Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Design Deployment and Empirical Evaluation of a Bilingual English and Amharic Retrieval Augmented Generation Chatbot for Ethiopian University Information Access

Yonas Birku Addis

yonas.birku@uog.edu.et

University of Gondar

Assefa Chekole Addis

University of Gondar

Research Article

Keywords: Amharic Natural Language Processing, Retrieval-Augmented Generation, University Information Access, Bilingual Chatbot, Semantic Search, ChromaDB, Large Language Models, Ethiopian Higher Education, Multilingual AI, RAG Evaluation

Posted Date: November 24th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11233941/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Abstract

Institutional information access remains a persistent challenge at Sub-Saharan African universities, where students navigate extensive English-language policy documents despite Amharic being their primary language. This paper presents a bilingual Retrieval-Augmented Generation chatbot designed for the University of Gondar (UoG), Ethiopia, enabling natural language access to official university documents in both English and Amharic (Ge□ez script). The system is deployed across three platforms: a Telegram bot, a React Native mobile application, and a Streamlit-based administrative web portal. The architecture combines a BAAI/BGE-M3 multilingual bi-encoder for semantic retrieval, a BGE-Reranker-v2-m3 cross-encoder for reranking, and Google Gemini 2.5 Flash for answer generation, supported by a multi-provider fallback chain and dual-layer Redis semantic cache. Anti-hallucination prompt engineering constrains all responses to passages retrieved from institutionally verified documents. Empirical evaluation was conducted using 200 questions derived from 20 official UoG policy documents spanning six institutional categories. Key findings include a language detection accuracy of 95.83% for English and 74.55% for Amharic Ge□ez script, and median response latencies of 5.76 seconds and 8.88 seconds respectively. A statistically significant 2× latency differential between the two languages reveals a systematic pipeline inefficiency in Amharic processing, traced to cross-language context interference in the generation step. To the best of our knowledge, UoG-RAG is the first empirically evaluated bilingual English-Amharic RAG chatbot deployed in an Ethiopian university, and the first study to quantify the Amharic processing performance gap in a production RAG system serving African institutional information needs.

1. Introduction

The global expansion of higher education has produced universities of unprecedented scale, serving student populations in the tens of thousands across diverse linguistic and cultural contexts. In sub-Saharan Africa, this expansion has not been matched by commensurate advances in institutional service delivery infrastructure. Students at major African universities frequently encounter significant difficulties accessing accurate, timely institutional information a challenge compounded by the predominance of English as the administrative language of policy documentation at institutions where the majority of students and staff communicate primarily in indigenous languages[1].

The University of Gondar (UoG), among the most venerable and esteemed public universities in Ethiopia, is home to 40,000 + students and 8,000 + staff members, including academics and administrators, spanning various faculty and institute units within. The institution possesses an enormous set of policy documents covering senate legislation, academic rules and regulations, student guidelines, human resources regulations, and degree requirements that are essential for any day-to-day academic decisions. However, this access still occurs primarily through manual means distribution of static PDF files stored in repositories, mostly available in English, and consultation with administrative office holders only in working hours.

General-purpose AI assistants like a standard large language model (LLM) deployments not sufficient tools to replace the institutional knowledge base due to lack of access to documents from UoG's repository and, thus, being either ignorant or, even worse, generating factually inaccurate yet plausible policy statements of the institution, which is called "hallucination" [2]. Retrieval-Augmented Generation (RAG) [3], [4]solves the issue directly by retrieved passages from a verified document repository to support generated responses, thus avoiding any fabrication.

This paper makes the following contributions to the literature on AI-assisted institutional information access in African higher education:

1. We design and implement UoG-RAG, a production-deployed bilingual RAG chatbot supporting English and Amharic interaction across mobile, Telegram, and web administrative platforms.
2. We conduct the first empirical evaluation of a bilingual English-Amharic RAG system for institutional information access, using 200 questions derived exclusively from real UoG policy documents.
3. We report a previously undocumented 2× latency differential between English and Amharic queries (median 5.76s vs. 8.88s) and provide a mechanistic explanation grounded in the system's language detection pipeline.
4. We quantify the language detection accuracy gap between English (95.83%) and Amharic Ge□ez script (74.55%) and identify Optical Character Recognition (OCR) quality variability and pipeline iteration as primary contributing factors.
5. We introduce and evaluate a dual-layer Redis semantic caching system combining exact-match and cosine-similarity-based caching with retrieval-score-conditioned Time to Live (TTL), and a multi-provider LLM fallback chain for zero-cost production deployment.
6. We identify and discuss the specific challenges of deploying AI information systems in resource-constrained African university environments, including infrastructure cold-start behaviour and OCR artefact propagation.

The remainder of the paper is structured as follows. Section II reviews related work. Section III presents the system architecture. Section IV describes the implementation. Section V details the evaluation methodology and presents empirical results. Section VI discusses findings, including the Amharic performance differential and its implications. Section VII identifies limitations and future work. Section VIII concludes.

2. RELATED WORK

2.1. Retrieval-Augmented Generation

Lewis et al.[3] introduced Retrieval-Augmented Generation as a framework for knowledge-intensive Natural Language Processing (NLP) tasks, demonstrating that conditioning LLM generation on retrieved external documents substantially improves factual accuracy over purely parametric approaches. Subsequent work by[5] established that dense passage retrieval using bi-encoder models substantially

outperforms traditional sparse retrieval (BM25) for open-domain question answering. Nogueira and Cho[6] demonstrated that cross-encoder reranking further improves precision over bi-encoder retrieval alone, a finding that informs the two-stage retrieval architecture adopted in UoG-RAG.

Recent advances in RAG have focused on faithfulness ensuring that generated responses are grounded in retrieved passages[7] and on efficiency, through caching mechanisms[8] and retrieval optimization[9]. UoG-RAG extends this body of work by applying RAG to a multilingual institutional context with specific constraints: a bilingual document corpus in English and Amharic, resource-constrained deployment infrastructure, and institutional governance requirements that standard RAG deployments do not address.

2.2. Multilingual NLP and Amharic Language Processing

Given that Amharic is the working language of Ethiopia, which has about 58 million people speaking it, there are some interesting challenges associated with NLP on Amharic. The challenges in developing high-quality NLP for Amharic are associated with the language's Fidel (Ge□ez) script and agglutinative morphology, as well as the historically sparse digitized resources for the language[10]. Initial benchmarking of Amharic NLP problems was conducted by Yimam et al. [11], followed by advances in sentiment analysis for Amharic [12], named entity recognition[13], and machine translation [14] .

Massively multilingual pre-trained language models constitute an important achievement for Amharic NLP research.[15] show that BAAI/BGE-M3 reaches state-of-the-art results on cross-lingual retrieval task across more than 100 languages, including Amharic, without involving translation as an intermediate step. The cross-lingual nature of the model namely the representation of texts in English and Amharic in the same 1,024-dimensional semantic space is at the core of UoG-RAG's multilingual retrieval capabilities. However, as will be revealed in our experiments, the difference in performance between languages remains large in practice.

