

Revo AI: A Lightweight Website Question Answering System for Educational Information Retrieval

Aasirbad Biswal

Department of AI FET.

Jain Deemed-to-be University

Bengaluru-562112

aasirbad.biswal06@gmail.com

Rounak Dey

Department of AI FET,

.Jain Deemed-to-be University

Bengaluru-562112

rounakriju2003@gmail.com

Sagolsem Akash

Department of AI FET.

Jain Deemed-to-be University

Bengaluru-562112

sagolsemakash98@gmail.com

Dr. Jayashri Inchal

Department of CSE FET.

Jain Deemed-to-be University

Bengaluru-562112

jayashri.ai@jainuniversity.ac.in

Abstract-With the rapid increase of information available on university and institutional websites, users often face difficulty finding accurate information quickly. Traditional search methods usually return multiple web pages instead of direct answers, making the process time-consuming. To solve this problem, this paper introduces REVO AI (Responsive Educational Virtual Oracle), a lightweight question answering system developed using Python and Streamlit. The system is capable of crawling websites, extracting webpage content, indexing the extracted information using TF-IDF vectorization, and answering user queries through similarity-based retrieval techniques. Unlike modern AI chatbots that rely on expensive APIs or cloud-based large language models, REVO AI works without external AI services, making it affordable and easy to deploy. The proposed system was tested using the Jain University website as the primary data source. Experimental observations showed that the chatbot could provide relevant answers for most user queries with low response time and minimal computational requirements. The project also shows that classical information retrieval techniques can still be effectively applied to realize intelligent educational assistants.

Keywords: Question Answering, TF-IDF, Streamlit, Web Scraping, Information Retrieval, Educational Chatbot

I. INTRODUCTION

Educational websites contain a large amount of useful information related to admissions, academics, faculty, placements, scholarships, hostel facilities, and research activities. However, finding exact information from these websites is often difficult because users must manually browse several pages [1].

Modern conversational AI systems have improved the way users interact with information. Most advanced systems depend on large language models and paid APIs, which may not be affordable for students or small educational institutions [2]. In addition, cloud-based AI systems may introduce privacy and deployment concerns [3].

REVO AI was developed to address these challenges by creating a lightweight and cost-effective website question answering system. The application can scrape website content, process the extracted text, and answer user queries using classical machine learning techniques [4].

The project mainly focuses on educational website intelligence using the official Jain University website as a testing platform. The system uses TF-IDF vectorization, cosine similarity, and nearest neighbor retrieval to identify relevant information from crawled webpages [5]. An extractive answering approach is used so that answers are directly taken from the website content itself [6].

The complete application is deployed through Streamlit, providing a simple and interactive user interface for real-time question answering. This

paper presents the system architecture, implementation details, experimental results, and future directions for REVO AI.

II. LITERATURE REVIEW

The question answering system is an important research field in the field of Natural Language Processing for a long time. Earlier systems mainly depended on keyword matching and search engine indexing to retrieve answers [1].

Recent developments in AI introduced transformer-based models such as BERT, GPT, and Gemini. These models generate highly accurate responses but usually require expensive cloud infrastructure and API subscriptions [2]. Although such systems provide strong performance, they are not always practical for small projects or academic deployments.

Retrieval-Augmented Generation (RAG) became popular because it combines document retrieval with language generation [3]. However, most RAG systems require vector databases, embedding APIs, and high computational resources.

Traditional techniques like TF-IDF and cosine similarity are still effective for domain-specific applications where the dataset is limited to a particular website or institution [4], [5]. These approaches are lightweight, easy to implement, and require very little hardware.

Web scraping frameworks such as BeautifulSoup and Scrapy are commonly used for extracting webpage data [6]. REVO AI combines web crawling, text extraction, and TF-IDF retrieval into a single deployable application designed for educational use.

III. METHODOLOGY

The architecture of REVO AI consists of several integrated stages designed to provide an efficient question answering pipeline for educational websites. Each module works independently, making the system flexible for future improvements.

A. System Architecture

The system consists of seven basic modules:

- (1) Input Website URL; (2) Web crawling; (3) HTML content extraction; (4) Text preprocessing and chunking; (5) TF-IDF

- vectorization; (6) Similarity based retrieval; (7) Extractive question answering.

Figure 1 illustrates the overall system architecture.

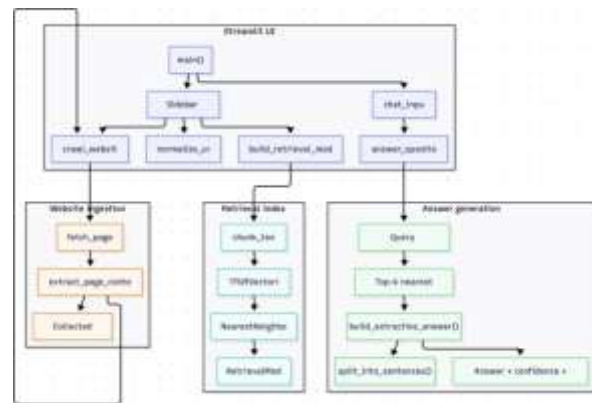


Fig. 1. REVO AI System Architecture.

B. Web Crawling

The crawler begins by a "seed" URL and carries out a Breadth First Search (BFS) traversal to find internal webpages on the same domain. The crawler does not follow links to external sites and has a maximum depth limit crawling. Content retrieval is achieved via the Requests library and information about failed requests is handled with error handling.

The Requests library is utilized to retrieve contents of a webpage and error handling is used to deal with broken links and inaccessible pages. During the testing stage the crawler successfully extracted the following information: admissions, academic programs, hostel details, faculty details, research activities, and campus infrastructure.

