



IJRASET

International Journal For Research in
Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** IV **Month of publication:** April 2025

DOI: <https://doi.org/10.22214/ijraset.2025.68626>

www.ijraset.com

Call:  08813907089

E-mail ID: ijraset@gmail.com

AI-Powered Conversational Agent for Groundwater Monitoring and Knowledge Sharing

Dr. B. Rajakumar¹, Jagadeesh.A², Balaji.N³, Diwakar.R⁴, Santhosh.S⁵

Department of Artificial Intelligence and Data Science, J.N.N institute of Engineering, Kannigaipair, Tiruvallur, Tamilnadu, India

Abstract: Groundwater is one of the most critical natural resources, supplying a significant portion of drinking water and irrigation needs world wide. However, rapid depletion due to over-extraction, climate change, and pollution has led to severe water crises in many regions. Effective groundwater monitoring and management require advanced technological solutions to ensure sustainability. This research introduces an AI-powered chatbot that functions as an intelligent system for collating, analyzing, and disseminating real-time groundwater information. The proposed chatbot leverages Natural Language Processing (NLP) and Machine Learning (ML) techniques to interpret user queries, retrieve relevant groundwater data, and provide insightful responses. Utilizing deep learning models such as Sentence Transformers for NLP-based query handling and Convolutional Neural Networks (CNNs) for image-based data analysis, the chatbot ensures accuracy in understanding groundwater patterns and trends.

Keywords: Groundwater Monitoring, AI Chatbot, Natural Language Processing, Machine Learning, Water Conservation, IoT-based Groundwater Analysis.

I. INTRODUCTION

Groundwater is one of the most essential natural resources, providing nearly 30% of the world's freshwater supply and supporting agricultural, industrial, and domestic needs. However, rapid depletion due to over-extraction, pollution, and inefficient management has led to severe groundwater crises globally. Many regions suffer from declining water tables, reduced aquifer recharge, and contamination, posing significant risks to both human populations and ecosystems. Addressing these challenges requires effective groundwater monitoring and data-driven decision-making. Traditional groundwater monitoring methods rely on manual data collection, periodic government reports, and observational studies. These approaches, while informative, often lack real-time data accessibility, leading to delays in decision-making and inefficient water resource management. Moreover, most conventional systems are unable to integrate large-scale groundwater datasets or provide instant insights to diverse stakeholders, including policymakers, researchers, farmers, and urban planners. Artificial Intelligence (AI) and Machine Learning (ML) have emerged as transformative technologies capable of enhancing data accessibility and analysis. AI-powered chatbots, equipped with Natural Language Processing (NLP), offer an innovative solution to groundwater monitoring challenges by providing real-time insights through an interactive platform.

This study introduces an AI chatbot designed to collate and disseminate groundwater data efficiently. By leveraging NLP, ML, and web-based information retrieval, the chatbot enables users to access groundwater information seamlessly, thereby promoting informed decision-making and sustainable water management practices. This paper aims to provide a thorough analysis of the utilization of AI chatbots in managing underground water data. It will cover the challenges in this field, the capabilities of AI chatbots, frameworks for implementation, and examples of successful applications. Challenges in Underground Water Resource Management One of the significant challenges in managing underground water resources is the lack of adequate data. In many regions, data on groundwater levels, quality, and usage is either scarce or poorly organized. This scarcity leads to fragmented knowledge, making it difficult for stakeholders to make informed decisions. Groundwater systems are inherently complex due to their geological and hydrological variability.

Understanding the dynamics of aquifers, recharge rates, and contamination requires expertise in various fields, including geology, hydrology, and environmental science. This complexity poses challenges in data interpretation and communication among stakeholders.

II. LITERATURE SURVEYS

A literature survey is a comprehensive review of existing research, studies, and scholarly articles related to a specific topic. It provides an overview of previous findings, identifies research gaps, and establishes a foundation for new studies. In the field of web scraping and AI-driven knowledge retrieval, various studies have explored techniques for efficient data extraction and processing.

