

Cross-Lingual RAG for Question Answering with Confidence-Aware Hallucination Detection

Asha G R¹, Saritha A N², Harshita R³, Jhashuva Udayagiri⁴, Subhadip Goswami^{5*}, Sasmita Lenka⁶

¹Professor, Department of CSE, B.M.S. College of Engineering, Bengaluru

Email: asha.cse@bmsce.ac.in

²Assistant Professor, Department of CSE, B.M.S. College of Engineering, Bengaluru

Email: saritha.cse@bmsce.ac.in

³Postgraduate Student, Department of CSE, B.M.S. College of Engineering, Bengaluru

E-mail : harshita.scs24@bmsce.ac.in

⁴Research Scholar, Department of CSE, B.M.S. College of Engineering, Bengaluru

Email: joshua537.nit@gmail.com

⁵Assistant Professor, Department of Electrical Engineering, Sandip Institute of Technology & Research Centre, Nashik Email: subhadip.goswami@sitrc.org

⁶Assistant Professor, Department of Electrical Engineering, Sandip Institute of Technology & Research Centre, Nashik Email: sasmita.lenka@sitrc.org

HOW TO CITE: | ABSTRACT:

Asha G R, Saritha A N, Harshita R, Jhashuva Udayagiri, Subhadip Goswami, Sasmita Lenka (2026). Cross-Lingual RAG for Question Answering with Confidence-Aware Hallucination Detection International Journal of Special Education, 41(2), 938 - 961

Access to the government welfare information still remains unequal in linguistically diverse countries such as India, where a significant portion of the population speaks primarily in regional languages and they are substantially underserved by existing NLP infrastructure. Kannada, spoken by over 40 million people and recognised as a classical language, is one such case and yet no document-grounded question answering system currently exists for Indian government schemes in Kannada. This paper presents SchemeQA-Kannada, a retrieval-augmented system built around government schemes or programs. We constructed a manually verified dataset of 315 QA pairs, including 63 unanswerable questions designed to evaluate a system's ability to abstain rather than provide hallucinated responses to user. We developed a section-boosted MuRIL retrieval pipeline over a FAISS index, incorporating a tuned section-awareness score and a three-stage scheme detection mechanism combining exact alias matching, fuzzy string similarity, and MuRIL semantic similarity. We introduce a cross-lingual hallucination detection mechanism that translates Kannada answers to English before NLI entailment verification via mDeBERTa-v3-base-xnli, and a Source-Aware Confidence Score (SACS) that weights entailment score alongside source credibility tier. Evaluation was performed on a held-out test set of 48 pairs which shows that our system reduces hallucination rate to 16.1% against baselines ranging from 31.6%

COPYRIGHT STATEMENT:

Copyright: © 2026 Authors.
Open access publication under the
terms and conditions
of the Creative Commons Attribution
(CC BY)
license (<http://creativecommons.org/licenses/by/4.0/>).

to 100%, while achieving a BERT Score of 0.7468, which is considerably above most comparators. Ablation analysis identifies that cross-lingual NLI verification as the most impactful component, which is responsible for the largest reduction in hallucination relative to naive retrieval. Existing systems for Indian government scheme advisory do not address hallucination detection or source credibility in low-resource Indic languages, this work targets that gap directly for Kannada language.

Keywords: Kannada question answering, retrieval-augmented generation, hallucination detection, cross-lingual NLI, government scheme information retrieval, source-aware confidence scoring.

Introduction

India's welfare architecture is very vast that hundreds of central and state government schemes collectively touch nearly every aspect of civic life such as agricultural support, health insurance, financial inclusion, and education savings, yet the population for whom these schemes are designed to serve often have the least access to the clear and reliable information about them. Consider a farmer in rural Karnataka who wants to know whether his/her land holdings make them eligible for PM-KISAN, or a parent who is looking for the documents which are needed to open a Sukanya Samriddhi Yojana account for their daughter, will expect to get a precise grounded answer. In reality, that expectation is rarely met. The Official portals are complex to navigate, their helpline wait times are long, and the information is fragmented across PDFs, government websites, and third-party aggregators are of varying credibility.

