Artificial Intelligence based Knowledge Organizer for Diverse Data Formats

Devansh Parapalli, Kaustubh Warade
Department of Computer Science
Government College of Engineering
Nagpur, Maharashtra
✉ dsparapalli@gcoen.ac.in
✉ kdwarade@gcoen.ac.in

Aditya Deshmukh, Yashasvi Thool
Department of Computer Science
Government College of Engineering
Nagpur, Maharashtra
✉ asdeshmukh@gcoen.ac.in
✉ ybthool@gcoen.ac.in

ABSTRACT

This system addresses knowledge organization challenges by managing diverse data formats, including images, audio, and various document types like PDFs, DOCX, and PPTX. Its modular architecture ensures seamless updates, using “plugins” to connect with large language models (LLMs) and ingest heterogeneous data. Svelte enables a fast, responsive interface, FastAPI provides asynchronous capabilities, and Pinecone serves as the vector database for embedding management. Semantic search leverages embeddings generated by transformer-based models. Retrieval-augmented generation (RAG) utilizes a unified text-only intermediate format, enabling accurate representations of image content (with OCR), transcribed audio, and extracted document text. Chunking further optimizes retrieval. Applications range from organizing class notes and schedules for students to enabling research content retrieval for academics and facilitating document access for corporate users. User feedback indicates a 75% reduction in search times, with documents found within 30 seconds on average. User studies show a 95% + accuracy for retrieval. Currently a proof of concept, the system aims to evolve into a self-hostable solution, ensuring all computations occur locally for enhanced privacy in addition to support for video formats. This work represents a significant advancement in efficient, scalable knowledge management for diverse data formats.

KEYWORDS *: Knowledge management, Information retrieval, Semantic search, Natural language processing, Machine learning.*

INTRODUCTION

The exponential growth of digital information in recent years has created significant challenges for individuals and organizations in efficiently managing, accessing, and utilizing data. As the volume and diversity of information continue to increase, traditional methods of data organization and retrieval struggle to meet the demands of modern users. This challenge, commonly referred to as information overload, affects productivity and decision-making across personal, academic, and corporate domains.

Previous studies have explored various approaches to knowledge management, including semantic search, machine learning, and AI-powered tools. While these solutions have improved information retrieval to some extent, they often lack the ability to effectively integrate diverse data types or provide intuitive, user- centred access. This highlights a critical gap in the development of tools that not only organize vast amounts of information but also ensure seamless retrieval and utilization, tailored to different contexts.

To address these gaps, this paper introduces AIKO (AI-powered Knowledge Organizer), a web-based system designed to simplify the organization, access, and use of information the proposed system leverages advanced natural language processing and modular technologies to enhance data retrieval while reducing search times.

Developed as part of a final-year Bachelor of Technology project at Government College of Engineering, Nagpur, India, the proposed system offers significant applications in personal, academic, and corporate knowledge management. The aim of this study is to present the design, development, and deployment of the proposed system, demonstrating its potential to transform information handling and retrieval.

LITERATURE REVIEW

The development of knowledge management systems capable of handling diverse data formats has garnered significant attention in recent years, owing to the widely known paper "Attention is all you need" [1]. Enhancements in Large Language Models and semantic search technologies have bettered the ability to improve information retrieval further. This review summarizes relevant works, examining some approaches to information retrieval and augmentation.

Recent advancements such as AssistRAG enhance LLMs with retrieval-augmented generation (RAG) capabilities by employing intelligent information assistants [2]. The paper also describes the pitfalls and shortcomings of the earliest "Retrieve-Read" techniques, prompt-based RAG techniques and Supervised Fine-Tuning (SFT) methods. To cope with these challenges, AssistRAG proposes an Assistant-based Retrieval Augmented Generation, which integrates newer use cases like tool-use to improve upon the previous techniques. Similarly, the ZeroG knowledge engine proposes a two model system to ground retrieval in validated sources. [3]. A case study on biodiversity publications demonstrated improved search relevance when integration LLMs with structured indexing methods [4].

Efficient information retrieval relies heavily on embedding representations and vector databases. Research on nearest neighbor search (NNS) in high- dimension spaces reveals that NNS is resilient to the "curse of dimensionality" [5]. Additionally, the paper explains the irrelevance of choice of the distance function and proposes methods for further optimization of dense vector-related applications.