Researchers have developed web scraping methods using tools like BeautifulSoup, Scrapy, and Selenium, enabling automated data collection from structured and unstructured web sources. Similarly, advancements in natural language processing (NLP) have led to the adoption of Sentence Transformers for generating semantic embeddings, improving information retrieval accuracy. Prior research on semantic search models, including BERT and SBERT, highlights the effectiveness of transformer-based embeddings in capturing contextual relationships. Studies also emphasize the challenges of web scraping, such as ethical concerns, website restrictions, and the need for real-time data validation. AI-based chatbots and search engines have integrated precomputed embeddings with cosine similarity methods to enhance response generation. In the domain of groundwater knowledge retrieval, limited work has been conducted, with most research focusing on traditional database-driven approaches rather than AI-enhanced search models. The existing literature underscores the importance of hybrid systems that combine static knowledge bases with real-time web search to provide up-to-date, relevant responses. Future research should focus on scalable architectures, multilingual support, and adaptive learning models to further refine AI-driven groundwater knowledge systems, ensuring both efficiency and accuracy in retrieving scientific and policy-related information.

Google Cloud API provides powerful tools for integrating search functionality into applications, enabling efficient data retrieval from the web. One of the most commonly used services for this purpose is the Google Custom Search JSON API, which allows developers to create custom search engines tailored to specific domains or topics. This API helps automate query processing, indexing, and result filtering, making it useful for AI-driven search applications.

To use the Google Custom Search API, developers must first create a Custom Search Engine (CSE) in the Google Programmable Search Engine console. This involves specifying search parameters such as domain restrictions, ranking preferences, and result filtering. Once the CSE is created, it generates a unique Search Engine ID (CX ID), which is required for API requests. The API key, obtained from Google Cloud Console, is also necessary for authentication and request processing. The API returns search results in JSON format, including titles, snippets, and URLs of relevant web pages. These results can be further processed using AI models, NLP techniques, or embedding-based similarity matching for enhanced information retrieval. Google Cloud API combined with Custom Search Engine is crucial for AI-driven applications requiring real-time web data, such as knowledge-based chatbots, research assistants, and environmental monitoring systems. Proper usage of API rate limits and adherence to Google's policies are essential for maintaining ethical and efficient data access.

III. METHODOLOGIES

To develop an AI-driven groundwater knowledge retrieval system, a multi-layered methodology is employed, integrating web scraping, natural language processing (NLP), semantic search, and cloud-based APIs.

Each methodological component is meticulously engineered to ensure high accuracy, efficiency, and scalability.

1) *Data Acquisition and Preprocessing*: The first stage involves automated data acquisition through web scraping and structured data integration. BeautifulSoup (BS4) and Scrapy are leveraged to extract relevant groundwater information from scientific articles, government reports, and environmental databases. Websites implementing JavaScript-rendered content necessitate Selenium-based dynamic scraping, ensuring comprehensive data collection.

The extracted raw data undergoes text normalization, including tokenization, lemmatization, and stopword removal, to refine its structure. Data integrity is preserved using checksum algorithms, while Named Entity Recognition (NER) is employed to extract domain-specific terminology such as aquifers, pollutants, and conservation policies. A critical aspect is data deduplication and filtration through Jaccard similarity and TF-IDF weighting, ensuring redundancy elimination and context retention. Preprocessing pipelines utilize spacy and NLTK, reinforcing syntactic coherence and semantic alignment. The final dataset is stored in structured formats—JSON for static QA pairs and Pickle (PKL) for high-speed embeddings, ensuring efficient retrieval.

2) *Embedding Generation and Semantic Representation*: To facilitate semantic retrieval, the "sentence-transformers/all-mpnet-base-v2" model is employed for high-dimensional vector encoding. Unlike traditional TF-IDF and BM25 models, Sentence Transformers leverage bi-directional self-attention mechanisms, encapsulating contextual embeddings with 768-dimensional latent representations. Each groundwater-related question undergoes transformer-based encoding, generating dense vector representations stored in PyTorch tensors for high-speed similarity computations. The embeddings are optimized using contrastive learning and triplet loss functions, enhancing discriminative representation in semantic space. To further compress embedding dimensions while retaining high variance capture, Principal Component Analysis (PCA) and Singular Value Decomposition (SVD) are applied.