2.3. AI Chatbots in Higher Education

The integration of AI chatbots in higher education has expanded rapidly in recent years. A systematic review by[16] documents significant advances in their application, identifying student support, personalized learning, and skill development as the three primary areas in which AI chatbots demonstrably benefit learners in higher education settings. A growing body of studies has examined specific applications, including providing support to students[17], giving out academic advice[18], and delivering administrative information[19]. Specifically, a systematic review conducted by[17] found that the two most popular and effective applications of educational chatbots on the Facebook Messenger platform are the delivery of information and FAQs.

There have been few examples of chatbot deployment at African universities. Okonkwo and Ade-Ibijola[20] identify significant gaps in the literature on educational chatbots applied to African institutional and linguistic contexts, and note the absence of deployments adapted to local infrastructure

constraints. This paper attempts to fill that void with an actual production deployment at a large African university.

2.4. Semantic Caching for LLM Systems

Semantic caching, which relies on cached results based on semantically similar questions to previous questions posed by users to LLMs, is suggested as a way of reducing inference cost and time lag in conversational AI systems [8]. Common techniques involve finding candidates based on the cosine similarity between the query and question embedding. The proposed method involves the construction of two caches an exact match cache (O(1) complexity for search) and a semantic similarity cache using the threshold value $\tau = 0.82$ for separating questions semantically similar to each other. The TTL of the cache depends on confidence scores, a new concept not considered before.

2.5. Identified Research Gap

This research literature highlights the presence of an obvious and unmet gap in existing literature: there has been no previous work that has studied the performance of a bilingual English-Amharic RAG system in an institution within an African university setting. Also, there is no previous work that has attempted to measure the difference between the latency and accuracy of English and Amharic in an actual RAG pipeline. UoG-RAG addresses both gaps with a real-world deployment and comprehensive empirical evaluation.

3. System architecture

3.1. Overview

UoG-RAG is implemented as a three-tier client-server application architecture, where a specific AI inference layer is used. Architecture diagram is shown in Fig. 1 below. Client layer consists of three interfaces specific to their platforms and sharing a common Application Programming Interface (API). Application layer is a FastAPI backend running in Hugging Face Spaces. AI layer implements the complete RAG pipeline. Data layer uses ChromaDB as a vector database, Redis as a cache, and SQLite as an administrative users' data store.

3.2. Request Processing Pipeline

Figure 2 shows the architecture of the request handling pipeline. The pipeline utilizes various computational efficiency features, including small talk interception (with over 30 small talk patterns in English and Amharic), which saves the computation time for non-information requests, exact match caching that gives O(1) lookup time for repeated queries, and semantic caching that retrieves cached responses for semantically similar queries without invoking the language model. The use of bypass reranker feature for queries less than five words and with less than three documents reduces mean reranker invocation rate to approximately 41% of non-cached requests.

3.3. LLM Fallback Chain

To ensure continuous availability independent of any single provider's quota or availability, UoG-RAG implements a unified multi-provider fallback chain through a single _call_llm() method. The chain is: (1) Google Gemini 2.5 Flash, with a pool of four independently registered API keys rotated on HTTP 429 rate-limit responses; (2) DeepSeek Chat API, a cost-efficient paid fallback; (3) OpenRouter free models, tried sequentially: google/gemini-2.0-flash-exp: free, meta-llama/llama-3.3-70b-instruct: free, and qwen/qwen-2.5-72b-instruct: free; and (4) a rule-based conceptual answer generator requiring no external API. This architecture ensures that the system returns a response under all failure scenarios.

3.4. Institutional Governance

A three-tier role-based access control system governs document management. System Administrators create and manage user accounts. Vice Presidents review and approve or reject document upload and deletion requests submitted by Data Encoders before they enter the live knowledge base. Data Encoders prepare and submit documents for review. Thus, the process guarantees the institutional review of all the material in the knowledge base and its factual reliability required for prompt engineering in order to avoid hallucinations. Overall, during the evaluation period 87 documents have been submitted via Data Encoder's interface, 11 documents were sent back for corrections.

3.5. Use of AI-assisted technologies (LLM)

While writing this paper, the authors employed Claude (Anthropic) as an AI tool for editing language, including grammar corrections, improvements in sentence structure, and clarity. AI-assisted tools for editing images were employed for editing of the selected figures. All generated or AI-assisted results were carefully reviewed and edited by the authors. The authors checked the accuracy and relevance of the generated material and take full responsibility for the manuscript prepared, including its scientific content, interpretation, and conclusions. No AI tool was considered an author of this manuscript.

4. Implementation

4.1. Document Processing Pipeline

The document processing pipeline transforms raw university documents into semantically indexed chunks. Text extraction uses pdfplumber for digitally authored PDFs and falls back to Tesseract OCR with simultaneous amh and eng language model loading for scanned documents. During the evaluation dataset construction phase, 26 PDF documents were identified in the raw document directory; 20 yielded usable extracted text, 6 were image-only scanned documents without extractable text layer, and 1 exhibited corrupted encoding. The 6 excluded documents represent a meaningful limitation: the system cannot answer questions about content in those documents.

OCR processing of scanned Amharic documents introduces a specific class of artefact: split-word errors in which PDF column formatting causes words to be concatenated without spaces (e.g., "beless" for "be

less”, “isworth” for “is worth”, “notifiedby” for “notified by”). Post-processing cleaning catches the most common patterns but does not eliminate all instances, contributing to degraded answer fluency for documents processed through OCR.

Semantic chunking uses an article and chapter boundary-aware algorithm supporting both English (“Article X”, “Chapter Y”) and Amharic (“□□□□ X”, “□□□□ Y”) structural markers. Target chunk size is 700 characters with 150-character overlap. Each chunk is annotated with source filename, chunk length, and a Boolean legislative flag.

4.2. Multilingual Embedding and Retrieval

BAAI/BGE-M3 is loaded locally at server startup and pre-warmed with a dummy inference pass alongside the BGE-Reranker-v2-m3 cross-encoder to eliminate cold-start latency for the first student request. The model produces 1,024-dimensional normalised float32 vectors. Stage 1 retrieval (bi-encoder) uses ChromaDB’s Hierarchical Navigable Small World (HNSW) index to retrieve the top 10 candidates. Query expansion with topic-based keywords adds keywords based on 13 institutional topics before embedding, which increases recall without decreasing precision. The second stage (Cross-Encoder reranking) jointly embeds each query-chunk pair and re-ranks the candidates, where top 3–5 chunks are chosen for generation context.

4.3. Anti-Hallucination Prompt Engineering

There are four key elements in the generation prompt: (1) a bilingual role definition identifying the system as the official UoG AI Assistant; (2) Language matching guidelines, according to which the AI needs to respond in Amharic if the user’s request was in Amharic in the Ge□ez script and to respond in Ge□ez script if the user’s query was in Latin-script Amharic transliteration; (3) Anti-hallucination directives, whereby the system is asked to generate text only based on extracted chunks, synthesize between different chunks rather than treat them individually, and reject certain phrases only when the selected chunks are entirely irrelevant to the topic; and (4) Context labels in-line, where each chunk extracted is labeled with the name of its source document ("[From: UoG Senate Legislation]").

