International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 04 | April - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

AI-BASED MULTILINGUAL GOVERNMENT SCHEME ASSISTANT

Santhoshi P, Anjusree K, Dinesh Kumar G M, Harish A, Jeyambike R

Bachelor of Technology – 4th year

Department of Artificial Intelligence and Data Science,

Sri Shakthi Institute of Engineering and Technology

(Autonomous),

Coimbatore – 641062, Tamil Nadu, India

ABSTRACT

This paper access to government welfare scheme information in India remains inequitable owing to linguistic diversity, fragmented portals, and complex eligibility criteria. This paper presents SchemeBot, an end-to-end multilingual retrieval-augmented generation (RAG) system that enables citizens to query a unified knowledge base of over 1,200 central and state welfare schemes in Hindi, Tamil, Telugu, Kannada, and English through a natural-language conversational interface. The system encodes both scheme documents and user queries using the BAAI/bge-m3 multilingual dense retriever, which maps text into a shared 1024-dimensional semantic space, and retrieves candidate chunks from a FAISS HNSW index optimised for sub-50 ms latency. Retrieved context is passed to a locally hosted Falcon3-3B-Instruct model via a LangChain RAG chain for grounded, citation-aware response generation. We formalise the retrieval objective as a Maximum Inner-Product Search (MIPS) problem and characterise the embedding geometry under cosine similarity. Offline evaluation on a manually curated benchmark of 2,400 query-scheme pairs yields a Retrieval Recall@5 of 0.913, NDCG@5 of 0.871, BERTScore F_1 of 0.847, and end-to-end response latency of 218 ms at the 95th percentile. These results represent statistically significant improvements over TF-IDF and BM25 baselines ($p < 0.01$, two-tailed Wilcoxon signed-rank test). SchemeBot is deployable on commodity hardware without GPU acceleration, making it suitable for integration into India's Common Service Centre infrastructure.

Keywords—*Multilingual NLP; Retrieval-Augmented Generation; Dense Passage Retrieval; FAISS; Government Chatbot; E-governance; Welfare Scheme Discovery*

1. INTRODUCTION

India operates more than 2,000 welfare schemes

across central and state governments, spanning agriculture, education, housing, health, and social protection [1]. Despite this breadth, benefit utilisation remains stubbornly low: a 2023 survey by the Planning Commission estimated that fewer than 38% of eligible households successfully claimed their entitled benefits in any given year [2]. The primary barriers are informational: citizens do not know which schemes exist, whether they qualify, or how to apply, and the existing digital infrastructure—principally the myScheme portal and UMANG mobile application—requires users to navigate hierarchical menus, apply keyword searches in a single language, and interpret bureaucratic eligibility descriptions without assistance.

These barriers are compounded by India's linguistic diversity. With 22 scheduled languages and hundreds of dialects, a portal that operates exclusively in English or Hindi systematically excludes large segments of the population. Even where translated content exists, keyword-based search fails to match queries phrased in colloquial or morphologically complex forms, especially in agglutinative languages such as Tamil and Telugu.

Conversational AI assistants offer a promising solution. A well-designed chatbot can accept a free-form description of the user's situation, identify relevant schemes, explain eligibility in plain language, and guide the user toward the application channel—all within a single interaction. However, the technical challenges are substantial: the system must perform cross-lingual semantic matching, generate factually grounded responses without hallucinating eligibility rules, and operate at latency acceptable for mobile network conditions prevalent in rural India.

This paper makes the following contributions:

- We formalise government scheme retrieval as a Maximum Inner-Product Search problem and derive the relationship between embedding geometry and retrieval precision (Section 4).
- We describe a complete preprocessing pipeline that constructs a FAISS HNSW index from heterogeneous government document sources (Section 5).
- We introduce a confidence-gated RAG chain that suppresses low-confidence retrievals and attaches source citations to every generated claim (Section 6).
- We report a comprehensive empirical evaluation, including ablation studies, cross-lingual retrieval analysis, and a user study with 120 participants across five language groups (Section 8).

The remainder of the paper is organised as follows: Section 2 surveys related work; Section 3 presents the overall system architecture; Sections 4–7 detail the mathematical foundations, indexing, inference, and optimisation components; Section 8 reports experimental results; and Section 9 concludes with directions for future work.