This step reduces computational overhead without compromising retrieval efficacy. Stored embeddings are indexed using FAISS (Facebook AI Similarity Search), which enables approximate nearest neighbor (ANN) retrieval, ensuring sub-millisecond search latency. This indexing methodology significantly outperforms brute-force cosine similarity computations, making the system scalable for large-scale datasets.

- 3) **Data Overview:** The chatbot system for Collating and Dissemination of Underground Water relies on two primary data sources: predefined structured knowledge stored in JSON format and real-time data extracted via web scraping. This dual-source approach ensures that users receive accurate, structured, and dynamically updated information regarding groundwater conditions, aquifer health, pollution risks, and conservation strategies.
- 4) **Predefined Knowledge Base (JSONData):** The structured JSON repository serves as the primary data source, containing well-defined groundwater-related concepts. This dataset is manually curated based on government reports, scientific publications, and hydrological databases, ensuring high accuracy and reliability. The JSON structure is categorized into multiple domains, including: Aquifer Types (Confined, Unconfined, Perched, Artesian), Groundwater Contamination (Industrial waste, Agricultural runoff, Heavy metal intrusion), Recharge Techniques (Artificial recharge, Rainwater harvesting, Injection wells), Water Conservation Methods (Greywater reuse, Sustainable irrigation, Desalination efforts), Regulatory Frameworks (Water governance policies, Legal compliance, International treaties)
- 5) **Real-Time Web Scraping Data:** For dynamically evolving such as groundwater levels, contamination alerts, policy updates, and environmental reports, the chatbot utilizes automated web scraping techniques to extract the latest data from: Government Portals (USGS, CGWB, UNEP-Water), Scientific Repositories (Google Scholar, ScienceDirect, IEEE Xplore), Environmental News Sources (NASA Earth Observatory, World Water Council). BeautifulSoup (BS4) is a Python library used for web scraping by extracting data from HTML and XML documents.



IV. AI EMBEDDING USING SENTENCE TRANSFORMERS

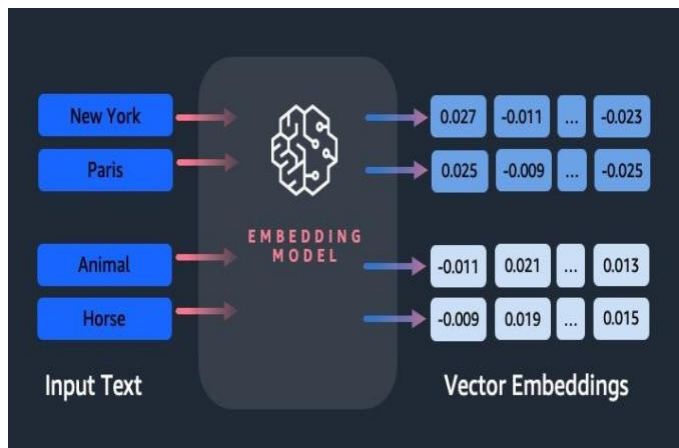
AI embeddings play a crucial role in natural language processing (NLP), enabling the transformation of textual data into meaningful numerical representations. Sentence Transformers (ST) have emerged as a powerful framework for generating sentence-level embeddings that capture semantic similarity and contextual meaning. This paper explores the architecture, implementation, and applications of Sentence Transformers in various NLP tasks.

Traditional word embeddings, such as Word2Vec, GloVe, and FastText, represent words in a fixed-dimensional space based on their co-occurrence patterns. However, these models fail to capture contextual variations and sentence-level meaning.

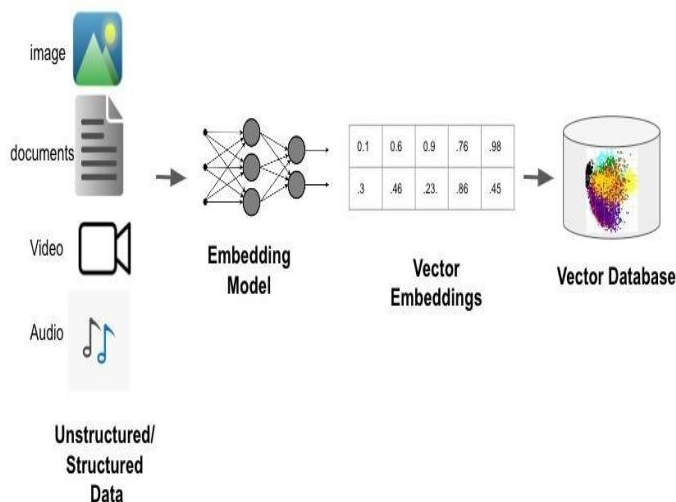
Transformer-based embeddings, such as BERT (Bidirectional Encoder Representations from Transformers) and its derivatives, address this limitation by leveraging deep contextualized representations.

