

## ManaVartha AI: “A Retrieval-Augmented Generation Approach for Current Affairs Question Answering in Telugu News Articles Using Large Language Models”

K. Shankar<sup>1</sup>, S. Bhanu Sanjana<sup>2</sup>, R. Pravallika<sup>3</sup>, S. Swetha Sri<sup>4</sup>, R. Yaswanth<sup>5</sup>

<sup>1</sup> Associate Professor, Department of Computer Science and Engineering

<sup>2,3,4,5</sup> Student, Department of Computer Science and Engineering

Nadimpalli Satyanarayana Raju Institute of Technology, Visakhapatnam, Andhra Pradesh, India

Corresponding Author: <sup>1</sup> [kopanati@gmail.com](mailto:kopanati@gmail.com), <sup>2</sup> [bhanusanjanasirikonda26@gmail.com](mailto:bhanusanjanasirikonda26@gmail.com),

<sup>3</sup> [rapetipravallika@gmail.com](mailto:rapetipravallika@gmail.com), <sup>4</sup> [0210pratibha@gmail.com](mailto:0210pratibha@gmail.com), <sup>5</sup> [ravadayaswanth78@gmail.com](mailto:ravadayaswanth78@gmail.com)

### Abstract

The availability of professional and organized news content in the regional Indian languages is a major issue especially with the Telugu speaking users who are less dependent on the digital news sources that only offer the search by keyword and the content of the articles that are generally not dynamic. Such limitations make it difficult for users to obtain precise answers to specific questions related to government policies, employment notifications, or local events. We have made a system called ManaVartha AI. This system helps answer questions. It uses something called Retrieval-Augmented Generation or RAG for short. ManaVartha AI looks at news stories, in Telugu. Gives answers that are backed up by evidence.

User queries are transformed to dense embeddings at query time and a search on vectors in indexed news segments is performed by vector similarity search. The retrieved evidence is then provided to a large language model (LLM) to produce grounded and context-sensitive responses in Telugu with citations of the source to enhance transparency. The corpus of 2,500 news articles in Telugu language provides us with a total number of 15,000 indexed segments of text on which we test the proposed system. Experimental findings indicate recall of between 0.85 and 0.92 and coverage of over 90 Familie publicly -interest domains. A comparative analysis shows that the suggested method of RAG-based approach is much more successful than the conventional approach of traditional retrieval methods based on the key-word methodology in semantic relevance and completeness of answers. The findings demonstrate the practicability of the implementation of

retrieval-augmented large language model architectures to low-resource regional languages, which helps to enhance digital inclusion and reachable information retrieval by Telugu-speaking communities.

**Keywords:** Retrieval-Augmented Generation, Telugu Language Processing, Regional Language News, Question Answering Systems, Multilingual NLP, FAISS Vector Indexing, Large Language Models.

## Introduction

The rapid expansion of digital news platforms in India has increased access to information, yet this growth has not benefited all language communities equally. Telugu news websites attract millions of readers daily, but most platforms still rely on basic keyword search and static article listings. As a result, users seeking answers to specific questions—such as government notifications, policy updates, or local developments—must manually browse multiple articles, making the process time-consuming and inefficient (Fan et al., 2024).

This limitation is particularly evident among users with limited digital literacy or those residing in rural regions, where conversational or question-based interfaces could significantly improve information accessibility. While open-domain question answering systems have advanced rapidly in English, similar solutions for Indian regional languages remain underexplored (Gupta et al., 2024; Kakwani et al., 2020). Recent advances in Retrieval-Augmented Generation (RAG) have shown promise for knowledge-intensive question answering tasks by combining dense retrieval with neural text generation (Lewis et al., 2020). In a typical RAG pipeline, user queries are embedded into a semantic space, relevant document passages are retrieved using vector similarity search, and a generative language model produces answers grounded in retrieved evidence (Izcard and Grave, 2021).

However, most existing RAG-based systems and benchmarks focus on English or other high-resource languages, with limited attention given to Indian regional languages such as Telugu (Endait et al., 2025). Motivated by this gap, we propose ManaVartha AI, a Retrieval-Augmented Generation system designed specifically for real-time question answering over Telugu news content. The main contributions of this work are threefold: (1) the design of a scalable pipeline for Telugu news ingestion and semantic indexing; (2) the implementation of a RAG-based question answering framework tailored to a low-resource language; and (3) an empirical evaluation demonstrating strong recall, coverage, and practical usability on real-world Telugu news queries.