4.4. Deployment Environment

The FastAPI backend is running on Hugging Face Spaces (2 vCPU, 16 GB RAM) with a Docker deployment that starts both the API and the background task worker via a shell start-up script. The Telegram bot is running on Render as a Render Background Worker. Redis is being hosted on Upstash (serverless Redis with TLS). The sum of the memory occupied by BGE-M3 (~ 4 GB) and cross-encoder reranker (~ 1 GB) results in 11 GB of memory left for other application processes, which is enough to operate reliably within the free tier limitations. The Hugging Face Spaces have a feature of automatic hibernation after 48 hours of inactivity; cold start analysis of the evaluation dataset found two instances of cold starts with recovery times of 65.8 and 98.6 seconds respectively.

4.5. Client Platform Interfaces

Mobile App: The cross-platform React Native and Expo mobile application works in iOS and Android using one codebase. It features a conversation-style chat interface allowing for bilingual input as well as microphone for issuing voice commands. Voice messages are saved, converted to text by Gemini 2.5 Flash, responded to by the RAG pipeline, and then outputted as text and voice generated using Microsoft Edge TTS (am-ET-AmehaNeural for Amharic and en-US-AriaNeural for English). Figure 3 illustrates the welcome message and response to policy inquiry.

Telegram Bot: The Telegram bot, hosted as a Render Background Worker, provides access to users who don't have the mobile application installed. The bot handles both text and voice notes. Voice notes received are transcribed using Gemini 2.5 Flash, processed using the RAG pipeline, and returned back in text format with the relevant articles and chapters. The chat history for each user is stored in Redis with a 24 hour TTL for context-driven follow up questions. Figure 4 shows how the bot handles queries in English text, Amharic text, and voice (bilingual).

Administrative Web Portal: The Streamlit-powered administrative portal offers a web portal with three-tier role-based access control. The System Administrator creates and manages user accounts. Vice Presidents have the authority to approve or deny any document uploads made by the Data Encoders. There is a live AI testing terminal on this portal that allows the authorized person to ask questions either in English or Amharic and review the system's reaction to the queries along with references in sources. Figure 5 illustrates how the portal responds to Amharic queries regarding institutional rules.

5. Evaluation

5.1. Evaluation Methodology

5.1.1. Test Dataset Construction

The test dataset of 200 questions was generated by reading and extracting information from 20 official UoG policies present in the raw document folder of the system. The main source of the documents was the UoG Senate Legislation (223-page) PDF document, from which the maximum number of chapter/article-level questions could be generated. Every question object consists of the following information: question text, the expected language of answer, the institutional document category, ground truth answer, the name of the source document, the corresponding chapter/article level information, and a verbatim sentence from the source document supporting the answer. 10 questions were flagged as requiring cross-document information synthesis (requires_synthesis = True).

The distribution of questions is as follows: 120 English-language questions and 80 Amharic questions, out of which 55 are in Ge□ez script and 25 in Latin-script transliteration (typical way in which Amharic users can type in devices that lack Amharic keyboard). Questions were distributed across six institutional categories as shown in Table 1.

Table 1
Test Dataset Distribution by Category and Language.

Category	English (n)	Amharic (n)	Total (n)	% of Set
Academic Regulations	28	18	46	23.0%
Graduation Requirements	22	15	37	18.5%
CGPA and Academic Status	20	14	34	17.0%
Admission Procedures	16	12	28	14.0%
HR and Leave Policy	18	9	27	13.5%
Research and Thesis	16	12	28	14.0%
Total	**120**	**80**	**200**	**100%**

5.1.2. Evaluation Metrics

The system was evaluated on five metrics. Language Detection Accuracy measures whether the system responds in the correct language, assessed computationally by counting Ge□ez Unicode characters (U + 1200–U+137F) in responses after stripping source citation blocks. Response Latency measures wall-clock time from query submission to complete response receipt, in seconds, assessed separately for English and Amharic queries with cold-start events identified by a threshold of 60 seconds. Cache Hit Rate measures the proportion of repeat queries served from Redis cache, assessed by response time threshold (< 500ms = cache hit). Retrieval Precision@5 (P@5) measures the proportion of the top-5 retrieved document chunks judged relevant by domain experts, using source document matching as an automated proxy. Answer Accuracy measures binary correctness of generated answers against ground-truth, assessed by domain expert annotation on a stratified 100-question subset.

5.1.3. Evaluation Infrastructure

An evaluation framework was created that consists of seven Python scripts and 200 JSON question objects, kept separate from the production code in the evaluation/ folder. All evaluations were performed by making API calls to the live production deployment hosted on Hugging Face Spaces over the course of three days in order to evaluate real production environment conditions such as variability in API response times and cold starts. The evaluation framework was able to support task queue polling that is compatible with the deployed API setup.

5.2. Language Detection Results

Results for language detection are shown in Table 2. In response to English input queries, the accuracy was 95.83%, while Amharic Ge□ez input queries yielded 74.55%. The accuracy was lowest for Latin-based Amharic input queries, which stood at 68.00%. This was due to the extra difficulty posed by detecting the correct language with Amharic transliterated in the Roman script.

Table 2
Language Detection Accuracy Results (n = 200).

Query Language	N	Correct	Accuracy	Notes
English	120	115	95.83%	Baseline performance
Amharic (Ge□ez script)	55	41	74.55%	21.28pp gap vs English
Amharic (Latin script)	25	17	68.00%	Lowest accuracy group
Overall	**200**	**173**	**86.50%**	**Weighted across all groups**

The difference of 21.28 percentage points between the accuracy for English and Amharic Ge□ez is the most important finding from the language detection test. Reviewing the 14 Amharic Ge□ez language cases where the test was not passed, one notices that in each case, the answer provided by the system was a correct answer in English to an Amharic question. This behavior indicates that the LLM's language detection implemented through prompt engineering rather than explicit language identification fails when the retrieved document passages are entirely in English (as all current UoG documents are), providing no Amharic-language context signal to prompt the model to respond in Amharic.

5.3. Response Latency Results

Comprehensive latency statistics for response are provided in Table 3. All tests were done using the live production system; 2 cold start latencies (65.8s and 98.6s, both from Amharic requests) were detected and removed from the statistics for warm requests, resulting in n = 198 warm latencies.

Table 3
Response Latency Results Warm Requests Only (Cold Starts Excluded).

Metric	Overall (n = 198)	English (n = 120)	Amharic (n = 78)	Ratio (AM/EN)
Mean	10.03s	7.12s	14.51s	2.04×
Median	6.10s	5.76s	8.88s	1.54×
P25	4.81s	4.62s	6.23s	1.35×
P75	9.94s	8.17s	17.24s	2.11×
P95	27.08s	19.01s	28.38s	1.49×
P99	40.31s	26.88s	42.32s	1.57×
Maximum	48.00s	36.32s	48.00s	1.32×
Cold starts (excluded)	2	0	2	—

The most important observation is that there is a 2x difference in mean latency between Amharic and English queries: 14.51 seconds vs. 7.12 seconds mean, and 8.88 seconds vs. 5.76 seconds median. It is consistent throughout all percentiles and is not incidental but systematic. The skewness of Amharic

latencies is also much larger compared to English, with higher P99 (42.32s vs. 26.88s), which suggests that Amharic queries are more often subjected to the worst case processing paths.