Conversational AI offers a realistic path forward here. Large language models (LLMs) have demonstrated impressive results across many languages, and retrieval-augmented generation (RAG) has improved factual grounding in knowledge-intensive tasks [3], but deploying such systems for government information access in India introduces a set of challenges that general-purpose solutions do not address sufficiently. Language is the main among them. India's 22 scheduled languages represent enormously diverse morphological and syntactic structures,

and most of the advances in multilingual NLP however significant they are, are concentrated on higher-resource languages. Kannada language, despite being spoken by over 40 million people and holding classical language status, still remains underexplored in NLP research. Retrieval models which are trained on English corpora struggle with Indian languages, and the evaluation infrastructure such as datasets, benchmarks and annotated corpora is sparse for such languages.

Hallucination is the other challenge. Consider a situation where a citizen queries about their eligibility for Ayushman Bharat health coverage and the model give inaccurate answers which will affect the users. So this is not a failure mode we can treat as acceptable in exchange for the coverage or fluency which these models offer but a systems operating in high-stakes advisory contexts and government scheme QA should precisely convey what they know and what they don't no and say it clearly in their responses.

Existing work on Indian languages for government services have made good progress, few of them are Jugalbandi [14] which provides government information access in Hindi and several other languages, Kisan e-Mitra targets agricultural queries, and BHASHINI provides a broader multilingual infrastructure layer. However, none of these systems combines Kannada document grounding with principled hallucination detection. Most of them, either operate without retrieval verification or lack the unanswerable detection capability which is

necessary to avoid fabricating answers for out-of-scope queries.

We address these gaps with SchemeQA-Kannada. The system is built around 4 technical contributions:

1. **Dataset** - It consists of 315 manually verified question-answer pairs for 4 schemes (PM-KISAN, Ayushman Bharat, PMJDY, and Sukanya Samriddhi Yojana), including 63 unanswerable questions. To our knowledge, this is the first such dataset for Kannada government QA.

2. **Section-Boosted MuRIL Retrieval** - A FAISS-based dense retrieval pipeline using MuRIL embeddings [4], augmented with section-aware score boosting that preferentially surfaces eligibility, benefit, and procedural content. Retrieval is further strengthened by a three-stage scheme detection mechanism which combines exact alias matching, fuzzy string similarity (SequenceMatcher ratio ≥ 0.75), and MuRIL semantic similarity (cosine ≥ 0.55) against pre-encoded scheme alias clusters that enables scheme-aware chunk filtering and mismatch-based rerouting to web search when retrieved chunks do not correspond to the scheme identified in the query.

3. **Cross-Lingual Hallucination Detection** - a translate-then-verify pipeline that translates generated Kannada answers into English before running NLI entailment verification against retrieved source chunks using mDeBERTa-v3-base-xnli. This cross-lingual framing is necessitated by the limited availability of Kannada-native NLI models, and as our ablation results show it is the single most impactful component in the system.

4. **Source-Aware Confidence Score (SACS)** - a composite confidence metric that jointly weights NLI entailment score, source credibility tier (official PDF, .gov.in website, educational source, news, or open web), and generation confidence. SACS provides an interpretable trust level that the system surfaces to users in Kannada as ಹೆಚ್ಚು (High), ಮಧ್ಯಮ (Medium), or ಕಡಿಮೆ (Low).

Evaluation on a test set of 48 pairs, shows that our system achieves a hallucination rate of 16.1% compared to 31.6%–100% across all baselines with a BERTScore of 0.7468 and correct unanswerable detection on 100% of out-of-scope queries in the full deployment configuration. The evaluation indicates that for government scheme advisory, hallucination reduction and source credibility matter more than the surface-level token overlap metrics.

The remainder of this paper is organised as follows. Section 2 reviews related work in multilingual QA, RAG-based systems, and Indic NLP. Section 3 describes the SchemeQA-Kannada dataset. Section 4 presents the system architecture in detail, section 5 outlines the experimental setup and baselines, section 6 reports and analyses the results, and Section 7 discusses limitations and directions for future work.

2. Related Work

Multilingual and Cross-Lingual Question Answering. The trajectory of multilingual NLP over the past several years has been shaped largely by the emergence of large pretrained transformer models. BERT [1] established the pretrained model, and its multilingual variant mBERT extended it to 104 languages. Though its widely documented, but its support for low-resource languages has to be improved. XLM-R [2] provides a cross-lingual pretrained model, and benchmarks like MLQA [11] and TyDiQA-GoldP [9] provides structural evaluation frameworks for multilingual extractive QA. Most of the multilingual QA research assume the problem as addressing the lower-resource languages, but we need to consider the morphological complexity of the agglutinative languages like Kannada, and its still underexplored.