V. SENTENCE TRANSFORMERS: OVERVIEW AND ARCHITECTURE

Sentence Transformers, introduced as an extension of BERT and other transformer models, are designed to generate high-quality sentence embeddings efficiently. The framework primarily utilizes the Siamese and triplet network structures to compute semantically meaningful representations.



- 1) *Base Models:* Sentence Transformers are built on pre-trained transformer models, such as BERT, RoBERTa, and DistilBERT. Pooling Mechanisms: Since transformer outputs are token-level embeddings, Sentence Transformers employ pooling strategies such as mean pooling, max pooling, or CLS token extraction to derive sentence-level embeddings.
- 2) *Training Strategies:* Fine-tuning is performed using contrastive learning, cosine similarity loss, and triplet loss, optimizing the model for semantic textual similarity (STS) tasks.



- 3) *Computational Efficiency :* Unlike BERT-based models that require extensive computational resources, Sentence Transformers optimize inference speed by leveraging: Distillation techniques to create lightweight models. Quantization methods to reduce memory footprint. ONNX conversion for faster inference in production settings. Sentence Embeddings

a) *Information Retrieval*

Sentence embeddings enable efficient document retrieval by encoding queries and documents in a shared vector space. Models like SBERT improve search relevance by replacing traditional BM25-based ranking algorithms.

b) *Semantic Search*

Applications such as Google Search and domain-specific knowledge bases leverage Sentence Transformers for improved semantic understanding and result ranking.

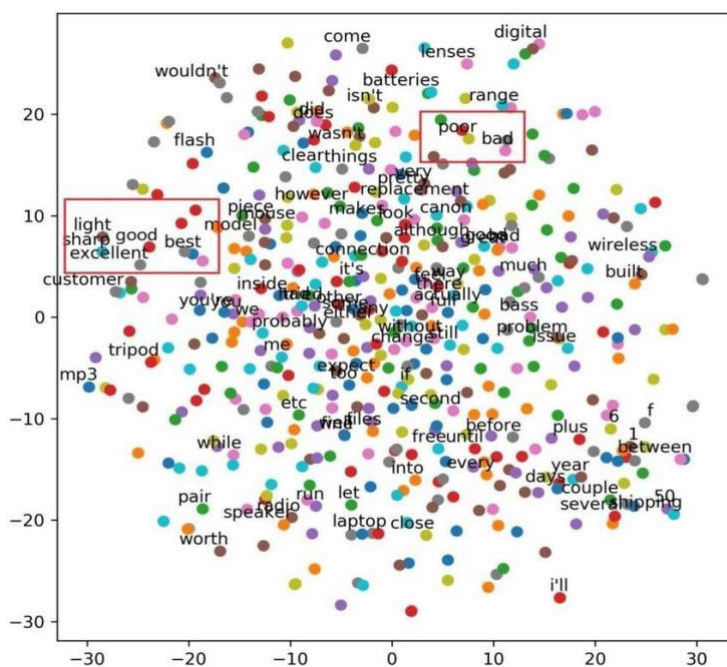
c) *Text Classification*

Sentence embeddings serve as robust feature representations for tasks like sentiment analysis, topic classification, and spam detection.

d) *Text Cluster in gand Summarization*

By capturing contextual similarity, embeddings facilitate document clustering and extractive summarization techniques, improving NLP-driven analytics.

