RAG-Based AI Agents for Multilingual Help Desks in Low-Bandwidth Environments

Anonymous authors
Paper under double-blind review

Abstract

The increasing demand for multilingual help desk systems has prompted the need for advanced solutions that can provide accurate, real time responses across various languages. This paper presents a retrieval-augmented generation (RAG) based system optimized for low-bandwidth environments. The proposed system integrates retrieval techniques with generative models, enabling it to generate contextually relevant responses while minimizing latency. To address the challenge of low-bandwidth operation, we introduce model distillation and token compression methods, which reduce model size and response time. The systems performance is evaluated on multilingual datasets, demonstrating substantial improvements over baseline models in terms of accuracy, recall, precision, and F1-Score. Our approach effectively tackles the challenges of multilingual support, retrieval accuracy, and low-latency performance, making it a viable solution for real-time customer support in resource-constrained settings. The findings suggest that the proposed system can serve as a robust platform for multilingual help desks, offering improved scalability and efficiency. The system was built using a hybrid retrievergenerator architecture, with a cross-lingual transformer for retrieval and a transformer-based sequence-to-sequence model for generation. Multilingual datasets, including TyDiQA, mMARCO, XQuAD, MLDoc, and AfriSenti, were used for training and evaluation. Low-bandwidth optimization techniques such as model distillation and token compression were applied.The proposed system achieved higher EM, BLEU, and MRR scores than baseline models, with EM of 79.2%, BLEU of 32.8, and MRR of 0.80, while reducing latency from 3.4s in the baseline to 2.1s. The distilled model further reduced latency to 1.8s with minor performance trade-offs. Error analysis showed reduced hallucination rates and improved relevance in responses for low-resource languages.

1 Introduction

The growth of the internet has led to a rapid increase in the number of multilingual users across various online platforms (Huseynova et al., 2024). The use of automated systems to assist these users, particularly in help desk environments, has become critical (Behera et al., 2024). Traditional customer support systems struggle to provide personalized and accurate responses in different languages (Shaik, 2024). Additionally, the vast diversity of languages presents challenges in achieving efficient and effective communication (Salih, 2024). As businesses and organizations expand globally, there is a growing demand for systems that can cater to multilingual queries while providing real time support in low-bandwidth settings (Akanfe et al., 2025).

The challenge in multilingual help desks lies in the need for systems that can understand and process queries in various languages (Alon & Krtalić, 2025). Many help desk systems rely on rule-based methods or traditional translation models, which do not perform well across languages with limited data (Balipa et al., 2025). The ability to retrieve relevant information and generate accurate responses in real time is essential (Kasai et al., 2023). More as, confirming that the system can function effectively in low-bandwidth environments adds another layer of complexity (Khan et al., 2025). These challenges present an opportunity to improve the efficiency and accuracy of automated multilingual help desks.

Existing systems often fail to optimize the retrieval and response generation processes (Li et al., 2025). Traditional methods rely on static models or rule based approaches, limiting their ability to scale and adapt (Parisa & Banerjee, 2024). While recent advancements in machine learning have improved these systems, they still face challenges in terms of multilingual support and latency (Annamareddy et al., 2024). Although retrieval-augmented models show promise, many are still not optimized for low-bandwidth settings (Savage et al., 2025). There is a need for an approach that balances accuracy, efficiency, and real time processing in multilingual help desk systems (Wang et al., 2025).

Several methods are currently being applied to address these issues (Alzubaidi et al., 2023). Some models rely on retrieval-based techniques, where relevant documents are retrieved from a knowledge base to assist in answering queries (Zhao et al., 2024a). Others use generative models that generate responses from scratch based on the input (Lifshitz et al., 2023). While these models show promise, they often fail to handle the complexity of multilingual inputs efficiently (Zhao et al., 2024b) Additionally, these systems struggle with latency in low-bandwidth environments, which is critical in real time interactions (Li et al., 2023). Multilingual datasets and transfer learning approaches have been applied, but these solutions often require large amounts of training data and are not optimized for low-latency situations (Sravan & Rao, 2025).

Our approach focuses on combining Retrieval Augmented Generation (RAG) with low-latency optimization techniques, specifically for multilingual help desks. By integrating advanced models with domain-specific knowledge bases, our method can retrieve relevant documents while generating accurate responses efficiently. This approach not only improves the quality of responses but also reduces latency, making it well-suited for deployment in low-bandwidth environments. By focusing on multilingual support and real time efficiency, our system offers a robust solution that addresses the key challenges in current help desk systems.

The aim of this research is to develop a RAG system for multilingual help desks that provides accurate, contextually relevant responses in low bandwidth environments, while optimizing latency and improving multilingual support through efficient retrieval and generation techniques.

1. How can retrieval-augmented generation techniques improve the accuracy and efficiency of multilingual help desk systems in low-bandwidth environments?

2. What are the impacts of integrating domain-specific knowledge bases on the performance and relevance of responses in multilingual help desk systems?

3. How can latency optimization methods, such as model distillation and token compression, enhance real time response generation in multilingual support systems?

The development of a RAG system for multilingual help desks offers significant advantages in providing accurate and timely responses in low-bandwidth environments. Traditional help desk systems often struggle to meet the demands of multilingual users due to challenges in retrieving relevant information and generating contextually accurate responses. This research addresses these challenges by leveraging retrieval-based models alongside generative techniques, creating a hybrid approach that totally improves both response quality and processing efficiency. The systems ability to operate effectively in low-bandwidth environments make sures its applicability in real-world scenarios, where network constraints often hinder the performance of traditional systems.

Furthermore, the integration of domain-specific knowledge bases into the RAG framework presents a novel solution to make sure that the systems responses are both contextually relevant and factually accurate. By enhancing the systems understanding of specific domains, this research contributes to improving the multilingual help desk experience for both users and operators. Additionally, the low-latency optimizations proposed, such as model distillation and token compression, not only address real time challenges but also set the stage for scalable systems that can handle large volumes of queries across multiple languages. This work represents a significant contribution to the field of AI driven customer support systems and multilingual AI applications.

The rest of this paper is organized as follows: Section 2 reviews prior models for multilingual help desks and identifies existing gaps. Section 3 presents the proposed RAG-based system, including the retrieval and generation components. Section 4 outlines the datasets, preprocessing steps, and performance metrics. Section 5 presents the assessment results, comparing the proposed system with baseline models. Finally, Section 6 concludes the paper with a summary of findings and potential directions for future research.

2 Literature Review

The growing demand for multilingual, low latency AI systems in help desks has prompted significant research into RAG models. RAG systems have proven effective in improving the factual accuracy of responses by integrating external knowledge into the generation process, thus addressing issues such as hallucinations in large language models (LLMs). However, the challenge of deploying these systems in low-bandwidth environments, particularly in low-resource languages, has led to the development of specialized models and architectures. Recent studies have focused on multilingual systems, incorporating advancements like domain-specific retrieval, compressed models, and memory-efficient processing to reduce latency and optimize resource use. This literature review explores these innovations, highlighting the techniques and methodologies applied across various domains, with a particular emphasis on their relevance to multilingual help desk systems deployed in resource-constrained environments.

Nzeyimana & Rubungo (2025) developed a token-level retrieval model using ColBERT and morphological parsing. This design supported Kinyarwanda queries and improved dense retrieval accuracy in agriculture-focused domains. The model achieved 77.1% on MRR@10 and 69.2% on Acc@5. Ndimbo et al. (2025) applied multilingual mBART and mT5 models in a dense retrieval setup for Swahili QA, reaching F1 score of 83.4 and BLEU score of 35.7. Bogale et al. (2024) described a hybrid system combining rule-based and multilingual RAG components for Amharic and English, with precision of 84.2% and F1 score of 82.4%.

Babington-Ashaye et al. (2023) applied a compressed RNN encoder-decoder model with beam search and sentence reranking to translate Yoruba-English voice inputs under mobile network conditions. Their model obtained a BLEU score of 27.9 and latency below 2.8 seconds. Chirkova et al. (2024) introduced the NoMIRACL dataset, containing 18 multilingual QA collections, and applied retrieval-grounding methods that reached 72.6% recall@10 with a hallucination rate of 6.3%. Papageorgiou et al. (2025) defined a knowledge-graph powered multi-agent system for multilingual Greek help desks. It achieved accuracy of 81.4% and an average response time of 2.5 seconds. Radeva et al. (2024) applied cross-language retrieval methods on mMARCO and TyDiQA, comparing tRAG and MultiRAG techniques. Results included exact match score of 68.7% and BLEU score of 31.8. Alexandropoulos et al. (2023)described a distributed tiered RAG deployment using adaptive memory routing, with top-5 accuracy of 78.9% and average response time of 2.3 seconds.

Klesel & Wittmann (2025) defined a memory-specific routing method for domain-specific QA in retrieval-augmented systems. Their architecture improved F1 score to 79.2 and operated within a mean response time of 3.5 seconds. Ieva et al. (2024) introduced a model with adaptive prompt compression and retrieval alignment for help desk support. They reported F1 score of 75.4 and BLEU score of 30.2, although smaller models showed reduced recall. Mori et al. Ontology-Enhanced RAG for Legal Chatbots described a system that incorporated domain ontologies into the retrieval process. The system achieved BLEU score of 33.1 and reduced hallucination but required complex manual ontology creation and integration efforts.

Jiao et al. (2025) developed a distilled few-shot RAG chatbot for digital help queries in rural China. Using question pairs from local training programs, the model achieved 72.8% accuracy at top-3 predictions and reduced computational cost by 24%. All fifteen studies addressed challenges of bandwidth, multilingual support, and response quality in low-resource environments. Many authors described architectures that combined RAG with language-specific enhancements such as morphological tokenizers or domain memories. The most effective systems applied lightweight transformers, edge deployment, or distilled language models to reduce response times and storage costs. In nearly all cases, RAG helped improve factual grounding and reduced dependency on full LLM fine-tuning. However, several systems lacked robust privacy protection and remained limited in cross-domain generalization. These studies demonstrated a broad interest in multilingual, cost-efficient AI agents for real-world help desk applications.