Furthermore, LLM-powered query generation along with NNS over the available tools allows for better retrieval statistics [6].

Challenges in ensuring factual accuracy and usability in search systems have been identified [7, 8]. While these insights are tangential to our primary focus, they highlight user-centric considerations crucial for adoption. Similarly, grounding hypothesis generation in reliable knowledge remains an ongoing challenge for all systems leveraging LLMs for information retrieval [9].

Existing research provides a robust foundation for the development of a modular, AI-powered knowledge organizer. By integrating insights from semantic search, vector database efficiency, and modularity, this system addresses the unique challenges of managing diverse data formats. While promising advancements have been made, the need for self-hosted solutions and improved user interfaces remains a future direction for exploration.

METHODOLOGY

The methodology for implementing the proposed system followed a structured Waterfall approach [10], with distinct phases enabling systematic development and validation of each subsystem. Rather than detail standard development practices, we focus on the novel architectural decisions and custom implementations that distinguish our system.

A notable deviation from standard practices was our decision to develop a custom LLM communication stack rather than utilizing LangChain. This decision was driven by the need for lower-level control and enhanced performance requirements. The custom implementation leverages WebSockets for communication, enabling efficient real-time data streaming while maintaining the system's stateless nature. The backend, developed in Python 3.12.6 [11], employs a novel plugin architecture that allows for seamless integration of new data connectors and transformation logic. This plugin architecture allows for the proposed system be very extensible while still maintaining the developer and user experience.

The models are used as-is from the respective vendors, with the system providing a unified interface for the models. The content ingested into the system is passed through a simple filter to remove unnecessary punctuation and limit the context window.

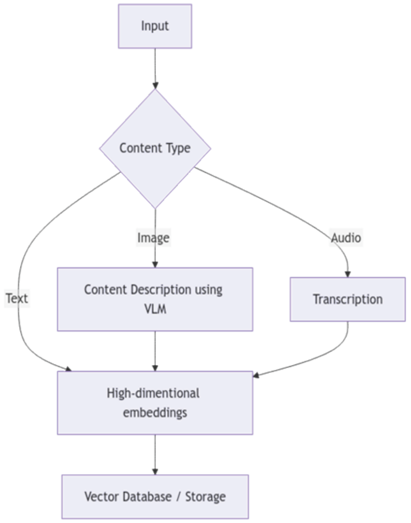

Input
Content Type
Image
Audio
Text
Content Description using VLM
Transcription
High-dimentional embeddings
Vector Database / Storage

Fig. 1: Ingest Flow

Fig. 1 shows the process of ingestion of various supported documents. We begin by separating the entity by content type, images are first passed into a vision language model such as QwenVL [12] with a custom prompt to extract as much detail as possible from the images. The audio is passed through to a model capable of extracting text from the audio sample [13]. The text is extracted from the input and then chunked and embedded using a high-dimensionality embedding model [14]. The resulting embeddings are stored in a vector database for efficient querying. The original source is saved in a storage medium and the identification number used to save the vectors, indexing with the chunk number. This maintains a two-way link between the saved source document and its embedded vectors.

The retrieval flow utilizes simple NSS to fetch the relevant vectors, and the link is used to fetch the metadata for the original document. These are presented to the user for further interaction.

Incase of retrieval-augmented-generation, the output chunks are used as part of the input prompt to a LLM with a large context window. The novel plugin and transformational logic allows for multiple LLMs to have a common interface with the proposed system. We use OpenAI's APIs as a reference and modify the data structures for use with other LLMs.

The system was built using a modular architecture, with each module responsible for its own functions, and connected to others using a strict interface. Each module was responsible for its own initialization, and is easily extensible and replaceable. Each part of the system can be independently scaled as per usage.

The proposed system was built as a proof- of-concept system to test the feasibility of a personalized knowledge management system with AI-powered features. The system was deployed as a technical preview, with user feedback collected. The highlights of user feedback are presented in the following section.

Experimental Setup: In this paper, we evaluate the application of such a proposed system for efficient knowledge management. We understand that such retrieval tasks have subjective performance metrics and utilize user feedback to gauge the effectiveness of the system.