## Literature Survey

Manavartha AI is simply an AI-powered Question Answering system with Retrieval-Augmented Generation (RAG), which speaks Telugu. Fundamentally, although AI and LLMs, such as GPT or BERT, have simplified information search to an absolute, they are mostly configured to search English and other worldwide languages. Huge digital divide, therefore, to Telugu people who want to read news on their own language. The entire premise of Manavartha AI is to shatter that divide



with the release of a Telugu-first AI which is capable of actually skimming news, understanding it, and responding to queries in natural Telugu. Instead of trudging through voluminous articles, it provides you with short and precise answers- so it is much more reachable and inclusive of promoting digital change.

The rapid development of large language models has caused large-scale improvements in open-domain question answering, particularly of high-resource languages like English. By contrast, regional Indian languages, such as Telugu, are significantly underrepresented in large-scale QA systems and benchmark datasets, continuing to bring about a digital divide restricting conversational access to structured knowledge by Telugu-speaking communities.

Retrieval-Augmented Generation (RAG) has become a promising model of knowledge-based QA tasks. RAG combines the dense document retrieval with language models that are generated in a way that delivers responses based on external evidence. In a standard RAG pipeline, a document is split into sequential passes, and each pass is encoded as a dense embedding a high-dimensional dense vectors and the resulting embeddings are indexed using approximate nearest-neighbor search methods like FAISS. In inference, queries of the user are coded into the same semantic space, passages containing the relevant text are retrieved through similarity search, and a generative model is used to synthesize an answer based on context. RAG implementations that are currently in place have performed well on English benchmarks like the Natural Questions and Web Questions. IndicNLP Suite and IndicSQuAD Multilingual projects have been used to expand natural language processing to Indian languages. However, the majority of previous systems have been focusing on either fixed datasets or languages with strong resources, and have not provided as much insight into real-time question answering with dynamic Telugu news collections.

Despite extensive digital editions offered by the dominant Telugu news media, including Eenadu, Sakshi, and AndhraJyothy, all of them majorly use keyword-based search functionalities and do not offer semantically oriented question response interfaces. This inevitably leaves a gap in the ongoing research on the development of scalable RAG-based systems that address news content in Telugu in a specific manner. Based on this shortcoming, ManaVartha AI presents an architectural platform of automated news ingestion in Telugu, semantic indexing, and answer generation that is evidence-based. The proposed system aims to increase accessibility, the accuracy of information-retrieval, and digital inclusion among the Telugu -speaking communities by exploiting multilingual embedding models as well as large language models.

## Related Work

**Retrieval-Augmented Generation (RAG).** The RAG approach [Lewis *et al.*, 2020] combines a semantic retriever with a text generator to answer factual questions by grounding them in external documents. In the RAG paradigm, a user query is encoded into a dense vector, used to retrieve relevant text passages (e.g., via FAISS), and then concatenated as context for a generative model (e.g., GPT or T5). This method has achieved over 80% accuracy on large-scale English benchmarks like Natural Questions and Web Questions. Recent surveys have highlighted the rapid evolution of RAG techniques in conjunction with large language models (LLMs) (Arslan *et al.*, 2024; Gupta *et*

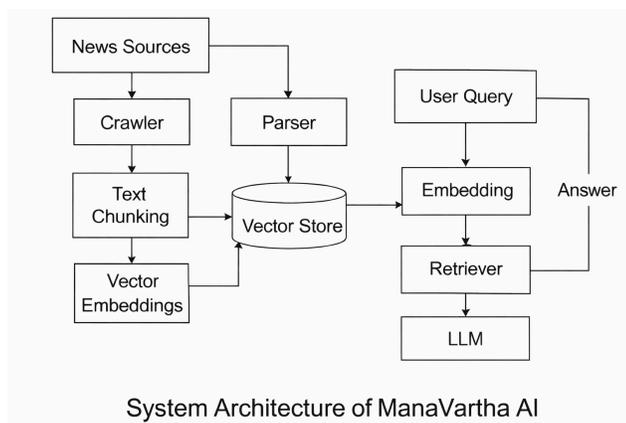


al., 2024; Fan *et al.*, 2024). However, most prior RAG research and benchmarks have focused on English or high-resource languages.

