SERBIAN JOURNAL OF ELECTRICAL ENGINEERING
Vol. 23, No. 2, June 2026, 227-256
UDC: 004.8:378.4 **DOI: https://doi.org/10.2298/SJEE2602227B**
Original scientific paper

A Multi-Agent RAG Architecture for University Information Management: A Comprehensive Analysis of Performance, Reliability, and Cost

Turgay Tugay Bilgin[1], Yusuf Eren Gül[1], Yücel Aytaç Akgün[1]

Abstract: University regulations and announcements at Bursa Technical University (BTU) are dispersed across PDF files, web pages, and notice boards, causing information retrieval inefficiencies for students and administrative staff. This study presents the development and quantitative evaluation of BTU-Chatbot, an Agentic Retrieval-Augmented Generation (RAG) powered conversational assistant that consolidates fragmented institutional information into a single citation-aware dialogue interface. Ninety-seven PDF documents were processed using PyPDFLoader and regular-expression-based preprocessing, then embedded into 1536-dimensional vectors using OpenAI's text-embedding-ada-v2 model and stored in a 29 MB ChromaDB collection. The vector-based retrieval layer selects the three most relevant passages per query using cosine similarity search. The upper layer implements a LangChain-orchestrated multi-agent ReAct loop, in which the retrieve tool accesses the vector database while the Google_search_univ tool performs domain-restricted searches limited to the "*.btu.edu.tr" domain. GPT-4o-mini, with a 128k context window, serves as the generative backbone. System reliability was measured using the RAGAS metric suite. The best performing run achieved Context Recall = 0.97, Context Precision = 0.99, and F1-RP (the F1 score of Context Recall and Precision) = 0.954, demonstrating near-perfect retrieval accuracy. The average cost per query was 6.6×10^{-5} USD, with 7.6 s of latency for typical 124-token exchanges. BTU-Chatbot demonstrates that an Agentic RAG pipeline can deliver source-grounded, citation-attributed answers to university-specific queries at low operating cost, although further improvements in generation faithfulness are needed before full deployment.

Keywords: Retrieval-Augmented Generation, Conversational Agents, Information Retrieval, Natural Language Processing, Vector Databases.

[1]Bursa Technical University, Bursa, Turkey
turgay.bilgin@btu.edu.tr, https://orcid.org/0000-0002-9245-5728
20360859056@ogr.btu.edu.tr, https://orcid.org/0009-0004-7378-5547
20360859059@ogr.btu.edu.tr, https://orcid.org/0009-0005-5780-8160

Colour versions of the one or more of the figures in this paper are available online at https://sjee.ftn.kg.ac.rs

1 Introduction

In contemporary artificial intelligence applications, the acceleration of digital transformation has made instant and accurate access to information crucial for institutional operations, particularly in higher education environments. Educational institutions face significant challenges in information dissemination, where critical documents such as regulations, academic policies, and administrative procedures are often scattered across multiple platforms and formats. This fragmentation creates substantial inefficiencies in information retrieval processes and reduces user satisfaction. The proliferation of institutional documents in PDF format, combined with the distribution of information across web portals, creates a complex information landscape that requires innovative technological solutions.

Recent advances in Large Language Models (LLMs) and Retrieval-Augmented Generation (RAG) architectures present promising opportunities to address these information accessibility challenges. RAG systems combine the contextual understanding capabilities of modern LLMs with external knowledge retrieval mechanisms, enabling more accurate and verifiable responses to user queries. The integration of multi-agent frameworks further enhances system capabilities by enabling complex reasoning workflows and tool utilization. This research presents the development and comprehensive evaluation of BTU-Chatbot, a Multi-Agent RAG-based conversational system specifically designed to address information retrieval challenges at Bursa Technical University. The system employs a multi-agent ReAct (Reasoning and Acting) framework to process institutional documents and provide contextually appropriate responses with verifiable source citations.

Our approach demonstrates how a modern AI architecture can be effectively adapted to institutional knowledge management requirements while maintaining reliability and cost-effectiveness. The primary contributions of this work include: (1) the development of a domain-specific multi-agent RAG architecture optimized for institutional document processing, (2) a comprehensive quantitative evaluation using established metrics for RAG system performance, (3) the implementation of hallucination reduction techniques achieving measurable improvements in response reliability, and (4) the demonstration of cost-effective deployment strategies suitable for educational institution budgets.

2 Related Work

2.1 Large language models

Large Language Models (LLMs) are neural networks with billions to trillions of parameters that acquire general-purpose language understanding and generation skills through self-supervised pre-training on web-scale corpora. Most

LLMs are built on the Transformer architecture, whose self-attention mechanisms enable the model to capture long-range contextual dependencies efficiently. Introduced by Vaswani et al. in 2017 [1], the Transformer laid the foundation for subsequent waves of large-scale language models. However, LLMs face several fundamental limitations that impact their deployment in institutional settings. Because a model's parameters are fixed after training, its internal knowledge quickly becomes outdated, creating temporal gaps between evolving regulations and the model's frozen cutoff [2]. Moreover, the facts stored within these parameters are implicit and source-agnostic, making it nearly impossible to trace or justify specific claims. Compounding these issues, LLMs can "hallucinate," i.e., generate confident yet spurious answers that undermine trust in automated advice. To overcome these shortcomings, researchers have turned to a variety of complementary techniques. One approach involves integrating knowledge graphs (KGs), which provide structured data that can augment the LLM's knowledge, enabling it to generate more accurate and explainable responses [3]. By linking LLMs with KGs, the model can access and utilize external knowledge to support its claims, making it easier to trace the sources of its outputs. Retrieval-Augmented Generation (RAG) is another technique that enhances the transparency and reliability of LLMs [4, 5]. RAG involves retrieving relevant information from external knowledge sources and incorporating it into the LLM's generation process [5]. This allows the LLM to access up-to-date information without needing to update its internal parameters, which is particularly useful in scenarios where information evolves rapidly [5]. By examining the retrieved documents, users can verify the factual accuracy of the LLM's claims and trace the origin of the information used [4]. R1-Searcher++ is a framework designed to train LLMs to adaptively leverage both internal and external knowledge sources, employing a two-stage training strategy to incentivize dynamic knowledge acquisition, which can further improve the reliability of RAG systems [6]. Direct Retrieval-augmented Optimization (DRO) can also synergize knowledge selection and language models, improving the factuality of LLM generation in knowledge-grounded tasks [7]. This joint training approach ensures that the retriever and the LLM work together effectively, enhancing the accuracy and relevance of the generated content [7]. Meta Knowledge for Retrieval Augmented Large Language Models is a novel data-centric RAG workflow for LLMs, transforming the traditional retrieve-then-read system into a more advanced read-then-retrieve system [8]. This approach constructs meta-knowledge to determine the most useful knowledge to retrieve, improving the synthesis of information from large and diverse sets of documents [8]. By understanding what knowledge is most relevant, the LLM can provide more accurate and reliable answers, enhancing trust and interpretability [8]. Furthermore, techniques such as BlendFilter can address the challenges of noisy knowledge retrieval [5]. BlendFilter integrates query generation blending with

knowledge filtering, improving the accuracy and relevance of the retrieved information [5]. By filtering out irrelevant or inaccurate information, the LLM can provide more reliable and verifiable responses, which is crucial in high-stakes institutional settings [5].

2.2 Retrieval augmented generation framework

Retrieval-Augmented Generation (RAG) extends a large language model by letting it consult an external knowledge store while generating text, thereby supplementing the static information encoded in its parameters [1]. The canonical RAG architecture comprises two interacting modules: a retriever, which is a vector search component that maps the user's query to a dense embedding and returns the k most similar passages from a large corpus (e.g., BTU's regulatory documents or Wikipedia); and a generator, which is a pretrained LLM that receives the retrieved passages as additional context and synthesizes the final answer. The joint probability of generating a response y given an input query x is formally expressed as:

$$p(y \mid x) = \sum_{z \in Z} p(z \mid x) \cdot p(y \mid x, z), \tag{1}$$

where *x* represents the input query, *y* denotes the generated response, *z* represents retrieved documents from the knowledge base $\boldsymbol{Z}$, $\boldsymbol{p}(\boldsymbol{z} \mid \boldsymbol{x})$ is the retrieval probability, and $\boldsymbol{p}(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{z})$ is the generation probability conditioned on both the input and the retrieved context.

In effect, RAG combines the LLM's parametric memory with a dynamic non-parametric memory supplied by the document store, enabling the model to ground responses in up-to-date evidence. This design alleviates several core limitations of standalone LLMs. However, it introduces challenges such as maintaining retrieval quality and ensuring robustness against noisy inputs [9]. Evaluating these systems requires protocols that assess both retrieval and generation aspects comprehensively. One key aspect of RAG evaluation is measuring the accuracy and faithfulness of the generated content [9]. Therefore, evaluation protocols should include metrics that specifically assess the presence of hallucinations and the traceability of information to the retrieved sources.