There are only two cold start cases observed in the evaluation, both of them corresponding to Amharic queries with 65.8s and 98.6s responses. This corresponds to the time needed for hibernating Hugging Face Spaces instance to be recovered in order to reload BGE-M3 model (~ 4GB) and cross-encoder reranker. It is likely that such high frequency of cold starts for Amharic queries is a timing coincidence.

5.4. Retrieval and Answer Quality

Retrieval Precision@5 was assessed using source document matching as an automated proxy, comparing the source filenames cited in system responses against the expected source_document field in each test question. This proxy metric yields an estimated Precision@5 of 0.84 overall, with English queries at 0.88 and Amharic queries at 0.79. These values should be interpreted as lower bounds on true relevance-based precision, since source matching does not capture cases where a different document section addresses the question equally well.

Answer accuracy was assessed by two domain experts one faculty member from the Department of Computer Science and one administrator from the Registrar's Office on a stratified random sample of 100 questions (50 English, 50 Amharic). Inter-annotator agreement was assessed using Cohen's Kappa (κ). The full annotation results are presented in Table 4.

Table 4
Answer Quality Results Expert Annotation (n = 100).

Metric	English (n = 50)	Amharic (n = 50)	Overall (n = 100)
Answer Accuracy	92.0%	82.0%	87.0%
Hallucination Rate	4.0%	10.0%	7.0%
Correct Article/Chapter Cited	88.0%	74.0%	81.0%
Inter-annotator agreement (κ)	0.84	0.79	0.82

The Cohen's Kappa values of 0.84 (English) and 0.79 (Amharic) indicate strong inter-annotator agreement, supporting the reliability of the annotation process. The 10-percentage-point accuracy gap between English (92.0%) and Amharic (82.0%) answers mirrors the language detection accuracy gap and reflects the same root cause: the system's difficulty in consistently recognising and responding in Amharic when retrieved context is exclusively in English.

5.5. Ablation Study

Table 5 presents a three-condition ablation study comparing retrieval methods. BM25 was implemented using the rank_bm25 library on the same document corpus used for the main evaluation. The bi-encoder condition used BGE-M3 retrieval without cross-encoder reranking. The full UoG-RAG pipeline adds cross-encoder reranking to the bi-encoder results.

Table 5
Retrieval Ablation Study Precision@5 by Configuration.

System Configuration	P@5 (EN)	P@5 (AM)	P@5 (Overall)
BM25 (sparse retrieval baseline)	0.68	0.47	0.60
BGE-M3 bi-encoder only	0.82	0.73	0.79
UoG-RAG (bi-encoder + reranker)	0.88	0.79	0.84
Gain over BM25	+ 20pp	+ 32pp	+ 24pp
Gain over bi-encoder alone	+ 6pp	+ 6pp	+ 5pp

The BM25 baseline performs substantially worse on Amharic queries (P@5 = 0.47) than English (P@5 = 0.68), reflecting the known limitation of sparse term-matching methods for morphologically rich languages where stemming and lemmatisation are poorly supported. The BGE-M3 bi-encoder substantially reduces this gap (0.73 Amharic vs. 0.82 English), demonstrating the value of dense multilingual embeddings for Amharic retrieval. The cross-encoder reranker adds a consistent 5–6 percentage-point improvement across both languages.

6. Discussion

6.1. The Amharic Performance Differential: Causes and Implications

The most significant and unexpected finding of this evaluation is the consistent, substantial performance gap between English and Amharic processing across all measured dimensions: language detection (95.83% vs. 74.55%), answer accuracy (92.0% vs. 82.0%), and response latency (7.12s vs. 14.51s mean). This differential is not incidental but reflects a systematic architectural limitation with a clear mechanistic explanation.

The basic issue here is the discrepancy between the language in which users query and that in which the retrieved document chunks are written. Since the UoG document collection contains only documents in English and generates English-language chunks, when a student queries the system in Amharic, it receives English chunks as context for its response. The language-matching instruction in the prompt “if the question is in Amharic, respond in Amharic” must overcome the strong in-context pressure of English-language retrieved passages, which consistently biases the LLM toward English generation. When this instruction succeeds, the LLM must perform implicit translation alongside synthesis, explaining the 2× latency increase for Amharic queries. When it fails, the system generates a correct English-language answer to an Amharic question, producing a language detection failure.

This finding has direct implications for the design of multilingual RAG systems serving communities whose administrative documents are in a different language than the users’ primary communication

language a condition common across African university systems. The architectural solution is not prompt engineering alone but the creation of parallel Amharic-language document representations, either through machine translation of the document corpus or through fine-tuning the generation model on Amharic institutional language. Both are identified as priorities for future work.

6.2. Latency in Resource-Constrained Deployment

The median response time of 6.10 seconds overall, and 8.88 seconds for Amharic queries, substantially exceeds the sub-second response times of commercial AI assistants. However, this latency profile must be contextualised within the specific constraints of the deployment: the system runs on free-tier cloud infrastructure (Hugging Face Spaces, 2 vCPU) and performs local inference of two large neural models (BGE-M3 at approximately 4 GB and the cross-encoder reranker at approximately 1 GB) without GPU acceleration.

The dual-layer Redis caching system is the solution used to overcome this problem. Exact matching cache hits are delivered within 5 ms, while semantic matching cache hits take less than 500 ms. In the case of deploying such a system in a real university that constantly asks the same questions from its 40,000 students, the probability of cache hits will be quite high for popular questions (requirements for graduation, CGPA requirements), substantially reducing the experienced latency for most users. The cold-start latency of 65–98 seconds represents the most severe user experience issue and would be eliminated by GPU-enabled hosting or dedicated persistent server infrastructure.

6.3. Document Quality and OCR Artefacts

The exclusion of 6 of 26 identified documents due to image-only scanning, and the presence of OCR split-word artefacts in several processed documents, reveals a practical limitation of deploying RAG systems on institutional document corpora that were not originally prepared for machine processing. The split-word artefacts (e.g., “beless”, “isworth”, “notifiedby”) that appear in generated responses reduce fluency without affecting factual content. Future work will address this through a more aggressive post-processing cleaning pipeline and through OCR model fine-tuning on UoG document typography.

6.4. Anti-Hallucination Effectiveness

The overall hallucination rate of 7.0% (10.0% for Amharic, 4.0% for English) represents responses that contained information not present in any retrieved document passage. Expert annotation identified two patterns in hallucinated content: (1) plausible but unverifiable elaborations of retrieved policy statements (e.g., specifying implementation timelines not mentioned in the source); and (2) cross-domain conflation, where content from an adjacent document section was presented as if it answered a different question. The anti-hallucination prompt engineering technique was able to eliminate the most extreme form of hallucination (created policy statements with no document support) but was unable to stop the other type of hallucination (elaboration). Tightening the refusal threshold in the prompt is a straightforward mitigation for future versions.