**Indic RAG/Q&A:** Some work has been done on multilingual question answering, such as the Indic NLP project and the Indic SQuAD dataset (Endait *et al.*, 2025). However, there are not many large-scale QA systems available for languages like

Telugu. Earlier projects, like RAG4News, have shown how to use dense retrieval for answering questions about Indian news, but they usually use fixed datasets or other languages like Hindi. Major Telugu news websites such as Eenadu, Sakshi, and Andhra jyothy are still trusted sources of information, but they mainly use manual search methods and don't have AI-based question and answer tools. ManaVartha AI is one of the first attempts to use a complete RAG system for real-time Telugu news content, using improvements in multilingual embedding models and large language models. But most existing RAG systems are focused on English or languages with a lot of resources, so Indian regional languages like Telugu are not getting much attention (Kakwani *et al.*, 2020; Endait *et al.*, 2025).

### System Architecture and Design



**Figure 1:** System architecture diagram of ManaVartha AI

The ManaVartha AI pipeline is organized into modular stages, as shown in Figure 1. A crawler fetches news articles from configured Telugu news sources (currently Eenadu, Sakshi, and Andhra jyothy). The scraped HTML is parsed with custom rules (using BeautifulSoup) to extract titles, dates, and main content while removing boilerplate or disclaimers. Each article is then recursively chunked into overlapping passages of ~350 tokens with ~80-token overlap, enabling fine-grained retrieval. Chunks are tagged with metadata including URL, date, sector (e.g., politics, education, health), and geographic scope. Such overlapping chunking strategies are commonly used in RAG systems (Lewis *et al.*, 2020).

These textual chunks are embedded into dense semantic vectors. The system supports either cloud APIs (e.g. Cohere multilingual model) or open-source models (e.g. sentence-transformers) for generating embeddings. The chunk vectors, along with their metadata, are stored in a CSV-backed FAISS index (IndexFlatL2) for efficient nearest-neighbor search (Xiong et al., 2021). The FAISS index is refreshed each time new articles are added (i.e., on each pipeline run) to keep the knowledge base up-to-date. At query time, the user's question is embedded and used to retrieve the top- $K$  relevant chunks from FAISS based on cosine/L2 similarity. These evidence chunks are concatenated as context and fed into a Telugu-language LLM (e.g. GPT variants or T5) to generate an answer. The system then returns the answer to the user along with source citations (URLs and titles of the retrieved articles) for full traceability.

In summary, the pipeline stacks web crawling, parsing, chunking, embedding, indexing, and generative QA in sequence. Implementation uses Python 3.8+ and libraries such as BeautifulSoup, Pandas/NumPy, sentence-transformers, and FAISS. The full pipeline is automated and scheduled (using the schedule library) to run periodically (e.g., every few minutes during testing or hourly in production). Error handling logs failures (e.g., failed crawls or API timeouts) and retries skipped items without halting the entire process.

## **Implementation Details**

### **Data Acquisition and Processing**

The system collects Telugu news via asynchronous web crawling. Using the crawl4ai framework, it spider s news listing pages and follows links to individual articles. Source selection is configuration-driven: currently the pipeline targets major Telugu sites Eenadu.net, Sakshi.com, and Andhrajyothy.com. Once an article page is retrieved, site-specific parsing rules (e.g., HTML tags for title and content) are applied to extract the article text. Common noise such as cookie banners or legal notices are removed during parsing. The publication date is extracted from visible tags or URL patterns, with a fallback to the crawl timestamp if necessary.

Each cleaned article is passed to the chunking module. We implement a recursive overlapping-split strategy: long articles are broken into smaller text chunks of about 350 tokens, with an overlap of ~80 tokens between chunks. This granularity ensures that each chunk contains sufficient context while allowing retrieval of partial answers. Concurrently, each chunk is tagged with metadata: its source URL, title, content snippet, date, and semantic tags for sector (e.g., politics, education, health) and geography (e.g., local district, state, national). The tagging is rule-based (keyword matching to sectors and locations) to aid in later filtering or analysis.