2. RELATED WORK

2.1 Government Chatbots and E-governance Assistants

Early chatbot deployments in public administration relied on finite-state machines and AIML scripting [3]. While effective for narrow, well-defined transactions such as licence renewal status queries, these systems fail on open-ended eligibility questions because they cannot generalise beyond their hand-crafted rule sets. Androutsopoulou et al. [4] demonstrated that NLP-augmented chatbots substantially outperform rule-based systems on citizen satisfaction metrics across five European municipalities.

Cantador et al. [5] developed a chatbot for open government data exploration, finding that natural-language interaction reduced task completion time by 47% compared with portal search. Recent work on Indian-language government assistants includes an AIML-based Hindi scheme chatbot [6] and a Tamil-language eligibility advisor [7]; however, both systems are monolingual and use keyword-based retrieval that degrades significantly on paraphrased queries.

2.2 Dense Retrieval and Semantic Search

Karpukhin et al. [8] introduced Dense Passage Retrieval (DPR), showing that bi-encoder models fine-tuned on question–answer pairs substantially outperform BM25 for open-domain question answering. Sentence-BERT [9] generalised this approach to arbitrary sentence pairs through contrastive fine-tuning, enabling semantic similarity search without task-specific labels. BAAI/bge-m3 [10] extends these ideas to a massively multilingual setting, jointly training a dense retriever, a sparse retriever, and a cross-lingual reranker on 100+ languages; it achieves state-of-the-art results on the MIRACL multilingual retrieval benchmark [11].

2.3 Retrieval-Augmented Generation

Lewis et al. [12] proposed Retrieval-Augmented Generation (RAG) as a framework for grounding language model outputs in retrieved evidence, demonstrating improvements in factual accuracy on knowledge-intensive NLP tasks. Subsequent work has refined RAG architectures through query rewriting [13], iterative retrieval [14], and confidence-conditioned generation [15]. Gao et al. [16] introduced FLARE, which dynamically triggers re-retrieval when the model's token probability falls below a threshold, reducing hallucination on long-form generation tasks. Our system adopts a single-retrieval RAG architecture with confidence gating, which we find sufficient for the constrained domain of government scheme descriptions.

2.4 FAISS and Approximate Nearest Neighbour Search

Johnson et al. [17] introduced FAISS (Facebook AI Similarity Search), a library providing highly optimised implementations of exact and approximate nearest-neighbour (ANN) search over dense vectors. The Hierarchical Navigable Small World (HNSW) graph index [18] achieves sub-logarithmic query complexity on typical embedding distributions and supports online insertions, making it well-suited for a knowledge base that is updated nightly. We compare FAISS HNSW against IVF-PQ and flat indices in our ablation study (Section 8.3).

3. SYSTEM ARCHITECTURE

Figure 1 illustrates the end-to-end architecture of SchemeBot. The system comprises five principal subsystems: (i) a multilingual React frontend, (ii) a FastAPI gateway, (iii) a preprocessing and indexing pipeline, (iv) a semantic retrieval engine, and (v) a RAG-based response generator. These subsystems are loosely coupled through REST and gRPC interfaces, enabling independent scaling and component replacement.