Indic Language NLP and MuRIL. IndicBERT and subsequent models in the Indic family [5] offers pretrained representations across 12 Indian languages, including Kannada, and have been applied to the tasks ranging from named entity recognition to sentiment analysis.

MuRIL [4] represents the most directly relevant prior work for our retrieval component trained specifically on 17 Indian languages using both monolingual and parallel corpora, it substantially outperforms mBERT on Indic language tasks, particularly for morphologically rich languages. Our system uses MuRIL as the backbone for dense retrieval, and the section-boosted scoring mechanism we introduce builds directly on its embedding quality. Kannada-specific NLP work remains relatively sparse in the literature, most existing efforts address morphological analysis, POS tagging, or machine translation, rather than end-to-end QA or information retrieval.

Retrieval-Augmented Generation. The RAG framework [3] has become the dominant paradigm for knowledge-intensive NLP tasks, combining dense retrieval with generative language models to produce grounded, and verifiable responses. Subsequent work has refined various components passage re-ranking, query reformulation, hybrid sparse-dense retrieval and RAG systems have been applied across domains including biomedical QA, legal document analysis, and customer service. For low-resource settings, however, RAG introduces a specific difficulty: retrieval quality degrades sharply when embedding models are not well-adapted to the target language, and poor retrieval cascades into poor generation. Our work addresses this directly through MuRIL-based retrieval and section-aware boosting, both of which are motivated by observed failure modes in naive RAG configurations for Kannada text.

Hallucination Detection in LLMs. Hallucination is the generation of fluent but factually unsupported content has emerged as one of the most actively studied failure modes of large language models [7]. Approaches to mitigation broadly fall into three categories: retrieval grounding (ensuring the model has access to relevant source material), post-hoc verification (checking generated outputs against reference sources), and training-time interventions. NLI-based verification has shown particular promise for post-hoc detection, where entailment models assess whether a generated

claim is supported by retrieved context. Our cross-lingual hallucination detection pipeline falls in this category, though it adds a translate-then-verify step necessitated by the absence of a reliable Kannada-native NLI model - a practical constraint that, one suspects, applies to many low-resource deployment scenarios and has received insufficient attention in the literature.

Government Scheme QA and Civic AI Systems. A small but growing body of work has addressed government information access through conversational AI. Jugalbandi[14], developed by AI4Bharat [6] in collaboration with partners, supports welfare scheme queries in Hindi and several other Indian languages using a RAG-based architecture, and has been deployed at meaningful scale. Kisan e-Mitra targets agricultural information specifically, with a focus on farmer-facing queries about PM-KISAN and related programs. BHASHINI provides a broader translation and speech infrastructure layer that provides the foundation for several government-facing applications. These applications represent important practical contributions and most of these systems focus on coverage.

3. Dataset Construction

There is no existing Kannada QA dataset which covers government schemes. The existing dataset are general-purpose Indic QA benchmarks which are either translated from English sources or focuses on encyclopedia content rather than procedural, eligibility-oriented queries which the citizens usually ask. Dataset construction is the primary contribution, and the dataset is built from scratch with manual verification across all QA pairs.

3.1 Scheme Selection

The four schemes included in this dataset are Pradhan Mantri Kisan Samman Nidhi (PM-KISAN), Ayushman Bharat PM Jan Arogya Yojana (PM-JAY), Pradhan Mantri Jan Dhan Yojana (PMJDY), and Sukanya Samriddhi Yojana (SSY) which were selected on the basis of few criteria's such as, population reach, and the diversity of query type. Altogether, these schemes

span agricultural income support, health insurance, financial inclusion, and girl child savings domains that together address the needs of a significant proportion of India's rural and semi-urban population, and most of the users here, would like to seek information in Kannada rather than English or Hindi. Official scheme PDFs were sourced directly from government portals[15]-[18], ensuring that the retrieval corpus reflects the actual documentation a citizen-facing system would reasonably be expected to use.