Relevant benchmarks should focus on domain-specific corpora to reflect real-world applications [9]. For instance, in the medical field, the ability of RAG systems to accurately process and interlink medical information is critical [3]. Evaluation in this context should involve datasets of medical literature and ontologies, with metrics that assess the accuracy and reliability of the generated medical information.

RAGLAB, a modular framework for RAG, offers a research-oriented platform for evaluating different RAG components and configurations [10]. Such frameworks can facilitate comprehensive comparisons and help identify the most

effective strategies for improving retrieval quality and reducing hallucinations. Consequently, the best evaluation protocols and benchmarks for RAG systems in high-stakes applications should include metrics for recall, relevance, passage diversity, hallucination detection, and traceability of citations [9, 10]. These protocols should be applied to domain-specific corpora and should consider the interplay of retrieval and generation components, as well as the impact of different augmentation techniques [3, 7]. Frameworks such as RAGLAB can facilitate comprehensive comparisons and drive the development of more reliable and accurate RAG systems [10].

2.3 Multi-agent systems and ReAct framework

The notion of Agentic Retrieval-Augmented Generation (RAG) elevates an LLM from a passive text generator to an agent that interacts with the external world and executes tasks via dedicated tools. Within this paradigm, the model emits, during problem solving, both (i) reasoning steps (intermediate texts that articulate its train of thought) and (ii) action commands that specify which tool should be invoked and how. In this way, the model forms its own planning and decision-making loop. The approach entered the academic literature in 2022 under the name ReAct (Reason + Act) [11]. ReAct is a prompting method that endows large language models with intertwined reasoning and acting capabilities. A model following ReAct produces successive blocks labelled "Thought:" and "Action:". The Thought segment captures the model's situational assessment and internal logic. The Action segment instructs the surrounding system to execute a concrete operation. Once the model outputs an action in the prescribed format, the host environment intercepts the command, runs the corresponding tool, and feeds the observation back to the model. Through this iterative thought-and-execution cycle, the LLM reaches a solution via multiple steps instead of a single monolithic response. ReAct's chief advantage is the higher performance and reliability it achieves compared with purely chain-of-thought or purely tool-use methods. In our implementation, the retrieve agent specializes in vector database queries, while the Google_search_univ agent handles domain-restricted web searches.

2.4 Educational chatbots

A chatbot is an artificial intelligence program that engages users in natural language dialogue, mimicking conversation with a human interlocutor. Advances in deep learning and natural language processing have matured chatbot technology. The availability of a retrieval-augmented institutional chatbot affects student learning outcomes, help-seeking behavior, and course satisfaction compared with traditional instructor-mediated support [12, 13]. However, instructor-mediated support provides depth and relational interaction that chatbots may lack [12]. AI chatbots offer personalized assistance, which enhances student engagement, autonomy, and satisfaction [12, 14]. Students can

use chatbots to obtain immediate answers to administrative questions about course schedules, materials, and assessment processes [15]. Chatbots can assist students in navigating college websites, predicting cut-off scores based on academic performance, and providing automated information retrieval services [16]. To maximize the benefits of AI chatbots, it is essential to integrate them thoughtfully into the learning environment [17].

AI chatbots should be seen as a supplement to, rather than a replacement for, traditional instructor-mediated support [12]. A blended approach that combines the strengths of both AI chatbots and instructor support may be the most effective way to enhance student learning outcomes, help-seeking behavior, and course satisfaction [18].

3 Methodology

3.1 Data collection and institutional document corpus

The foundation of BTU-Chatbot rests upon a comprehensive corpus of institutional documents sourced from Bursa Technical University's official regulatory framework. The dataset encompasses 97 PDF documents totalling 465 pages, published between 2003 and 2025, representing the complete set of active regulations, directives, and procedural guidelines governing university operations. A detailed statistical breakdown of the document corpus, including its composition by document type and temporal distribution, is provided in **Table 1**.

Table 1
Evaluation Dataset Characteristics.

Characteristic	Value
Total Documents	97 PDF files
Document Types	Administrative (67%), Academic / Student Affair (33%)
Temporal Range	2003-2025
Total Document Length	465 pages
Total Article Count	1,425 individual articles
Language	Turkish (institutional documents)

The document collection was systematically acquired from the Electronic Public Information Management System (KAYSİS), which serves as the official repository for BTU's regulatory documents. The corpus composition includes 33% of documents concerning academic and student affairs and 67% covering administrative or governance topics. Document characteristics reveal significant structural diversity, with individual documents ranging from 1 to 26 pages in length (mean $\mu = 4.7$). The temporal distribution spans 15 years of institutional evolution, capturing policy changes, procedural updates, and regulatory refinements that reflect the university's development trajectory. Language consistency is maintained throughout the corpus, with all documents authored in

Turkish using standardized institutional terminology and formatting conventions. The segmentation process identified 1,425 regulatory articles ($\mu = 15$ per document) using the "MADDE" pattern.

3.2 System architecture and data processing pipeline

The BTU-Chatbot architecture implements a two-phase processing pipeline that transforms unstructured institutional documents into an intelligent conversational interface. The system design follows a modular approach that separates document ingestion, vector indexing, retrieval processing, and response generation into distinct computational stages.

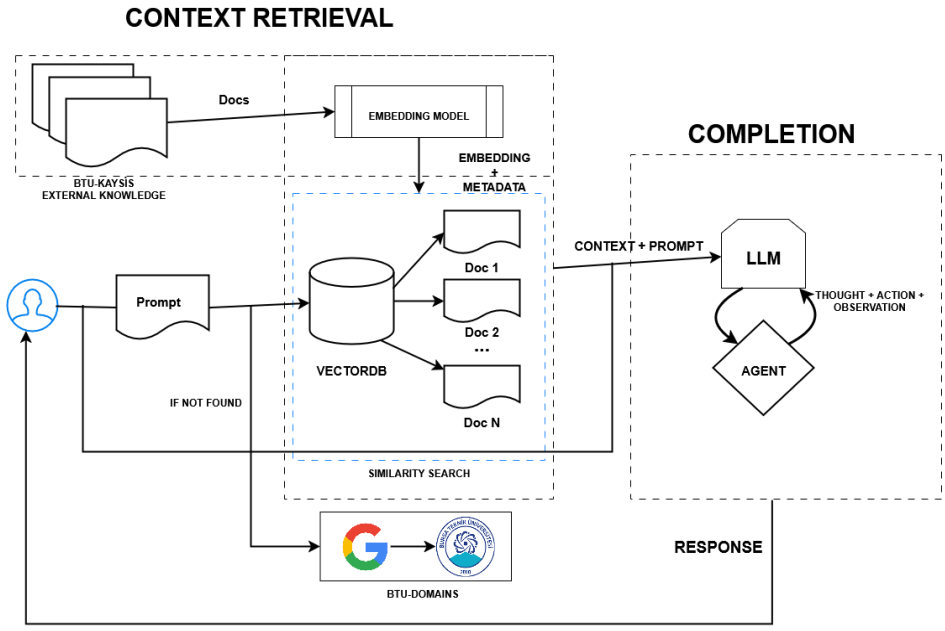

Fig. 1 – *Two-phase architecture of the BTU-Chatbot approach.*

Fig. 1 illustrates the comprehensive system architecture, divided into two primary phases: Context Retrieval and Response Generation. The Context Retrieval phase manages the transformation of raw PDF documents into a searchable vector database, while the Response Generation phase orchestrates the multi-agent reasoning workflow that produces contextually grounded responses. The first phase, Context Retrieval, begins with systematic document ingestion through PyPDFLoader, which extracts textual content while preserving structural metadata essential for source attribution. Document preprocessing employs regular-expression-based segmentation using the "MADDE" pattern. Following segmentation, the OpenAI text-embedding-ada-v2 model transforms each segment into 1536-dimensional vectors that capture semantic relationships within

the institutional domain. These embeddings are stored in ChromaDB, a persistent vector database that implements indexing for efficient similarity search operations. The database configuration maintains approximately 29 MB of vector data. The second phase, Response Generation, orchestrates the multi-agent ReAct framework through LangChain's orchestration capabilities. When users submit queries through the interface, the system simultaneously engages both tools. The retrieve tool performs a vector similarity search against the institutional document database, while the google_search_univ tool conducts domain-restricted web searches limited to the "*.btu.edu.tr" domain for current information. The central LLM (GPT-4o-mini) processes the retrieved context through the ReAct reasoning loop, generating structured responses that include explicit thought processes, tool utilization decisions, and observation integration. Final responses include comprehensive source citations that enable users to verify information authenticity and access original regulatory documents. The specific versions and configuration parameters for all system components, including the language model, embedding model, and vector database, are detailed in **Table 2**.

Table 2
System Configuration Details.