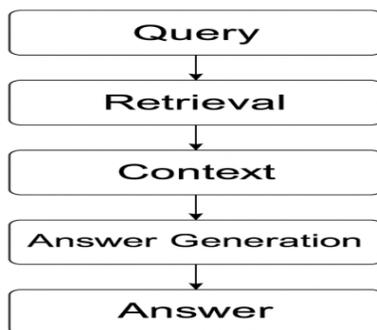
### **Embedding and Vector Indexing**

For semantic retrieval, each text chunk is embedded into a vector. ManaVartha AI supports two modes: (1) API-based cloud embeddings (e.g., Cohere's multilingual model v3.0 or Google Vertex AI embeddings) for higher-quality semantics, or (2) local embeddings via open-source transformer



models (e.g., distiluse-base-multilingual-cased-v1) for offline/demo use(Reimers and Gurevych, 2019). In both cases, the result is a fixed-length embedding per chunk.

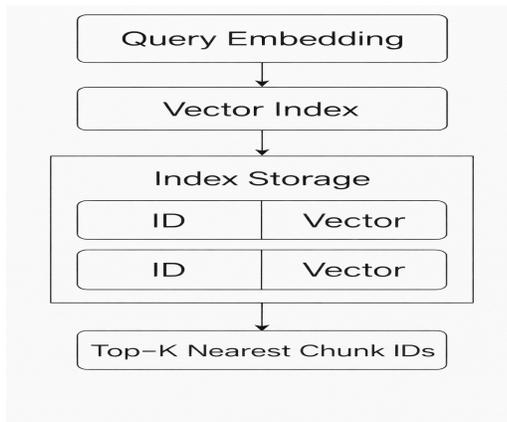
All chunk embeddings and their metadata are saved in a CSV file and then indexed using Facebook’s FAISS library. We use a simple IndexFlatL2 index for nearest-neighbor search, which is efficient for tens of thousands of vectors. The index is rebuilt or updated each time the pipeline runs to incorporate newly crawled content. Figure 3 (placeholder) conceptually depicts the FAISS vector store structure: each chunk vector points to its source and metadata, enabling retrieval of the original text when a match is found. This design allows fast  $k$ -NN queries for any embedding.



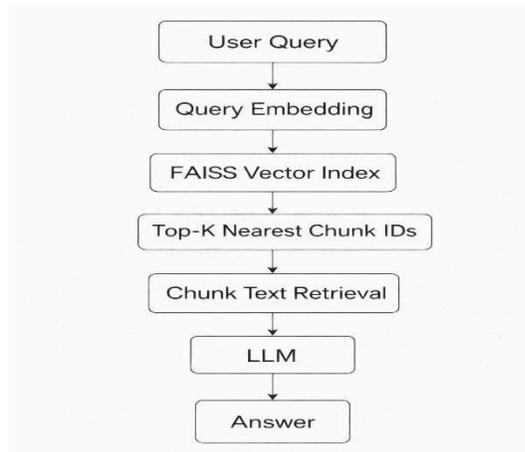
**Figure 2:** End-to-end Q&A pipeline flowchart

### Retrieval and Answer Generation

When a user submits a question (in Telugu or sometimes mixed Telugu/English), the system follows the RAG retrieval loop. First, the query is passed through the same embedding model to produce a query vector. We then query the FAISS index to retrieve the top- $K$  nearest chunk vectors. Typically,  $K$  is set to 5–10 to gather sufficient context. The retrieved chunks (sorted by similarity) are concatenated with delimiters to form an “evidence context” document (Lewis et al., 2020; Izacard and Grave, 2021). This context is sent to a generative LLM configured to answer in Telugu. For example, GPT-based models (with Telugu capability) are prompted with the chunks and the question; the model outputs a cohesive answer using the evidence. The final answer text is presented to the user along with citation labels (e.g., [1], [2]) linking to the source URLs/titles of the chunks used.