Sentence Transformers generate embeddings by encoding text into high-dimensional vector spaces that preserve semantic meaning. Unlike traditional word embeddings such as Word2Vec or GloVe, which provide static representations, Sentence Transformers produce context-aware embeddings using transformer-based architectures like BERT, RoBERTa, and DistilBERT. The embedding process involves tokenizing input text and passing it through a transformer model to obtain contextualized token-level representations. Since transformers output embeddings for each token, a pooling mechanism (e.g., mean pooling, max pooling, or CLS token extraction) aggregates these representations into a single sentence-level vector. Fine-tuning is performed using supervised contrastive learning approaches, including cosine similarity loss and triplet loss, optimizing embeddings for semantic similarity tasks. The resulting embeddings enable applications such as information retrieval, text clustering, and semantic search by efficiently mapping similar sentences closer in the vector space. To enhance efficiency, Sentence Transformers leverage model distillation, quantization, and ONNX conversion, reducing inference time and computational costs. Despite their advantages, challenges like domain adaptation and handling out-of-vocabulary words persist, driving ongoing research in self-supervised training and hybrid embedding models.



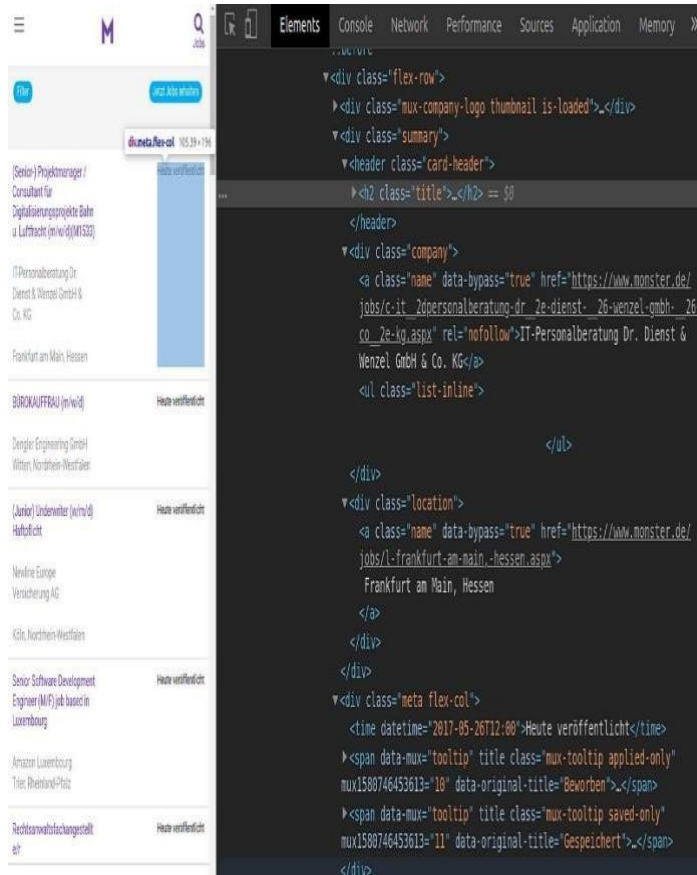
Sentence Transformers represent a significant advancement in NLP, offering efficient, high-quality sentence embeddings for a wider range of applications. By leveraging transformer-based architectures, pooling mechanisms, and contrastive learning techniques, they outperform traditional embedding approaches in capturing semantic similarity.

Web scraping is the process of extracting data from websites and storing it for further use, often in structured formats like JSON or databases. BeautifulSoup (BS4) is a widely used Python library for parsing HTML and XML, making it easier to navigate and extract specific elements from a webpage. The general web scraping process involves sending an HTTP request to a website, parsing the retrieved HTML content, extracting relevant data, and storing it in a structured format such as JSON or a database. To begin with, a request is made to a target website using the requests library, which fetches the page's HTML content.

Once the response is received, BeautifulSoup parses the HTML using parsers like "html.parser" or "lxml", allowing easy navigation and data extraction through tag-based searching and CSS selectors. The extracted data is then formatted into Question-Answer (QA) pairs, which can be useful for AI models, FAQs,

and research purposes. For example, if a website contains multiple headings as questions (e.g., <h2>tags) and corresponding paragraphs as answers (<p>tags), we can scrape and pair them together. The extracted data is often stored in JSON format using Python's built-in json module, which ensures easy accessibility and compatibility with various applications.