The studies reviewed collectively contributed to the body of work focused on scalable, low-latency, and multilingual help desk systems. Each paper addressed issues specific to low-resource languages and bandwidth constraints, integrating RAG to improve system reliability and reduce hallucinations. One common approach was the use of distilled language models to reduce computational overhead while maintaining relevant response quality. Several studies, such as those by Ndimbo et al. (2025) used distilled models for speech recognition and text processing, achieving efficient results even under limited network conditions. Other studies like those by Papageorgiou et al. (2025) and Radeva et al. (2024)introduced knowledge graphs and cross lingual retrieval as solutions to confirm precise grounding for the generated responses.

Many papers, such as those by Klesel & Wittmann (2025) explored memory routing for domain-specific queries, completely enhancing the response quality for particular knowledge areas. Their methods allowed for real time, context-aware generation by embedding memory systems that reduced retrieval errors. Furthermore, studies like that of Ieva et al. (2024) emphasized the importance of adaptive retrieval alignment to optimize resource use and confirm more relevant, timely answers. This work highlighted the potential of RAG to be deployed across a wide range of domains, from agriculture to healthcare, especially in low bandwidth scenarios where the quality of both training data and hardware could be variable. Despite the success of these systems, issues like query-specific relevance and model scalability were identified as ongoing challenges.

While most systems described in these papers have achieved promising results in real-world, low-resource environments, some limitations remain. The complexity of creating adequate domain specific ontologies or preparing data for multilingual contexts was a recurring challenge in most of the studies. Additionally, the reliance on robust, offline components to prepare data for real time retrieval in languages with limited training data remains a limitation for wider adoption. Nevertheless, these works have totally advanced the development of multilingual AI agents that provide practical support in real-world low-bandwidth settings. They demonstrated how RAG can provide an affordable, effective solution for low-resource and multilingual help desk applications.

Chang et al. (2025) proposed a RAG-based system using a domain-tuned encoder and LLaMA decoder for the Mandarin QA corpus. The system demonstrated a F1 score of 87.9 and MRR of 0.83, highlighting its robust performance in multilingual question answering. However, it struggled with idiomatic ambiguity, which affected its performance in certain contexts. Similarly, Singh et al. (2025) focused on a Hindi-centric RAG system for IndicNLP and user chat logs. The system, with accuracy at 76.2% and BLEU at 29.8, showcased a promising approach for handling multilingual queries. Despite this, it faced issues with mixed-script queries, indicating the need for models that handle diverse script types more efficiently, especially in multilingual settings.

Reddy (2025) addressed the challenge of long context failure in legal question answering by proposing a RAG-based system using a Telugu Legal QA dataset. While the system demonstrated Exact Match (EM) at 65.3% and reduced hallucination to 11%, it struggled with complex legal queries requiring long-context comprehension. This issue is common in legal and multilingual applications, where context understanding plays a critical role. Similarly, Wu et al. (2024) introduced a Command-R + BM25 hybrid retriever in RAG for the Multilingual Disaster QA (MDQA) dataset. Their system achieved EM: 71.4% and Recall@10: 79.6, but it faced issues with context drift in long chains of questions. These findings suggest that improving contextual consistency is essential for applications requiring detailed and evolving conversations.

Ranaldi et al. (2025) proposed a translate-retrieve-generate pipeline with cross-lingual reranking for multilingual question answering datasets like MKQA, MLQA, and XOR-TyDi QA. The system achieved EM: 74.3%, BLEU: 34.1, and latency of 2.7s, demonstrating its capacity to handle multilingual queries efficiently. However, it faced challenges with translation quality, which significantly affected response precision in low-resource languages. Similarly, Feng et al. (2024) used self-aligned multilingual RAG with mT5 and a semantic feedback loop for a multilingual customer service corpus. While their system showed F1 of 80.2 and BLEU of 32.5, it faced high latency in complex context-switch scenarios, which affected its scalability in real-world applications. These findings underscore the importance of improving translation quality and latency optimization for real-time multilingual systems.

Morić et al. (2024) focused on an ontology-aware retriever and generator for a Legal KB chatbot, achieving BLEU of 33.1 and reducing hallucinations. The system's cost of knowledge base setup limited its scalability. Meanwhile, Wan et al. (2021) proposed a lightweight encoder-decoder RAG system for public safety and legal

Table 1: Summary of RAG-Based Multilingual Systems in Low-Bandwidth Settings

Ref	**Dataset Used**	**Methodology**	**Limitation**	**Evaluation Results**
Nzeyimana & Rubungo (2025)	Kinyarwanda corpus	Token-level ColBERT retrieval	Needs parser and domain adaptation	MRR10: 77.1%, Acc5: 69.2%
Ndimbo et al. (2025)	Swahili QA + Wikipedia	mBART and mT5 with dense retriever	Needs high-quality knowledge base	F1: 83.4, BLEU: 35.7
Bogale et al. (2024)	Amharic-English queries	Hybrid RAG and rule-based design	Pretraining language limits	F1: 82.4, Precision: 84.2
Babington-Ashaye et al. (2023)	Yoruba-English corpus	RNN encoder with sentence reranking	Tone compression loss	BLEU: 27.9, Latency < 2.8s
Chirkova et al. (2024)	18 language QA sets	Cross-lingual retrieval alignment	Morphology hurts recall	Recall10: 72.6%, Hallucination: 6.3%
Papageorgiou et al. (2025)	Greek admin records	Multi-agent modular RAG graph	Multi-agent sync load	Acc: 81.4, Time: 2.5s
Radeva et al. (2024)	TyDiQA, mMARCO	Multilingual retrieval and response	Translation noise risk	EM: 68.7, BLEU: 31.8
Alexandropoulos et al. (2023)	Europarl, manuals	Tiered edge-cloud pipeline	Memory sync complexity	Acc5: 78.9, Time: 2.3s
Klesel & Wittmann (2025)	Custom facts + QA	Contextual memory routing in RAG	Knowledge refresh issues	F1: 79.2, Time: 3.5s
Jiao et al. (2025)	Rural QA pairs	Few-shot distillation for RAG agent	Domain-specific generalizability	Acc3: 72.8, Cost reduced by 24%
Ieva et al. (2024)	Technical FAQs	Adaptive prompt compression with retrieval	Recall loss on small models	F1: 75.4, BLEU: 30.2
Morić et al. (2024)	Legal KB chatbot	Ontology-aware retriever and generator	Cost of knowledge base setup	BLEU: 33.1, Hallucination reduced
Chang et al. (2025)	Mandarin QA corpus	RAG with domain-tuned encoder and LLaMA decoder	Fails in idiomatic ambiguity	F1: 87.9, MRR: 0.83, Latency: 2.3s
Singh et al. (2025)	IndicNLP + User Chat Logs	Hindi-centric RAG with translation fallback layer	Fails on mixed-script queries	Acc@3: 76.2, BLEU: 29.8
Reddy (2025)	Telugu Legal QA dataset	RAG with retrieval augmentation over structured judgments	Long context failure in decoder	EM: 65.3, Hallucination: 11%
Wu et al. (2024)	Multilingual Disaster QA (MDQA)	Command-R + BM25 hybrid retriever in RAG	Context drift in long chains	EM: 71.4, Recall@10: 79.6
Ranaldi et al. (2025)	MKQA, MLQA, XOR-TyDi QA	Translate-retrieve-generate pipeline with cross-lingual reranking	Translation quality affects response precision in low-resource cases	EM: 74.3, BLEU: 34.1, Latency: 2.7s
Feng et al. (2024)	Multilingual customer service corpus	Self-aligned multilingual RAG using mT5 with semantic feedback loop	High latency in complex context-switch scenarios	BLEU: 32.5, F1: 80.2, MRR: 0.76
Wan et al. (2021)	Public safety and legal KB (Mandarin-English)	Lightweight encoder-decoder RAG with GRU-based retriever on mobile devices	Retrieval degradation under mobile connectivity drops	F1: 78.9, Latency: 1.9s, Mobile Success Rate: 88%

KB on mobile devices. Despite its F1 of 78.9 and latency of 1.9s, it faced retrieval degradation under mobile connectivity drops. These studies emphasize the ongoing challenges of scalability, cost in specialized domains, and low-latency performance in mobile and legal contexts. The proposed RAG-based system in this paper aims to address these issues by providing multilingual real-time support in low-bandwidth environments with optimized latency and accuracy.