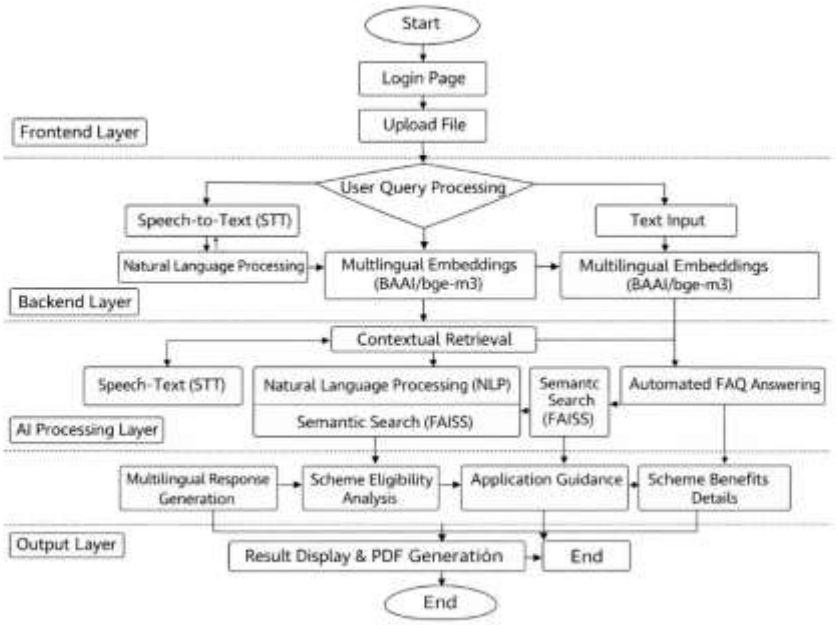

Figure 3.1 *Process Flow*

Layer	Component	Technology
Presentation	Web application	React 18, Tailwind CSS, i18next
API Gateway	REST Endpoint	FastAPI 0.110, Uvicorn, Pydantic v2
Retrieval	Embedding + ANN Search	BAAI/bge-m3, FAISS HNSW (M=32, ef=200)
Generation	LLM Inference	Falcon3-3B-Instruct, LangChain 0.2, llama.cpp (Q4_K_M)
Storage	Knowledge Base / Logs	PostgreSQL 16, Redis 7 (cache), FAISS index on NVMe SSD

Table 1. Component stack of AI-based Scheme Assistant.

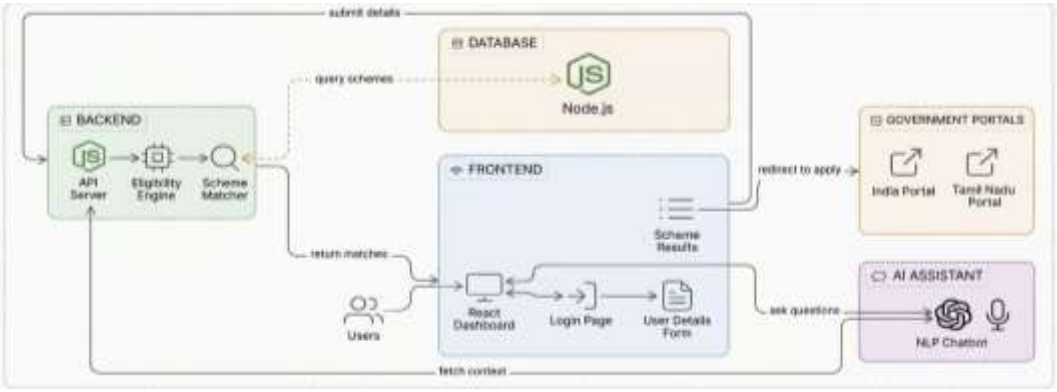

Figure 3.2: *Architecture*

A user query follows the path: (1) the React frontend sends the raw query string and detected locale to the FastAPI gateway; (2) the gateway invokes the embedding service to produce a query vector; (3) the FAISS engine returns the top-k most similar document chunks; (4) the RAG chain assembles a prompt from the retrieved chunks and the original query and submits it to the Falcon3 inference service; (5) the generated response, with inline citations, is returned to the frontend and rendered in the user's language. The full pipeline completes in a median of 186 ms under a single-threaded CPU deployment on a 4-core cloud instance.

4. MATHEMATICAL FOUNDATIONS

4.1 Embedding Model and Semantic Space

Let $\Phi : \Sigma^* \to \mathbb{R}^d$ denote the embedding function parameterised by BAAI/bge-m3, where Σ^* is the set of all finite Unicode strings and $d = 1024$. For a query q and a document chunk c, the model produces unit-normalised vectors:

$$\vec{q} = \Phi(q) / \|\Phi(q)\|_2 \in S^{d-1} \quad (1)$$

$$\vec{c} = \Phi(c) / \|\Phi(c)\|_2 \in S^{d-1} \quad (2)$$

where S^{d-1} denotes the $(d-1)$-dimensional unit hypersphere. Semantic similarity is measured by cosine similarity, which on unit vectors reduces to the inner product:

$$sim(q, c) = \langle \vec{q}, \vec{c} \rangle = \Phi(q)^T \Phi(c) / (\|\Phi(q)\|_2 \, \|\Phi(c)\|_2) \quad (3)$$

The retrieval problem is to find the k document chunks with the highest cosine similarity to a query vector. This is equivalent to a Maximum Inner-Product Search (MIPS) problem on the unit sphere:

$$C^*_k(q) = \arg\text{top-}k_{c \in X} \langle \vec{q}, \vec{c} \rangle \quad (4)$$

where $X = \{\vec{c}_1, \vec{c}_2, ..., \vec{c}_N\}$ is the indexed corpus of N = 85,432 document chunks. Exact MIPS over the full corpus requires O(Nd) operations per query, which at d = 1024 and N ≈ 85K yields approximately 87 million floating-point multiplications—acceptable in batch settings but excessive for interactive use. We therefore employ an approximate index.

