International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

AI Policy Intelligence System Using Retrieval Augmented Generation, Semantic Search and Knowledge Graphs

Mohd Faeez Ahmed[1], Krish Bansal[2], Chitraksh Tuli[3]

[1,2,3]Department of Information Technology, Delhi Technological University, New Delhi, India

Abstract

Government welfare schemes and public policies are introduced for the sole purpose of supporting citizens in areas such as healthcare, education, employment, agriculture, financial assistance and what not. However, many people are still not able to access a majority of suitable schemes because they have low awareness, or scattered information is available, or due to complex eligibility conditions. Also the existing systems mainly rely on keyword-based searches which is a bit primitive at these times of advancements, they often fail to provide personalized recommendations. This research presents an AI-Powered Government Policy Intelligence and Recommendation System that uses Retrieval Augmented Generation, semantic vector search, BM25 retrieval, and large language models to provide accurate and personalized policy recommendations. The system combines semantic understanding with keyword-based retrieval to improve policy discovery and relevance. We have developed a modern conversational interface using Streamlit, while ChromaDB and Sentence Transformers were used for vector storage and embedding generations. openAI language models were integrated into this platform to generate structured and user-friendly responses as per the need. The proposed system supports intelligent policy search, eligibility checking, personalized recommendations, analytics dashboards, policy similarity analysis all according to the user's needs. The solution improves accessibility and usability of government schemes while reducing the effort required for citizens to identify relevant policies. This project solves a real world problem that has not been looked into before.

Keywords: Retrieval Augmented Generation, Artificial Intelligence, Semantic Search, Government Policies, Recommendation System, ChromaDB, NLP, Knowledge Graph, BM25 Retrieval, Cosine Similarity

1. Introduction

Government policies and welfare schemes plays an important role in improving the quality of life of citizens. In India, thousands of schemes are introduced by both the central and state governments every year for the students, farmers, women, entrepreneurs, senior citizens, workers, and economically weaker sections of society. Despite the availability of vast schemes, a major challenge still exists that is of the low awareness of the citizens. Most citizens are not aware of the policies relevant to them. Information related to policies is spread across multiple websites and documents, making it difficult for the users to search and understand eligibility conditions.

Traditional search systems mostly depend on exact keyword matching, if the keywords do not match then there is no output given to the users. These systems often fail when users ask questions in natural language.

For example, a user may search for "financial support for female students in Haryana," while the actual policy may not contain the same exact words. This creates a large gap between the user's intent and retrieved policy information. To solve this real issue, this research project proposes an AI Driven Government Policy Intelligence System which is based on Retrieval Augmented Generation. The system combines semantic retrievl, keyword searches, vector databases, knowledge graph reasoning and large language models to provide accurate and context-aware recommendations to the fellow users. The proposed platform acts as an intelligent assistant which is capable of understanding user's intent and retrieving relevant schemes, presenting information in a structured and user friendly format.

2. Literature Review

- Several research studies and applications have explored the intelligent information retrieval systems and recommendation engines.
- Traditional information retrieval systems mainly depend on TF-IDF and keyword matching algorithms which is insufficient as of these times. While these methods are simple and computationally efficient, they very much struggle to understand semantic meaning and user intent.
- Modern Natural Language Processing techniques such as transformers and embeddings have significantly improved semantic understanding and helps in improving this system at a great extent. Models such as BERT and Sentence Transformers allows systems to convert text into dense vector representations which then helps in enabling semantic similarity search.
- Retrieval Augmented Generation has emerged as a very effective method for combining external knowledge retrieval with large language models. RAG efficiently improves factual accuracy because responses are generated using retrieved context instead of relying only on model memory.
- Vector databases such as ChromaDB and Pinecone are nowadays widely used for semantic search apps because they allow us to efficiently store and retrieve embeddings.
- Hybrid retrieval systems that combine semantic vector search with keyword-based retrieval methods such as BM25 have shown better performance up to great extent than using only a singular retrieval method.
- This research helps combine these modern AI technologies into a single unified platform for government policy intelligence system.

3. Methodology

The proposed AI Driven Government Policy Intelligence System was developed using a vast combination of semantic searches, Retrieval Augmented Generation and large language models, LLMs. Firstly, the government policy data was collected and preprocessed from JSON datasets. Many important policy fields such as the policy's title, summary, eligibility, and benefits were combined into structured text documents. These textual documents were then swiftly converted into vector embeddings using the Sentence Transformer model which goes by the name: **all-MiniLM-L6-v2**. These embeddings were then stored in ChromaDB, which acts as a vector database for semantic similarity search.