3.2 Question Construction and Annotation

QA pair generation followed a two-stage pipeline combining automated candidate generation with manual human verification. In the first stage, each chunk in the document corpus comprising a scheme name, section label, and up to 600 characters of passage text, was passed to Large Language Model (LLM) as a generation prompt. For answerable questions, the model was asked to generate extractive QA pairs, where the answers are explicitly present in the passage. They considered six types of questions (eligibility, benefits, process, documents, amount, and contact), and three levels of difficulty (easy, medium and hard). The questions had to be written in Kannada script and reflected what a real citizen interacting with a scheme for the first time would want to know. For unanswerable questions, LLM was supplied with the same passage with explicit instructions to generate plausible questions with answers that are truly unanswerable from the passage, e.g. no mention of benefits, contact details, procedures, or other schemes that are related but distinct. The passage-relative construction suggests that the unanswerable questions are semantically grounded rather than arbitrarily out-of-scope.

In the second stage, each generated pair was manually verified by a human annotator against the source passage next to it. QA pairs for which the generated answer was partially correct but incorrect due to lack of detail, imprecision, or inconsistency with the passage text were edited before inclusion. During this process a human-edited flag was used to record which pairs were

corrected. Malformed pairs are the pairs with invalid question or answer fields, or pairs that could not be verified against the source passage were discarded. The final dataset of 315 pairs represents only those pairs that passed this source-grounded verification step. Verification was performed by a single annotator.

3.3 Unanswerable Questions

Including the 63 unanswerable questions in the dataset was a deliberate design choice motivated by a concern that appears frequently in deployment discussions but less often in academic evaluation, that is, a system that always attempts an answer is, in many ways, more dangerous than the one that occasionally refuses. SQuAD 2.0 [10] also has included unanswerable questions to test QA systems. For government scheme QA, the failure mode of confident generation of plausible-sounding answer, for example, incorrect eligibility condition, is arguably worse than a refusal. Unanswerable questions were constructed to be semantically plausible and grammatically well-formed, they contain about things a reasonable person might genuinely want to know, but for which the answer is not present in the official scheme documents. This includes questions about schemes not covered in the corpus, queries about edge cases which are not addressed in official documentation, and questions that combine details across schemes in ways that have no correct answer.

3.4 Dataset Statistics

The full dataset comprises 315 manually verified QA pairs, which are split into a development set of 47 pairs used for hyperparameter tuning and a test set of 48 pairs used for all reported evaluations. Due to rate limits on the free tier of the Groq API which is used for generation and NLI verification, the evaluation was restricted to a small subset of the full dataset. Table 1 summarizes the distribution across schemes, Table 2 shows the dataset distribution and Figure 1 shows the dataset distribution across schemes and question types.

Table 1. Dataset Distribution by Scheme

Scheme	Total QA Pairs	Answerable	Unanswerable
Ayushman Bharat (PM-JAY)	87	73	14
PM-KISAN	77	65	12
PMJDY	78	59	19
Sukanya Samriddhi Yojana	73	55	18
Total	315	252	63

Table 2. Dataset Distribution by Question Type

Question Type	Count	Percentage
Scheme Overview	54	17.1%
Eligibility	46	14.6%
Application Process	45	14.3%
Numerical / Factual	36	11.4%
Required Documents	34	10.8%
Benefits	27	8.6%
Contact	10	3.2%
Unanswerable	63	20.0%

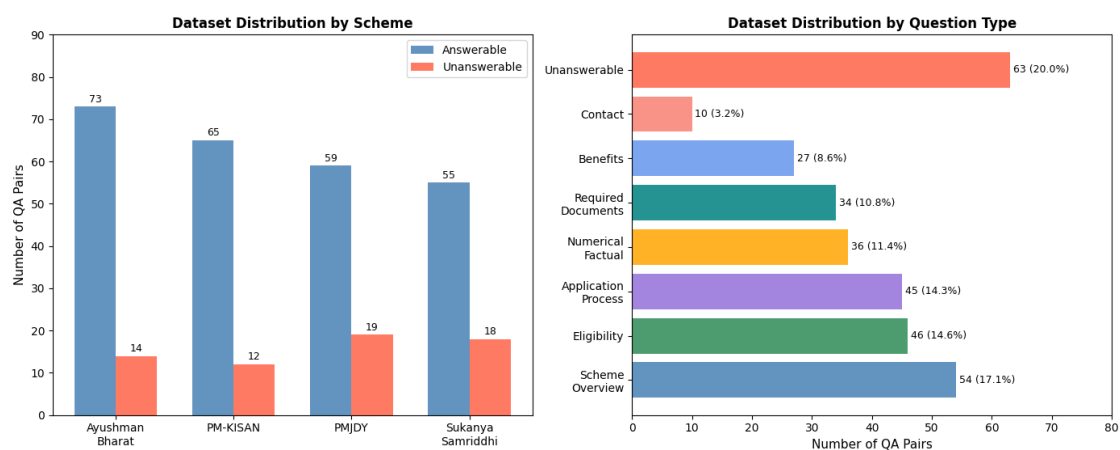
SchemeQA-Kannada Dataset Statistics (315 QA Pairs)