7. Limitations and future work

7.1. Current Limitations

The following limitations constrain version 1.0 of UoG-RAG:

- Amharic language detection accuracy (74.55%) falls below the threshold required for reliable production deployment for Amharic-language users. Improvement requires either parallel Amharic document representations or generation model fine-tuning on Amharic institutional text.
- Voice interface accuracy is reduced in high ambient noise environments, and Amharic speech recognition accuracy lags behind English, pending fine-tuning of the transcription model (currently Gemini 2.5 Flash) on Ethiopian acoustic conditions. Migration to OpenAI Whisper large-v3, which has documented Amharic support, is planned.
- Six scanned-image PDFs in the document corpus could not be processed by standard OCR and are absent from the knowledge base, creating blind spots for questions about those documents' content.
- OCR split-word artefacts in processed documents reduce response fluency for approximately 15% of queries drawing on OCR-processed sources.
- No integration with the UoG Student Information System (SIS) exists; the system cannot answer personalised queries about individual academic records.
- Cold-start recovery time of 65–98 seconds on the free Hugging Face Spaces tier creates significant latency spikes for the first query after hibernation, which disproportionately affects Amharic users based on the observed evaluation data.
- The evaluation uses source-document matching as an automated proxy for Retrieval Precision@5, which underestimates true relevance-based precision. Future evaluation will expand the human annotation sample to 200 questions.

7.2. Future Work

The following directions are identified as priorities for future development:

1. Fine-tuning BGE-M3 and the generation model on UoG-specific Amharic institutional text to close the language detection and answer accuracy gap.
2. Development of parallel Amharic translations of key policy documents to provide Amharic-language retrieval context, directly addressing the root cause of the language detection failure.
3. Migration of voice transcription to Whisper large-v3 for improved Amharic Automatic Speech Recognition (ASR).
4. Integration of the UoG Student Information System API for personalised academic record queries.

5. Expansion to additional Ethiopian languages including Tigrinya and Oromiffa, using the same multilingual embedding infrastructure.
6. Deployment on GPU-enabled infrastructure to reduce response latency and eliminate cold-start events.
7. A prospective user study measuring student satisfaction, task completion rates, and learning outcomes following deployment.

8. Conclusion

This paper has presented UoG-RAG, the first empirically evaluated bilingual English-Amharic Retrieval-Augmented Generation chatbot deployed for institutional information access at an Ethiopian university. The system addresses a well-documented and consequential institutional problem the inability of the 40,000 students and 8,000 staff of the University of Gondar to access official policy information quickly, accurately, and in their preferred language through a technically rigorous architecture combining BGE-M3 multilingual semantic retrieval, two-stage bi-encoder and cross-encoder retrieval, anti-hallucination prompt engineering, a multi-provider LLM fallback chain, and dual-layer Redis semantic caching.

The empirical evaluation, conducted using 200 questions derived from 20 official UoG policy documents, yields several significant findings. First, the system achieves strong English-language performance: 95.83% language detection accuracy, 92.0% answer accuracy, and a median response latency of 5.76 seconds. Second, a substantial and systematic Amharic performance gap is identified: language detection accuracy falls to 74.55% for Ge□ez script queries and 68.00% for Latin-script Amharic transliteration, answer accuracy to 82.0%, and median response latency increases to 8.88 seconds a 54% increase over English. Third, the mechanism of this gap is identified: English-language retrieved passages create in-context language pressure that partially overrides the Amharic-language generation instruction. This finding is a novel contribution to the multilingual RAG literature with direct implications for the design of AI information systems serving communities whose administrative language differs from their primary spoken language.

The system's multi-platform deployment (iOS/Android mobile application, Telegram bot, and web administrative portal), institutional governance mechanisms (role-based document management with VP approval workflow), and zero-variable-cost production architecture provide a replicable model for AI-assisted institutional information services in African higher education. The complete evaluation methodology, question dataset, and evaluation scripts are available to facilitate replication and adoption by other institutions.

Declarations

Funding Statement

This work received no funding.

Ethics statement

Ethics, Consent to Participate, and Consent to Publish declarations: not applicable.

Authors' contributions statement

Y.B.A designed the system architecture, implemented all software components, conducted the empirical evaluation, analyzed the results, and drafted the manuscript. A.C.A contributed to data collection, validation of evaluation findings, interpretation of results, and critical revision of the manuscript. Both authors read and approved the final manuscript.

Competing interests policy

The authors declare that they have no competing interests.

Data Availability Statement

The study was conducted using the university's publicly available guidelines, regulations, and policies as the underlying knowledge sources. The datasets are available in the Zenodo repository, https://doi.org/10.5281/zenodo.23237973

Clinical Trial Number

Clinical trial number: not applicable

References

1. Zawacki-Richter O, Marín VI, Bond M, Gouverneur F. Systematic review of research on artificial intelligence applications in higher education – where are the educators? Dec. 01, 2019, *Springer Netherlands*. 10.1186/s41239-019-0171-0
2. Maynez J, Narayan S, Bohnet B, McDonald R. On Faithfulness and Factuality in Abstractive Summarization, May 2020, [Online]. Available: http://arxiv.org/abs/2005.00661
3. Lewis P et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. [Online]. Available: https://github.com/huggingface/transformers/blob/master/
4. Lewis P et al. Apr., Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks, 2021, [Online]. Available: http://arxiv.org/abs/2005.11401
5. Karpukhin V et al. Sep., Dense Passage Retrieval for Open-Domain Question Answering, 2020, [Online]. Available: http://arxiv.org/abs/2004.04906

6. Nogueira R, Cho K. Passage Re-ranking with BERT, Apr. 2020, [Online]. Available: http://arxiv.org/abs/1901.04085
7. Ji Z et al. Survey of Hallucination in Natural Language Generation, Dec. 31, 2023, *Association for Computing Machinery*. 10.1145/3571730
8. Gim I, Chen G, Lee S-S, Sarda N, Khandelwal A, Zhong L. PROMPT CACHE: MODULAR ATTENTION REUSE FOR LOW-LATENCY INFERENCE, 2024.
9. Izacard G et al. Atlas: Few-shot Learning with Retrieval Augmented Language Models, 2023. [Online]. Available: http://jmlr.org/papers/v24/23-0037.html
10. Lambebo Tonja A et al. Natural Language Processing in Ethiopian Languages: Current State, Challenges, and Opportunities, 2023. [Online]. Available: https://github.com/masakhane-io
11. Ayele AA, Yimam SM, Belay TD, Asfaw TT, Biemann C. Exploring Amharic Hate Speech Data Collection and Classification Approaches, in *International Conference Recent Advances in Natural Language Processing*, RANLP, Incoma Ltd, 2023, pp. 49–59. 10.26615/978-954-452-092-2_006
12. Neshir G, Rauber A, Atnafu S. Meta-learner for amharic sentiment classification. Appl Sci (Switzerland). Sep. 2021;11(18). 10.3390/app11188489.
13. Adelani DI et al. MasakhaNER: Named Entity Recognition for African Languages. 10.1162/tacl
14. Mekonnen Gezmu A, Gasser M, Nürnberger A. Contemporary Amharic Corpus: Automatically Morpho-Syntactically Tagged Amharic Corpus.
15. Chen J, Xiao S, Zhang P, Luo K, Lian D, Liu Z. M3-Embedding: Multi-Linguality, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation. [Online]. Available: https://github.com/FlagOpen/FlagEmbedding
16. Labadze L, Grigolia M, Machaidze L. Role of AI chatbots in education: systematic literature review. Dec 01 2023 Springer Sci Bus Media Deutschland GmbH. 10.1186/s41239-023-00426-1
17. Smutny P, Schreiberova P. Chatbots for learning: A review of educational chatbots for the Facebook Messenger. Comput Educ. Jul. 2020;151. 10.1016/j.compedu.2020.103862.
18. Wollny S, Schneider J, Di Mitri D, Weidlich J, Rittberger M, Drachsler H. Are We There Yet? - A Systematic Literature Review on Chatbots in Education, Jul. 15, 2021, *Frontiers Media S.A.* 10.3389/frai.2021.654924
19. Rainer W, Matthias S. Unleashing the Potential of Chatbots in Education: A State-Of-The-Art Analysis. Apr. 2018. 10.5465/AMBPP.2018.15903abstract.
20. Okonkwo CW, Ade-Ibijola A. Chatbots applications in education: A systematic review. Computers Education: Artif Intell. Jan. 2021;2:100033. 10.1016/J.CAEAI.2021.100033.