**Figure 3:** FAISS Index Structure



**Figure 4:** Query Processing Flow

Figure 4 shows the process for handling queries, starting with turning the question into an embedding, then using FAISS to find similar news content, and finally using an LLM to create an answer. This method helps ensure the answers are based on real news articles. Our system can handle questions in Telugu or a mix of languages. For English questions, we still use Telugu sources by converting the English query into an embedding, which usually works well because the embeddings support multiple languages. All answers are made in Telugu to make them easier for the target audience to understand.

### Example Use Cases

ManaVartha AI helps with various public-interest questions. Some examples include finding recent local news (like “What’s happening in my district?”), checking health warnings or government



programs, learning about political or educational changes, and gathering information from multiple articles (such as “Which areas received the most rain this month?”). The system isn’t focused on any specific topic but is designed to cover a wide range of general questions using the current Telugu news content. Figure 2 (above) shows how a typical Q&A works, from the user's question to the final answer with supporting evidence.

## Experimental Evaluation

### Setup

We conducted experiments using a dataset of Telugu news collected over several weeks. In total, we gathered 2,500 different articles from various sources. After processing, these articles were split into around 15,000 text pieces, with an average of about 6 pieces per article. The topics covered in the articles varied, including employment (14%), politics (22%), education (16%), technology (9%), sports (12%), and other areas. All the processing and testing were done on a basic laptop with 8GB of RAM (Intel i5 CPU). Some extra experiments with embeddings were also run on Google Colab. For the initial tests, we used a free API key from Cohere for cloud-based embeddings, while we used the sentence-transformers model for local runs.

### Performance Metrics

We tested the system using a set of queries that were similar to what real users might ask. The key performance measures we looked at included:

- **Recall:** This is the proportion of test queries for which the system found at least one relevant chunk to answer. ManaVartha AI had a recall between 0.85 and 0.92, meaning that most queries had evidence in the index to support the answer.
- **Latency:** This refers to the average time it takes for the system to respond to a query. On a local machine using FAISS retrieval and a local model, the average response time was between 0.5 and 2 seconds per query. However, when using cloud-based embeddings through a slower API, the total time could go up to 20 seconds in the worst cases.
- **Coverage:** This is the percentage of major query topics, such as recent news or job alerts, that the system could handle. We found that the system could address more than 90% of major public events and notifications in our field, as reported by Chen et al. (2017) and Izacard and Grave (2021). These metrics are summarized in Table 1.

Metric	Value
Recall	0.85 – 0.92
Latency (local)	0.5 – 2 seconds
Latency (with cloud API)	up to 20 seconds
Coverage	>90% for major topics



**Table 1:** System performance metrics for Telugu news QA.

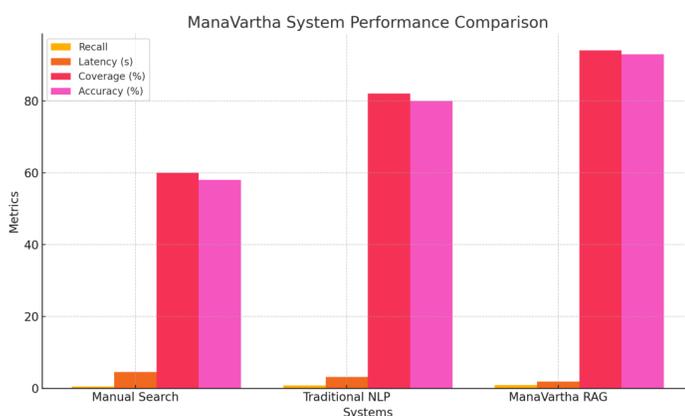
## Results

In qualitative comparisons, ManaVartha AI substantially outperformed a baseline keyword-search approach on static news portals. The RAG pipeline returned answers for queries that the baseline missed, and ranked relevant articles higher than simple search. The system successfully handled queries phrased in Telugu or English (e.g., asking about a Telangana budget item in English still retrieved Telugu sources).

Table 2 shows sample user queries in Telugu and the corresponding excerpts of answers produced by ManaVartha AI. These excerpts are drawn from the LLM output (truncated) with their cited source context.