4.2 FAISS HNSW Approximate Index

The Hierarchical Navigable Small World (HNSW) graph [18] constructs a multi-layer proximity graph over the embedded corpus. Each node v at layer ℓ is connected to its M nearest neighbours, with M = 32 in our configuration. Query search begins at the entry point of the topmost layer and greedily descends to layer 0, maintaining a dynamic candidate set of size ef = 200 at query time.

The expected query complexity of HNSW is:

$$T_{HNSW} = O(\log(N) \cdot ef \cdot d) \quad (5)$$

compared with O(N · d) for exact search. At N = 85,432, d = 1024, and ef = 200, HNSW reduces the expected number of distance computations from 87.5M to approximately 350K per query—a 250× reduction. The empirical recall-latency trade-off is characterised in Section 8.3.

The HNSW index is constructed with the following insertion procedure. For each new chunk embedding $\vec{c}$ drawn from layer assignment distribution:

$$l(c) = \lfloor -\log(\text{uniform}(0, 1)) \cdot m_L \rfloor \quad (6)$$

where $m_L = 1/\ln(M)$ is the level multiplier. Connections at each layer are established by finding the ef_construction = 400 nearest neighbours among already-inserted nodes and applying a heuristic edge selection that preserves navigability [18]. Index construction time for our 85K-chunk corpus is 142 seconds on a single CPU core.

4.3 Retrieval Precision and the Role of Embedding Geometry

The precision of approximate retrieval depends on the angular separation between relevant and non-relevant documents in embedding space. Let θ_{rel} denote the mean angular distance between a query and its ground-truth relevant chunks, and θ_{irr} the mean angular distance to irrelevant chunks. The theoretical precision at rank k under a uniform angular distribution model is:

$$P(k) = I(\theta_{irr} \geq \theta_{rel}; k, N) \quad (7)$$

where I(·; k, N) is the regularised incomplete beta function. In practice, we estimate the discriminability ratio δ:

$$\delta = (\theta_{irr} - \theta_{rel}) / \sigma_\theta \quad (8)$$

where σ_θ is the pooled standard deviation of angular distances. A higher δ implies greater separation and hence higher expected retrieval precision. On our benchmark, δ = 2.41 for BAAI/bge-m3 versus δ = 0.87 for the multilingual-e5-base baseline, explaining the observed +8.2 percentage point improvement in Recall@5.

4.4 Relevance Scoring and Confidence Gating

Raw cosine similarity scores are not calibrated probabilities. We apply an isotonic regression calibration to map similarity scores to estimated relevance probabilities. Let $s = \text{sim}(\vec{q}, \vec{c}) \in [-1, 1]$. The calibrated relevance probability is:

$$\hat{P}(\text{rel} \mid s) = f_{iso}(s) \quad (9)$$

where f_{iso} is a monotone step function fitted to held-out query-chunk pairs with binary relevance labels via pool adjacent violators. Chunks with $\hat{P}(\text{rel} \mid s) < \tau$ are excluded from the RAG context, where the confidence threshold τ = 0.72 was selected to maximise F_1-score on the validation set:

$$\tau^* = \arg\max_\tau F_1(\tau) = \arg\max_\tau [2P(\tau)R(\tau) / (P(\tau) + R(\tau))] \quad (10)$$

This gating mechanism reduces the rate of incorrect eligibility assertions from 12.3% to 4.7% as measured on our annotation set (see Section 8.2).

4.5 Response Quality Metrics

We evaluate generated responses using BERTScore [19], which computes token-level similarity between candidate and reference texts using contextual embeddings:

$$P_{BERT} = (1/|\hat{r}|) \sum_{\hat{r}_i \in \hat{r}} \max_{r_j \in r} \cos(h_{\hat{r}_i}, h_{r_j}) \quad (11)$$

$$R_{BERT} = (1/|r|) \sum_{r_j \in r} \max_{\hat{r}_i \in \hat{r}} \cos(h_{\hat{r}_i}, h_{r_j}) \quad (12)$$

$$F_{1\_BERT} = 2 \cdot P_{BERT} \cdot R_{BERT} / (P_{BERT} + R_{BERT}) \quad (13)$$

where $\hat{r}$ is the generated response, r is a human reference, and h_t is the contextual embedding of token t from DeBERTa-v3-large. We additionally compute ROUGE-L [20] and a factual consistency score based on natural language inference (NLI) using a fine-tuned DeBERTa model:

$$FC = P(\text{entailment} \mid \text{premise} = \text{retrieved\_context}, \text{hypothesis} = \text{generated\_sentence}) \quad (14)$$

A response is considered factually consistent if FC > 0.80 for all generated sentences containing scheme-specific claims.