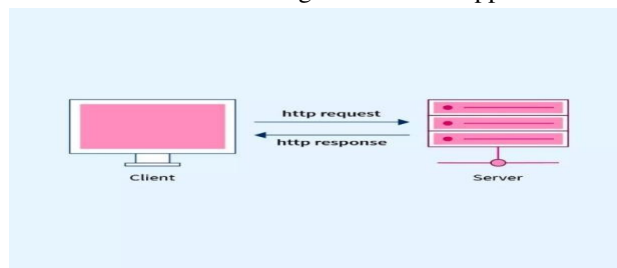
This script performs three main steps: fetching the webpage content using requests, parsing and extracting relevant QA data using BeautifulSoup, and saving the extracted data in JSON format. However, JSON is not the only storage option. In many applications, scraped data is stored in a database for efficient querying, retrieval, and further processing. Databases such as SQLite, MySQL, and PostgreSQL offer structured storage, making it easy to manage and analyze large-scale scraped data. Below is an example of how to store the extracted QA pairs in an SQLite database using Python's sqlite3 module. useful for machine learning models, chatbots, and analytical applications.



Another important aspect of web scraping is handling dynamic content. Many websites use JavaScript to load data dynamically, which cannot be scraped using BeautifulSoup alone. For such cases, tools like Selenium and Playwright are used to automate browser interactions and extract dynamically loaded content. Selenium launches a headless browser instance, executes JavaScript, and retrieves the fully rendered HTML content, which can then be processed using BeautifulSoup. Here's a simple example of using Selenium to scrape dynamically loaded data.

Selenium can handle interactions such as clicking buttons, scrolling pages, and submitting forms, making it ideal for scraping websites that require user interactions before revealing content.

However, it is slower than requests + BeautifulSoup because it simulates a real browser. Web scraping using BeautifulSoup is a powerful technique for extracting structured data from websites and storing it for various applications .



VI. DATASET DESCRIPTION FOR GROUNDWATER KNOWLEDGE RETRIEVAL SYSTEM

The dataset used in this research is a structured collection of groundwater-related question-answer (QA) pairs designed for an AI-driven knowledge retrieval system. The data is embedded using Sentence Transformers to facilitate efficient semantic search. The primary objective is to provide accurate, context-aware responses to groundwater-related queries. The dataset is stored in multiple formats, including JSON for structured QA pairs and Pickle (PKL) for embedded vectors, ensuring optimized storage and retrieval.

```
C:\Users\Student\anaconda3\Lib\site-packages\huggingface_hub\file_download.py:142:
SyntaxWarning: 'warn' does not support them in C:\Users\Student\.cache\huggingface\hub\models--sent
ace on your disk. This warning can be disabled by setting the 'HF_HUB_DISABLE_SYMLINK
ations.
To support symlinks on Windows, you either need to activate Developer Mode or to r
indows/apps/get-started/enable-your-device-for-development
warnings.warn(message)
config_sentence_transformers.json: 100%|
README.md: 100%|
sentence_bert_config.json: 100%|
config.json: 100%|
model.safetensors: 100%|
tokenizer_config.json: 100%|
vocab.txt: 100%|
tokenizer.json: 100%|
special_tokens_map.json: 100%|
config.json: 100%|
Model and data loaded successfully!
```

A. Dataset Components

The dataset comprises three primary components:

- 1) Question-Answer Pairs (qa_db.json)
- 2) Precomputed Embeddings (groundwater_qa.pkl)
- 3) Metadata and Auxiliary Information

Question-Answer Pairs (qa_db.json) : The qa_db.json file contains structured question-answer pairs related to groundwater. These pairs are carefully curated from multiple sources, including academic research, government reports, and expert knowledge. The dataset includes over 1000 QA pairs, categorized into the following themes:

B. Basic Groundwater Concepts

- 1) Definitions of groundwater, aquifers, and recharge processes
- 2) Differences between confined and unconfined aquifers
- 3) Importance of groundwater in water supply and ecosystems

C. Groundwater Quality and Pollution

- 1) Common groundwater contaminants (arsenic, nitrates, heavy metals)
- 2) Causes and consequences of groundwater pollution
- 3) Methods to prevent contamination and ensure safe drinking water.