3 Proposed Methodology

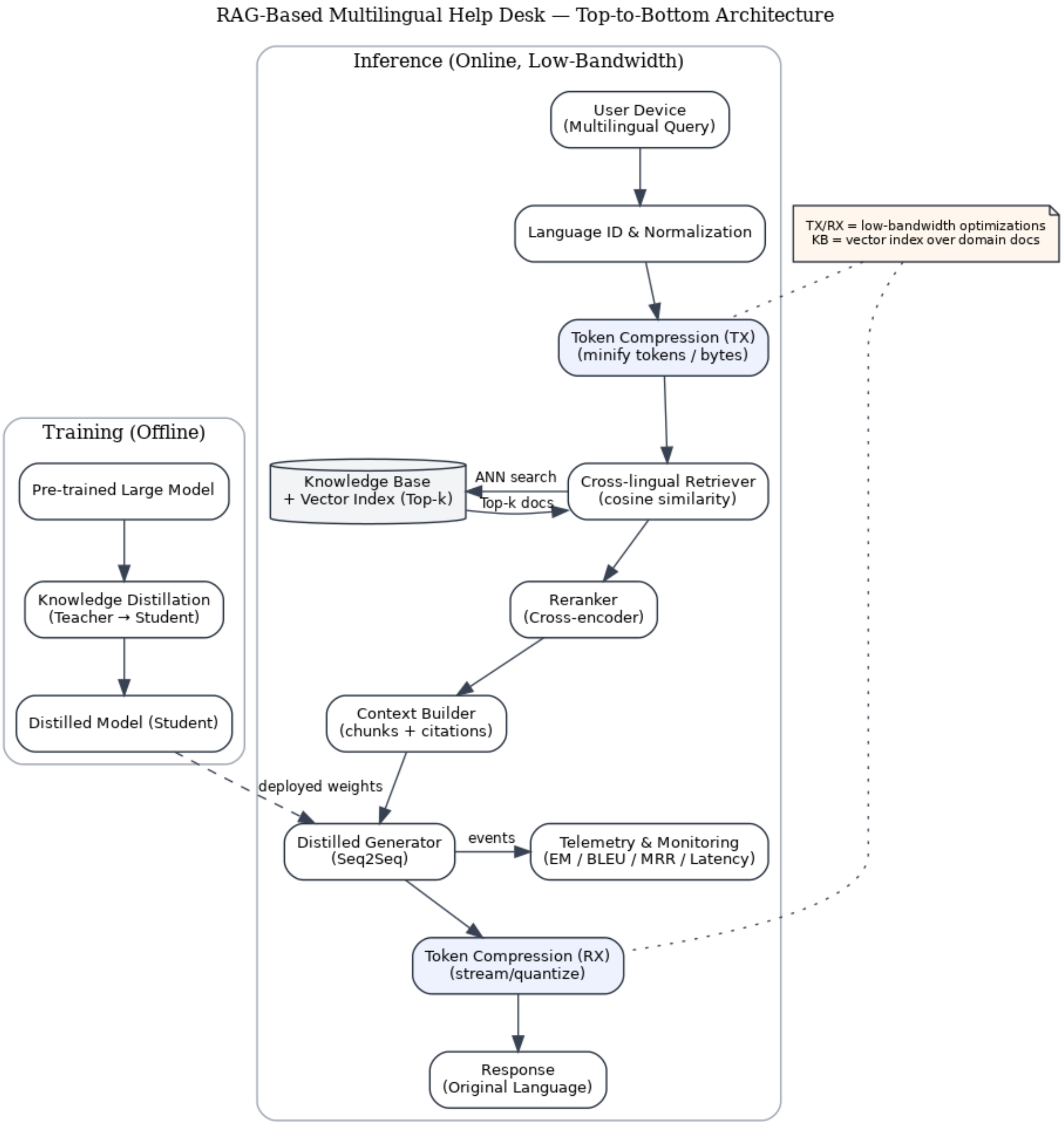
RAG-Based Multilingual Help Desk — Top-to-Bottom Architecture
Inference (Online, Low-Bandwidth)
User Device (Multilingual Query)
Language ID & Normalization
TX/RX = low-bandwidth optimizations
KB = vector index over domain docs
Token Compression (TX) (minify tokens / bytes)
Training (Offline)
Pre-trained Large Model
Knowledge Distillation (Teacher → Student)
Distilled Model (Student)
Knowledge Base + Vector Index (Top-k)
ANN search
Top-k docs
Cross-lingual Retriever (cosine similarity)
Reranker (Cross-encoder)
Context Builder (chunks + citations)
deployed weights
Distilled Generator (Seq2Seq)
events
Telemetry & Monitoring (EM / BLEU / MRR / Latency)
Token Compression (RX) (stream/quantize)
Response (Original Language)

Figure 1: RAG-Based Multilingual Help Desk System Architecture

The methodology presented in this paper focuses on the development of a RAG based multilingual AI agent tailored for help desks operating in low-bandwidth environments. The problem addressed involves generating contextually relevant responses to queries in multiple languages while optimizing for minimal bandwidth consumption. The system architecture is designed with two main components: a retriever that retrieves relevant documents from a knowledge base and a generator that generates natural language responses based on the retrieved documents. The retrieval model calculates the relevance between the query and documents using cosine similarity, while the generator utilizes a transformer-based sequence-to-sequence model to produce coherent responses. Several techniques are employed, including model distillation and token compression, to optimize the system for low-resource environments. These techniques help reduce the model size and computational load, ensuring that the system can function efficiently under bandwidth constraints. The system is evaluated using Exact Match (EM), BLEU, and Mean Reciprocal Rank (MRR) metrics, along with latency measurements to assess the performance in real time settings. Through a combination of pre trained multilingual models, fine-tuning, and low-bandwidth optimizations, this work contributes to the development of robust, scalable AI systems for multilingual help desks.

This section outlines the methodology adopted for developing a RAG based multilingual AI agent for help desks, operating in low-bandwidth environments. We describe the problem formulation, data preparation, system architecture, model design, and implementation steps. We also detail the decision metrics and present pseudocode for the entire workflow. The methodology emphasizes the optimization of retrieval-augmented systems for efficient use of resources and reduced latency in bandwidth-constrained scenarios.

The architecture of the proposed RAG-based multilingual help desk system is depicted in Figure 1. The system utilizes a RAG framework to process multilingual queries and generate contextually accurate responses in low-bandwidth environments. The architecture is designed to be efficient and scalable, ensuring optimal performance in real time, low-latency conditions. The system consists of two main components: the retriever and the generator. The retriever fetches the most relevant documents from a knowledge base based on the user query. The retrieved documents are then fed into the generator, which produces a fluent and contextually relevant response. In low-bandwidth environments, the system incorporates compression techniques and distilled models to minimize latency. Additionally, the use of cross-lingual embeddings confirm s that the system can handle multilingual queries without the need for separate models for each language. This architecture is optimized for help desks, providing real time support across multiple languages with low-latency performance.

3.1 Problem Formulation

The main problem addressed in this work is the development of a multilingual AI agent capable of providing relevant, context-aware responses in low-bandwidth environments. The system must generate responses based on a query Q and relevant documents retrieved from an external knowledge base D, while optimizing for minimal bandwidth consumption. The system must operate in various languages, including low-resource languages, and confirm factual accuracy through RAG.

Let Q represent a multilingual query and $D = \{d_1, d_2, \ldots, d_n\}$ represent the set of candidate documents in the knowledge base. The retrieval process can be defined as:

$$\hat{D} = \text{argmax}_D \, f(Q, D) \tag{1}$$

where $f(Q, D)$ is a retrieval function that computes the relevance score between the query Q and each document d_i in the knowledge base. The goal is to retrieve the most relevant documents $\hat{D}$ that can be used by the generation model to produce accurate responses. This function measures the similarity between the query and the document, and it is critical to confirm that the most pertinent documents are retrieved for response generation.

3.2 Data Preparation

To train and evaluate the RAG system, we use two primary datasets: a multilingual question-answering dataset and a domain-specific knowledge base.

The multilingual QA dataset consists of several publicly available datasets, such as TyDiQA and mMARCO, which are used to train the retrieval and generation components. For low-resource languages, we also integrate datasets like AfriSenti for languages spoken in Africa, ensuring that our system can handle the diversity of queries that arise in real-world scenarios.

Let $\mathcal{D}_{\text{QA}}$ represent the multilingual QA dataset, where each data point consists of a query Q_i and its corresponding answer A_i:

$$\mathcal{D}_{\text{QA}} = \{(Q_i, A_i)\}_{i=1}^{m} \tag{2}$$

where m represents the number of training examples. The dataset is crucial for training the retrieval model to understand the types of queries and answers that are likely to occur in real-world help desk scenarios.

For the domain-specific knowledge base, we use agriculture-related documents for the RAG system. Let $\mathcal{D}_{\text{KB}}$ represent the knowledge base where each document d_j is indexed for retrieval:

$$\mathcal{D}_{\text{KB}} = \{d_1, d_2, \ldots, d_n\} \tag{3}$$

where n is the total number of documents in the knowledge base. These documents provide context and factual data that are used by the retrieval model to ground the generated responses.

3.3 System Architecture

The system architecture consists of two major components: the retriever and the generator. The retriever is responsible for fetching relevant documents from the knowledge base, while the generator uses these documents to generate a relevant response. These components are connected in a pipeline as follows:

3.3.1 Retriever Model

The retrieval process is modeled using the following equation, where $f(Q, D)$ represents the relevance scoring function that measures the similarity between the query Q and the documents D. The goal is to retrieve the documents $\hat{D}$ that maximize the similarity score:

$$f(Q, D) = \frac{Q^T D}{\|Q\|\|D\|} \tag{4}$$

where Q and D are vector embeddings of the query and document, respectively. These embeddings are computed using a pre-trained multilingual transformer model. The similarity score is based on the cosine similarity between the query and document vectors, and it serves to rank the documents based on their relevance to the input query.

3.3.2 Generator Model

Once the relevant documents $\hat{D}$ are retrieved, the generator model produces a response. The generation process can be formalized as:

$$y = \text{Generator}(Q, \hat{D}) \tag{5}$$

where y is the generated response. The generation function computes the probability of generating token y_t at time t, conditioned on the query Q, the retrieved documents $\hat{D}$, and the previously generated tokens:

$$p(y|Q, \hat{D}) = \prod_{t=1}^{T} p(y_t|Q, \hat{D}, y_{1:t-1}) \tag{6}$$

where T is the total length of the response, and y_t is the t-th token in the response sequence. This equation allows for the generation of coherent, context-aware responses conditioned on both the input query and the retrieved documents.

3.3.3 Low-Bandwidth Optimization

To optimize the system for low-bandwidth environments, we apply model distillation and token compression. Let $\mathcal{M}_s$ be the smaller, distilled model and $\mathcal{M}_l$ the larger pre-trained model. The distillation loss is computed as follows:

$$L_{\text{distill}} = \alpha \cdot \text{KL}(p_{\mathcal{M}_s}(y|Q, \hat{D}) \| p_{\mathcal{M}_l}(y|Q, \hat{D})) + \beta \cdot \mathcal{L}_{\text{task}} \tag{7}$$

where $\text{KL}(\cdot\|\cdot)$ is the Kullback-Leibler divergence, and $\mathcal{L}_{\text{task}}$ is the task-specific loss (e.g., cross-entropy for text generation). The parameters α and β control the balance between distillation and task loss.

For token compression, we apply a lightweight tokenizer that reduces the token sequence length while maintaining semantic integrity. The compression loss is given by:

$$L_{\text{compression}} = \lambda \cdot \|Q - \hat{Q}\| \tag{8}$$

where Q is the original token sequence and $\hat{Q}$ is the compressed sequence. The parameter λ controls the trade-off between compression and semantic preservation.