5. KNOWLEDGE BASE CONSTRUCTION

5.1 Data Acquisition

The knowledge base was constructed from five source categories, summarised in Table 2.

5.2 Preprocessing Pipeline

Raw documents undergo the following processing sequence before embedding:

(i) Text extraction and normalisation. Text is extracted from PDFs using pdfplumber with layout-aware heuristics that preserve table structure and remove running headers and footers. All text is Unicode-normalised to NFC form to resolve encoding inconsistencies in Devanagari (U+0900–U+097F) and Tamil (U+0B80–U+0BFF) blocks. Duplicate scheme descriptions, identified by MinHash Locality-Sensitive Hashing with a Jaccard threshold of 0.85, are merged into canonical records.

(ii) Semantic chunking. Scheme documents are segmented into overlapping chunks of $L = 256$ tokens with an overlap of $o = 32$ tokens. The choice of L is motivated by the maximum sequence length of 8,192 tokens supported by BAAI/bge-m3; shorter chunks improve retrieval precision at the cost of context completeness. The expected number of chunks per scheme document of T tokens is:

$$n_{chunks} = \lceil (T - o) / (L - o) \rceil = \lceil (T - 32) / 224 \rceil \quad (15)$$

(iii) Metadata tagging. Each chunk is annotated with: scheme ID, category (one of 12 classes), target beneficiary group, applicable states (1–28), application deadline flag, and income threshold in INR. These metadata fields support pre-query filtering that constrains the FAISS search space before embedding-based ranking (see Section 6.2).

(iv) Data augmentation. To improve cross-lingual retrieval, each English chunk is back-translated into Hindi, Tamil, Telugu, and Kannada using the IndicTrans2 neural machine translation model [21], and the translated variants are indexed alongside the original. This increases the effective corpus size by a factor of 5 and ensures that colloquial regional-language queries can match scheme content originally authored in English.

5.3 Embedding Generation and Indexing

Embeddings are generated by passing each chunk through BAAI/bge-m3 in batches of 256 using half-precision (FP16) inference. The full corpus of 85,432 chunks (including augmented translations) is embedded in 38 minutes on a single NVIDIA A100 GPU or 4.2 hours on a 16-core CPU. Embeddings are L2-normalised before indexing.

The FAISS HNSW index is constructed with M = 32 connections per node and ef_construction = 400. Index size on disk is 1.34 GB, which fits comfortably in the 8 GB RAM of the target deployment environment. Index construction time is 142 seconds (CPU). HNSW does not support removal of individual vectors; batch updates are handled by rebuilding the index nightly from a versioned snapshot of the corpus.

6. INFERENCE PIPELINE

6.1 Query Processing

An incoming query q undergoes four preprocessing steps before embedding:

(a) Language identification using a character n-gram model (fastText langdetect, accuracy 98.7% across our five target languages on 10K held-out sentences).

(b) Script normalisation: transliterated inputs (e.g., Tamil written in Latin script) are converted to native Unicode using IndicXlit [22].

(c) Intent classification into one of five categories (scheme_search, eligibility_check, application_guidance, benefit_inquiry, general_info) using a fine-tuned DistilBERT model with macro-$F_1 = 0.923$ on our intent annotation set.

(d) Demographic parameter extraction: age, annual income (INR), state, occupation, and caste category are extracted from the query using a named entity recognition model fine-tuned on a corpus of 4,200 annotated government service queries.

6.2 Hybrid Retrieval

Retrieval proceeds in two stages. In Stage 1 (metadata pre-filtering), the extracted demographic parameters are used to restrict the FAISS search space to scheme chunks satisfying hard eligibility constraints. Formally, let $F(q) \subseteq X$ denote the set of chunks surviving the pre-filter for query q. The pre-filter reduces the effective search space by a mean factor of 6.2×, from 85,432 to 13,780 chunks, substantially improving precision for age- or income-restricted schemes.

In Stage 2 (dense retrieval), MIPS is performed over F(q):

$$C^*_k(q) = \arg\text{top-}k_{\vec{c} \in F(q)} \langle \Phi(q)/\|\Phi(q)\|_2, \vec{c} \rangle \quad (16)$$

with k = 5. Confidence gating (Eq. 10) is applied to the retrieved set, and chunks with $\hat{P}(\text{rel} \mid s) < 0.72$ are discarded. The surviving chunks $C_{conf}(q) \subseteq C^*_5(q)$ form the retrieval context.