Parameter	Value
Vector Database Configuration	
ChromaDB version	0.4.15
Similarity metric	Cosine similarity
Vector dimensionality	1536
Collection persistence	Local file system
Language Model Configuration	
Primary model	GPT-4o-mini (gpt-4o-mini-2024-07-18)
Context window	128,000 tokens
Temperature	0.7
Maximum output tokens	1000
API timeout	60 seconds
Embedding Model Specifications	
Model	text-embedding-ada-002
Dimensionality	1536
Batch processing	100 documents per request

3.3 Multi-agent ReAct implementation

The multi-agent architecture implements the ReAct framework through LangChain's agent orchestration capabilities, enabling systematic reasoning and tool utilization workflows. The Retrieve tool implements vector similarity search through the ChromaDB interface, executing k-nearest-neighbor queries with

$k = 3$ to balance information completeness with context window constraints. The similarity computation employs the cosine similarity given in (2).

$$\boldsymbol{similarity}(\boldsymbol{q}, \boldsymbol{v}) = \frac{\boldsymbol{q} \cdot \boldsymbol{v}}{\| \boldsymbol{q} \| \ \| \boldsymbol{v} \|}, \tag{2}$$

where $\boldsymbol{q}$ and $\boldsymbol{v}$ represent the query and document embeddings, respectively. The top k most similar documents are selected based on this similarity metric and provided as context to the generation component.

The Google_search_univ tool implements domain-restricted web search through the Google Custom Search API, limiting results to the "*.btu.edu.tr" domain space. This restriction ensures information freshness while maintaining institutional authority and relevance. The tool processes search results through structured parsing that extracts title, URL, and snippet information for integration into the reasoning context. The ReAct loop orchestrates these tools through iterative cycles of reasoning, action execution, and observation processing. Each cycle begins with the model generating a "Thought" that analyzes the current state and determines the next action. This process continues until the model determines that sufficient information has been gathered to produce a comprehensive response. Memory management within the multi-agent framework employs ConversationBufferMemory to maintain complete interaction history. The ConversationBufferMemory object retains the context of successive queries within each session in its chat_history field and feeds it back to the LLM. This keeps consecutive follow-up questions (e.g., "So what are those questions?") anchored to their context.

3.4 Prompt engineering and constraint implementation

The prompt design implements a structured template system that combines system instructions, tool definitions, and conversation history into coherent input representations for the language model. The template architecture enables consistent behavior across interaction sessions while maintaining flexibility for diverse query types and complexity levels. System-level constraints enforce mandatory source citation requirements through explicit prompt instructions that prohibit response generation without appropriate tool utilization. The constraint implementation employs conditional generation logic that requires successful tool execution prior to final answer synthesis. This behavior is formalized as follows:

$$\boldsymbol{P}(\boldsymbol{response} \mid \boldsymbol{Q}) = \begin{cases} \boldsymbol{P}(\boldsymbol{response} \mid \boldsymbol{Q}, \boldsymbol{tools\_used}), & \text{if } \boldsymbol{tools\_executed}; \\ \boldsymbol{P}(\boldsymbol{no\_info\_message} \mid \boldsymbol{Q}), & \text{otherwise.} \end{cases} \tag{3}$$

The prompt structure incorporates few-shot examples that demonstrate proper tool usage patterns and citation formatting requirements. These examples provide behavioral templates that guide model responses toward desired output

formats while maintaining natural language fluency. The few-shot approach proves particularly effective for establishing consistent citation formats that include source document names, article numbers, and relevant excerpts. The prompt engineering strategy also incorporates domain-specific constraints that guide the model toward institutional language patterns and formal communication styles appropriate for regulatory contexts. These constraints help maintain a professional tone while ensuring that responses align with institutional communication standards and user expectations.

3.5 Hallucination detection and mitigation

The system adopts a lightweight, two-stage verification pipeline that relies exclusively on the hand-curated test_cases.json file. The first stage passes each tuple to the EnhancedRagasEvaluator, which converts the data into Dataset objects and returns four RAGAS metrics: Context Precision, Context Recall, Answer Relevancy, and most importantly, Faithfulness, defined as:

$$\text{Faithfulness} = \frac{|\text{supported\_claims}|}{|\text{total\_claims}|}, \tag{4}$$

where the numerator counts claims fully backed by the context. Each metric is reported on a continuous 0–1 scale, quantifying how closely the answer adheres to the supporting evidence. The second stage invokes a Hallucination Checker, which submits the same context-answer pair to GPT-4o-mini with a forced-choice prompt that returns a single token “yes” when every claim is supported and “no” otherwise.

3.6 Performance evaluation framework

The evaluation methodology employs the RAGAS (Retrieval-Augmented Generation Assessment) framework, which provides comprehensive metrics for assessing different components of RAG system performance. The RAGAS suite evaluates retrieval quality, generation faithfulness, response relevance, and overall system effectiveness through distinct measurement approaches that enable systematic performance analysis. Context Recall, as shown in (5), measures the proportion of relevant information successfully retrieved from the knowledge base relative to the ground-truth requirements.

$$\text{Context Recall} = \frac{|\text{retrieved\_relevant} \cap \text{ground\_truth\_relevant}|}{|\text{ground\_truth\_relevant}|}. \tag{5}$$

This metric evaluates the completeness of information retrieval, ensuring that the system successfully identifies all relevant documents needed to answer user queries. High Context Recall values indicate that the retrieval mechanism effectively searches the document collection and selects appropriate source materials.

Context Precision evaluates the proportion of retrieved information that proves relevant to the query, measuring the accuracy of the retrieval filtering process:

$$\text{Context Precision} = \frac{\left|\text{retrieved\_relevant} \cap \text{ground\_truth\_relevant}\right|}{\left|\text{retrieved\_relevant}\right|}. \quad (6)$$

This metric identifies whether the system successfully avoids including irrelevant or tangential information that might confuse the generation process or provide misleading context to users. The F1-RP metric combines recall and precision into a single measure that balances retrieval completeness with accuracy:

$$\text{F1-RP} = 2 \cdot \frac{\text{Context Recall} \cdot \text{Context Precision}}{\text{Context Recall} + \text{Context Precision}}. \quad (7)$$

Answer Relevancy assesses the semantic alignment between generated responses and input queries through embedding similarity analysis:

$$\text{Answer Relevancy} = \frac{1}{\boldsymbol{N}} \sum_{\boldsymbol{i}=1}^{\boldsymbol{N}} \boldsymbol{cos}\left(\text{emb}(\boldsymbol{Q}), \text{emb}(\boldsymbol{A}_i)\right), \quad (8)$$

where $\boldsymbol{Q}$ represents the original query, $\boldsymbol{A_i}$ represents generated questions from the response, and cos denotes the cosine similarity between embedding vectors. This metric ensures that responses remain focused on user questions rather than providing tangential or overly broad information. $\boldsymbol{QA_icos}$

Finally, the four primary scores, Context Recall (R), Context Precision (P), Faithfulness (F), and Answer Relevancy (V), are combined into a single harmonic-mean indicator (HM-all) that captures system-wide quality while penalizing any weak component:

$$\text{HM-all} = \frac{4}{\frac{1}{\text{Recall}} + \frac{1}{\text{Precision}} + \frac{1}{\text{Faithfulness}} + \frac{1}{\text{Relevancy}}}. \quad (9)$$

Since the harmonic mean is driven by the smallest term, a low score in any dimension drags the aggregate down. HM-all therefore answers the practical question: “Is the system simultaneously good at retrieval, grounding, and relevancy?” Higher values signal that no single stage constitutes a bottleneck, whereas a depressed HM-all highlights the need for targeted improvement before the model can be considered reliably trustworthy. The composite RAGAS score is computed with the harmonic mean, as this aggregation cannot increase unless the lowest-performing metric itself improves, thereby preventing high values in other dimensions from obscuring weak components. By contrast, an arithmetic mean would permit large scores to mask deficiencies, yielding a less discriminative assessment of end-to-end system quality. The evaluation

framework also incorporates cost analysis metrics that assess the economic viability of system deployment. Token consumption analysis tracks both input and output token usage across different query types, enabling precise cost calculations based on current language model pricing structures. Latency measurements assess user experience quality by monitoring end-to-end response times across system components.

Having established the methodology for document processing, system architecture, and performance evaluation, the following section describes the practical implementation of these components in a deployable web application.

4 Implementation and User Interface

The BTU-Chatbot implementation employs a modern web application architecture that prioritizes user experience and institutional security requirements. The system integrates multiple technological components to deliver a cohesive conversational interface that supports complex institutional query workflows while maintaining responsive performance across diverse device platforms.

4.1 Frontend architecture and user interface design

The frontend implementation utilizes React.js with contemporary functional component patterns and hooks-based state management to create an intuitive conversational interface optimized for institutional users. The component architecture implements responsive design principles that ensure consistent functionality across desktop computers, tablets, and mobile devices, accommodating the diverse technology preferences of university stakeholders.