Figure 1. SchemeQA-Kannada dataset distribution across schemes (left) and question types (right), the unanswerable fraction (20%) is distributed across all schemes to reflect realistic deployment conditions.

The distribution across the schemes is roughly proportional to the volume of official documentation available for each, with Ayushman Bharat's having comparatively better

documentation covering both the broader PM-JAY framework and state-level implementation guidelines yielding the largest subset. The unanswerable fraction was set to reflect a realistic

proportion of out-of-scope queries one might expect in deployment, rather than a synthetically balanced split.

3.5 Limitations of the Dataset

The dataset covers only four central government schemes; There are dozens of additional schemes that are entirely absent here. Question construction, while informed by real informational needs, was ultimately researcher-driven rather than derived from logs of actual citizen queries a limitation shared by most domain-specific QA datasets, but one that may affect how well the distribution generalizes. Average answer lengths are relatively short, which means token F1 is sensitive to verbosity

mismatches between gold and generated answers may underestimate system quality; this is part of why we treat BERTScore as the primary quality metric in our evaluation.

4. System Architecture

SchemeQA-Kannada is a multi-stage pipeline that routes an incoming query through a sequence of specialized components intent classification, retrieval or web search, answer generation, hallucination verification, and confidence scoring each of which addresses a specific failure mode observed in simpler configurations. Figure 2 provides an overview of the full pipeline. We describe each component in turn.

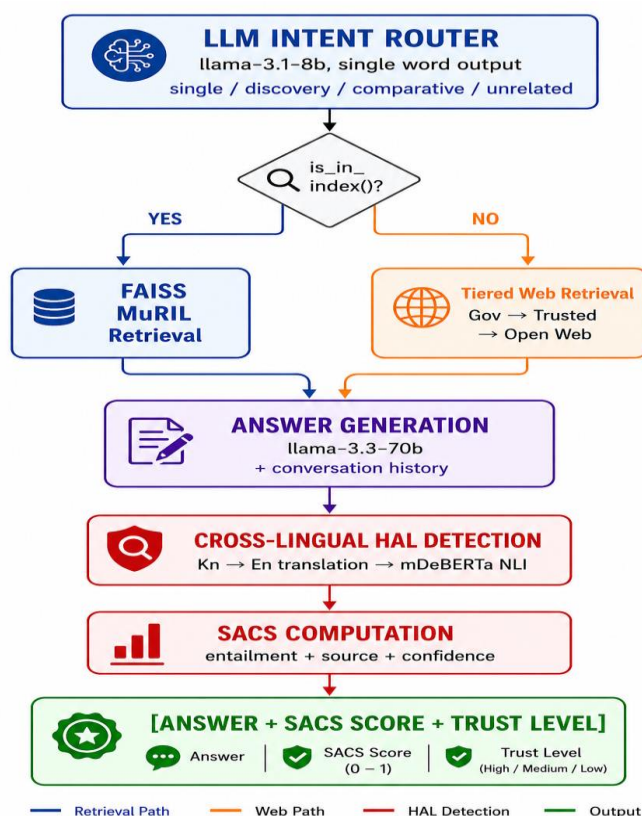


Figure 2. System Architecture

4.1 LLM-Based Intent Routing

The first decision the system makes upon receiving a query is not retrieval, it is routing. Not all questions about government schemes are answerable from the indexed documents, and treating every query as a retrieval problem leads predictably to either hallucination (the model confabulates an answer from weak retrieval signal)

or false refusal (the retrieval fails and the system incorrectly declines). To avoid both, we introduce an LLM-based intent classifier as the entry point to the pipeline.