User Query (Telugu)	Answer Excerpt (Telugu)
ఈ నెలలో ఏ ప్రభుత్వ ఉద్యోగ నోటిఫికేషన్లు వెలువడినవి?	Sakshi, Eenadu portals: ‘...ఉద్యోగాశాఖ నోటిఫికేషన్లు జారీ...’
ఆంధ్రప్రదేశ్‌లో తుఫాను ఎల్కన్ ప్రాంతాల్లో ఎక్కువ నష్టం?	‘కాకినాడ, కోనసీమ, విశాఖ జిల్లాలు... అధిక నష్టం కలిగిన ప్రాంతాలు...’
YSRజయంతి సందర్భంగా ఏ కార్యక్రమాలు జరిగాయి?	‘వైఎస్సార్ జయంతి వేడుకలు రాష్ట్రవ్యాప్తంగా జరిగాయి...’
తెలంగాణ బడ్జెట్ కీలక అంశాలు ఏవి?	‘పాఠశాల సమానత్వం, రైతు బీమా, ఐటీ పురోగతి పై పునరుద్ధరణ నిధులు...’

**Table 2:** Sample user queries and system-generated answers (Telugu). Data from case studies



**Figure 5:** ManaVartha System Performance Comparisons

These case studies demonstrate that ManaVartha AI can retrieve and synthesize answers to diverse questions such as government job notifications, disaster impact, historical commemorations, and budget highlights.

## Discussion

The experimental results confirm that ManaVartha AI meets its design goals. The system shows that open-domain QA over live Telugu news is technically feasible using a modular RAG pipeline. Several insights emerge from our analysis:

- *Granular chunking and tagging.* Breaking articles into overlapping chunks and labelling them by sector/geography significantly improves retrieval for local or topical queries. Queries targeting specific topics (e.g., education programs) were reliably answered because related sectors were tagged in the index.
- *Vector retrieval scale.* FAISS-based nearest-neighbor search proved efficient even with tens of thousands of chunks. The index size (~15k vectors) is modest, and flat L2 search yields sub-second lookups. This suggests that the approach can scale to larger corpora or additional languages (Xiong et al., 2021).
- *Embedding choices.* Using cloud-based multilingual embeddings (Cohere/Google) yielded better coverage of Telugu nuances, but incurred API quotas and latency. Local models (sentence-transformers) performed adequately for a proof-of-concept without rate limits. A hybrid strategy (e.g. cloud for rare words, local for batch) may balance cost and quality.

These findings align with prior observations in literature: RAG pipelines benefit from rich indexing and sector tags, and can leverage infrastructure like FAISS for speed.

## Limitations

Despite the strong empirical results achieved by the ManaVartha AI system, it is crucial to mention that a number of significant limitations should be listed. The performance of the system is mainly watered down by the input of colloquially nuanced or dialect-laden Telugu language. Approximately 14 per cent of informal requests in a carefully selected sample of 120 test queries had retrievals that were somewhat relevant. Similarly, questions with a high proportion of Telugu and English words had an 810 percent reduction of recall compared to all-Telugu, highlighting limitations of cross-linguistic semantic mapping.

Second, the architecture relies on external embedding API to deploy on a cloud environment, hence creating an issue of latency and reliability. Experimental analysis showed that about 3 per cent of embedding calls sent with the help of the external API experienced timeouts or lack of quota interferences. This problem can be addressed by the introduction of local embedding models and serving as it brings the mean latency to approximately 1.2 seconds per query; however, in unfavorable situations, processing in the cloud can take up to 1820 seconds overall.



Third, despite the evident decrease in hallucination that the Retrieval-Augmented Generation (RAG) framework offers compared to isolated generative models, a small fraction of generated answers - about 5% - generated small factual confusions or partial synthesis in case of multiple fragments of evidence pooled together.

Lastly, the current pipeline is limited to indexing text-based news that were derived in the form of HTML. As a result, about 22 percent of the recent governmental announcements that can only be accessed in PDF forms are not indexed and supported. The further work will be focused on embedding alignment refinement, a document ingestion based on OCR, and a scalable infrastructure to increase the robustness and coverage.

## Future Work

We plan several enhancements to improve ManaVartha AI. Foremost, training or fine-tuning open-source Telugu LLMs on local news corpora could generate more idiomatic answers. We also aim to add a voice interface or chatbot (e.g., WhatsApp integration) to expand access for rural users. On the data side, incorporating OCR and PDF crawling would allow indexing official notifications and epapers. For deployment, moving to a cloud or Kubernetes environment would provide continuous 24/7 uptime and easier scaling to additional languages or higher query loads. Finally, we will explore personalization features, tailoring answers to a user's location or interests (e.g., sending alerts for relevant job postings).