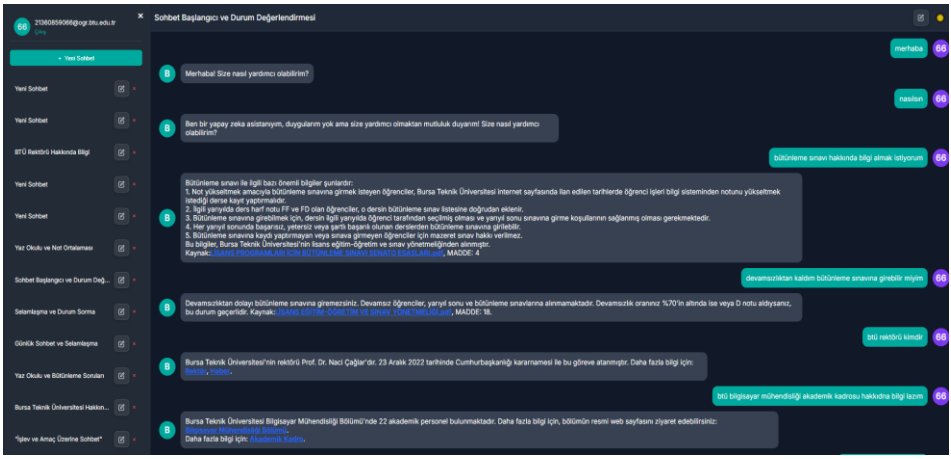

Fig. 2 – *BTU-Chatbot user interface.*

Fig. 2 demonstrates the complete user interface design, showcasing the multi-session conversation management system that enables users to maintain concurrent discussions across different regulatory topics. The interface design incorporates Material Design principles with customizable themes (light and dark mode).

The conversation interface implements several advanced features that enhance user productivity and information verification. The multi-session management system allows users to view conversations by topic, with automatic session titling based on conversation content and manual editing capabilities for user customization.

Citation integration represents a critical interface feature that distinguishes BTU-Chatbot from generic conversational systems. Source references appear as interactive elements within the response text, enabling users to directly access referenced document sections through embedded PDF viewers or external links. This bidirectional linking system supports institutional accountability requirements and enables users to verify information accuracy through primary source consultation.

The responsive design implementation ensures optimal performance across screen resolutions, from 320 px mobile devices to 2560 px desktop displays. The component hierarchy utilizes CSS Grid and Flexbox layouts for adaptive content organization, while progressive loading strategies minimize initial page load times and optimize perceived performance for users with varying network connectivity.

4.2 Authentication and security implementation

User authentication is integrated with Firebase Authentication services to provide enterprise-grade security appropriate for institutional deployments. The authentication system restricts access to verified university email addresses (the btu.edu.tr and @ogr.btu.edu.tr domains), ensuring that system usage remains limited to authorized university personnel and students.

The registration process implements multi-factor verification through email confirmation workflows that prevent unauthorized account creation while maintaining user convenience. Account verification must be completed within defined time limits (5 minutes), after which unverified accounts are automatically purged to maintain database hygiene and security standards.

Session management employs secure token-based authentication with automatic expiration and renewal mechanisms that balance security requirements with user convenience. All user interactions are encrypted during transmission using HTTPS protocols, while conversation data receives AES-256 encryption before storage in Firestore databases.

4.3 Backend services and API architecture

The backend service layer implements RESTful API patterns through FastAPI, providing asynchronous request handling capabilities that support concurrent user sessions without performance degradation.

Database operations utilize a hybrid storage approach that optimizes different data types for their specific access patterns. Firestore manages user authentication, session metadata, and conversation histories with real-time synchronization capabilities, while ChromaDB handles vector storage and similarity search operations with specialized indexing optimized for dense vector retrieval.

4.4 Deployment infrastructure and scalability

The deployment strategy leverages cloud infrastructure services to ensure system availability and performance. Amazon Web Services EC2 instances host the backend processing services with configurations optimized for vector operations and language model inference. The frontend deployment utilizes Vercel's content delivery network to minimize latency.

With the system fully deployed and operational, the following section presents the experimental results obtained through systematic evaluation of all architectural components.

5 Experimental Result and Analysis

5.1 Dataset characteristics and processing metrics

The experimental evaluation utilized a corpus of 97 institutional documents from Bursa Technical University, encompassing regulations, directives, and procedural guidelines published between 2003 and 2025. The document collection represents approximately 29.1 MB of raw text content, segmented into 1,425 individual articles through automated parsing procedures. The document processing pipeline achieved complete extraction success rates across all PDF sources, with no significant extraction errors requiring manual intervention. Vector embedding generation produced 1536-dimensional representations for each segment. The embedding generation process was completed using OpenAI's text-embedding-ada-v2 API. ChromaDB persistence mechanisms successfully maintained index integrity across system restart cycles, confirming operational reliability for production deployment scenarios.

5.2 Retrieval performance analysis

The retrieval subsystem evaluation employed systematic testing across five independent experimental runs, each incorporating different configuration parameters and model settings. The evaluation methodology assessed both

retrieval accuracy and computational efficiency across diverse query types and complexity levels.

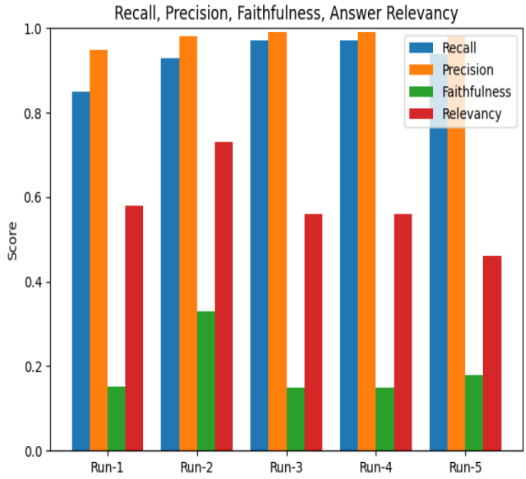

Fig. 3 – *Comparison of basic metrics across runs.*

Fig. 3 presents a comprehensive comparison of the four fundamental RAGAS metrics across all experimental runs. The visualization demonstrates the consistent excellence of retrieval performance, with Recall and Precision maintaining values between 93 % and 99 % across all studies, indicating that required documents were identified almost completely and appropriate fragments were selected accurately. Faithfulness emerges as the weakest component, revealing the presence of hallucination or incomplete information in responses, while Answer Relevancy shows moderate levels, suggesting medium to good conceptual overlap between responses and queries. Context Recall performance was consistently high across all experimental conditions, with values ranging from 0.85 to 0.97. The mean Context Recall of 0.932 ($\sigma = 0.044$) indicates robust retrieval capabilities that successfully identify relevant information sources for the majority of user queries. Context Precision achieved exceptional performance levels, with all experimental runs exceeding 0.95 and a maximum observed value of 0.99. The mean Context Precision of 0.978 ($\sigma = 0.014$) demonstrates effective filtering of irrelevant content and suggests that the embedding model successfully captures semantic relationships within the institutional domain. The combined F1-RP metric achieved a maximum value of 0.980, with a mean across all runs of 0.9542 ($\sigma = 0.031$). These results indicate balanced performance between recall and precision objectives, suggesting that the retrieval system successfully optimizes both information completeness and accuracy simultaneously.

For a complete run-by-run breakdown of all evaluated RAGAS metrics, including F1 scores and the harmonic mean (HM-all), see the detailed retrieval performance tables in **Table 3** and **Table 4**, respectively.

Table 3
Retrieval Performance by Query Category.

Run	Context Recall	Context Precision	Faithfulness	Answer Relevancy	F1-RP	F1-FR	HM-all	Answer Correctness	#
Run-1	0.850	0.950	0.151	0.580	0.897	0.238	0.377	0.50	20
Run-2	0.930	0.980	0.330	0.730	0.954	0.455	0.616	0.50	60
Run-3	0.970	0.990	0.150	0.560	0.980	0.237	0.381	0.65	69
Run-4	0.970	0.990	0.150	0.560	0.980	0.237	0.381	0.54	73
Run-5	0.970	0.980	0.180	0.460	0.960	0.259	0.408	0.50	60

Table 4
Computational Performance Metrics.

Run	#	Mean Input Token	Mean Output Token	Mean Total Token	Total Token	Mean Cost	Total Cost
Run-1	20	30.05	128.20	158.25	3165	0.000081	0.00162
Run-2	60	31.52	152.90	184.41	11065	0.000096	0.00576
Run-3	69	20.07	118.78	136.86	9588	0.000074	0.00511
Run-4	73	20.07	120.04	140.12	10229	0.000075	0.00548
Run-5	60	31.52	104.52	136.03	8162	0.000067	0.00402

Table 5 summarizes the descriptive statistics and 95% confidence intervals for all RAGAS metrics across the five experimental runs. To quantify the reliability of these estimates, confidence intervals were computed using the t-distribution as follows:

Table 5
Descriptive Statistics and 95 % Confidence Intervals for RAGAS Metrics Across Five Experimental Runs.