Participants were recruited through voluntary self- selection from the undergraduate student population of Government College of Engineering Nagpur, specifically targeting students within the Computer Science & Engineering branch. All students within the branch were invited to participate in the study, with participation being entirely voluntary and free from any academic or institutional pressure. The participant pool consisted of college students, aged 20 to 24, with a balanced representation of gender. The experimental period consisted of the penultimate semester, wherein students are asked to create a "Mega-Project" as part of the academic requirements. All participants were anonymized for the survey to prevent any bias during the evaluation of subjective questions.

Participants were tasked to keep note of the references, documents, images, and any other relevant pieces of info saved using their usual methodology for creation of the project synopsis. This comprises the baseline experience.

Participants were asked to use the proposed system during creation of the final report for the "Mega- Project". The feedback was gathered through a form with both general and specific questions.

The survey includes both qualitative and quantitative questions, with participants asked to rate their experience on a scale of 1 to 5, with 1 being the lowest and 5 being the highest. Participants were asked to rate the system's various features along with their overall satisfaction with the system. The survey also included open-ended subjective questions aimed at gathering detailed feedback on the difficulties faced, most useful feature, and suggestions towards improvements.

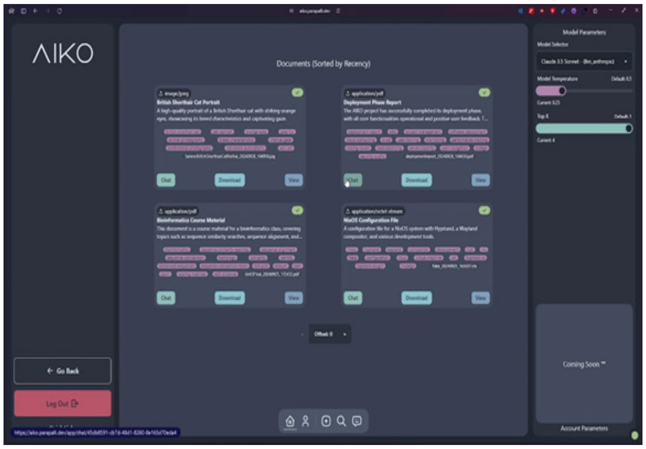
RESULTS AND DISCUSSION

Main Result: The user interface for the system was built using Svelte [15], TypeScript [16] and TailwindCSS [17]. The backend was built using Python [11], FastAPI [18] and the APIs for the various LLMs.

Fig. 2 gives a overlook of the user interface for the proposed system. The user interface is completely decoupled from the backend and only uses APIs to communicate with it.

The backend contains the logic for the ingest, retrieval and retrieval-augmented-generation flows. It connects the various components into one accessible endpoint. The plugins are active on the backend and dictate the LLMs as well as the files that can be understood by the system.

Current limitations include the inability to handle video formats, and documents with embedded graphics, such as charts and graphs. The system is also unable to handle scanned documents as PDF files. The intermediate representation being a text-only format limits the amount, quality and medium of information that can be processed through the system at any given time-step.



A high-quality portrait of a British Shorthair cat with striking orange eyes, showcasing its breed characteristics and captivating gaze.

The AIKO project has successfully completed its deployment phase, with all core functionalities operational and positive user feedback. T...

This document is a course material for a bioinformatics class, covering topics such as sequence similarity searches, sequence alignment, and...

A configuration file for a NixOS system with Hyprland, a Wayland compositor, and various development tools.

Source: Screen capture from https://aiko.parapalli.dev

Fig. 2: Dashboard User Interface

Feedback Results: The feedback gathered after the use of the proposed system gave a few interesting observations.

The proposed system was instrumental in efficiently locating and retrieving the relevant pieces of content from the user's personalized corpus. The addition of retrieval-augmented-generation allowed for chat based interaction with the documents, allowing for various tasks that would have required the user to first obtain the document and then load it into a platform which supports the media and then proceed to accomplish the task. Tasks such as Query Answering, Summarization were able to be accomplished in under a mean time of 30 seconds while in the normal flow it would have taken the user a mean time of 2 minute 30 seconds.

Manual testing with Wikipedia documents of various topics and lengths demonstrated an accuracy of 97% on first query and 99% after refining the query.

Quantitative analysis of user feedback revealed strong performance metrics, with search accuracy and core features averaging 4.83/5.0. System response times were efficient, with 83.3% of users receiving results in under 3 seconds. The system demonstrated substantial time savings, with 33.3% of users saving over 2 minutes per search operation. User satisfaction was notably high, reflected in a Net Promoter Score of 9.50/10.0 and an overall experience rating of 4.67/5.0.

