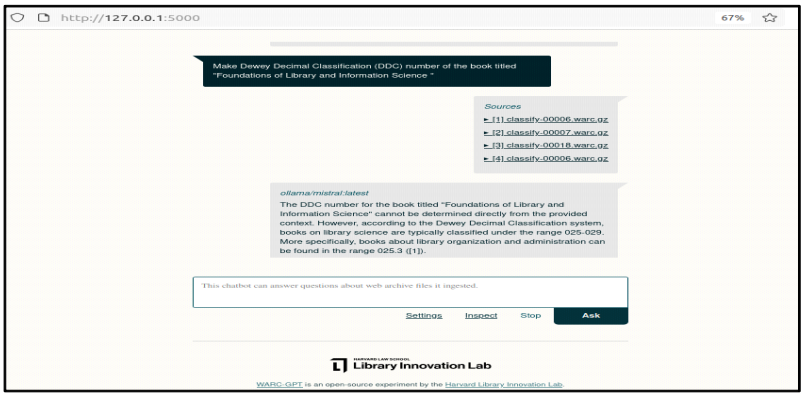


**Figure 4: Response based on Practical Query**

**5.3 CS3**, Practical Question: To the question "Make the Dewey Decimal Classification (DDC) number of this Bengali novel, titled 'Gora' by Rabindranath Tagore". In Bengali, "রবীন্দ্রনাথ ঠাকুরের "গোরা" শিরোনামের এই বাংলা উপন্যাসটির ডিউই দশমিক শ্রেণিবিন্যাস (ডিডিসি) সংখ্যা তৈরি করুন।". As per this query on Bengali language our locally trained GPT partially failed to find and make the appropriate answer, because of unavailable content. This BiblioGPT response on the basis of ingested content. Similarly others widely used GPTs can access number of content freely available in the web.

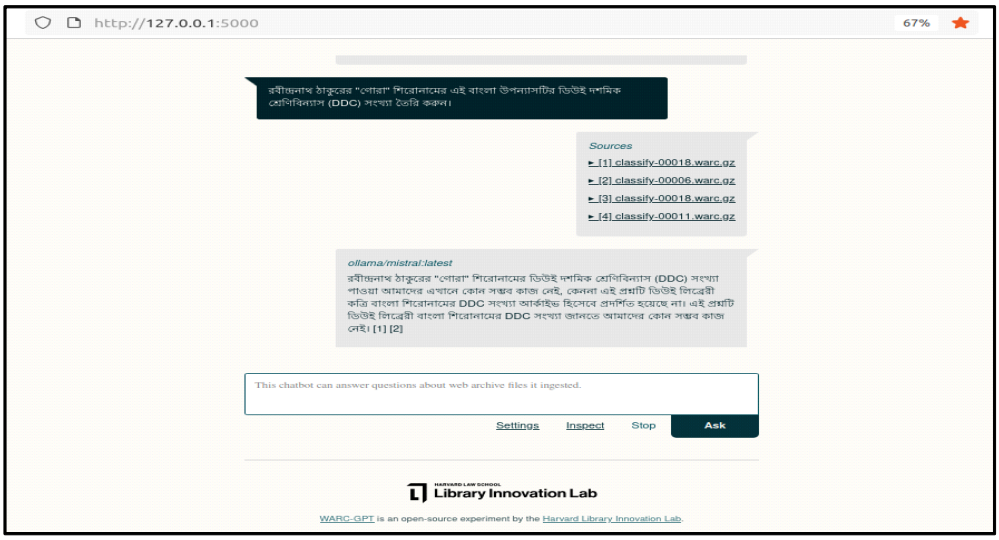


**Figure 5: Response based on a Practical Query on the Bengali Language**

**5.4 CS4**, Practical Question: To the question “Make the DDC number of this novel named ‘The God of Small Things’ by Arundhati Roy.”. Locally installed BiblioGPT responds like: “The Dewey Decimal Classification (DDC) number for the novel The God of Small Things by Arundhati Roy is 823.914. This number is found in the saved web file named classify-00002.warc.gz. According to the section on English and Old English literature, English novels are classified under 823, and modern or contemporary novels, like this one, are assigned the number 823.914. As per the above response, we may be satisfied with our BiblioGPT, as it works perfectly in some cases. We can say that it’s similar to other widely used GPT models, as indicated by the responses.