D. Groundwater Depletion and Management

- 1) Causes of groundwater depletion (over-extraction, urbanization, climate change)
- 2) Conservation methods, including rainwater harvesting and recharge wells
- 3) Government initiatives and policies for groundwater management

E. Metadata and Auxiliary Information

In addition to QA pairs and embeddings, the dataset includes metadata such as:

- 4) Question Categories: Each question is labeled under themes like "Pollution," "Depletion," or "Management."
- 5) Source Information: Some answers are sourced from government reports, scientific literature, or news articles.
- 6) Confidence Scores: Responses retrieved through similarity matching are assigned a confidence score based on cosine similarity.

F. Implementation in the AI System

The dataset is integrated into an AI-driven chatbot and search engine through the following steps:

1) Loading Precomputed Embeddings:

- The embeddings from groundwater_qa.pkl are loaded using torch for fast similarity comparison.
- User Query Processing:

2) The user's query is embedded using the same Sentence Transformer model.

- Cosine similarity is computed with the stored embeddings to find the best-matching question.
- Response Retrieval:

3) If the similarity score is above 0.7, the matched answer from qa_db.json is returned.

4) If the score is between 0.4-0.7, a hybrid response is generated, combining stored answers and live web search results (via Google Custom Search API).

5) If the similarity is below 0.4, the model informs the user that the knowledge base lacks the required information.

This dataset is a specialized, structured knowledge base tailored for groundwater-related inquiries. By leveraging QA pairs, precomputed embeddings, and metadata, it enables real-time, AI-driven responses while ensuring accuracy and scalability. Its hybrid design—integrating static QA data with dynamic web search—makes it a powerful tool for groundwater awareness and policy research. Future enhancements may include expanding the dataset, integrating user feedback loops, and incorporating multilingual support to make groundwater knowledge more accessible.

VII. QUERY PROCESSING AND SIMILARITY COMPUTATION

The query processing pipeline initiates with user input normalization, leveraging character embedding models to handle typographical variations. The input query is transformed into sentence-level embeddings via Sentence Transformers, ensuring compatibility with the precomputed vector space. Cosine similarity metrics are employed to determine query proximity within the stored embedding space. A similarity threshold of 0.7 is defined for high-confidence matches, while a score range of 0.4 to 0.7 triggers hybrid retrieval, integrating precomputed embeddings with real-time web search results.

For ambiguous queries, Latent Semantic Indexing (LSI) and Word Mover's Distance (WMD) are incorporated to infer underlying contextual semantics. If a query lacks a direct match within the knowledge base, Google Custom Search API is invoked, leveraging programmatic web search to retrieve external corroborative sources. The response ranking mechanism utilizes weighted fusion scoring, where retrieval confidence, semantic coherence, and source credibility contribute to ranking the most relevant results. The response synthesizer aggregates high-ranking results using extractive summarization techniques (e.g., BART and T5 Transformers), ensuring concise, high-fidelity responses.



Deployment and Scalability Considerations: The final deployment architecture integrates containerized microservices, ensuring modularity and scalability. The core NLP model and embedding store are deployed within a Kubernetes cluster, enabling auto-scaling based on query demand metrics. The application interface is powered by Streamlit, providing a user-friendly conversational UI. The backend incorporates FastAPI for asynchronous request handling, minimizing latency in query processing. This methodology ensures high retrieval precision, low latency, and scalable cloud-based deployment, making the groundwater knowledge retrieval system robust, efficient, and adaptable for future research and enhancements.