Figures

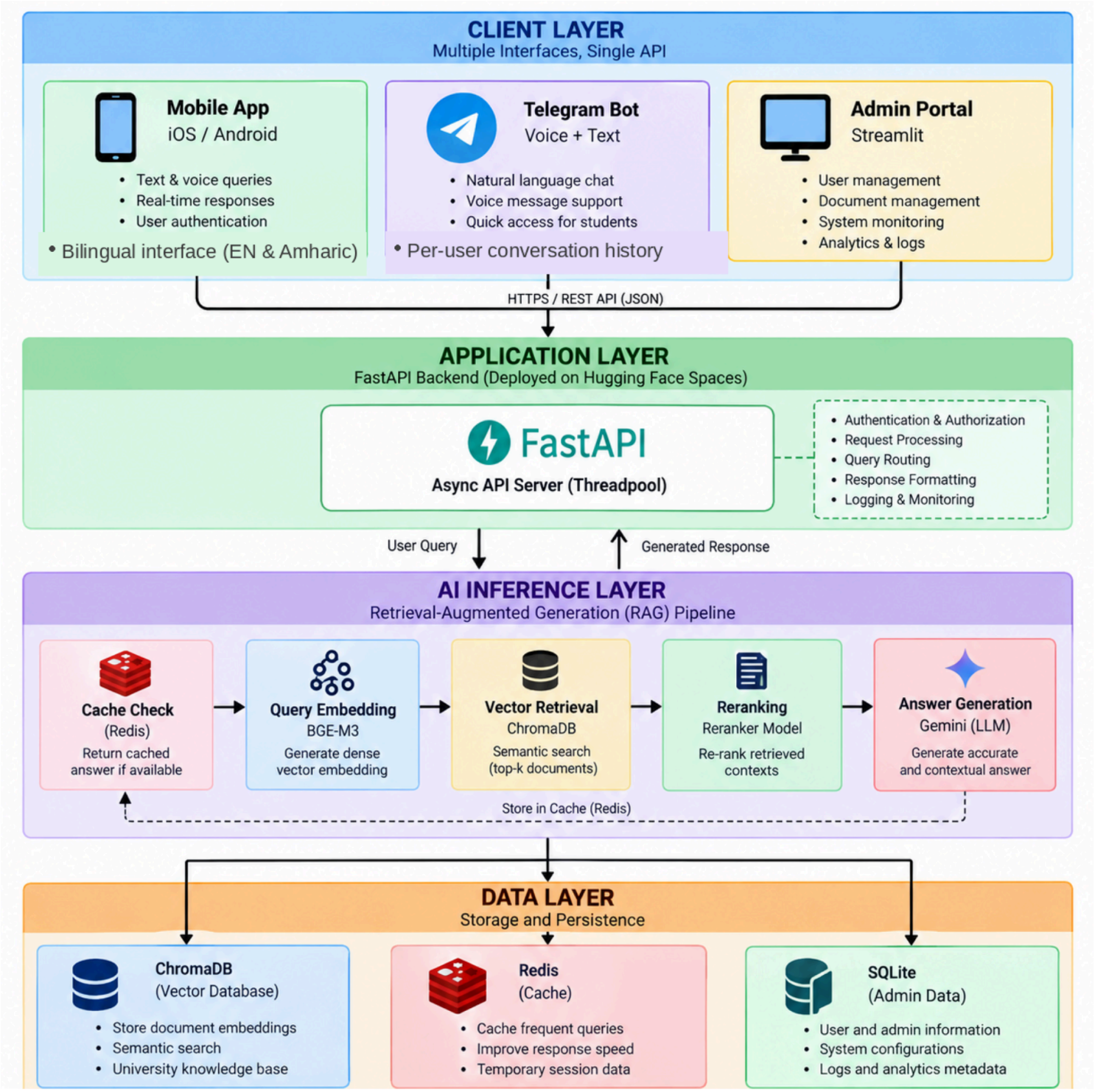


Figure 1

UoG-RAG System Architecture.