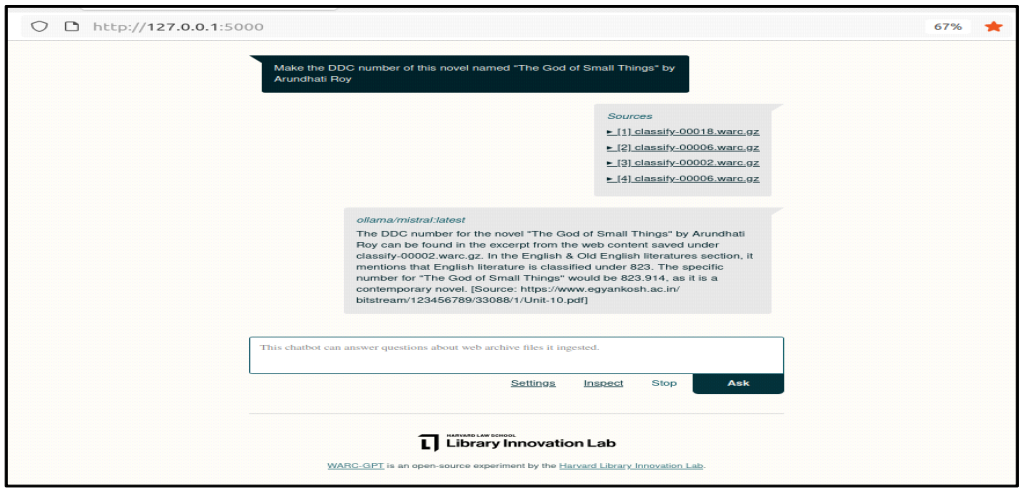


**Figure 6: Response based on Practical Query**

## 6. Comparative Study on GPTs' Responses

For a better understanding of the impact and usability of locally installed 'BiblioGPT', here we attached a comparative study of responses among two globally used GPT models, ChatGPT and Gemini, based on a practical question (CS4), "Make the DDC number of this novel named 'The God of Small Things' by Arundhati Roy.".

| Criteria | ChatGPT Response | Gemini Response | BiblioGPT (Mistral) Response |
|---|---|---|---|
| Identified DDC Number | 823.914 (with optional Cutter: R888) | 823.914 | 823.914 |
| Level of Detail | Very detailed: Includes classification steps, themes, Cutter number, and optional abridged DDC info | Concise: Provides DDC breakdown and general explanation | Minimal: Gives correct DDC and source reference |
| Classification Breakdown | 800 (Literature) → 820 (English literature) → 823 (Fiction) → 823.914 (Post-1945 English fiction) | Same breakdown as ChatGPT, explained briefly | Mentions 823.914 as a classification under English literature |
| Author-Specific Cutter Number | Provided: R888 (for Roy, optional) | Not mentioned | Not mentioned |
| Contextual Justification | Mentions themes, setting, period, and publication date to justify classification | Notes: it is post-1945 literature and specific to the author | States it is a contemporary novel, from stored web data |
| Reference or Source Used | General DDC logic + optional local practice | Standard DDC logic | Refers to "classify-00002.warc.gz" file for DDC info |
| Usefulness for Cataloging | Very helpful, practical for library use | Useful but lacks detail on the application | Minimal explanation; requires external lookup |
| Response Depth | High (step-by-step classification logic with notes) | Medium (explained classification hierarchy) | Low (just gives final number with vague reference) |