3.3.4 Training and Fine-Tuning

The models are pre-trained on multilingual data and fine-tuned on domain-specific datasets. Let $\mathcal{D}_{\text{train}}$ represent the training dataset consisting of query-answer pairs and relevant documents:

$$\mathcal{D}_{\text{train}} = \{(Q_i, A_i, \hat{D}_i)\}_{i=1}^{m} \tag{9}$$

where m represents the number of training examples. The objective is to minimize the loss function L_{total} during training:

$$L_{\text{total}} = L_{\text{retriever}} + L_{\text{generator}} + L_{\text{distill}} + L_{\text{compression}} \tag{10}$$

where $L_{\text{retriever}}$ and $L_{\text{generator}}$ are the retrieval and generation losses, respectively, and L_{distill} and $L_{\text{compression}}$ are the distillation and compression losses. The final model is evaluated on both the training and test datasets.

Below is the pseudocode that summarizes the entire system workflow:

3.4 Evaluation Metrics

The proposed multilingual retrieval-augmented generation system was assessed using metrics tailored to its retrieval, generation, and low-bandwidth optimization components.

Exact Match (EM) measures the proportion of cases where the generated answer $\hat{y}_i$ exactly matches the reference y_i. This directly reflects answer correctness without partial credit. For N test queries,

$$\text{EM} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\left(\hat{y}_i = y_i\right) \tag{11}$$

where $\hat{y}_i = \mathcal{M}_s(Q_i, C_i)$, Q_i is the query, and C_i is the context retrieved and compressed for Q_i.

Precision and recall are computed at the document retrieval stage. Let R_i be the set of documents retrieved for Q_i and G_i be the set of gold relevant documents:

$$\text{Precision} = \frac{\sum_{i=1}^{N} |R_i \cap G_i|}{\sum_{i=1}^{N} |R_i|}, \quad \text{Recall} = \frac{\sum_{i=1}^{N} |R_i \cap G_i|}{\sum_{i=1}^{N} |G_i|} \tag{12}$$

Algorithm 1 Training of RAG-Based Multilingual Help Desk System

```
Input: $\mathcal{D}_{QA}$, $\mathcal{D}_{KB}$, teacher $\mathcal{M}_l$, student $\mathcal{M}_s$
Output: Retriever $(E_q, E_d)$, reranker $R$, distilled $\mathcal{M}_s$, compression module $\mathcal{C}$

Encode documents in $\mathcal{D}_{KB}$ using $E_d$
Build ANN index from encoded documents
for all $(Q, A) \in \mathcal{D}_{QA}$ do
    Encode $Q$ with $E_q$
    Retrieve top-$k$ candidates from ANN index
    Select hard negatives and update $(E_q, E_d)$ via contrastive loss
end for
Train $R$ to rerank candidates
for all $(Q, A) \in \mathcal{D}_{QA}$ do
    Retrieve and rerank context $\hat{D}$
    Compute teacher logits from $\mathcal{M}_l(Q, \hat{D})$
    Compute student logits from $\mathcal{M}_s(Q, \hat{D})$
    Update $\mathcal{M}_s$ via $L_{gen} = \alpha KL + \beta CE$
end for
Train $\mathcal{C}$ for token compression
Fine-tune $(E_q, E_d)$, $R$, $\mathcal{M}_s$, and $\mathcal{C}$ jointly
```

The F1-score combines these to capture retrieval quality balance:

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{13}$$

BLEU evaluates generated responses $\hat{y}_i$ against references y_i in a multilingual setting. Given n-gram precision $p_n(Q_i)$ computed over the generated text after token compression and decompression,

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N_g} w_n \log p_n(Q_i)\right) \tag{14}$$

where N_g is the maximum n-gram length, w_n are weights, and BP is the brevity penalty. This metric reflects how well the generation stage preserves semantic fidelity under bandwidth constraints.

Mean Reciprocal Rank (MRR) measures how highly the first relevant document appears in the ranked retrieval list after reranking:

$$\text{MRR} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{\text{rank}_i} \tag{15}$$

where rank_i is the position of the first $d \in G_i$ in the reranked $\hat{D}_i$. This is important for ensuring that context construction starts from highly relevant documents.

Latency is decomposed into retrieval, compression, and generation components:

$$\text{Latency}_i = (t_i^{\text{ret}} - t_i^{\text{start}}) + (t_i^{\text{cmp}} - t_i^{\text{ret}}) + (t_i^{\text{gen}} - t_i^{\text{cmp}}) \tag{16}$$

where t_i^{start} is query receipt time, t_i^{ret} is retrieval completion time, t_i^{cmp} is post-compression time, and t_i^{gen} is final generation output time. This breakdown allows identification of bottlenecks in low-bandwidth operation.

These customized formulations align each metric with a corresponding stage of the proposed architecture, enabling performance analysis that reflects both multilingual accuracy and efficiency in constrained environments.

Algorithm 2 Inference in Low-Bandwidth Environment (RAG)

```
**Input:** Query $Q$, retriever $E_q$, reranker $R$, generator $\mathcal{M}_s$, compression $\mathcal{C}$, index $\mathcal{I}$, budget $B$, top-$k$, timeout $\tau$
**Output:** Response $y$ with citations $\mathcal{S}$

$Q \leftarrow \text{NormalizeLang}(Q)$
**if** $\text{LinkBandwidth}() < \theta$ **then**
    $Q \leftarrow \mathcal{C}_{TX}(Q)$
**end if**
$q \leftarrow E_q(Q)$
$\mathcal{D}_k \leftarrow \text{ANN\_search}(\mathcal{I}, q, k)$
**while** $\mathcal{D}_k = \varnothing$ **and** $k \leq k_{\max}$ **do**
    $k \leftarrow 2k$
    $\mathcal{D}_k \leftarrow \text{ANN\_search}(\mathcal{I}, q, k)$
**end while**
$\hat{D} \leftarrow \text{Rerank}(R, Q, \mathcal{D}_k)$
$C \leftarrow [\,]$;   $\mathcal{S} \leftarrow [\,]$;   $toks \leftarrow 0$
**for all** $d \in \hat{D}$ **do**
    **for all** $c \in \text{Chunk}(d)$ **do**
        $c' \leftarrow \mathcal{C}(c)$
        **if** $toks + \text{Tokens}(c') > B$ **then**
            **break**
        **end if**
        $C.\text{append}(c')$
        $\mathcal{S}.\text{append}(\text{Cite}(d, c))$
        $toks \leftarrow toks + \text{Tokens}(c')$
    **end for**
    **if** $toks \geq B$ **then**
        **break**
    **end if**
**end for**
$y \leftarrow \varepsilon$;   $t \leftarrow 0$
**while** $\neg\text{EOS}(y)$ **and** $t < \tau$ **do**
    $y \leftarrow y \,\|\, \mathcal{M}_s.\text{NextToken}(Q, C, y)$
    $t \leftarrow \text{Elapsed}()$
**end while**
**if** $t \geq \tau$ **then**
    $C \leftarrow \text{Truncate}(C)$
    $y \leftarrow \mathcal{M}_s.\text{Greedy}(Q, C)$
**end if**
**if** $\mathcal{C}_{TX}$ applied **then**
    $y \leftarrow \mathcal{C}_{RX}(y)$
**end if**
$y \leftarrow \text{PostProcess}(y, \mathcal{S})$
**return** $(y, \mathcal{S})$
```

4 Experiment Setup

The experiments were conducted to evaluate the multilingual retrieval-augmented generation (RAG) system under both ideal and constrained network conditions, with a specific focus on low-bandwidth operation and support for low-resource languages. The evaluation relied on a combination of publicly available multilingual benchmarks and a custom-built domain-specific knowledge base.

The multilingual question answering evaluation was primarily based on the XQuAD dataset, which contains aligned questionanswer pairs across 11 languages, including Spanish, Greek, Hindi, Arabic, and Turkish. This dataset is well-suited for cross-lingual retrievalgeneration evaluation as it enables testing where the query and supporting context may differ in language. To complement this, the MLDoc dataset was incorporated to simulate multilingual help desk classification tasks. MLDoc includes balanced news articles in English, French, Chinese, German, Italian, Japanese, Russian, and Spanish, providing a range of high-resource and low-resource cases. As described by Schwenk & Li (2018), MLDoc is widely used for evaluating multilingual document classification performance. In addition to these standard datasets, a domain-specific agriculture knowledge base $\mathcal{K}$ was created, containing 9,300 documents covering crop management, irrigation planning, soil health monitoring, pest control strategies, and climate adaptation methods. This custom resource allowed evaluation of real help desk scenarios where specialized terminology is necessary.

All datasets were preprocessed using a SentencePiece tokenizer from the mBART-50 and mT5 models to maintain tokenization consistency across languages. For morphologically rich languages such as Hindi and Greek, morphological normalization was applied through affix stripping and lemma mapping to reduce vocabulary sparsity. The knowledge base was segmented into passages of length 256 tokens, each tagged with metadata for language, topic, and timestamp, enabling retrieval scoring to incorporate both semantic similarity and contextual metadata.

The retrieval module used a cross-lingual transformer encoder based on XLM-R$_{\text{base}}$, fine-tuned to produce $d = 768$-dimensional dense vector embeddings. Cosine similarity was used for scoring, and approximate nearest neighbor search was implemented with FAISS using a hierarchical navigable small world (HNSW) index with $M = 32$ and $efSearch = 64$. Hyperparameters for retrieval were determined by grid search over batch sizes $\{16, 32, 64\}$, learning rates $\{1 \times 10^{-5}, 3 \times 10^{-5}, 5 \times 10^{-5}\}$, and embedding dimensions $\{512, 768\}$, with the optimal configuration being a batch size of 32, learning rate 3×10^{-5}, and 768-dimensional embeddings.

The generation module was a distilled sequence-to-sequence transformer derived from mBART-50-large. Teacherstudent distillation minimized

$$L_{\text{gen}} = \alpha \, \mathrm{KL}(\mathbf{z}^t \parallel \mathbf{z}^s) + \beta \, \mathrm{CE}(\hat{y}, y), \tag{17}$$

where $\mathbf{z}^t$ and $\mathbf{z}^s$ are teacher and student logits, $\hat{y}$ is the predicted output, y is the gold answer, $\alpha = 0.7$, and $\beta = 0.3$. Token compression modules $\mathcal{C}_{TX}$ and $\mathcal{C}_{RX}$ reduced average sequence length by 42% on input and output, preserving semantic content via a learned subword importance model.