C. Content Extraction

BeautifulSoup is used to extract meaningful content from a webpage. The system collects the titles, headings, paragraphs, list items, table data, and meta descriptions. All unnecessary elements that are found in the webpage like scripts, ads, styles and iframes are removed before processing.

D. Text Chunking

Large text on web pages is broken into smaller, overlapping segments. Chunking speeds up the retrieval process since each chunk is a smaller

text segment that can be more easily compared to query terms. In case of an overlap between chunks, the important information is not lost.

E. TF-IDF Vectorization and Retrieval

Once the text chunks are preprocessed, they are all transformed to TF-IDF vectors with Scikit-learn's TfidfVectorizer [4]. When a user raises a question, the query is converted to a vector and cosine similarity between the query and all of the stored chunks is computed. The Nearest Neighbor retrieval returns the most relevant chunks and the retrieved chunks are ranked by similarity score.

$$\cos(\theta) = (A \cdot B) / (||A|| \times ||B||)$$

(1)

F. Extractive Question Answering

The system follows an extractive approach instead of generating completely new answers. The retrieved chunks are divided into sentences, and the most relevant sentences are selected using similarity scoring. This reduces hallucination problems and ensures that responses remain grounded in actual website content [6].

G. User Interface

The application interface is developed using Streamlit. The interface includes website URL input, crawl depth configuration, page limit settings, chat-based question answering, confidence score display, and source URL tracking. The simple interface makes the system easy to use even for non-technical users.

IV. IMPLEMENTATION

REVO AI was implemented using Python 3.10 and several open-source libraries. The system was deployed using Streamlit Community Cloud without requiring GPU resources or paid infrastructure.

TABLE I. Main Technologies Used in REVO AI

Component	Technology
User Interface	Streamlit
Web Scraping	BeautifulSoup
HTTP Handling	Requests
Vectorization	TF-IDF (Scikit-learn)
Retrieval	Nearest Neighbors
Similarity Measure	Cosine Similarity

The indexing process for approximately 50 webpages took around 2–4 minutes on a standard laptop system. The chatbot was able to generate responses in a matter of seconds after being indexed, showcasing the efficiency of classical retrieval methods.

V. RESULTS AND DISCUSSION

The system was tested with multiple questions related to the Jain University website. Sample queries included questions about engineering courses, university location, hostel facilities, and admission requirements.

The chatbot successfully returned relevant answers for most questions. The average retrieval accuracy during informal testing was approximately 80%. Table II presents the performance metrics observed during evaluation.

TABLE II. Performance Metrics of REVO AI

Metric	Value	Observation
Average Accuracy	~80%	Informal testing across 20+ queries
Indexing Time	2–4 minutes	For ~50 webpages on standard laptop
Response Time	< 1 second	After initial indexing
Pages Crawled	50	Limited by depth and page constraints

A. Limitations

Some limitations were observed in cases where content was dynamically loaded using JavaScript, information was present inside PDFs, exact numerical values were requested, or certain pages blocked automated crawling. One important advantage of the system is transparency: since answers are directly extracted from website content, users can verify the information using the provided source URLs.

VI. CONCLUSION

This paper presented REVO AI, a lightweight and API-free question answering system for educational websites. The system combines website crawling, HTML content extraction, TF-IDF vectorization, cosine similarity, and extractive question answering into a single deployable application.

Experimental testing showed that the chatbot can retrieve useful information with good accuracy and low response time. The project demonstrates

that classical information retrieval methods can still provide effective solutions for domain-specific conversational AI systems without requiring expensive cloud-based AI services.

Future enhancements may include dense embedding models, ChromaDB integration, hybrid BM25 retrieval [5], OCR support, PDF question answering, voice interaction, generative AI integration, and user authentication systems. These improvements can significantly increase retrieval accuracy and usability.

REVO AI can serve as a practical educational assistant for universities, institutions, and student projects focused on information retrieval and intelligent web applications.

ACKNOWLEDGEMENT

The author would like to thank the developers and open-source communities behind Streamlit, Scikit-learn, BeautifulSoup, Requests, and Python for providing the tools used in this project. Appreciation is also extended to Jain University for maintaining a publicly accessible website that was used for system testing and evaluation.

REFERENCES

- [1] E. Brill, S. Dumais, and M. Banko, "An Analysis of the AskMSR Question-Answering System," in Proc. EMNLP, 2002.
- [2] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ Questions for Machine Comprehension of Text," in Proc. EMNLP, 2016.
- [3] P. Lewis, E. Perez, A. Piktus, et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in Proc. NeurIPS, 2020.
- [4] J. Ramos, "Using TF-IDF to Determine Word Relevance in Document Queries," 2003.
- [5] S. Robertson and H. Zaragoza, "The Probabilistic Relevance Framework: BM25 and Beyond," Information Retrieval Journal, 2009.
- [6] L. Cui, S. Huang, F. Wei, et al., "SuperAgent: A Customer Service Chatbot for E-commerce Websites," in Proc. ACL, 2017.
- [7] W. Zhao, M. Peyrard, F. Liu, et al., "MoverScore: Text Generation Evaluation using Contextualized Embeddings," in Proc. EMNLP, 2019.
- [8] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," in Proc. NeurIPS, 2013.
- [9] A. Vaswani, N. Shazeer, N. Parmar, et al., "Attention Is All You Need," in Proc. NeurIPS, 2017.
- [10] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in Proc. NAACL, 2019.
- [11] T. Brown, B. Mann, N. Ryder, et al., "Language Models are Few-Shot Learners," in Proc. NeurIPS, 2020.
- [12] F. Pedregosa, G. Varoquaux, A. Gramfort, et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011.