When a user enters a query, this system converts the query into vector embeddings and retrieves the most relevant policies using both semantic vector search and BM25 keyword retrieval. This hybrid retrieval approach highly improves the accuracy of the policy recommendations. The retrieved policies are then passed to a GPT-based large language model or we can say LLMs, using the Retrieval Augmented Generation (RAG) framework. This model generates personalized and context-aware responses or we can say

the policies based on the user's query and retrieved policy information. Fig 1 shows the whole architecture of how the user's query and the policy data works together to give the needed result.

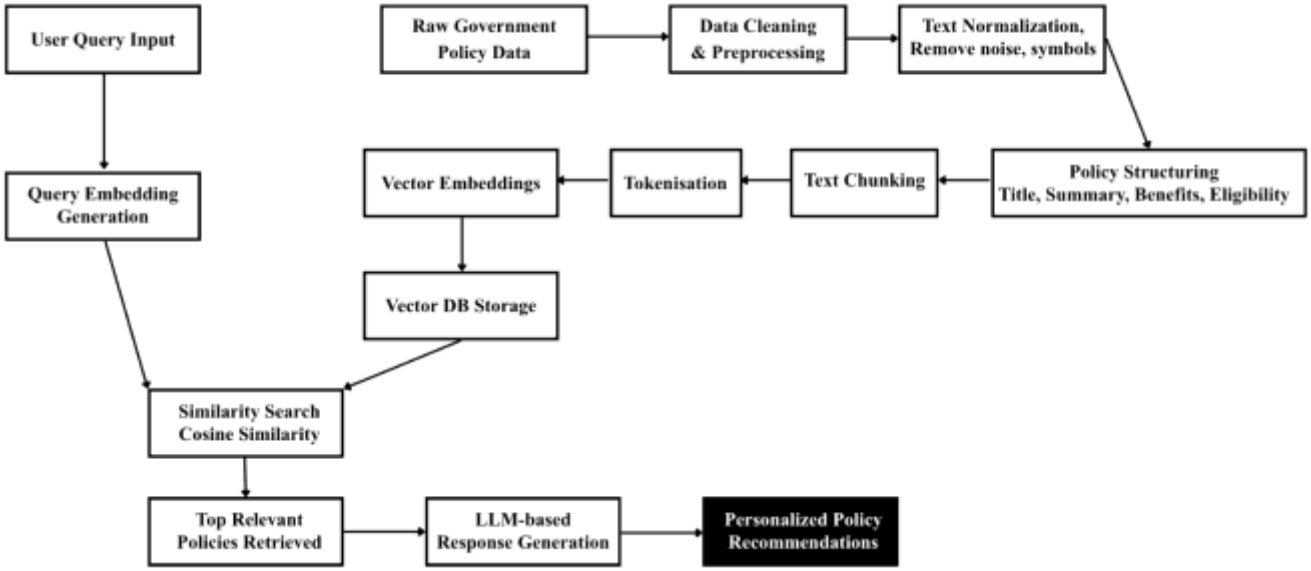

Fig 1: System Architecture of Proposed Framework

Finally, the results are then displayed through an interactive Streamlit-based web interface containing modules such as AI Policy Assistant, Policy Recommender, Eligibility Checker and also the Analytics Dashboard.

4. System Architecture And Implementation Details

The proposed system consists of various AI and retrieval components working together.

4.1 Overall Workflow

The workflow of the system is as follows:

- User enters a natural language query as per his/her needs.
- User's profile is extracted using NLP.
- Query embeddings are generated using the user's query.
- Semantic search retrieves relevant policies from ChromaDB as per the input given to it.
- BM25 keyword retrieval is performed afterwards.
- Results from both the retrieval methods are combined to give the appropriate result.
- Knowledge graph expansion identifies related policies which matches the user's criteria.
- Context filtering selects the most relevant policies for the user.
- OpenAI's GPT model generates final recommendations which is as close to the user's needs.
- Results are then displayed using Streamlit UI (works as the frontend of this sytem).

4.2 Technology Stack

The system uses multiple technologies for frontend, backend, AI processing, and vector retrieval. These technologies are shown below in Table 1.

Technology	Purpose
Python	Core programming language

Streamlit	Frontend web interface
ChromaDB	Vector database
Sentence Transformers	Text embedding generation
OpenAI GPT-4o-mini	Response generation
Pandas	Data processing
Plotly	Analytics visualization
BM25	Keyword retrieval
JSON	Dataset storage
NetworkX	Knowledge graph generation

Table 1: Technology Stack

4.3 Implementation Details

4.3.1 Data Preprocessing

The implementation process started with the pre-processing of the government policy dataset, which is in JSON format. The dataset was available in JSON format and contains information such as policy title, summary, eligibility, benefits, state, and tags. Since the raw data was not directly suitable for semantic retrieval, the policies were cleaned and converted into structured textual documents.