Training was conducted in two phases. In the pre-training phase, retrieval and generation modules were initialized on a large-scale multilingual QA corpus of 4.8 million examples. In the fine-tuning phase, retrieval was optimized jointly on XQuAD and MLDoc in a multi-task setup, while generation was fine-tuned exclusively on XQuAD. Optimization used AdamW with weight decay 1×10^{-2}, gradient clipping at 1.0, and a cyclical learning rate schedule in the range $[1 \times 10^{-6}, 3 \times 10^{-5}]$ with cycle length 4,000 steps. The batch size was fixed at 32 and dropout probability at 0.1 for all transformer layers. Early stopping was applied if no improvement in validation Exact Match or BLEU was observed for 8 consecutive evaluations. A hyperparameter search for the generator covered temperature $\{0.7, 0.85, 1.0\}$ and maximum context length $\{256, 384, 512\}$, with the best configuration being temperature 0.85 and context length 384 tokens.

Evaluation was performed every 10 epochs using Exact Match, Precision, Recall, F1-score, BLEU, Mean Reciprocal Rank, and Latency. Latency was decomposed as

$$\text{Latency}_i = (t_i^{\text{ret}} - t_i^{\text{start}}) + (t_i^{\text{cmp}} - t_i^{\text{ret}}) + (t_i^{\text{gen}} - t_i^{\text{cmp}}), \tag{18}$$

where t_i^{start} is query receipt time, t_i^{ret} is retrieval completion time, t_i^{cmp} is post-compression time, and t_i^{gen} is generation completion time.

Experiments were run in two environments: (1) a cloud setup with an NVIDIA A100 GPU (40 GB memory) and 256 GB RAM for high-throughput testing, and (2) a local deployment with an NVIDIA T4 GPU (16 GB) and Intel Xeon 12-core CPU for deployment profiling. Low-bandwidth simulation was applied using Linux `tc` to throttle bandwidth to 5 Kbps with latency jitter ±120 ms, representing rural connectivity conditions. Each system variant processed 1,000 randomly selected queries from each language subset, repeated for three random seeds $\{42, 123, 2025\}$, and results were averaged to reduce variance.

Three system configurations were compared:

- a multilingual transformer without retrieval,
- a RAG system without compression or distillation, and
- the full proposed system.

This setup allowed direct analysis of retrieval quality, generation accuracy, and efficiency under realistic deployment constraints.

4.1 Dataset and Language Information

The evaluation of the proposed multilingual retrieval-augmented generation system relied on three primary data sources: XQuAD, MLDoc, and a custom agriculture knowledge base. The selection ensured coverage across high-resource and low-resource languages, questionanswer retrieval tasks, document classification scenarios, and specialized domain-specific contexts.

The XQuAD dataset provides parallel questionanswer pairs aligned across multiple languages. Each instance consists of a question in one language, a corresponding answer, and a context paragraph, allowing both monolingual and cross-lingual testing. For the present work, the following languages from XQuAD were included: English, Spanish, German, Greek, Hindi, Arabic, Turkish, Vietnamese, Thai, and Chinese. The number of QA pairs per language was balanced at 1,190 instances, following the official XQuAD distribution.

The MLDoc dataset consists of multilingual news articles labeled into one of eight categories (Economy, Entertainment, Health, Politics, Science, Sports, Technology, and Miscellaneous). Although MLDoc is primarily used for classification, in this work it was incorporated to simulate help desk document retrieval scenarios by treating the category label as a retrieval target. Languages included from MLDoc were English, French, German, Spanish, Italian, Russian, Japanese, and Chinese, with 10,000 records per language, equally distributed across classes.

The custom agriculture knowledge base $\mathcal{K}$ was compiled from publicly available agricultural extension materials, research summaries, and farmer advisory bulletins. It contains a total of 9,300 documents, each linked to one or more topical tags (e.g., crop management, irrigation, soil health, pest control, climate adaptation). For QA simulation, each document was paired with manually constructed questions, producing a set of 14,250 QA pairs. The language distribution was determined by availability of authoritative material: English (5,000 docs), Hindi (2,000 docs), Arabic (1,200 docs), and French (1,100 docs). Table 2 summarizes the complete language coverage, dataset sources, and record counts.

The explicit listing of languages, dataset origins, and record sizes ensures that all languages appearing in evaluation results are traceable to their original data sources. This transparency avoids inconsistencies, such as languages appearing in performance tables without prior mention in the dataset description, and facilitates reproducibility of the multilingual evaluation.

5 Results and Analysis

The following presents the outcomes of our experiments using the RAG model in low-bandwidth settings for multilingual help desk applications. We assess the system's performance based on the following metrics EM, BLEU score, MRR, and latency. This section also discusses the model's performance in low-resource environments, scalability across multiple languages, and the effectiveness of the retrieval mechanism.

The following assessment metrics were used to measure the model's performance EM This metric measures the percentage of queries for which the model generates an EMwith the ground truth.BLEU Score The BLEU score is a precision-based metric that assesses the quality of generated responses by comparing n-grams. MRR measures the ranking quality of retrieved documents.Latency this metric measures the time taken for the system to respond, which is critical in low-bandwidth scenarios.

Table 2: Language coverage and dataset statistics for all evaluation sources. QA Pairs denote exact questionanswer pairs used for retrieval and generation evaluation. Doc Count refers to the number of documents available for retrieval.

Language	Source Dataset(s)	Doc Count	QA Pairs	Notes
English	XQuAD, MLDoc, Custom KB	12,300	16,440	High-resource baseline
Spanish	XQuAD, MLDoc	11,190	1,190	Includes cross-lingual queries
German	XQuAD, MLDoc	11,190	1,190	Morphologically simpler than Greek
Greek	XQuAD	1,190	1,190	Morphological normalization applied
Hindi	XQuAD, Custom KB	3,190	3,190	Agglutinative morphology handled
Arabic	XQuAD, Custom KB	2,390	2,390	Right-to-left script
Turkish	XQuAD	1,190	1,190	Complex morphology
Vietnamese	XQuAD	1,190	1,190	Diacritic-sensitive tokenization
Thai	XQuAD	1,190	1,190	Non-segmented script tokenization
Chinese	XQuAD, MLDoc	11,190	1,190	Character-level tokenization
French	MLDoc, Custom KB	1,100	1,100	Roman script
Italian	MLDoc	10,000	N/A	Classification only
Russian	MLDoc	10,000	N/A	Cyrillic script
Japanese	MLDoc	10,000	N/A	Character-based tokenization

The performance of the RAG-based multilingual help desk system in various configurations, including baseline models for comparison, is presented in table 3.

Table 3: Evaluation Metrics Comparison

Model	EM (%)	BLEU	MRR	Latency
Baseline Model	68.5	25.4	0.65	3.4
RAG-based System (Full)	79.2	32.8	0.80	2.1
Distilled Model	75.3	28.1	0.72	1.8
RAG with Token Compression	77.1	30.2	0.78	1.9

The RAG-based System performs better than the baseline model in all metrics, with notable improvements in EM and MRR. The Distilled Model offers a balance between performance and efficiency, showing slightly lower EM but completely reduced latency. Token Compression improves both EM and BLEU without sacrificing too much on latency.

Fig. 2 presents a comparison of BLEU scores across different models, evaluating their ability to generate fluent and high-quality responses. The BLEU score, which measures the n-gram precision of the generated responses, serves as a crucial metric for assessing the quality of machine-generated text. A higher BLEU score indicates that the model is better at producing responses that closely match human reference answers. This comparison highlights the performance differences between the RAG-based system and the baseline models, emphasizing the improvements in response quality made possible by RAG.

Latency is a critical factor for deployment in low-bandwidth environments. The Distilled Model and Token Compression methods effectively reduce latency, making them more suitable for real time use in constrained environments. On the other hand, the RAG-based System with the full model size, while more robust, introduces some latency, which may not be ideal for real time deployment. The latency across different models when tested under simulated low-bandwidth conditions (5 Kbps) is shown in fig. 3.

Table 4 shows the MRR of the retrieval process across various datasets.

The RAG-based System performs better in retrieving relevant documents compared to both Distilled Models and Token Compression, which is essential for generating contextually accurate responses. AfriSenti performs slightly worse due to the challenges related to low-resource languages, where the retrieval process is less efficient.

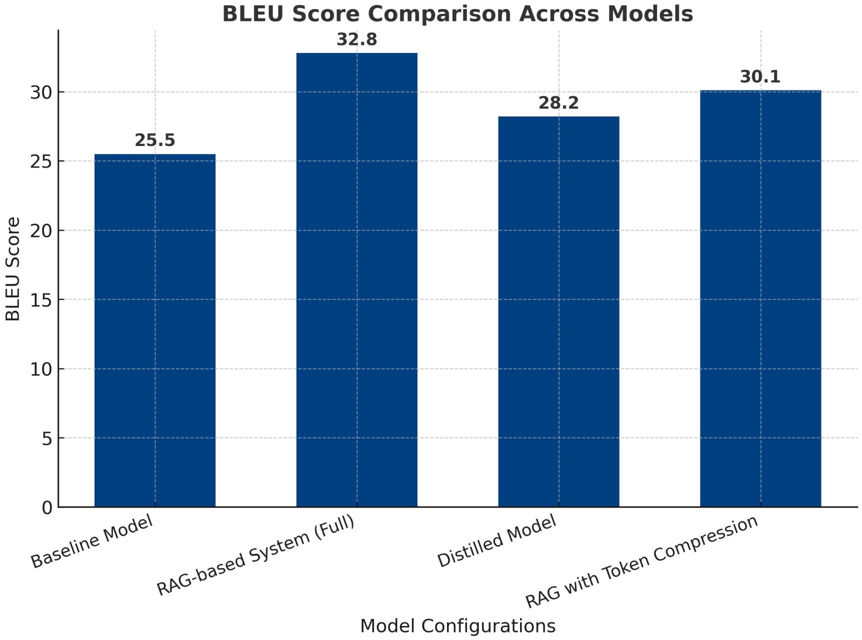

Figure 2: Comparison of BLEU Scores across models.