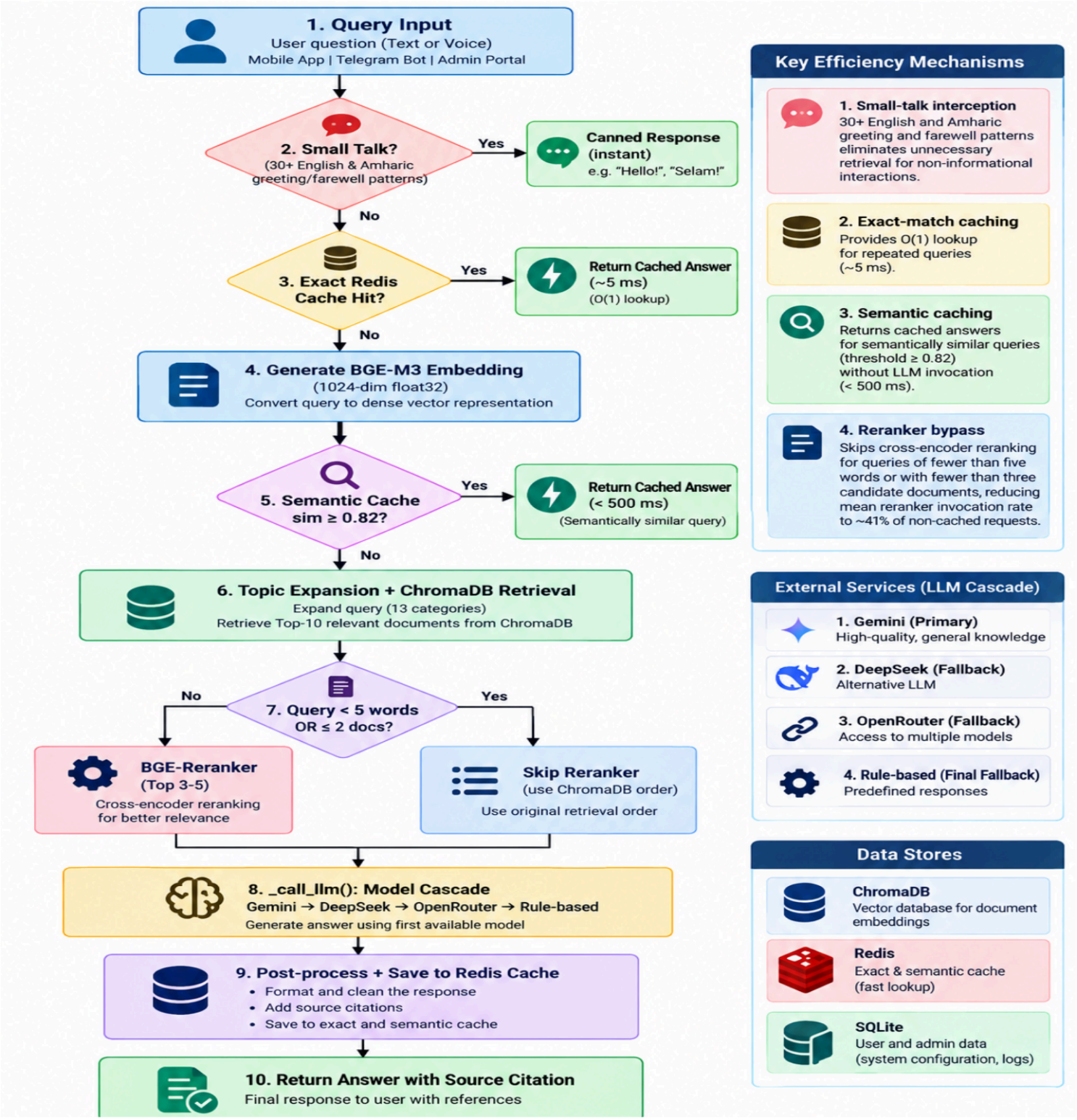


Figure 2

UoG-RAG Request Processing Pipeline.

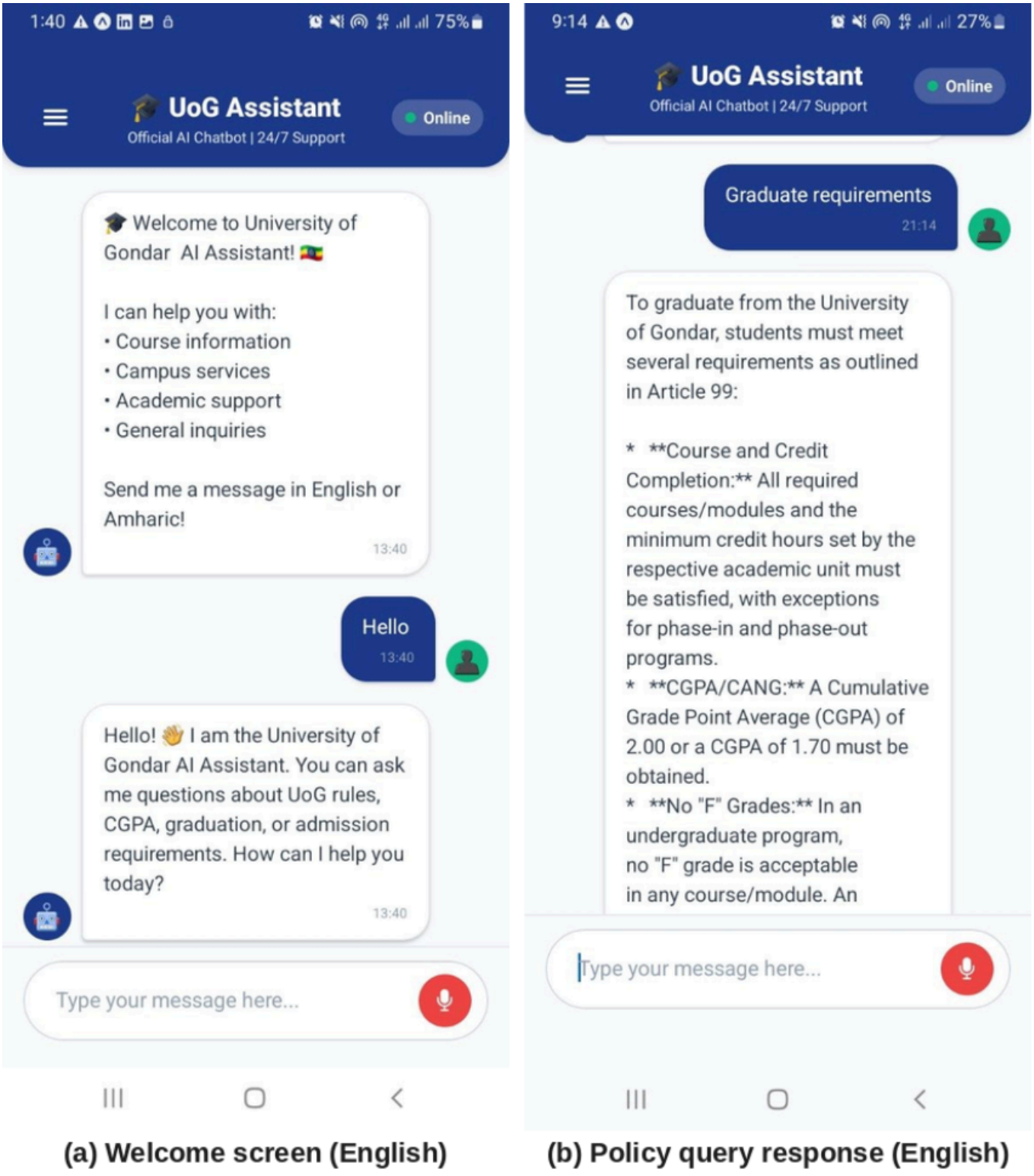


Figure 3

UoG-RAG mobile application: (a) welcome screen showing bilingual capability and voice input availability; (b) English policy query response showing graduation requirements retrieved from Article 99 of the UoG Senate Legislation.