Important fields like title, summary, eligibility, and benefits were merged into a single text block. This helped the system capture complete contextual information during embedding generation.

Example Code:

```
text = " ".join([
    policy.get("title", ""),
    policy.get("summary", ""),
    policy.get("eligibility", ""),
    policy.get("benefits", "")
])
```

This pre-processing step highly improved the quality of the semantic understanding and the retrieving output's accuracy.

4.3.2 Embedding Generation

After the pre-processing stage, semantic embeddings were then generated using Sentence Transformers. The model used in the project was: **all-MiniLM-L6-v2**

This model converts textual information into dense vector representations that capture semantic meaning instead of relying only on exact keywords which is the core idea behind this research.

The model was selected because it is very lightweight, fast, and simply suitable for real-time applications. It also performs efficiently on large datasets while maintaining good semantic understanding, large datasets also helps in training the model even better for more accurate results.

The generated embeddings represents each policy in vector form, allowing the system to perform semantic similarity search, which highlights the use of cosine similarity.

4.3.3 Vector Database using ChromaDB

ChromaDB was used as the vector database for storing policy embeddings from the JSON dataset. It enables efficient vector similarity search and provides persistent local storage for embeddings. ChromaDB was selected because of the following advantages:

- It has fast semantic retrieval capabilities.
- Architecture is very light-weight.
- Easily integrates with Python.
- Supports persistent storage option.
- It's open-source nature improves flexibility.

The policy embeddings were stored along with metadata such as policy title, state, and category.

Example Code:

```
collection.add(
   ids=ids,
   embeddings=embeddings,
   documents=documents,
   metadatas=metadatas
)
```

This allowed the system to retrieve relevant policies quickly in due time during user queries.

4.3.4 Semantic Search

The semantic search module enables the system to understand the meaning and intent behind user queries, who does not have enough knowledge about the policies. When a user enters a query, the same embedding model converts the user's query into vector form. Cosine similarity is a fundamental approach which is used to compare the query embedding with stored policy embeddings in ChromaDB. This approach allows the system to retrieve relevant policies for the user even if the exact words are not present.

For **example** if User Query is "support for women startups in Rajasthan" then Retrieved Policy is "Women Entrepreneurship Development Scheme in Rajasthan"

Even though the wording may differ, semantic similarity helps identify the correct policy that the user requires. This significantly improves the retrieval quality compared to traditional keyword search systems.

4.3.5 BM25 Keyword Retrieval

To further improve the retrieval performance, BM25 keyword-based retrieval was integrated into the system. While semantic search is effective in understanding context, BM25 helps retrieve policies containing exact keywords, especially when the schemes are rare and hard to catch the names or technical terms. The hybrid retrieval system combines:

- Semantic Vector Search, and BM25 Keyword Search

This combination improves both precision and recall, resulting in more accurate recommendations for the citizens.

4.3.6 Retrieval Augmented Generation (RAG)

The core intelligence of the system is based mostly on Retrieval Augmented Generation (RAG).

Instead of generating responses only from model memory, the system first retrieves relevant policies from the database and then provides them as context to the Large Language Model. This approach offers several advantages:

- Reduces hallucinations, which helps in achieving more promising results for the user.
- Improves factual correctness of the policies.
- Generates context-aware responses according to the user's query.
- Supports domain-specific recommendations which is very crucial for the model

The retrieved policies, user profile, and user query are then combined into a structured prompt before being passed to the GPT model by openAI.

Example code:

```
prompt = f"""
User Profile:
{user_profile}
User Query:
{query}
Retrieved Policies:
{context}
"""
```

The GPT model then generates structured and meaningful responses containing:

- Relevant policy names as per the verified data.
- Eligibility details for the policies.
- Benefits offered by the same.
- Personalized explanations if the policy is right for the user or not.

This ensures that the generated response remains grounded in actual government policy data.

4.3.7 User Profile Extraction

The system also includes a lightweight user profile extraction mechanism. It identifies important details from user queries such as:

- State in which the user requires the policy.
- Occupation of the user
- User's category
- Gender-related information to segregate the policies conditions.

Fig 2 shows the vital information the system asks the user to give meaningful results.

Fig 2: User Information Extraction Panel

For **example**, if a user asks:

"I am a student from Haryana and I'm looking for scholarships in my area"

The system will extract the following:

```
profile["state"] = "Haryana"
profile["role"] = "student"
```

This profile information is stored temporarily and to be reused during future interactions to provide more personalized recommendations to the user.