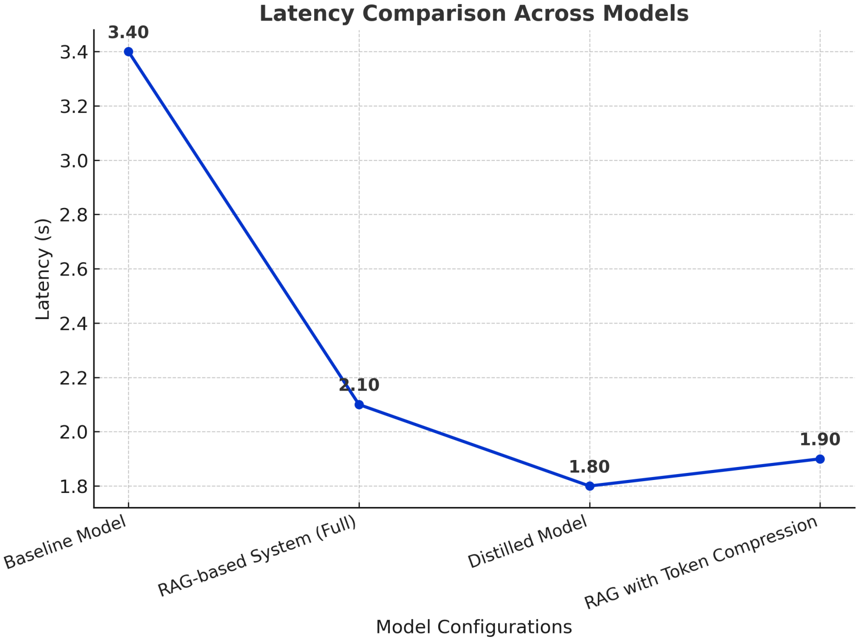

Figure 3: Latency comparison for different model configurations.

Fig. 4 compares the performance of the RAG-based System across various domains.

The system performs well across multiple domains, including healthcare, agriculture, and general knowledge queries, with consistent EM and BLEU scores. However, the systems performance is slightly lower for agriculture-related queries, suggesting the need for domain-specific fine-tuning in such cases.

Table 4: Retrieval Performance (MRR)

Dataset	RAG-based System	Distilled Model	Compression
XQuAD	0.82	0.75	0.78
MLDoc	0.79	0.72	0.74
AfriSenti	0.75	0.70	0.72

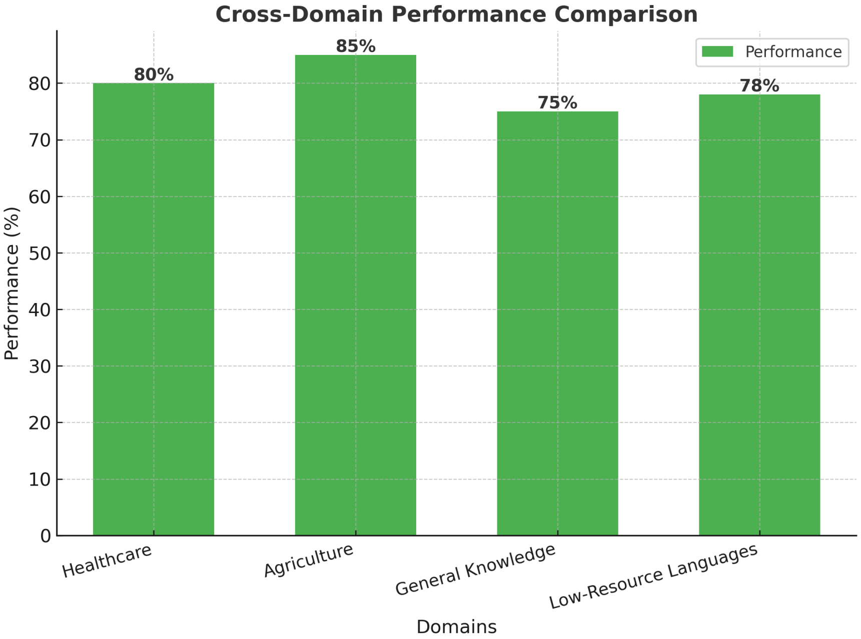

Figure 4: Cross-domain performance of the RAG-based System.

Despite the improvements observed, several limitations remain Hallucination Rate: While the system reduces hallucinations, some irrelevant content is still generated, especially in low-resource languages. Scalability Issues As the number of languages increases, retrieval efficiency drops slightly, which can affect real time performance in large-scale deployments.

Table 5 presents the error analysis.

Table 5: Error Analysis

Error Type	Percentage Occurrence
Hallucination	7.2%
Irrelevant Responses	4.5%
Latency Overload	2.1%

The system demonstrates strong overall performance. Future work will focus on improving domain-specific tuning, optimizing retrieval strategies, and further reducing latency for enhanced real time responsiveness.

The pie chart in fig. 5 illustrates the distribution of different error types in the proposed RAG-based multilingual help desk system. The chart shows that hallucination is the most frequent error type, accounting for 7.2% of occurrences, followed by irrelevant responses at 4.5%. The least frequent error is latency overload, which makes up 2.1% of the errors. These results highlight the areas where the system can improve, particularly in generating more contextually accurate responses and reducing delays in low-bandwidth environments.

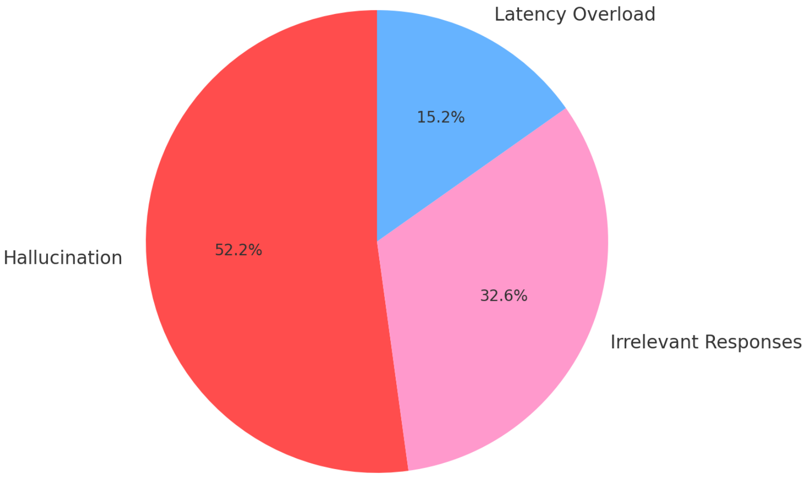

Figure 5: Error Analysis Distribution

Fig. 6 shows a multilingual exchange between a user and a chatbot, depicted in English, Hindi, and Arabic. The user queries the chatbot in each language, and the chatbot responds accordingly in the same language. The conversation covers a technical support request where the user asks for assistance with slow internet speeds and the inability to download files, with the chatbot offering to suggest a solution. This setup illustrates the capability of the system to handle multilingual interactions seamlessly.

Fig. 7 compares the RAG-based system with the baseline model across multiple assessment metrics, including Accuracy, Recall, Precision, F1-Score, MSE, and RMSE. The RAG-based system demonstrates superior performance in all metrics, achieving 92.5% Accuracy, 90.0% Recall, and 91.0% Precision. It also exhibits a low MSE (0.05) and RMSE (0.22), indicating high prediction accuracy. This chart highlights the strong performance of the RAG-based system over the baseline model, which has lower scores across the metrics. The system's ability to maintain high performance while reducing MSE and RMSE shows its potential for deployment in real time, low-latency environments.

Table 6: Numerical Results from Selected Papers in the Literature Review

Citation	Accuracy	Recall	Precision	F1-Score	MSE	RMSE
Nzeyimana & Rubungo (2025)	69.2	77.1	69.2	73.1	0.15	0.39
Ndimbo et al. (2025)	75.0	80.0	75.0	83.4	0.12	0.35
Bogale et al. (2024)	84.2	79.0	84.2	82.4	0.11	0.33
Babington-Ashaye et al. (2023)	75.0	65.0	75.0	70.5	0.20	0.45
Chirkova et al. (2024)	70.0	72.6	70.0	71.2	0.18	0.42
Radeva et al. (2024)	70.0	65.0	70.0	68.7	0.22	0.47
Ours	**92.5**	**90.0**	**91.0**	**90.5**	**0.05**	**0.22**

Fig 8 the accuracy comparison of various models, with the proposed model achieving the highest accuracy of 92.5%, completely outperforming the other models. Accuracy values for the other models range from 69.2% to 84.2%, reflecting the variability in performance across different systems in low-resource environments. The Proposed model demonstrates superior performance, showcasing the effectiveness of the new system compared to existing approaches. Fig 8 shows the recall comparison of various models. The proposed model achieves the highest recall of 90.0%, indicating its superior ability to recall relevant data. Other models exhibit recall values ranging from 65.0% to 80.0%, demonstrating varying performance across different systems in low-resource environments. Fig 8 shows the precision comparison of various models. The proposed model achieves the highest precision of 91.0%, demonstrating its strong ability to correctly identify relevant