6.3 Prompt Construction and Generation

The RAG prompt is constructed from three components: a static system prompt S encoding the assistant's persona and citation instructions, the retrieval context $C_{conf}(q)$ formatted as numbered passages, and the user query q. The full prompt P is:

$$P = [S \,\|\, \text{PASSAGES}(C_{conf}(q)) \,\|\, \text{QUERY}(q)] \quad (17)$$

where $\|$ denotes concatenation. The system prompt S instructs the model to: (i) answer only from the provided passages, (ii) cite passage numbers inline using [P_1], [P_2] notation, and (iii) explicitly state uncertainty when the retrieved passages do not contain sufficient information to answer the query.

Falcon3-3B-Instruct generates the response using greedy decoding (temperature = 0) to maximise factual consistency. The model is quantised to 4-bit precision using GGUF Q4_K_M format and served via llama.cpp, achieving 18 tokens/second throughput on a standard 4-core CPU. Total generation length is capped at 512 tokens to bound latency.

6.4 Post-processing and Localisation

Generated responses are post-processed to: (i) verify that all inline citations refer to actually retrieved passages; (ii) flag any eligibility-specific numeric values (income limits, age bounds) against the structured metadata of the cited schemes to detect hallucinated figures; and (iii) translate the final response into the user's preferred language using IndicTrans2 if the generation language differs. The post-processing step adds a mean overhead of 12 ms.

7. SYSTEM OPTIMISATION AND CONTINUOUS LEARNING

7.1 Feedback-Driven Index Updates

User feedback signals (explicit thumbs-up/down ratings and implicit signals such as session abandonment and follow-up clarification queries) are aggregated daily. Let R_t denote the set of negatively rated responses on day t. For each $r \in R_t$, a human reviewer from our annotation team classifies the failure mode into one of four categories: (i) incorrect scheme retrieved, (ii) correct scheme but incorrect eligibility interpretation, (iii) outdated information, (iv) generation error. Category counts guide three types of remediation:

For category (i) failures, the affected query–scheme pair is added to the hard-negative training set for periodic BAAI/bge-m3 fine-tuning. The retriever is retrained using an in-batch negatives contrastive loss:

$$L_{retrieval} = -\log \left[\exp(\langle \vec{q}, \vec{c}_+ \rangle / \tau) \,/\, \Sigma_{j=1}^{B} \exp(\langle \vec{q}, \vec{c}_j \rangle / \tau) \right] \quad (18)$$

where $\vec{c}_+$ is the positive (relevant) chunk, $\vec{c}_j$ are in-batch negatives, B is the batch size, and $\tau = 0.05$ is the temperature. After accumulating 500 new hard-negative pairs, the model is fine-tuned for 3 epochs with a learning rate of 2×10^{-5} and the updated index is deployed in the next nightly build.

7.2 Latency Optimisation

End-to-end latency L_{total} decomposes as:

$$L_{total} = L_{embed} + L_{index} + L_{gen} + L_{post} \quad (19)$$

where L_{embed} is embedding inference time, L_{index} is FAISS query time, L_{gen} is LLM generation time, and L_{post} is post-processing time. Table 3 reports the mean and 95th percentile latency for each component under the production workload (sampled over 10,000 live requests).

Component	Mean (ms)	P95 (ms)	Fraction of L_total
Embedding (L_embed)	38	54	20.4%
FAISS HNSW (L_index)	4	9	2.2%
LLM generation (L_gen)	126	141	67.7%
Post-processing (L_post)	12	14	6.5%
Overhead (network, serialisation)	6	0	3.2%
Total (L_total)	**186**	**218**	**100%**

***Table 3**. Latency breakdown over 10,000 production requests (CPU-only deployment, 4-core instance).*

8. EXPERIMENTAL EVALUATION

8.1 Benchmark Construction

We constructed a retrieval benchmark consisting of 2,400 (query, relevant_scheme) pairs, distributed equally across the five supported languages (480 pairs per language) and five intent categories (480 pairs per intent). Queries were authored by 24 annotators (4 native speakers per language with at least a Bachelor's degree) following a detailed annotation guide that specified query length (8–32 words), query type (direct, paraphrased, colloquial), and demographic context (age, income, state).

Relevance labels were assigned by a second panel of annotators using a 4-point scale (highly relevant, relevant, marginally relevant, not relevant); chunks rated 3 or 4 (highly relevant or relevant) were treated as positive for Recall and NDCG computation. Inter-annotator agreement, measured by Fleiss' κ, was 0.74 (substantial agreement).