## 7. Conclusion

The deployment of BiblioGPT marks a significant leap forward in the utilization of conversational AI by libraries for enhanced information discovery and improved end-user engagement. By integrating open-source large language models, such as Mistral, with Retrieval-Augmented Generation (RAG) and the WARC-GPT architecture, BiblioGPT demonstrates the ability of libraries to create secure, private, and intelligent search systems optimized for domain-specific knowledge, including classification schemes. This environment provides users with natural language interfaces, rendering library materials, especially those associated with intricate classification schemes, such as DDC and LCC, more manageable and understandable. The inclusion of WARC files as a basis for knowledge compels greater emphasis on digital preservation and presents an exciting means to recover and integrate archived materials. The case studies included in this

paper demonstrate the model's ability to address both theoretical and practical questions with impressive accuracy, even when trained on a small dataset. It demonstrates the promise of locally executed GPT systems to imitate and, in some settings, surpass the capabilities of large-scale cloud-based models without sacrificing user privacy.

Additionally, the system's modular architecture can be scaled up and enhanced later on, for example, by adding more domain-specific data or supporting additional languages. Essentially, BiblioGPT is not merely a prototype but a blueprint for the future of library information seeking, where users are no longer limited by keyword search but are enabled to interact with library knowledge through natural language and interactive interfaces. As libraries evolve by digitizing and preparing for emerging technologies, platforms such as BiblioGPT demonstrate a significant leap toward more intelligent, convenient, and privacy-driven library services.

This paper primarily focuses on evaluating the performance of BiblioGPT, a conversational AI tool designed to facilitate information retrieval in the specialized area of library classification systems. Through case studies, the paper demonstrates BiblioGPT's effectiveness in understanding and categorizing domain-specific queries, highlighting its value as a privacy-driven, scalable solution for enhancing access to complex bibliographic information.

## 8. References

1. Arslan, M., Ghanem, H., Munawar, S., & Cruz, C. (2024). A Survey on RAG with LLMs. Procedia Computer Science, 246, 3781–3790. https://doi.org/10.1016/j.procs.2024.09.178

2. Bevara, R., Lund, B. D., Nishith Reddy Mannuru, Karedla, S. P., Mohammed, Y., Kolapudi, S. T., & Aashrith Mannuru. (2025). Prospects of Retrieval Augmented Generation (RAG) for Academic Library Search and Re-trieval. https://doi.org/10.13140/RG.2.2.33848.25607

3. Botello, J. G., Frew, L., Padilla, J. J., & Weigle, M. C. (2024). Exploring Large Language Models for Analyzing Changes in Web Archive Content: A Retrieval-Augmented Generation Approach. 2024 IEEE International Conference on Big Data (BigData), 2410–2418. https://doi.org/10.1109/BigData62323.2024.10826069

4. Chan, L. M., Intner, S. S., & Weihs, J. (2016). Guide to the Library of Congress Classification. Bloomsbury Publishing USA.

5. Di Marcantonio, G. (2024). Artificial Intelligence, LLM and RAG supporting Library and Information Sci-ence: Challenges and opportunities. In NAVIGARE LA COMPLESSITÀ (pp. 24–28). Consortium GARR. https://doi.org/10.26314/GARR-Conf24-proceedings-04

6. Dicheva, D., & Dichev, C. (2004). A Framework for Concept-Based Digital Course Libraries. Journal of Inter-active Learning Research, 15(4), 347–364.

7. Gummadi, V., Udayaraju, P., Sarabu, V. R., Ravulu, C., Seelam, D. R., & Venkataramana, S. (2024). Enhancing Communication and Data Transmission Security in RAG Using Large Language Models. 2024 4th Interna-tional Conference on Sustainable Expert Systems (ICSES), 612–617. https://doi.org/10.1109/ICSES63445.2024.10763024

8. Gupta, S., Ranjan, R., & Singh, S. N. (2024). A Comprehensive Survey of Retrieval-Augmented Generation (RAG): Evolution, Current Landscape and Future Directions (No. arXiv:2410.12837). arXiv. https://doi.org/10.48550/arXiv.2410.12837

9. Islam, M. M., Jha, A. K., Mahmoud, M., Akhmetov, I., & Nadi, S. (2025). Using LLMs for Library Migration (No. arXiv:2504.13272). arXiv. https://doi.org/10.48550/arXiv.2504.13272

10. Jana, Mr. A., & Rout, R. (2024). From Archives to Insights: Using WARC-GPT to Advance Library Metalitera-cy (pp. 216–223).