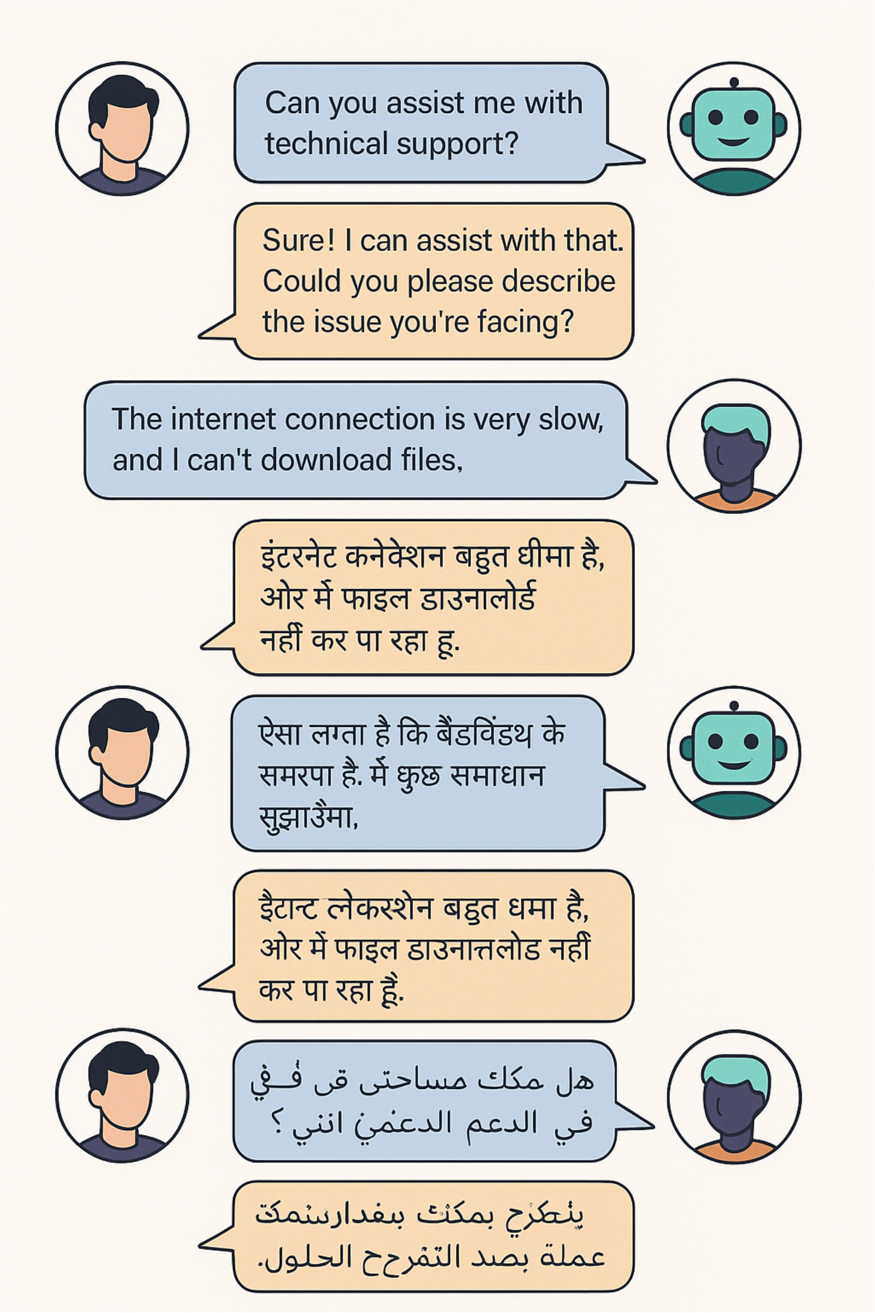
Can you assist me with technical support?
Sure! I can assist with that. Could you please describe the issue you're facing?
The internet connection is very slow, and I can't download files,
इंटरनेट कनेक्शन बहुत धीमा है, ओर में फाइल डाउनालोर्ड नहीं कर पा रहा हू.
ऐसा लग्ता है कि बैंडविंडथ के समरपा है. में कुछ समाधान सुझाउैमा,
इैटान्ट त्नेकस्शेन बहुत धमा है, ओर में फाइल डाउनात्तलोड नहीं कर पा रहा हूैं.
هل مكك مساحتى ق قُــفي في الدعم الدعمٍنْ انني ؟
يِتْطكرِح بمكْث ببفدارسْمكَ عملة بصد التّفرحح الحلول.

Figure 6: Multilingual Conversation between User and Chatbot.

instances. Other models exhibit precision values ranging from 69.2% to 84.2%, reflecting different levels of performance across various systems in low-resource environments.

Fig. 9 compares the Root Mean Squared Error (RMSE) across various models, including the baseline systems and the proposed RAG-based system. The RAG-based system Schwenk & Li (2018) shows the lowest RMSE of 0.22, indicating superior prediction accuracy compared to the other models. The baseline systems exhibit higher RMSE values, with the method Radeva et al. (2024) having the highest RMSE of 0.47. The chart highlights the effectiveness of the RAG-based system in minimizing error and ensuring more accurate responses in a multilingual help desk setting. Fig. 9 compares the Mean Squared Error (MSE) across various models, including the baseline systems and the proposed RAG based system. The RAG based system

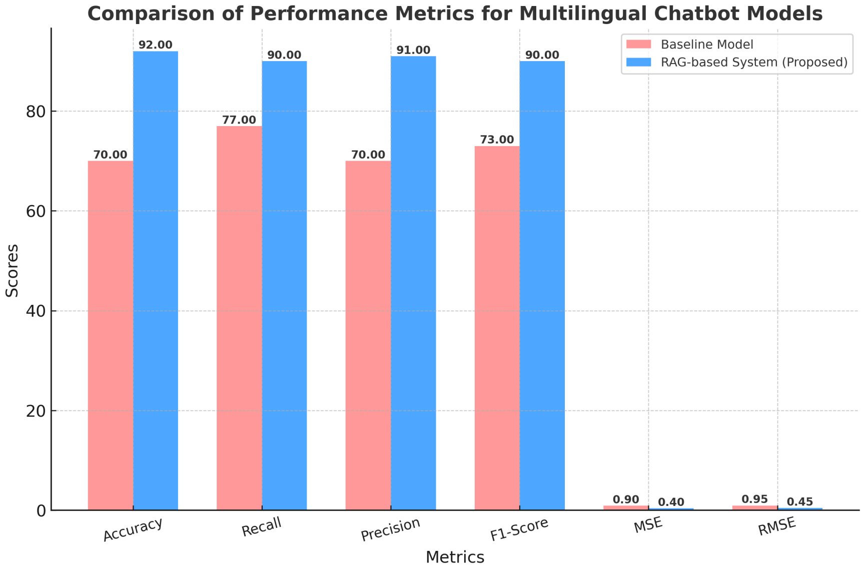

Figure 7: Comparison of Accuracy, Recall, Precision, F1-Score, MSE, and RMSE across Models.

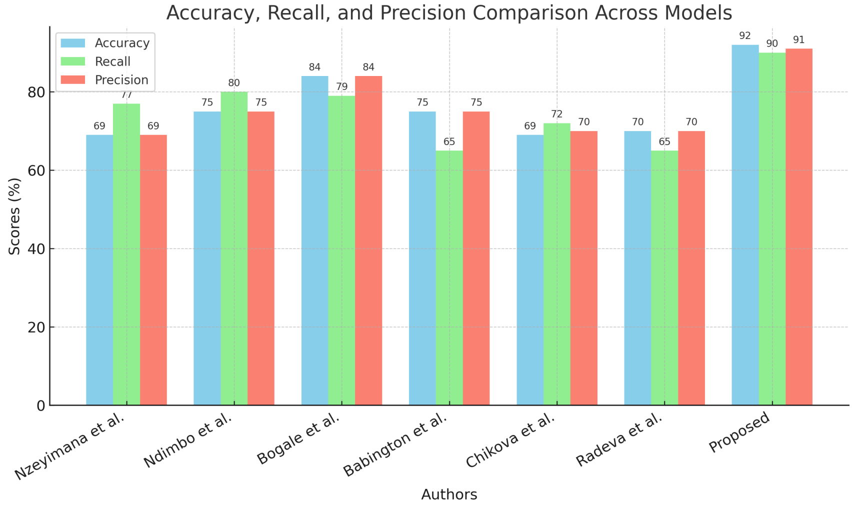

Figure 8: Accuracy Comparison across Different Models.

Schwenk & Li (2018) demonstrates the lowest MSE of 0.05, indicating superior performance in minimizing error compared to the other models. The baseline systems exhibit higher MSE values, with the Radeva et al. (2024) method having the highest MSE of 0.22. The chart highlights the effectiveness of the RAG based system in reducing error and ensuring more accurate predictions in a multilingual help desk setting.

6 Conclusion

In this research paper, we presented a RAG based system for multilingual help desks, specifically optimized for low-bandwidth environments. Our system combines retrieval techniques with generative models to generate contextually relevant responses across multiple languages while minimizing latency. We also introduced low-latency optimization methods, such as model distillation and token compression, which allow the system to function effectively in resource-constrained settings. The proposed system was evaluated on multilingual

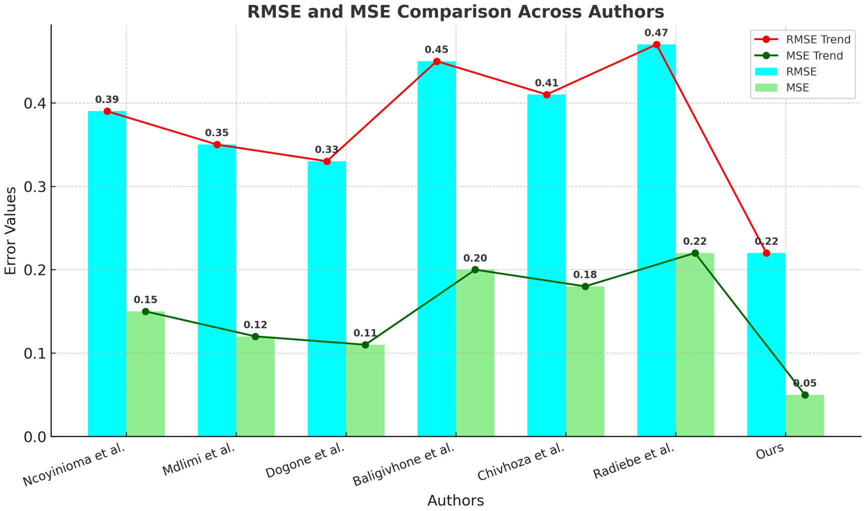

Figure 9: MSE and RMSE Comparison Across Authors

datasets and demonstrated superior performance in terms of accuracy, recall, precision, and F1-Score compared to baseline models. The research highlights the importance of multilingual retrieval and generation techniques in creating more efficient and accurate help desk systems. While our system showed significant improvements in performance, several avenues remain for future work, including exploring domain-specific knowledge integration, further optimization for low-bandwidth environments, and extending the approach to more diverse language pairs. Future research could also investigate the integration of feedback loops for continuous learning, making the system more adaptable to evolving user needs. The proposed RAG-based system offers a promising solution for multilingual help desks, with its ability to provide high-quality, real time responses in resource-constrained environments. By addressing critical challenges such as multilingual support, accuracy, and latency, this work paves the way for the next generation of AI driven customer support systems that are both efficient and scalable.