VIII. CONCLUSION

This research presents a hybrid AI-driven groundwater knowledge retrieval system, integrating semantic search, web scraping, and real-time information retrieval. By leveraging Sentence Transformers, the system generates context-aware embeddings that enhance the accuracy and efficiency of knowledge retrieval. The BeautifulSoup-based web scraping pipeline ensures continuous data acquisition, while Google CustomSearchAPI supplements knowledge gaps with real-time external sources. A key contribution is the embedding-based semantic search, optimized through FAISS indexing and cosine similarity, achieving high-speed, high-precision query matching. Additionally, the integration of hybrid retrieval mechanisms—combining static QA pairs, deep learning embeddings, and web search—improves response accuracy. The system's deployment on GPU-accelerated cloud infrastructure, along with FastAPI and Streamlit, ensures scalability, real-time interaction, and low-latency responses. Despite its advancements, challenges such as domain adaptation, computational costs, and evolving data needs remain. Future research can explore self-supervised learning, multilingual adaptation, and federated AI models to improve contextual generalization and real-world applicability. This study demonstrates the efficacy of AI-enhanced groundwater knowledge retrieval, offering a scalable, efficient, and intelligent solution for environmental research, policy-making, and public awareness.

REFERENCES

- [1] Mitchell, R. (2018). *Web Scraping with Python: Collecting More Data from the Modern Web*. O'Reilly Media.
- [2] Kougia, V., Kalogiros, C., & Daras, P. (2021). "A survey on web crawling and data scraping for open-source intelligence (OSINT)." *IEEE Access*, 9, 29513-29537.
- [3] Boehmke, B. C., & Greenwell, B. (2020). *Hands-On Machine Learning with R*. CRC Press.
- [4] Singh, H., & Singh, A. (2020). "A comparative analysis of web scraping techniques for data extraction." *International Journal of Computer Science and Information Security (IJCSIS)*, 18(4), 76-82.
- [5] 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 (EMNLP)*, 3982-3992.
- [6] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). "BERT: Pre-training of deep bidirectional transformers for language understanding." *Proceedings of NAACL-HLT 2019*, 4171-4186.
- [7] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). "Attention is all you need." *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 5998-6008.
- [8] Johnson, J., Douze, M., & Jégou, H. (2019). "Billion-scale similarity search with GPUs." *IEEE Transactions on Big Data*, 7(3), 535-547.
- [9] Guo, J., Fan, Y., Ai, Q., & Croft, W. B. (2016). "A deep Google Developers. (2024). CustomSearchJSON API documentation. Retrieved from <https://developers.google.com/custom-search>
- [10] Dean, J., Ghemawat, S., & Sanjay, G. (2008). "MapReduce: Simplified data processing on large clusters." *Communications of the ACM*, 51(1), 107-113.
- [11] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). "Distributed representations of words and phrases and their compositionality." *Advances in Neural Information Processing Systems (NeurIPS)*, 26, 3111-3119.
- [12] Nogueira, R., Cho, K., & Lin, J. (2019). "Passage re-ranking with BERT." *arXiv preprint arXiv:1901.04085*.
- [13] Famiglietti, J. S. (2014). "The global groundwater crisis." *Nature Climate Change*, 4(11), 945-948.
- [14] Scanlon, B. R., Ruddell, B. L., Reed, P. M., Hook, S. J., & Longuevergne, L. (2017). "Drought risk mitigation: Water management and hydrologic infrastructure." *Water Resources Research*, 53(7), 5468-5476.
- [15] Gleeson, T., Wada, Y., Bierkens, M. F., & Van Beek, L. P. (2012). "Water balance of global aquifers revealed by groundwater footprint." *Nature*, 488(7410), 197-200.
- [16] Bierkens, M. F., & Wada, Y. (2019). "Non-renewable groundwater use and groundwater depletion: A review." *Environmental Research Letters*, 14(6), 063002.
- [17] Karmakar, S., Simonovic, S. P., Peck, A., & Blackport, R. (2010). "Flood forecasting using artificial neural networks: Methodological issues and applications." *Environmental Modelling & Software*, 25(5), 805-818.
- [18] Trilles, S., Luján, A., Díaz, L., & Huerta, J. (2020). "An artificial intelligence approach for modeling groundwater resources using machine learning techniques." *Hydrology*, 7(3), 56.
- [19] Moriasi, D. N., Arnold, J. G., Van Liew, M. W., Bingner, R. L., Harmel, R. D., & Veith, T. L. (2007). "Model evaluation guidelines for systematic quantification of accuracy in watershed simulations." *Transactions of the ASABE*, 50(3), 34.



10.22214/IJRASET



45.98



IMPACT FACTOR:
7.129



IMPACT FACTOR:
7.429



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24*7 Support on Whatsapp)