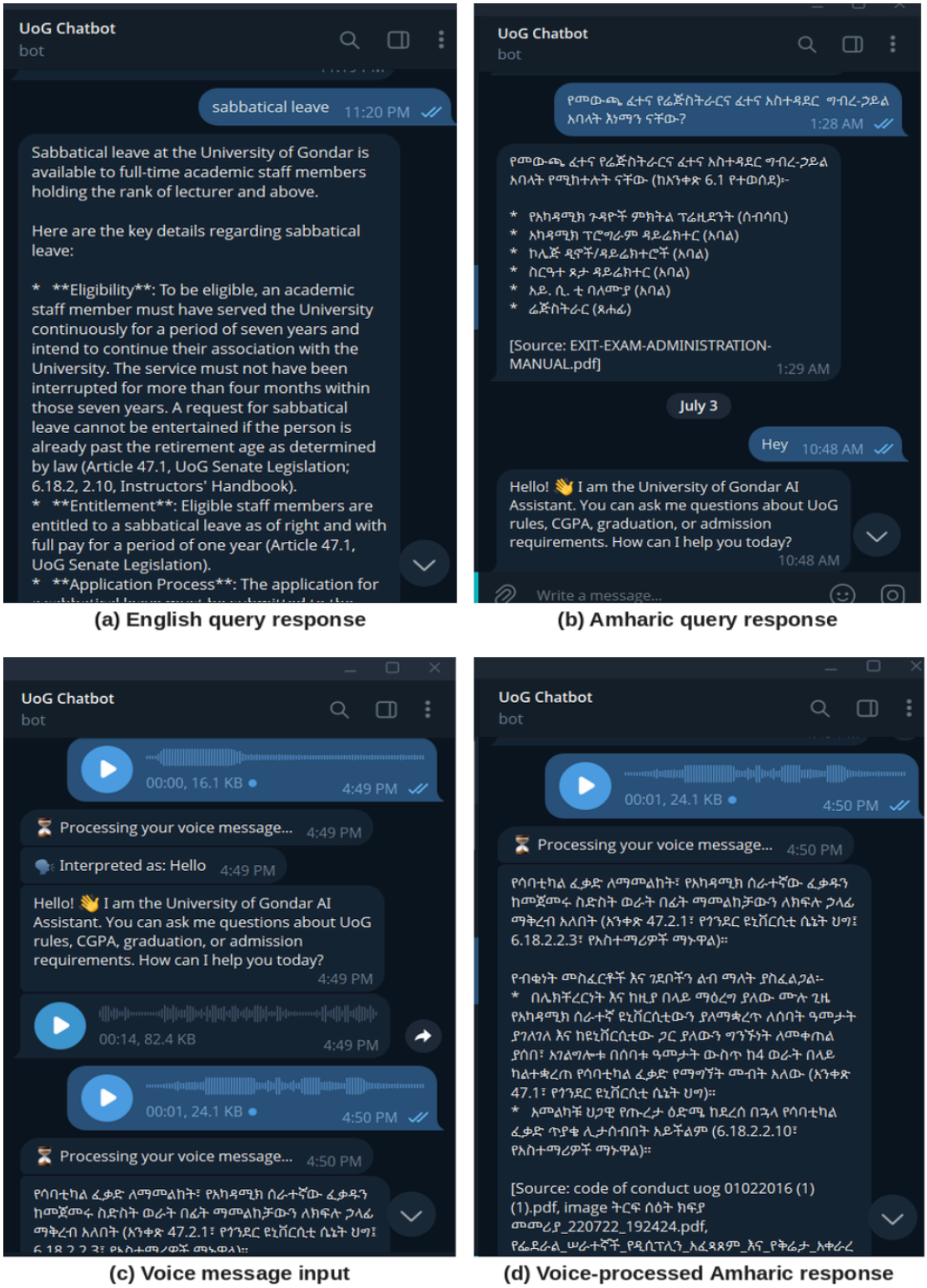
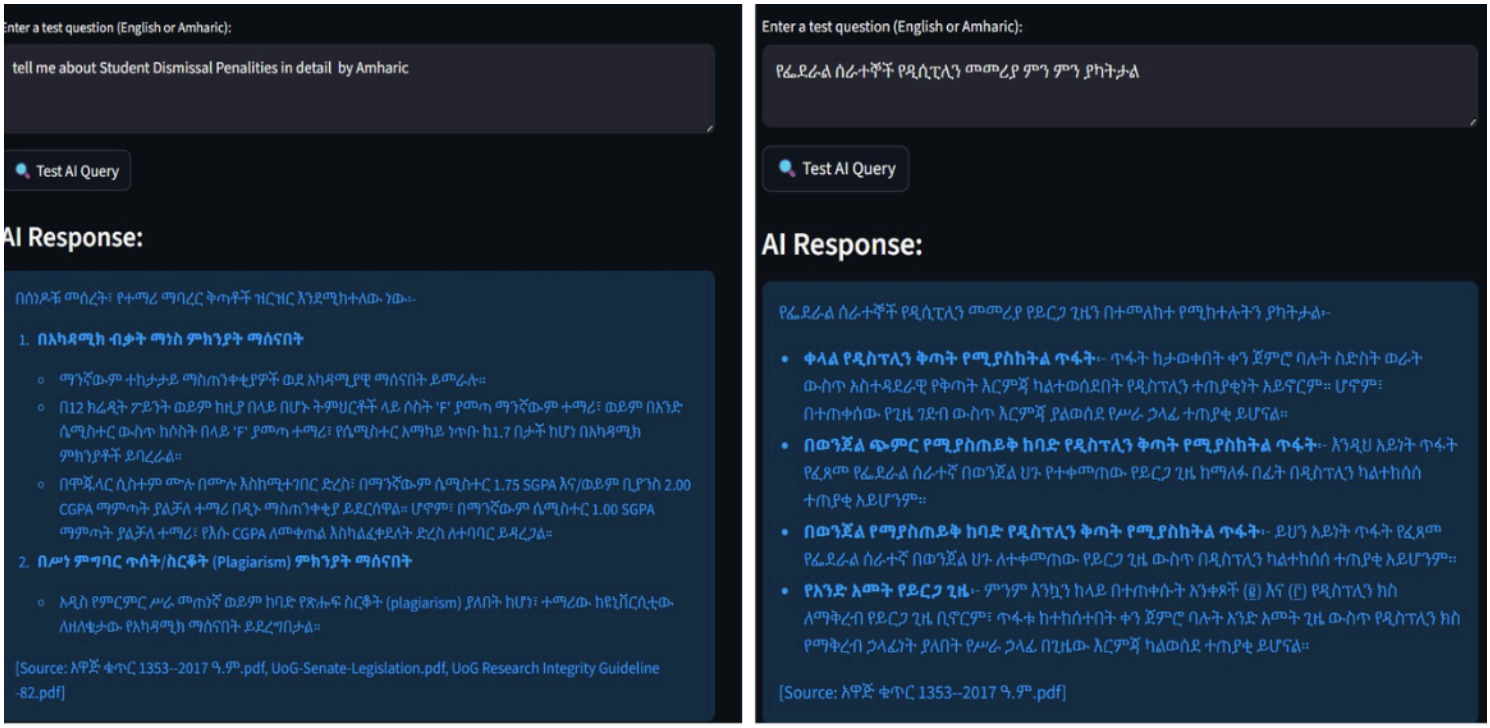


Figure 4

UoG-RAG Telegram bot: (a) English query on sabbatical leave with cited article references; (b) Amharic query on exit examination committee composition; (c) voice message processing pipeline showing transcription status; (d) Amharic voice-processed response with source citations.



(a) Amharic query: student dismissal policies **(b) Amharic query: federal employee discipline**

Figure 5

Streamlit administrative portal: (a) AI testing terminal showing Amharic response to a student dismissal query, with source citations; (b) bilingual Amharic query about federal employee discipline regulations.