4.3.8 Streamlit Frontend Development

The frontend interface was developed using Streamlit. Multiple intelligent modules were implemented, including: AI Policy Assistant, AI Policy Recommender, Eligibility Checker, Policy Explorer, Analytics Dashboard, Knowledge Graph Visualizer.

The frontend was designed using modern UI principles to make the UI more interactive and appealing such as glass morphism, neon gradients, responsive layouts, and interactive policy cards to improve user experience. Custom CSS styling was integrated into Streamlit for enhanced visual appearance.

4.3.9 Knowledge Graph Integration

A policy relationship graph was implemented to improve contextual retrieval and to measure the accuracy of the response to the respective query of the user. Policies were connected based on:

- Similar tags
- Common states
- Related benefits
- Similar target audiences

This helped the system recommend related policies even when they were not directly retrieved through semantic similarity search.

4.3.10 Analytics Dashboard

An analytics dashboard was developed using Plotly to visualize policy-related insights as shown in Fig 2,3. The dashboard provides:

- State-wise policy distribution, Policy category analysis, Query analytics, policy statistics

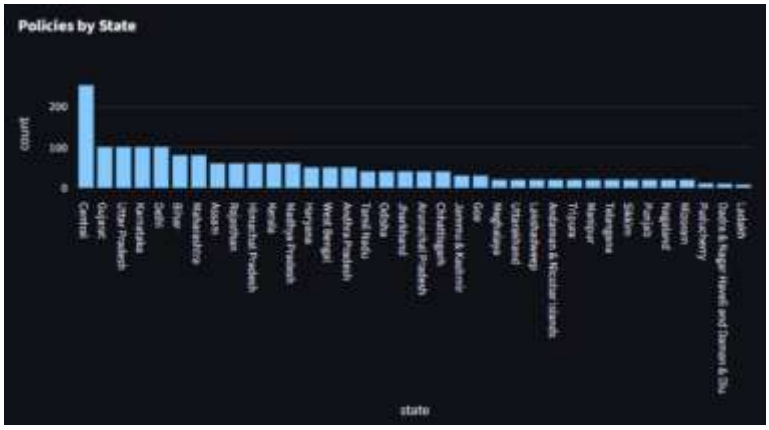

Fig 3: Distribution of Policies by State

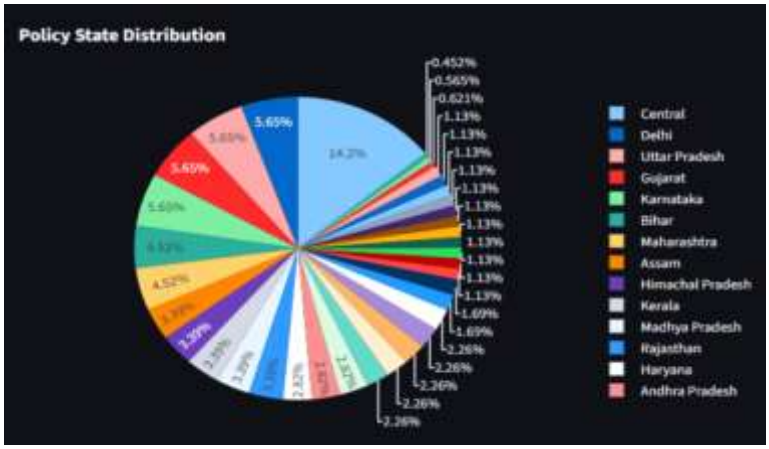

Fig 4: Percentage of Policies by State

These visualizations improve administrative monitoring and help users better understand the dataset and to apply for suitable policies according to their needs.

4.3.11 Error Handling and Optimization

Several technical issues were identified and resolved during implementation. Some major issues included duplicate vector IDs, dataset parsing errors, schema inconsistencies, and retrieval noise since the dataset was not clean in the earlier stages of the developments, Optimization techniques such as caching, bulk vector insertion, and filtered retrieval were also implemented to improve performance and system stability.

4.3.12 Final System Integration

Finally, all the modules were integrated into a single unified AI-powered platform. The completed system successfully provides:

- Intelligent policy recommendations as per the user's needs.
- Semantic policy retrieval way beyond the primitive keyword matching technique.
- Personalized eligibility analysis for better understanding and even administrative monitoring.
- Interactive analytics to have an in-depth idea of the right policy.
- Real-time conversational assistance provided by the administrative team.