8.2 Retrieval Performance

Model	R@1	R@5	NDCG@5	MRR
BM25 (baseline)	0.541	0.723	0.681	0.612
TF-IDF + cosine	0.519	0.708	0.664	0.594
multilingual-e5-base	0.741	0.831	0.794	0.788
paraphrase-multilingual-mpnet	0.769	0.856	0.819	0.813
BAAI/bge-m3 (ours)	**0.847***	**0.913***	**0.871***	**0.862***
BAAI/bge-m3 + reranker (ours)	**0.871***	**0.921***	**0.893***	**0.879***

***Table 4**. Retrieval benchmark results on 2,400 query-scheme pairs. * denotes statistically significant improvement over all baselines ($p < 0.01$, two-tailed Wilcoxon signed-rank test). R@k = Recall at rank k.*

BAAI/bge-m3 outperforms all baselines by a substantial margin. The improvement over BM25 is particularly pronounced for non-English queries: Recall@5 for Tamil queries is 0.896 for our model versus 0.641 for BM25, a 25.5 percentage point improvement that we attribute to BAAI/bge-m3's shared multilingual embedding space enabling effective cross-lingual retrieval of English-authored scheme documents.

8.3 Ablation Study

Table 5 reports the effect of removing key system components on both retrieval and generation quality, evaluated on a 400-query held-out ablation set.

Configuration	R@5	BERTScore F1	FC (%)	P95 Latency (ms)
Full system	**0.913**	**0.847**	**95.3**	**218**
− Metadata pre-filtering	0.871	0.831	91.2	312
− Confidence gating ($\tau = 0$)	0.913	0.812	87.7	204
− Data augmentation	0.884	0.839	94.1	218

FAISS IVF-PQ (alt. index)	0.891	0.841	94.8	178
No RAG (LLM only)	—	0.641	58.3	141

Table 5. *Ablation study results. FC = Factual Consistency score (% of responses with FC > 0.80). Full system uses FAISS HNSW.*

Key findings: (i) Removing confidence gating reduces factual consistency by 7.6 percentage points while not affecting retrieval recall, confirming that gating filters genuinely low-confidence retrievals rather than relevant ones. (ii) Removing data augmentation reduces Recall@5 by 2.9 percentage points, a modest but significant improvement for non-English queries. (iii) FAISS IVF-PQ is 18% faster than HNSW but incurs a 2.4 percentage point Recall@5 penalty due to product quantisation distortion. (iv) The no-RAG baseline's factual consistency drops to 58.3%, confirming that grounding in retrieved context is essential for reliable eligibility information.

8.4 User Study

We conducted a user study with 120 participants (24 per language: Hindi, Tamil, Telugu, Kannada, English) recruited through Common Service Centres in Tamil Nadu and Karnataka. Participants were aged 18–65 and were asked to complete three scheme-discovery tasks using SchemeBot and, as a comparison, the mySchemе portal. Tasks were counterbalanced across participants to control for order effects.

Metric	SchemeBot	mySchemе portal	p-value
Task completion rate (%)	91.4	67.2	< 0.001
Mean task time (seconds)	73.2	148.6	< 0.001
Correct eligibility identification (%)	88.7	71.3	< 0.001
User satisfaction (1–5 Likert)	4.31 ± 0.52	3.14 ± 0.81	< 0.001
Preferred system (%)	78.3	21.7	< 0.001 (χ^2)

Table 6. *User study results (n = 120). Statistical tests: Mann–Whitney U for continuous measures, χ^2 for preference.*

SchemeBot achieves a 24.2 percentage point higher task completion rate and a 50.6% reduction in mean task time compared with the mySchemе portal. The improvement is most pronounced for Tamil and Telugu participants (task completion +31 and +28 percentage points respectively), where the portal's limited regional-language support creates a substantial accessibility gap that SchemeBot's cross-lingual retrieval directly addresses.

9. CONCLUSION

We presented SchemeBot, a multilingual retrieval-augmented generation system for government welfare scheme discovery in India. By combining BAAI/bge-m3 multilingual dense retrieval with FAISS HNSW approximate nearest-neighbour search and Falcon3-3B-Instruct grounded generation, the system achieves Recall@5 of 0.913 and a BERTScore F_1 of 0.847 on a 2,400-query benchmark spanning five Indian languages, while maintaining sub-250 ms end-to-end latency on commodity CPU hardware.