Metric	Mean	SD	95% CI Lower	95% CI Upper	CV%
Context Recall	0.938	0.052	0.873	1.003	5.6
Context Precision	0.978	0.016	0.958	0.998	1.7
Faithfulness	0.192	0.078	0.095	0.289	40.6
Answer Relevancy	0.578	0.097	0.457	0.699	16.8
F1-RP	0.954	0.034	0.912	0.996	3.6
HM-all	0.433	0.103	0.304	0.561	23.9

$$CI = \bar{x} \pm t_{\alpha/2,n-1} \cdot \frac{s}{\sqrt{n}}, \tag{10}$$

where $\boldsymbol{x}$ is the sample mean, s is the sample standard deviation, n is the number of experimental runs, and $\boldsymbol{t}_{\alpha/2,n-1}$ is the critical value from the $\boldsymbol{t}$-distribution with n–1 degrees of freedom at a significance level $\boldsymbol{\alpha} = 0.05$ (for $n = 5$, $\boldsymbol{t}_{0.025,4} = 2.776$). Context Precision yielded the tightest interval [0.958, 0.998], reflecting highly consistent retrieval filtering across all runs (CV = 1.7%). Context Recall showed similarly stable performance [0.873, 1.003] with a coefficient of variation of 5.6%. In contrast, Faithfulness exhibited the widest confidence interval [0.095, 0.289] and the highest coefficient of variation (CV = 40.6%), confirming that generation grounding is not only low on average but also highly unstable across experimental conditions. Answer Relevancy showed moderate variability (CV = 16.8%), suggesting sensitivity to query formulation differences across runs. It should be noted that with $n = 5$ experimental runs, these confidence intervals are necessarily wide and should be interpreted as indicative rather than definitive estimates; larger-scale replication with additional runs and a broader question set would strengthen these statistical conclusions.

$$\bar{\boldsymbol{x}} \boldsymbol{s} \boldsymbol{n} \boldsymbol{t}_{\boldsymbol{\alpha}/2,\boldsymbol{n}-1} \boldsymbol{n} - 1 \boldsymbol{\alpha} = 0.05 \boldsymbol{n} = 5 \boldsymbol{t}_{0.025,4} = 2.776 .$$

Fig. 4 illustrates the relationship between the harmonic mean of all four fundamental metrics (HM-all) and the actual accuracy rate (1–hallucination rate) across experimental runs.

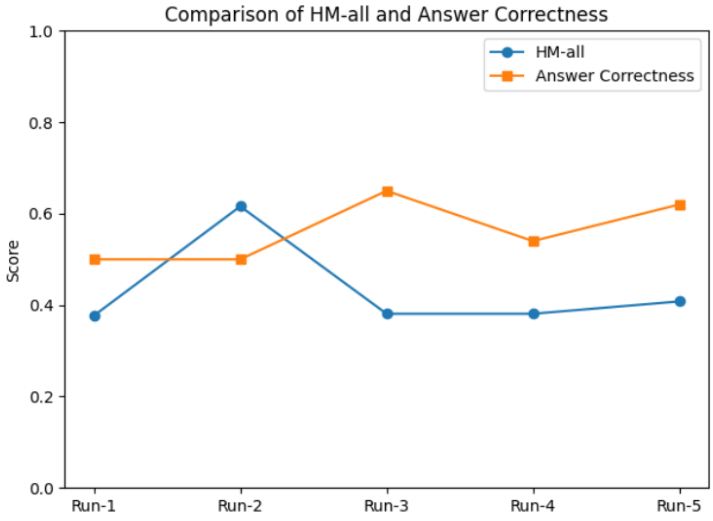

Fig. 4 – *Relationship between HM-all and Answer Correctness.*

This analysis demonstrates whether the comprehensive HM-all score correlates with the accuracy rate that users experience. Notably, Run-3 achieved the highest answer correctness despite relatively low HM-all scores, attributable

to the reduction in hallucination rate to less than 38%. This parallel relationship suggests that improving Faithfulness and reducing hallucinations will enhance both HM-all scores and overall accuracy.

5.3 Generation quality and faithfulness assessment

Response generation quality assessment employs comprehensive evaluation across multiple dimensions, including factual accuracy, source grounding, and response completeness. The evaluation process incorporated both automated metrics and systematic manual review to ensure comprehensive assessment coverage.

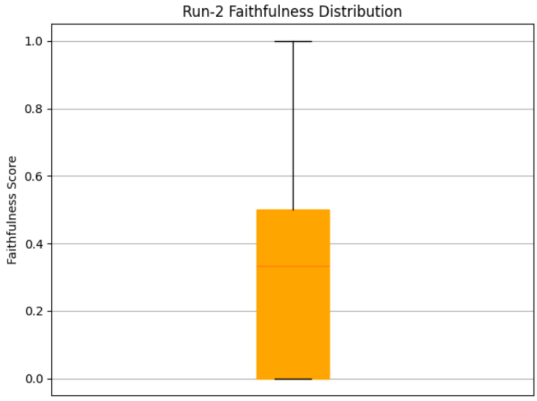

Fig. 5 – *Distribution of Faithfulness and outliers.*

Fig. 5 presents a box-and-whisker plot displaying the distribution of faithfulness scores from the Run-2 test results. The visualization reveals that the median value is approximately 0.33, indicating that most responses demonstrate moderate or low faithfulness to source documents. The upper whisker extends to 1.0, suggesting that some responses achieved complete accuracy when the system successfully utilized the most reliable reference sources. Outlier points represent system queries that achieved optimal source utilization and response reliability. Faithfulness scores demonstrated significant variation across experimental conditions, with values ranging from 0.15 to 0.33. The observed mean of 0.192 ($\sigma = 0.069$) indicates a critical weakness in response grounding. This low score implies that a significant majority of generated responses are not fully supported by the retrieved context.

Several architectural and domain-specific factors likely contribute to this low faithfulness. First, the RAGAS faithfulness metric performs claim-level verification: it decomposes each generated response into individual claims and checks whether each claim is explicitly supported by the retrieved context. When the system retrieves only the three most similar passages ($k = 3$), the retrieved context may cover the core answer but omit peripheral details that the language

model, drawing on its parametric memory, nonetheless includes in its response. These parametric additions are flagged as unsupported claims even when they are factually correct, systematically depressing the faithfulness score. Second, the institutional documents are written in Turkish, while the RAGAS evaluation pipeline operates primarily through English-language LLM reasoning. This cross-lingual verification step may introduce misalignments between the semantic content of retrieved Turkish passages and the English-language claim decomposition process, further reducing faithfulness scores. Third, the generative model (GPT-4o-mini) is prompted to produce comprehensive, citation-rich responses. Longer and more elaborate responses naturally contain a greater number of individual claims, increasing the probability that at least some claims will lack direct contextual support.

Beyond metric methodology, a qualitative inspection of hallucinated responses reveals three recurring patterns. The first pattern is context boundary violations: the model correctly identifies the relevant regulatory article but extends its answer beyond what the retrieved passage explicitly states, inferring procedural implications that are not textually present. The second pattern is temporal extrapolation: when queried about deadlines or dates, the model occasionally generates plausible but unverified dates, particularly for documents whose temporal metadata was incomplete or ambiguous during ingestion. The third pattern is cross-document conflation: in queries requiring synthesis across multiple regulatory articles, the model sometimes merges conditions from two separate articles into a single response, producing a composite answer that is not directly supported by any single retrieved passage. These three patterns collectively suggest that hallucinations in BTU-Chatbot are primarily a result of generative overreach rather than retrieval failure, consistent with the near-perfect retrieval scores observed across all runs.

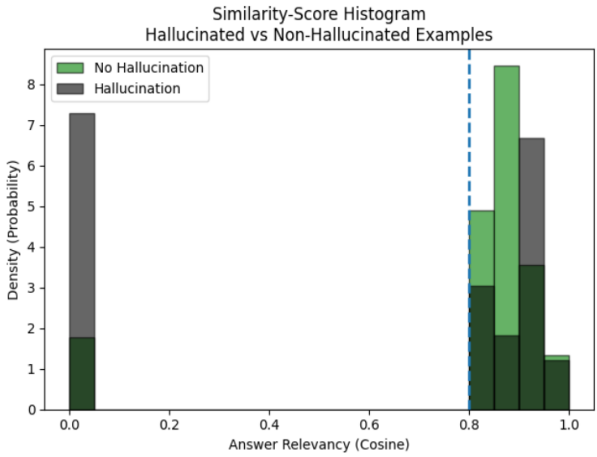

Fig. 6 – *Comparison of similarity-score histograms with and without hallucinations.*

The horizontal axis (x-axis) represents answer relevancy, operationalized as the cosine similarity between the embedding of the model-generated answer and that of the corresponding ground-truth answer. Scores span the closed interval [0, 1], where 1 denotes perfect semantic alignment and 0 indicates no directional overlap. The vertical axis (y-axis) depicts the probability density of these similarity scores rather than raw counts. The histogram has therefore been normalized so that the total area under each distribution integrates to unity, allowing the relative prevalence of hallucinated and non-hallucinated instances to be compared on a common scale regardless of sample size differences. Fig. 6 compares the distribution of similarity scores between hallucinated and non-hallucinated response examples through overlapping histograms. The analysis reveals that most non-hallucinated examples are concentrated in the 0.80 to 1.00 range. A considerable portion of the hallucinated samples are also concentrated in the 0.85-and-above interval, and an accumulation is further observed around 0.0. This indicates that a high similarity score alone does not guarantee that the answer is free of hallucination. Answer Relevancy metrics achieved moderate performance levels, with mean scores of 0.578 ($\sigma = 0.086$) across experimental conditions. The observed variance suggests sensitivity to query formulation and domain specificity, with higher scores achieved for queries that closely match document vocabulary and structure patterns. The harmonic-mean composite metric (HM-all) achieved values ranging from 0.377 to 0.616, with an overall mean of 0.4346 ($\sigma = 0.092$). These results indicate balanced but improvable performance across the complete evaluation framework, with faithfulness limitations serving as the primary constraint on overall system effectiveness.