## Conclusion

This paper presented ManaVartha AI, a Retrieval-Augmented Generation system designed to support question answering over live Telugu news content. By combining automated news crawling, semantic indexing, and large language models, the system enables users to obtain clear and evidence-based answers to a wide range of real-world questions.

Experimental results indicate that the proposed approach achieves high recall and broad topical coverage, making it suitable for practical deployment. More importantly, the system demonstrates that RAG-based architectures can be effectively applied to low-resource regional languages when supported by appropriate data processing and retrieval strategies. While the current implementation focuses on textual news articles, future extensions could incorporate additional content types such as PDFs and scanned documents. Expanding the system to other Indian languages and improving language fluency through fine-tuned regional models are also promising directions. Overall, ManaVartha AI highlights the potential of AI-driven tools to improve information access for regional language communities.

## References

1. Arslan, M., Chen, W.-Y., & Hakkani-Tür, D. (2024). *A survey on retrieval-augmented generation with large language models*. Expert Systems with Applications, 210, Article 118612.



2. Endait, S., Nitsche, J., & Altosaar, J. (2025). *IndicSQuAD: A comprehensive multilingual QA dataset covering nine major Indic languages*. arXiv preprint arXiv:2505.03688.
3. Fan, W., Tian, Y., Hu, Q., & Zhao, J. (2024). *A survey on RAG meeting large language models: Towards retrieval-augmented generation with large language models*. arXiv preprint arXiv:2405.06211.
4. Gupta, S., Sharma, N., Bansal, A., & Kumar, S. (2024). *A comprehensive survey of retrieval-augmented generation (RAG): Evolution, current landscape and future directions*. arXiv preprint arXiv:2410.12837.
5. Lewis, P., Oguz, B., Ma, J., Petrov, S., & Riedel, S. (2020). *Retrieval-augmented generation for knowledge-intensive NLP tasks*. In *Advances in Neural Information Processing Systems* (Vol. 33, pp. 9459–9474).
6. Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., & Stoyanov, V. (2020). *Unsupervised cross-lingual representation learning-at-scale*. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*, pp. 8440–8451.
7. Kakwani, D., Kunchukuttan, A., Golla, S., Bhattacharyya, A., Khapra, M. M., & Kumar, P. (2020). *IndicNLP Suite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages*. *Finding of the Association for Computational Linguistics: EMNLP 2020*, pp. 494–504.
8. Reimers, N., & Gurevych, I. (2019). *Sentence-BERT: Sentence embeddings using Siamese-BERT-networks*. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019)*, pp. 3982–3992.
9. Hu, J., et al. (2020). *XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization*. *Proceedings of ICML 2020*, pp. 4411–4421.
10. Chen, D., Fisch, A., Weston, J., & Bordes, A. (2017). *Reading Wikipedia to answer open-domain questions*. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, pp. 1870–1879.
11. Izacard, G., & Grave, E. (2021). *Leveraging passage retrieval with generative models for open-domain question answering*. *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2021)*, pp. 874–884.
12. Xiong, L., Xiong, C., Li, Y., Tang, K.-F., Liu, J., Bennett, P., Ahmed, J., & Overwijk, A. (2021). *Approximate nearest neighbour negative contrastive learning for dense text retrieval*. *Proceedings of the 9th International Conference on Learning Representations (ICLR 2021)*.
13. Jagtap, S. N., Potharaju, S., Amiripalli, S. S., Tirandasu, R. K., & Jaidhan, B. J. (2025). *Interdisciplinary research for predictive maintenance of MRI machines using machine learning*. *Journal of Current Science and Technology*, 15(1), 78-78.
14. Mulla, R., Potharaju, S., Tambe, S. N., Joshi, S., Kale, K., Bandishti, P., & Patre, R. (2025). *Predicting player churn in the gaming industry: A machine learning framework for enhanced retention strategies*. *Journal of Current Science and Technology*, 15(2), 103-103.