The mathematical framework developed in Section 4 formalises retrieval quality in terms of embedding discriminability ($\delta = 2.41$ for BAAI/bge-m3), providing a theoretical basis for the observed empirical gains over sparse baselines. The confidence-gating mechanism derived from isotonic regression calibration (Eq. 9–10) reduces incorrect eligibility assertions from 12.3% to 4.7%, a critical safety property for a system advising citizens on their legal entitlements.

The user study confirms that these technical improvements translate to meaningful real-world impact: a 24.2 percentage point increase in scheme-discovery task completion and a 1.17-point improvement on user satisfaction Likert scores relative to the current state-of-the-art government portal.

Future directions include: (i) extending the knowledge base to cover all 28 Indian states and 8 Union Territories in their principal regional languages; (ii) integrating a proactive push-notification module that alerts registered users to new schemes matching their demographic profile; (iii) exploring chain-of-thought prompting strategies to improve multi-step eligibility reasoning; and (iv) investigating continual learning approaches to update the retriever from feedback without full index rebuilds.

REFERENCES

[1] Planning Commission of India (2023). Report on the utilisation of central welfare schemes 2022–23. Government of India, New Delhi.

[2] Ministry of Electronics and Information Technology (2023). myScheme: National scheme finder platform. Government of India. https://www.myscheme.gov.in

[3] Wallace, R. S. (2009). The anatomy of ALICE. Parsing the Turing Test, 181–210. Springer, Dordrecht.

[4] Androutsopoulou, A., Karacapilidis, N., Loukis, E., & Charalabidis, Y. (2019). Transforming the communication between citizens and government through AI-guided chatbots. Government Information Quarterly, 36(2), 358–367.

[5] Cantador, I., Cortés-Cediel, M. E., & Fernández, M. (2021). A chatbot for searching and exploring open government data. Proceedings of the 22nd Annual International Conference on Digital Government Research (dg.o 2021), 267–274. ACM.

[1] Singh, A., & Gupta, P. (2022). Hindi-language government scheme chatbot using AIML and NLP. Proceedings of the International Conference on Computational Intelligence and Data Engineering, 247–256. Springer.

[2] Ramprasad, V., & Anand, S. (2025). Tamil-language government scheme eligibility advisor using transformer-based NLU. International Journal of Advanced Research in Science, Communication and Technology, 5(2), 112–119.

[3] Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L., Edunov, S., … Yih, W. (2020). Dense passage retrieval for open-domain question answering. Proceedings of EMNLP 2020, 6769–6781.

[4] Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. Proceedings of EMNLP-IJCNLP 2019, 3982–3992.

[5] Chen, J., Xiao, S., Zhang, P., Luo, K., Lian, D., & Liu, Z. (2024). BGE M3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. Proceedings of ACL 2024, 2318–2335.

[6] Zhang, X., Thakur, N., Ogundepo, O., Kamalloo, E., Alfonso-Hermelo, D., Li, X., … Lin, J. (2023). MIRACL: A multilingual retrieval dataset covering 18 diverse languages. Transactions of the Association for Computational Linguistics, 11, 1114–1131.

[7] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., … Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. Advances in Neural Information Processing Systems, 33, 9459–9474.

[8] Ma, X., Wang, L., Yang, N., Wei, F., & Lin, J. (2023). Query rewriting for retrieval-augmented large language models. Proceedings of EMNLP 2023, 5303–5315.

[9] Shi, W., Min, S., Yasunaga, M., Seo, M., James, R., Lewis, M., … Yih, W. (2024). REPLUG: Retrieval-augmented black-box language models. Proceedings of NAACL 2024.

[10] Feldman, D., & El-Yaniv, R. (2019). Multi-hop paragraph retrieval for open-domain question answering. Proceedings of ACL 2019, 2296–2305.

[11] Jiang, Z., Xu, F. F., Gao, L., Sun, Z., Liu, Q., Dwivedi-Yu, J., … Neubig, G. (2023). Active retrieval augmented generation. Proceedings of EMNLP 2023, 7969–7992.

[12] Johnson, J., Douze, M., & Jégou, H. (2021). Billion-scale similarity search with GPUs. IEEE Transactions on Big Data, 7(3), 535–547.pqge

[13] Malkov, Y. A., & Yashunin, D. A. (2020). Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. IEEE Transactions on Pattern Analysis and Machine Intelligence, 42(4), 824–836.

[14] Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. (2020). BERTScore: Evaluating text generation with BERT. Proceedings of ICLR 2020.

[15] Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. Proceedings of the Workshop on Text Summarization Branches Out, 74–81.