5.4 Computational efficiency and cost analysis

The cost analysis framework evaluated both computational resource requirements and API usage costs across different language model configurations. The evaluation considered input token consumption, output token generation, and processing latency as primary efficiency indicators. The monetary component of this evaluation is expressed in the equation below, which scales raw token counts to the vendor's 1,000-token pricing granularity.

$$\boldsymbol{Cost}_{\boldsymbol{query}} = \frac{\boldsymbol{n}_{\text{in}}}{1000}\boldsymbol{C}_{\text{in}} + \frac{\boldsymbol{n}_{\text{out}}}{1000}\boldsymbol{C}_{\text{out}}, \tag{11}$$

where $\boldsymbol{n}_{\text{in}}$ is the number of input tokens, $\boldsymbol{n}_{\text{out}}$ is the number of output tokens, $\boldsymbol{C}_{\text{in}}$ is the price per 1,000 input tokens, and $\boldsymbol{C}_{\text{out}}$ is the price per 1,000 output tokens quoted by the API provider. $\boldsymbol{n}_{\text{in}}\boldsymbol{n}_{\text{out}}\boldsymbol{C}_{\text{in}}\boldsymbol{C}_{\text{out}}$.

Fig. 7 demonstrates token consumption patterns across different experimental runs and model configurations. The visualization clearly shows that output tokens constitute the major portion of each request, as evidenced by the substantial orange sections in each bar. This pattern confirms that output token generation

represents the primary driver of both cost and latency in the system architecture. Token consumption analysis revealed mean input token counts of 26.6 per query ($\sigma = 5.39$). Output token generation averaged 124.8 tokens per response ($\sigma = 15.9$).

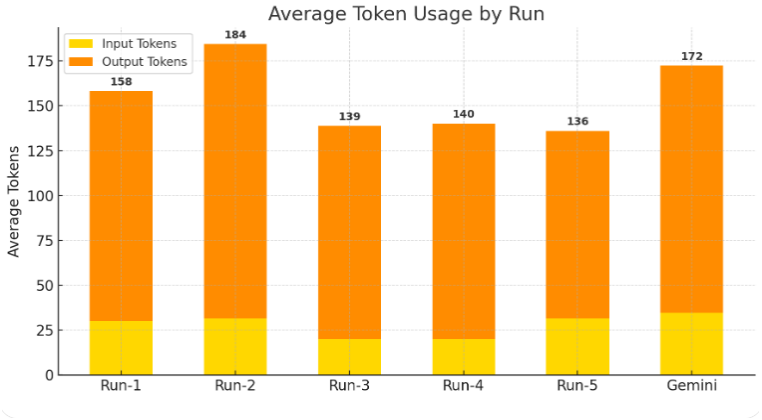

Fig. 7 – *Token usage differences by runs and models.*

Fig. 8 presents latency measurements across different runs and model configurations, demonstrating a direct relationship between total token count and processing delay.

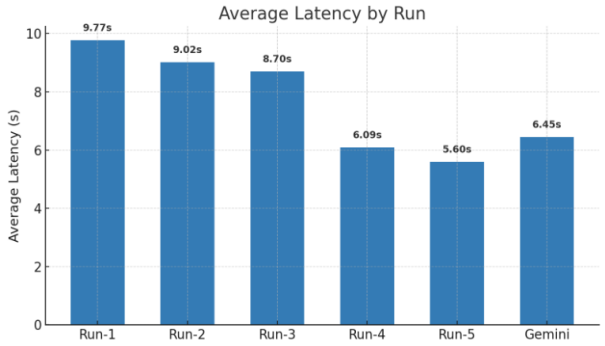

Fig. 8 – *Comparison of latency metrics across runs and models.*

The visualization confirms that response generation and thought processing consume the majority of processing time. API cost calculations based on OpenAI GPT-4o-mini pricing demonstrated a mean per-query cost of 6.6×10^{-5} USD, suggesting economically viable operation for institutional deployment scenarios. The cost structure analysis revealed that output token generation represents approximately 80% of total API costs, highlighting the importance of response

length optimization for cost management. Processing latency measurements indicated mean end-to-end response times of 7.6 seconds ($\sigma = 1.609$), encompassing retrieval operations, generation processing, and hallucination verification. The precise mean token counts, total consumption, and cost calculations for each of the five experimental runs are itemized in the computational performance section in **Table 3** and **Table 4**, respectively.

The cost analysis reveals strategic implications for model selection: "Flash" or "Mini" variants prove optimal for scenarios requiring low latency and brief responses, while GPT or Claude models demonstrate superior performance for applications demanding comprehensive response quality. Comparative analysis across different language model configurations demonstrated significant cost variations, with premium models exhibiting 5 to 15 times higher per-token costs while achieving only marginal improvements in faithfulness and relevance metrics. These findings suggest that model selection strategies should carefully balance cost considerations with performance requirements based on specific deployment contexts and quality thresholds. This cost-performance trade-off is further illuminated by an examination of current pricing strategies across the language model market. As illustrated in Figs. 9 – 12, significant price disparities exist among leading providers such as DeepSeek, OpenAI, Google, Anthropic, Mistral, and Cohere. For instance, premium models such as Anthropic's Claude 3 Opus feature output costs exceeding $70 per million tokens, while more economical "mini" or "flash" alternatives, such as the GPT-4o-mini used in this study, present a viable option. Likewise, the output cost for Google's Gemini 2.5 Pro is markedly higher than that of the "Flash" versions within the same model family. This landscape underscores the criticality of model selection for budget-constrained projects and highlights the tiered pricing structures offered even within a single provider's portfolio.

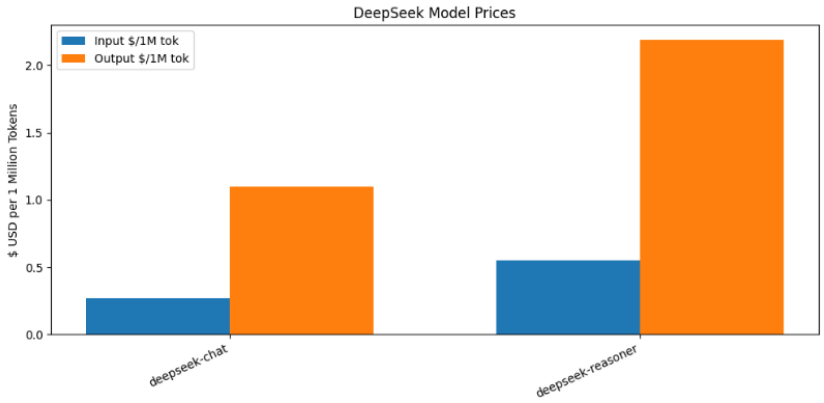

Fig. 9 – *Comparison of input/output token metrics for DeepSeek.*

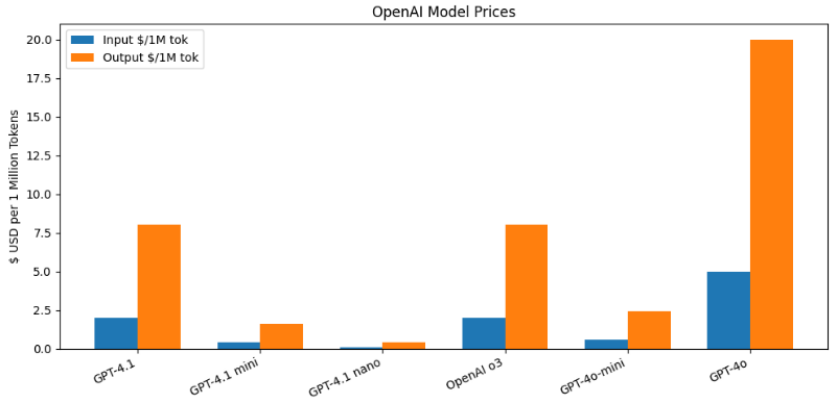

Fig. 10 – *Comparison of input/output token metrics for different OpenAI models.*