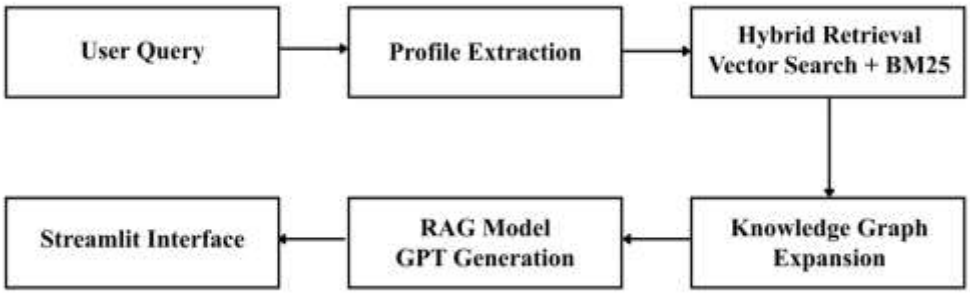

Fig 5: Data Flow in the System

Fig 5 shows the implementation of the practical application of Retrieval Augmented Generation and semantic AI systems in improving accessibility and awareness of government welfare schemes.

5. Results and Discussion

The proposed research project successfully improved government policy retrieval and recommendation quality of the policies requested by the user by combining semantic search, BM25 retrieval, and Retrieval Augmented Generation. Unlike traditional keyword-based systems, the proposed framework was able to understand user intent in a highly efficient and promising manner and provides more relevant policy suggestions.

Fig 6: Proposed AI Policy Recommendation System

The system also generated personalized recommendations based on user details such as state, occupation, and category of the user's policy . The AI-generated responses were structured, informative, and easy to understand. In addition, the Streamlit based interface provided a smooth and interactive user experience through modules such as the AI Policy Assistant, Eligibility Checker, and Policy Recommender which improves the user's experience as well as provides with the utilization of the governments initiatives for the good of the citizens.

Overall, the results demonstrate that the proposed framework improves accessibility, personalization, and intelligent discovery of government welfare schemes that created for the betterment of the citizens.

6. Conclusion

This research simply presented an AI-powered Government Policy Intelligence System designed to simplify access to government schemes and welfare policies and to raise awareness of the schemes that are till now under the knowledge of a few. The system combines Retrieval Augmented Generation, semantic search, BM25 retrieval, and knowledge graph techniques to provide accurate and personalized policy recommendations. By using AI-based retrieval and natural language processing, the platform improves policy discovery, eligibility analysis, and user interaction. Modules such as the AI Policy Assistant, Policy Recommender, Eligibility Checker, and Analytics Dashboard make the system more accessible and user-friendly upto a great extent. The proposed solution demonstrates how modern AI technologies can exponentially improve awareness and accessibility of government welfare schemes. In the future, the system can be further enhanced with multilingual support, voice interaction, and real-time government data integration to make the user's experience as seamless as it can be, for the sole purpose of the good for mankind.

7. References

1. P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," Advances in Neural Information Processing Systems (NeurIPS), 2020.
2. T. Mikolov et al., "Efficient Estimation of Word Representations in Vector Space," International Conference on Learning Representations (ICLR), 2013.
3. J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," NAACL-HLT, 2019.
4. S. Robertson and H. Zaragoza, "The Probabilistic Relevance Framework: BM25 and Beyond," Foundations and Trends in Information Retrieval, vol. 3, no. 4, pp. 333–389, 2009.
5. J. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT Networks," EMNLP, 2019.
6. M. Lewis et al., "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation," ACL, 2020.
7. H. Chen, X. Liu, and Y. Yin, "Hybrid Information Retrieval using Semantic and Keyword-Based Techniques," IEEE Access, vol. 8, pp. 12345–12360, 2020.
8. ChromaDB, "Chroma: The Open-Source Embedding Database," 2023. [Online]. Available: https://www.trychroma.com
9. LangChain, "Building Applications with LLMs through Composable Components," 2023. [Online]. Available: https://www.langchain.com
10. T. K. Landauer and S. T. Dumais, "A Solution to Plato's Problem: The Latent Semantic Analysis Theory," Psychological Review, 1997.

11. Government of India, "National e-Governance Plan (NeGP)," Ministry of Electronics and Information Technology, 2015.
12. Digital India Programme, Government of India, 2015. [Online]. Available: https://www.digital-india.gov.in
13. Elasticsearch, "BM25 Similarity Algorithm," 2022. [Online]. Available: https://www.elastic.co
14. J. Leskovec, A. Rajaraman, and J. Ullman, Mining of Massive Datasets, Cambridge University Press, 2020.