CONCLUSIONS

The proposed system demonstrates significant potential in revolutionizing document interaction and information retrieval in academic settings. The consistently high satisfaction ratings across users of varying technical proficiency suggests that AI-powered document management systems can effectively bridge the gap between complex information processing and user-friendly interfaces. The notable time savings reported by users (over 2 minutes per search) points to substantial productivity gains possible through AI-augmented document interaction.

Future directions for this research include several promising avenues. First, expanding support to include video formats would broaden the system's multimedia capabilities. Second, developing a fully self-hostable version would address privacy concerns and enable offline usage, particularly important for sensitive corporate or academic environments. Third, integrating more sophisticated document comparison and version control features could help track the evolution of knowledge over time. Finally, developing domain-specific plugins or data connectors for specialized fields like legal, medical, or technical documentation could enhance the system's utility in professional settings.

ACKNOWLEDGMENT

We are very thankful to our project guide, Dr. Devchand J. Chaudhari, whose support and guidance were instrumental in the successful completion of this project. We would also like to express our gratitude to the faculty and staff of the Department of Computer Science at Government College of Engineering, Nagpur, for their assistance and encouragement throughout the project. We also extend a special thanks to Dr. Rewatkumar P. Borkar, Principal, Government College of Engineering, Nagpur. Finally, we would like to thank participants of the user experience and volunteers who filled out the feedback forms.

REFERENCES

1. A. Vaswani et al. (2017) "Attention is All you Need" arXiv (Cornell University) Vol. 30 Pp 5998– 6008

2. Y. Zhou, Z. Liu, Z. Dou (2024) "AssistRAG: Boosting the Potential of Large Language Models with an Intelligent Information Assistant"

3. A. Sharma, S. E. John, F. R. Nikroo, K. Bhatt, M. Zambre, A. Wikhe (2024) “Mitigating Hallucination with ZeroG: An Advanced Knowledge Management Engine”

4. V. K. Kommineni, B. König-Ries, S. Samuel (2024) “Harnessing multiple LLMs for Information Retrieval: A case study on Deep Learning methodologies in Biodiversity publications”

5. Z. Chen, R. Zhang, X. Zhao, X. Cheng, X. Zhou (2024) “Exploring the meaningfulness of Nearest Neighbor Search in High-Dimensional Space” arXiv (Cornell University)

6. M. Kachuee, S. Ahuja, V. Kumar, P. Xu, X. Liu (2024) “Improving tool retrieval by leveraging large language models for query generation” arXiv (Cornell University)

7. P. N. Venkit, P. Laban, Y. Zhou, Y. Mao, C.- S. Wu (2024) “Search engines in an AI era: The false promise of factual and verifiable Source-Cited responses” arXiv (Cornell University)

8. D. Dukić, M. Petričević, S. Ćurković, J. Šnajder (2024) “TakeLab Retriever: AI-Driven Search Engine for Articles from Croatian News Outlets” arXiv (Cornell University)

9. G. Xiong, E. Xie, A. H. Shariatmadari, S. Guo, S. Bekiranov, A. Zhang (2024) “Improving Scientific Hypothesis Generation with Knowledge Grounded Large Language Models”

10. K. Petersen, C. Wohlin, D. Baca, The waterfall model in Large-Scale development, 2009

11. Welcome to Python.org. Accessed: July 14, 2024. [Online]. Available: https://python.org/

12. J. Bai et al. (2023) “Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond” arXiv preprint arXiv:2308.12966

13. A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, I. Sutskever (2022) “Robust Speech Recognition via Large-Scale Weak Supervision”

14. NV-Embed: Improved Techniques for Training LLMs as Generalist Embedding Models. Accessed: August 23, 2024. [Online]. Available: https://arxiv. org/abs/2405.17428

15. SvelteKit. Accessed: July 14, 2024. [Online]. Available: https://kit.svelte.dev/

16. Gavin Bierman, Mart\in Abadi, Mads Torgersen “Understanding typescript” in “European Conference on Object-Oriented Programming” organized by during 2014

17. Tailwind CSS - Rapidly build modern websites without ever leaving your HTML. Accessed: July 14, 2024. [Online]. Available: https://tailwindcss.com/

18. M. Lathkar, High-Performance Web Apps with FastAPI, 2023.