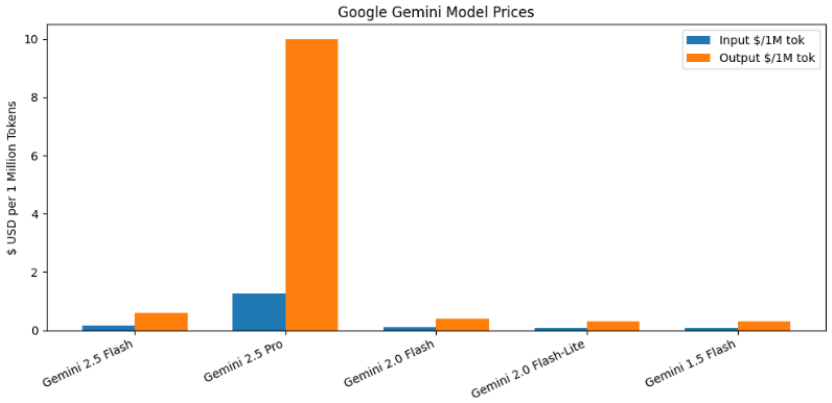

Fig. 11 – *Comparison of input/output token costs for different Google Gemini models.*

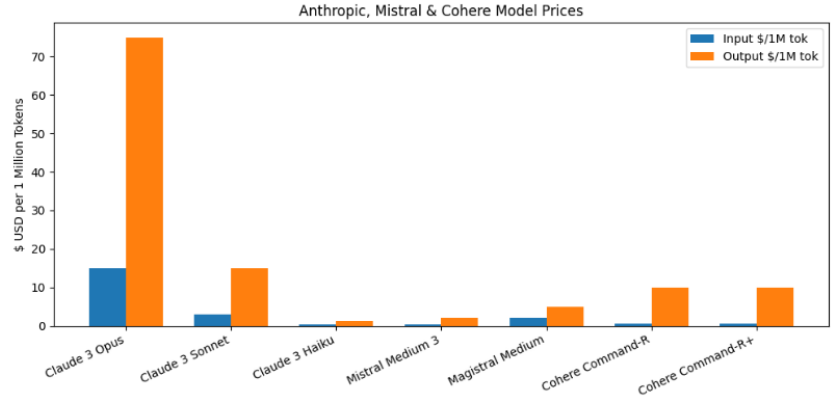

Fig. 12 – *Comparison of input/output token costs for different LLM models.*

Fig. 13 facilitates a more strategic comparison by consolidating models with costs under $10 per million tokens. This aggregated view confirms that models designated as “Flash” (Gemini), “Mini” (GPT), and “Haiku” (Claude) are optimal for scenarios where low latency and cost are primary considerations. A further consistent finding across all charts is that output tokens are almost universally more expensive than input tokens. This observation reinforces our earlier analysis that the length of the model-generated response is the primary driver of both cost and latency in the system architecture. Consequently, for institutional applications such as BTU-Chatbot, optimizing response length emerges as a critical strategy for ensuring both economic sustainability and a positive user experience.

Taken together, these experimental findings reveal a clear performance asymmetry between the retrieval and generation components of the system, the implications of which are examined in detail in the following section.

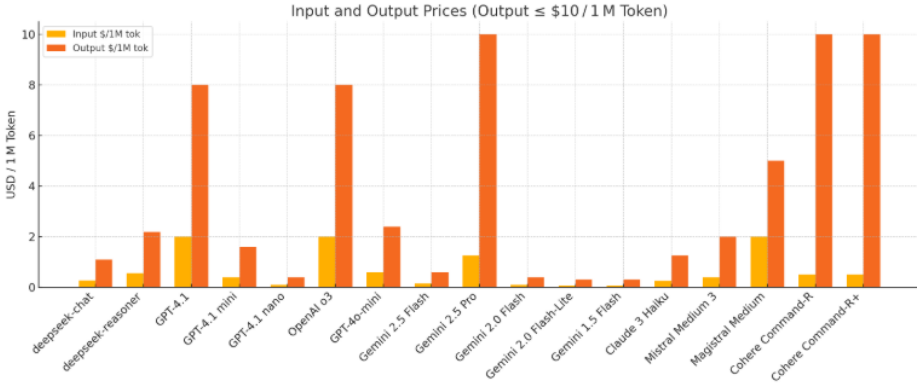

Fig. 13 – *Comparison of latency metrics across runs and models.*

6 Discussion

6.1 System performance and limitations

The experimental results demonstrate that BTU-Chatbot achieves strong performance in information retrieval accuracy while facing significant challenges in response generation reliability. The exceptional Context Recall and Context Precision values (0.97 and 0.99, respectively) confirm that the vector-based retrieval mechanism successfully identifies relevant institutional documents with high accuracy. However, the substantially lower Faithfulness scores (mean = 0.1922) reveal persistent challenges in maintaining factual grounding during response generation.

Furthermore, the analysis of generation quality reveals a critical limitation in relying solely on semantic similarity metrics for evaluation. Fig. 6 illustrates this paradox clearly: while non-hallucinated examples consistently score high in Answer Relevancy, a significant portion of hallucinated responses also cluster in the high-similarity range (0.85 and above). This indicates that an answer can be semantically and stylistically very close to the ground truth, perhaps differing only in a critical date, number, or specific term, and still be factually incorrect. For example, a response stating a deadline of 'May 25th' may be semantically very similar to the correct answer of 'May 15th,' yet its factual inaccuracy is what truly matters to the user. This overlap implies that metrics such as Answer Relevancy, while useful for gauging contextual appropriateness, are insufficient on their own for verifying the factual reliability of a system. It reinforces why metrics such as Faithfulness are essential and highlights the need for evaluation protocols that move beyond semantic similarity to perform more stringent, claim-by-claim verification against the source documents. This finding suggests that a truly reliable RAG system must not only be relevant but, more importantly, verifiably accurate, a challenge that remains central to future research.

The low faithfulness scores are therefore best understood not solely as evidence of hallucination, but as a consequence of the interaction between three factors: the strict claim-level scoring methodology of RAGAS, the cross-lingual nature of the evaluation pipeline, and the verbosity of GPT-4o-mini responses under the current prompt configuration. This interpretation is supported by the Answer Correctness scores in **Table 3**, which remain considerably higher than Faithfulness across all runs, indicating that the system frequently produces factually correct answers that nonetheless receive low faithfulness scores due to unsupported peripheral claims.

The observed performance pattern suggests a fundamental asymmetry between retrieval capabilities and generation quality that characterizes current RAG system implementations. While dense vector representations effectively capture semantic relationships within an institutional document corpus, the language model component struggles to maintain strict adherence to the retrieved context during response synthesis. The current hallucination rate and faithfulness metrics therefore remain insufficient to ensure full factual reliability.

On the other hand, the cost and efficiency analysis reveals encouraging results for institutional deployment. With a per-query cost of 6.6×10^{-5} USD and a 7.6-second average response latency, the system demonstrates economic viability for educational institutions with limited technology budgets. While the latency is acceptable for many use cases, it may require further optimization for scenarios demanding immediate information access.

A limitation of the current evaluation design is the absence of a stratified analysis by question type. The test sets across runs contain varying numbers of

questions (20 to 73) drawn from different regulatory categories, but the evaluation framework does not systematically disaggregate performance by question complexity or document type. Future work should construct a balanced evaluation set with explicit question-type stratification, distinguishing, for example, between factual lookup queries, multi-article synthesis queries, and procedural questions, to enable a more granular understanding of where retrieval and generation failures concentrate.

Given these limitations in reliability, future work should prioritize more robust mitigation strategies. These strategies could include pre-validating generated claims against a structured knowledge graph extracted from institutional documents, or incorporating a human-in-the-loop review process for sensitive queries to guarantee factual accuracy before delivering answers to the end user.

6.2 Architectural insights and design implications

The multi-agent ReAct architecture demonstrated effective tool coordination and reasoning capabilities, successfully integrating vector database retrieval with domain-restricted web search to provide comprehensive information coverage. The tool selection mechanisms functioned as designed, appropriately choosing retrieval sources based on query characteristics and information availability. The domain restriction strategy for web search (limiting to "*.btu.edu.tr") proved effective in maintaining information authority while enabling access to current institutional announcements and updates not yet incorporated into the static document collection. This hybrid approach addresses temporal limitations inherent in static knowledge bases while preserving institutional credibility and relevance. The prompt engineering strategies employed in this study successfully established behavioral constraints that guided model responses toward desired output formats and citation requirements. However, the persistence of hallucination issues suggests that prompt-based constraints alone are insufficient for ensuring complete factual reliability in specialized domains. The modular system architecture enabled effective separation of concerns and facilitated independent optimization of retrieval and generation components. This design approach proved valuable during experimental iterations and suggests that similar architectural patterns would benefit related research and development efforts.