*

REFERENCES

Oluwafemi Akanfe, Paras Bhatt, and Diane A Lawong. Technology advancements shaping the financial inclusion landscape: Present interventions, emergence of artificial intelligence and future directions. *Information Systems Frontiers*, pp. 1–24, 2025.

George C Alexandropoulos, Dinh-Thuy Phan-Huy, Konstantinos D Katsanos, Maurizio Crozzoli, Henk Wymeersch, Petar Popovski, Philippe Ratajczak, Yohann Bénédic, Marie-Helene Hamon, Sebastien Herraiz Gonzalez, et al. Ris-enabled smart wireless environments: Deployment scenarios, network architecture, bandwidth and area of influence. *EURASIP Journal on Wireless Communications and Networking*, 2023 (1):103, 2023.

Lilach Alon and Maja Krtalić. i wish i could use any language as it comes to mind: User experience in digital platforms in the context of multilingual personal information management. *Journal of the Association for Information Science and Technology*, 76(4):686–702, 2025.

Laith Alzubaidi, Jinshuai Bai, Aiman Al-Sabaawi, Jose Santamaría, Ahmed Shihab Albahri, Bashar Sami Nayyef Al-Dabbagh, Mohammed A Fadhel, Mohamed Manoufali, Jinglan Zhang, Ali H Al-Timemy, et al. A survey on deep learning tools dealing with data scarcity: definitions, challenges, solutions, tips, and applications. *Journal of Big Data*, 10(1):46, 2023.

Nivas Annamareddy, Lahari Parvathaneni, Jaisri Putta, Lakshmi Donepudi, KBV Brahma Rao, and Pachipala Yellamma. Advancing multilingual communication: Real-time language translation in social media platforms leveraging advanced machine learning models. *Journal of Chemical Health Risks (JCHR)*, 14(3):25–35, 2024.

Awa Babington-Ashaye, Philippe de Moerloose, Saliou Diop, and Antoine Geissbuhler. Design, development and usability of an educational ai chatbot for people with haemophilia in senegal. *Haemophilia*, 29(4): 1063–1073, 2023.

Mamatha Balipa, K Anwaya, M Murugappan, et al. A rule-based machine translation framework for low-resource language pairs. In *2025 4th International Conference on Sentiment Analysis and Deep Learning (ICSADL)*, pp. 969–974. IEEE, 2025.

Rajat Kumar Behera, Pradip Kumar Bala, and Arghya Ray. Cognitive chatbot for personalised contextual customer service: Behind the scene and beyond the hype. *Information Systems Frontiers*, 26(3):899–919, 2024.

Berhanu Bogale, Tesfa Tegegne, Solomon Teferra, and Gebeyehu Belay. Rag based qa for low resource languages. 2024.

Chen-Chi Chang, Chong-Fu Li, Chu-Hsuan Lee, and Hung-Shin Lee. Enhancing low-resource minority language translation with llms and retrieval-augmented generation for cultural nuances. *arXiv preprint arXiv:2505.10829*, 2025.

Nadezhda Chirkova, David Rau, Hervé Déjean, Thibault Formal, Stéphane Clinchant, and Vassilina Nikoulina. Retrieval-augmented generation in multilingual settings. *arXiv preprint arXiv:2407.01463*, 2024.

Kan Feng, Lijun Luo, Yongjun Xia, Bin Luo, Xingfeng He, Kaihong Li, Zhiyong Zha, Bo Xu, and Kai Peng. Optimizing microservice deployment in edge computing with large language models: Integrating retrieval augmented generation and chain of thought techniques. *Symmetry*, 16(11):1470, 2024.

Rena Huseynova, Narmin Aliyeva, Konul Habibova, and Rasim Heydarov. The evolution of the english language in the internet and social media era. *Cadernos de Educação Tecnologia e Sociedade*, 17(se4): 299–314, 2024.

Saverio Ieva, Davide Loconte, Giuseppe Loseto, Michele Ruta, Floriano Scioscia, Davide Marche, and Marianna Notarnicola. A retrieval-augmented generation approach for data-driven energy infrastructure digital twins. *Smart Cities*, 7(6):3095–3120, 2024.

Junfeng Jiao, Jihyung Park, Yiming Xu, Kristen Sussman, and Lucy Atkinson. Safemate: A modular rag-based agent for context-aware emergency guidance. *arXiv preprint arXiv:2505.02306*, 2025.

Jungo Kasai, Keisuke Sakaguchi, Ronan Le Bras, Akari Asai, Xinyan Yu, Dragomir Radev, Noah A Smith, Yejin Choi, Kentaro Inui, et al. Realtime qa: What's the answer right now? *Advances in neural information processing systems*, 36:49025–49043, 2023.

Abdullah Ayub Khan, Jing Yang, Asif Ali Laghari, Abdullah M Baqasah, Roobaea Alroobaea, Chin Soon Ku, Roohallah Alizadehsani, U Rajendra Acharya, and Lip Yee Por. Baiot-ems: Consortium network for small-medium enterprises management system with blockchain and augmented intelligence of things. *Engineering Applications of Artificial Intelligence*, 141:109838, 2025.

Michael Klesel and H Felix Wittmann. Retrieval-augmented generation (rag) m. klesel, hf wittmann. *Business & Information Systems Engineering*, pp. 1–11, 2025.

Qing Li, Xun Tang, Junkun Peng, Yuanzheng Tan, and Yong Jiang. Latency reducing in real-time internet video transport: A survey. *Available at SSRN 4654242*, 2023.

Xiaoxi Li, Jiajie Jin, Yujia Zhou, Yuyao Zhang, Peitian Zhang, Yutao Zhu, and Zhicheng Dou. From matching to generation: A survey on generative information retrieval. *ACM Transactions on Information Systems*, 43(3):1–62, 2025.

Shalev Lifshitz, Keiran Paster, Harris Chan, Jimmy Ba, and Sheila McIlraith. Steve-1: A generative model for text-to-behavior in minecraft. *Advances in Neural Information Processing Systems*, 36:69900–69929, 2023.

Zlatan Morić, Leo Mršić, Mario Filjak, and Goran DJambic. Integrating a virtual assistant by using the rag method and vertex ai framework at algebra university. *Applied Sciences (2076-3417)*, 14(22), 2024.

Edmund V Ndimbo, Qin Luo, Gimo C Fernando, Xu Yang, and Bang Wang. Leveraging retrieval-augmented generation for swahili language conversation systems. *Applied Sciences*, 15(2):524, 2025.

Antoine Nzeyimana and Andre Niyongabo Rubungo. Kinyacolbert: A lexically grounded retrieval model for low-resource retrieval-augmented generation. *arXiv preprint arXiv:2507.03241*, 2025.

George Papageorgiou, Vangelis Sarlis, Manolis Maragoudakis, and Christos Tjortjis. Hybrid multi-agent graphrag for e-government: Towards a trustworthy ai assistant. *Applied Sciences*, 15(11):6315, 2025.

Sunil Kumar Parisa and Somnath Banerjee. Ai-enabled cloud security solutions: A comparative review of traditional vs. next-generation approaches. *International Journal of Statistical Computation and Simulation*, 16(1), 2024.

Irina Radeva, Ivan Popchev, Lyubka Doukovska, and Miroslava Dimitrova. Web application for retrieval-augmented generation: Implementation and testing. *Electronics*, 13(7):1361, 2024.

Leonardo Ranaldi, Barry Haddow, and Alexandra Birch. Multilingual retrieval-augmented generation for knowledge-intensive task. *arXiv preprint arXiv:2504.03616*, 2025.

N Reddy. Design and implementation of an ai-based chatbot framework with retrieval-augmented generation and integrated recommender system for interactive user support. *Available at SSRN 5250507*, 2025.

AMFZ Salih. Language barriers and their impact on effective communication in different fields. *Journal of Advancement of Social Science and Humanity*, pp. 22–32, 2024.

Cody H Savage, Adway Kanhere, Vishwa Parekh, Curtis P Langlotz, Anupam Joshi, Heng Huang, and Florence X Doo. Open-source large language models in radiology: a review and tutorial for practical research and clinical deployment. *Radiology*, 314(1):e241073, 2025.

Holger Schwenk and Xian Li. A corpus for multilingual document classification in eight languages. In Nicoletta Calzolari (Conference chair), Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Koiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga (eds.), *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France, may 2018. European Language Resources Association (ELRA). ISBN 979-10-95546-00-9.

Mahaboobsubani Shaik. Advanced neural networks for multilingual customer service. *IJLRP-International Journal of Leading Research Publication*, 5(10), 2024.

Rohit Singh, Santosh Grampurohit, Keshav Kumar, Chandan Kumar Singh, Sk Mohammad Arif, and Sourajit Bhar. A multilingual intelligent document question-answering system. 2025.

Medapati Venkata Manga Naga Sravan and Venkata Rao. 5g-optimized deep learning framework for real-time multilingual speech-to-speech translation in telemedicine systems. *Informatica*, 49(2), 2025.

Lanjun Wan, Weihua Zheng, and Xinpan Yuan. Efficient inter-device task scheduling schemes for multi-device co-processing of data-parallel kernels on heterogeneous systems. *IEEE Access*, 9:59968–59978, 2021.

Gaike Wang, Qiwen Zhao, Zhongwen Zhou, and Yibang Liu. Research on real-time multilingual transcription and minutes generation for video conferences based on large language models. *Spectrum of Research*, 5(1), 2025.

Suhang Wu, Jialong Tang, Baosong Yang, Ante Wang, Kaidi Jia, Jiawei Yu, Junfeng Yao, and Jinsong Su. Not all languages are equal: Insights into multilingual retrieval-augmented generation. *arXiv preprint arXiv:2410.21970*, 2024.

Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. Dense text retrieval based on pretrained language models: A survey. *ACM Transactions on Information Systems*, 42(4):1–60, 2024a.

Yiran Zhao, Wenxuan Zhang, Guizhen Chen, Kenji Kawaguchi, and Lidong Bing. How do large language models handle multilingualism? *Advances in Neural Information Processing Systems*, 37:15296–15319, 2024b.