11. Kukreja, S., Kumar, T., Purohit, A., Dasgupta, A., & Guha, D. (2024). A Literature Survey on Open Source Large Language Models. Proceedings of the 2024 7th International Conference on Computers in Management and Business, 133–143. https://doi.org/10.1145/3647782.3647803

12. Kumar, M., Yadav, A., & Sinha, A. (2024). Harnessing Large Language Models for Enhanced Library Services: A Strategic Framework for Indian Libraries. International Journal of Engineering Research And, 13. https://doi.org/10.17577/IJERTV13IS050234

13. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. (2021). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks (No. arXiv:2005.11401). arXiv. https://doi.org/10.48550/arXiv.2005.11401

14. Mazumder, J., & Mukhopadhyay, P. (2024). Designing Question-Answer Based Search System in Libraries: Application of Open Source Retrieval Augmented Generation (RAG) Pipeline. Journal of Information and Knowledge, 255–260. https://doi.org/10.17821/srels/2024/v61i5/171583

15. Mukhopadhyay, P. (2025). Designing Conversational Search for Libraries: Retrieval Augmented Generation through Open Source Large Language Models. DESIDOC Journal of Library & Information Technology, 45(2), Article 2. https://doi.org/10.14429/djlit.20206

16. Packowski, S., Halilovic, I., Schlotfeldt, J., & Smith, T. (2025). Optimizing and Evaluating Enterprise Retriev-al-Augmented Generation (RAG): A Content Design Perspective. Proceedings of the 2024 8th International Conference on Advances in Artificial Intelligence, 162–167. https://doi.org/10.1145/3704137.3704181

17. Randolph, M. C., Michaleas, A., & Ricke, D. O. (2024). LLMs for Closed-Library Multi-Document Query, Test Generation, and Evaluation. In Review. https://doi.org/10.21203/rs.3.rs-5038817/v1

18. Satija, M. P. (2013). The Theory and Practice of the Dewey Decimal Classification System. Elsevier.

19. Sivasothy, S., Barnett, S., Kurniawan, S., Rasool, Z., & Vasa, R. (2024). RAGProbe: An Automated Approach for Evaluating RAG Applications (No. arXiv:2409.19019). arXiv. https://doi.org/10.48550/arXiv.2409.19019

20. Yang, L., & Zhang, Z. (2025). Using RAG, LLMs, and LangChain to Create a Query Application on Open Re-positories: Demo Application on DSpace-7. Proceedings of the 24th ACM/IEEE Joint Conference on Digital Libraries, 1–2. https://doi.org/10.1145/3677389.3702488

**About Authors**

**Patit Paban Santra,** Research Scholar, Department of Library & Information Science, Vidyasagar University, Midnapore, West Bengal.
Email: ppsantra95@gmail.com
ORCID: https://orcid.org/0000-0002-8916-7290

**Rupsikha Kashyap,** Research Scholar, Department of Library & Information Science, Vidyasagar University, Midnapore, West Bengal.
Email: rupsikhakashyap043@gmail.com
ORCID: https://orcid.org/0000-0001-6426-0010

**Anima Mahata,** Research Scholar, Department of Library & Information Science, Vidyasagar University, Midnapore, West Bengal.
Email: anima18391@gmail.com
ORCID: https://orcid.org/0009-0001-9385-1861

**Prof. Durga Sankar Rath,** Professor, Department of Library & Information Science, Institution Affiliation with Address: Vidyasagar University, Midnapore, West Bengal.
Email: dsrath@mail.vidyasagar.ac.in
ORCID: https://orcid.org/0000-0002-0951-3908

**Dr. Md. Ajimuddin SK,** Assistant Professor, Department of Library & Information Science, Vidyasagar University, Midnapore, West Bengal.
Email: ajimsk10@gmail.com
ORCID: https://orcid.org/0009-0004-0304-2334