6.3 Implications for educational technology

The demonstrated capabilities of BTU-Chatbot suggest significant potential for improving information accessibility in educational institutions facing similar document management and information retrieval challenges. The system's ability to provide source-attributed responses addresses key requirements for institutional accountability and verification needs. The cost-effectiveness analysis indicates that RAG-based conversational systems can provide valuable

services within typical educational technology budgets, potentially democratizing access to advanced AI capabilities for institutions with limited resources. The observed per-query costs suggest that such systems could support thousands of daily interactions while maintaining reasonable operational expenses. The user experience improvements demonstrated through the multi-session interface and citation tracking capabilities represent meaningful advances in educational technology interfaces. These features address common user workflows in institutional settings and suggest design patterns applicable to broader educational software development efforts. The evaluation methodology developed for this study provides a replicable framework for assessing similar systems, contributing to standardization efforts in educational AI system evaluation. The RAGAS-based assessment approach offers objective metrics that enable comparison across different implementations and architectural approaches.

Although BTU-Chatbot was developed and evaluated within the specific institutional context of Bursa Technical University, the underlying architecture is designed with adaptability in mind. The document ingestion pipeline accepts standard PDF input, the vector database is institution-agnostic, and the domain-restricted web search tool can be reconfigured for any institutional domain. Scaling the system to a larger university with a broader document corpus would primarily require additional embedding storage and may benefit from hierarchical retrieval strategies to manage increased index size. Adapting the system to a different institution involves three main steps: (1) replacing the document corpus with the target institution's regulatory documents, (2) updating the domain restriction parameter in the web search tool to match the target domain, and (3) rerunning the embedding pipeline. The modular architecture therefore positions BTU-Chatbot as a transferable template for institutional RAG deployment rather than merely a single-institution solution. Future work will empirically validate this claim by testing the architecture against corpora from institutions with different languages, document structures, and regulatory frameworks.

6.4 Future research directions

Several promising research directions emerge from the limitations and opportunities identified in this study. The faithfulness challenges suggest investigating alternative generation approaches that enforce stricter adherence to retrieved context, potentially through constrained generation techniques or retrieval-augmented fine-tuning approaches. A critical priority is the development of more advanced hallucination detection mechanisms tailored for regulatory text. This could involve machine learning models trained on institutional fact-checking patterns or the creation of a knowledge graph to validate relationships between entities and procedures mentioned in responses. Additionally, addressing the three hallucination patterns identified in this study,

namely context boundary violations, temporal extrapolation, and cross-document conflation, calls for targeted mitigation strategies. Context boundary violations could be reduced by implementing output length constraints tied to the retrieved passage length. Temporal extrapolation errors could be mitigated by enriching document metadata during ingestion to flag date-sensitive articles for stricter grounding requirements. Cross-document conflation could be addressed through a post-generation verification step that checks each synthesized claim against its attributed source passage individually before the final response is delivered to the user. Integrating these techniques with constrained generation that strictly limits the model's vocabulary to the retrieved context could significantly improve faithfulness.

7 Conclusion

This research presented BTU-Chatbot, an Agentic RAG system designed to consolidate and streamline access to Bursa Technical University's institutional information. The study confirms that such an architecture can effectively tackle information fragmentation in a university setting. Our evaluation demonstrates a key performance dichotomy: the system achieves near-perfect retrieval accuracy, with Context Recall and Precision scores of 0.97 and 0.99, respectively, ensuring that relevant documents are almost always found. However, this strength is contrasted by a significant weakness in generation reliability. The multi-agent ReAct framework proved successful for complex query processing, and the system is economically viable for institutional budgets. The primary contribution of this work is the detailed demonstration of both the promise and the current pitfalls of deploying Agentic RAG in a specialized, high-stakes domain. While BTU-Chatbot is a valuable step forward, achieving the level of factual faithfulness required for full deployment remains a critical challenge for future research.

8 Acknowledgments

The authors wish to express their gratitude to Bursa Technical University for providing access to the comprehensive corpus of institutional documents, including all laws, directives, and regulations governing student affairs and processes.

9 References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin: Attention is All You Need, Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS), Long Beach, USA, December 2017, pp. 6000 – 6010.

[2] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, J. Gao: Large Language Models: A Survey, arXiv:2402.06196v3 [cs.CL], March 2025, pp. 1 – 44.

[3] S. Gilbert, J. N. Kather, A. Hogan: Augmented Non-Hallucinating Large Language Models as Medical Information Curators, npj Digital Medicine, Vol. 7, April 2024, p. 100.

[4] F. Ye, S. Li, Y. Zhang, L. Chen: R²AG: Incorporating Retrieval Information into Retrieval-Augmented Generation, Proceedings of the Conference on Empirical Methods in Natural Language Processing - Findings of the Association for Computational Linguistics (EMNLP), Miami, USA, November 2024, pp. 11584 – 11596.

[5] Y. Chen, D. Röder, J.- J. Erker, L. Hennig, P. Thomas, S. Möller, R. Roller: Retrieval-Augmented Knowledge Integration into Language Models: A Survey, Proceedings of the 1st Workshop on Towards Knowledgeable Language Models (KnowLLM), Bangkok, Thailand, August 2024, pp. 45 – 63.

[6] H. Song, J. Jiang, W. Tian, Z. Chen, Y. Wu, J. Zhao, Y. Min, W. X. Zhao, L. Fang, J.- R. Wen: R1-Searcher++: Incentivizing the Dynamic Knowledge Acquisition of LLMs via Reinforcement Learning, arXiv:2505.17005v1 [cs.CL], May 2025, pp. 1 – 14.

[7] Z. Shi, L. Yan, W. Sun, Y. Feng, P. Ren, X. Ma, S. Wang, D. Yin, M. de Rijke, Z. Ren: Direct Retrieval-Augmented Optimization: Synergizing Knowledge Selection and Language Models, arXiv:2505.03075v1 [cs.IR], May 2025, pp. 1 – 26.

[8] L. Mombaerts, T. Ding, A. Banerjee, F. Felice, J. Taws, T. Borogovac: Meta Knowledge for Retrieval-Augmented Large Language Models, Proceedings of the Workshop on Generative AI for Recommender Systems and Personalization (KDD), Barcelona, Spain, August 2024, pp. 1 – 8.

[9] C. Sharma: Retrieval-Augmented Generation: A Comprehensive Survey of Architectures, Enhancements, and Robustness Frontiers, arXiv:2506.00054v1 [cs.IR], May 2025, pp. 1 – 32.

[10] X. Zhang, Y.- Z. Song, Y. Wang, S. Tang, X. Li, Z. Zeng, Z. Wu, W. Ye, W. Xu, Y. Zhang, X. Dai, S. Zhang, Q. Wen: RAGLAB: A Modular and Research-Oriented Unified Framework for Retrieval-Augmented Generation, Proceedings of the Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP), Miami, USA, November 2024, pp. 408 – 418.

[11] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, Y. Cao: ReAct: Synergizing Reasoning and Acting in Language Models, Proceedings of the 11th International Conference on Learning Representations (ICLR), Kigali, Rwanda, May 2023, pp. 1 – 33.

[12] Z. Liu, H. Zuo, Y. Lu: The Impact of ChatGPT on Students' Academic Achievement: A Meta-Analysis, Journal of Computer Assisted Learning, Vol. 41, No. 4, August 2025, p. e70096.

[13] J. Yin, H. Xu, Y. Pan, Y. Hu: Effects of Different AI-Driven Chatbot Feedback on Learning Outcomes and Brain Activity, npj Science of Learning, Vol. 10, April 2025, p. 17.

[14] N. D. Septiyanti, M. I. Luthfi, D. Darmawansah: Effect of Chatbot-Assisted Learning on Students' Learning Motivation and Its Pedagogical Approaches, Khazanah Informatika: Jurnal Ilmu Komputer dan Informatika, Vol. 10, No. 1, April 2024, pp. 69 – 77.

[15] M. G. Forero, H. J. Herrera-Suárez: ChatGPT in the Classroom: Boon or Bane for Physics Students' Academic Performance?, arXiv:2312.02422v1 [physics.ed-ph], December 2023, pp. 1 – 9.

[16] C.- C. Liu, W.- J. Chen, F.- Y. Lo, C.- H. Chang, H.- M. Lin: Teachable Q&A Agent: The Effect of Chatbot Training by Students on Reading Interest and Engagement, Journal of Educational Computing Research, Vol. 62, No. 4, July 2024, pp. 902 – 934.

[17] B. A. Dünya, H. Y. Durak: Hi! Tell Me How to Do It: Examination of Undergraduate Students' Chatbot-Integrated Course Experiences, Quality & Quantity: International Journal of Methodology, Vol. 58, No. 4, August 2024, pp. 3155 – 3170.

[18] R. Kandakatla, E. J. Berger, J. F. Rhoads, J. DeBoer: Student Perspectives on the Learning Resources in an Active, Blended, and Collaborative (ABC) Pedagogical Environment, International Journal of Engineering Pedagogy, Vol. 10, No. 2, March 2020, pp. 7 – 31.