The classifier uses Llama-3.1-8B Instant model accessed via the Groq API, configured with a maximum of 10 output tokens and temperature set to zero essentially forcing a categorical output

from a small, fast model rather than invoking the heavier generation model unnecessarily. Each query is classified into one of four intent categories: single scheme where the query pertains to one specific known scheme, comparative - the user is comparing two or more schemes, discovery - the user does not know which scheme applies to their situation, or unrelated - the query has no connection to the covered schemes. Alongside intent classification, a keyword-based index check matching against a

$$(q) = \begin{cases} \text{FAISS}(q, k = 5) & \text{if intent}(q) = \text{single_scheme} \wedge q \in \mathcal{I} \\ \text{FAISS}(q, k = 10) & \text{if intent}(q) \in \{\text{comparative}, \text{discovery}\} \wedge q \in \mathcal{I} \\ \text{WebSearch}(q) & \text{if intent}(q) = \text{unrelated} \vee q \notin \mathcal{I} \end{cases}$$

where $\text{intent}(q)$ is the intent category assigned by the LLM classifier, $\mathcal{I}$ denotes the set of schemes present in the FAISS index, and k is the number of retrieved chunks.

Queries involving a known scheme with single-scheme intent are routed to FAISS retrieval with $\text{top-k}=5$. Comparative and discovery queries - which typically require drawing on multiple scheme documents - are routed to a broader FAISS search across all indexed documents with $\text{top-k}=10$. Queries mentioning no known scheme, or classified as unrelated, bypass FAISS entirely and proceed directly to tiered web search. This design avoids the common failure mode of attempting dense retrieval over a corpus that simply does not contain the relevant information.

A persistent failure mode in retrieval over government scheme corpora arises from what we term MuRIL score saturation, because documents across schemes share substantial domain-specific vocabulary terms like *yojane(scheme)*, *labhartiaru(beneficiaries)*, and scheme-generic benefit descriptors. MuRIL embeddings for semantically distinct chunks tend to cluster tightly, with cosine similarities frequently exceeding 0.99, causing FAISS retrieval to rank scheme-incorrect chunks at the top and the LLM to generate coherent but wrong answers that can pass NLI verification due to superficial lexical overlap. To address this, we introduce a scheme mismatch guard that

curated list of known scheme names in both Kannada and English determines whether the queried scheme is present in the FAISS index.

The routing logic that follows from this dual signal is mentioned below, and it has meaningful downstream effects on system behaviour. The routing decision for an incoming query q is defined as:

R

intercepts between retrieval and generation: after `retrieve()` returns the top-k chunks, `detect_scheme_from_query()` independently resolves the query's intended scheme through a three-stage pipeline. Stage 1 performs exact alias matching against a curated SCHEME_ALIASES dictionary covering Kannada script variants, English names, abbreviations, and misspellings. Stage 2 applies fuzzy matching via `difflib.SequenceMatcher` over the query unigrams and bigrams at a ratio ≥ 0.75 to handle transliteration inconsistencies. Stage 3 encodes alias clusters with MuRIL and computes cosine similarity at a threshold of 0.55 to resolve paraphrased or purpose-based references, such as "girl child scheme" correctly mapping to Sukanya Samriddhi programme, despite containing no recognizable form of the scheme name. If the retrieved chunk's metadata doesn't match the detected scheme, the pipeline routes directly to the tiered web search, bypassing the answer generation that a confidently wrong answer is more harmful than a transparent fallback.

4.2 Section-Boosted MuRIL Retrieval

For queries routed through the document path, retrieval is performed using MuRIL embeddings [4] indexed with FAISS IndexFlatIP, which computes cosine similarity via inner product over L2-normalised vectors. MuRIL was chosen over mBERT and other multilingual alternatives specifically because of its training on Indic

language corpora as the baseline comparisons in Section 6 make clear, mBERT's generic multilingual representations produce substantially higher false refusal rates for Kannada text, likely due to poor coverage of Kannada's morphological variation.

The FAISS index spans 151 chunks extracted from the four scheme PDFs, with chunk boundaries determined by section-aware parsing rather than fixed token windows. This is a deliberate choice: government scheme documents have a reasonably consistent internal structure eligibility criteria appear in one section, benefit amounts in another, application procedures in a third and chunking that respects this structure produces more coherent retrieved passages than naive sliding-window approaches.

When a specific scheme is detected through the three-stage pipeline, retrieval applies scheme-aware filtering: the system fetches top- $k \times 3$ candidates from the FAISS index and retains only those whose scheme metadata field matches the detected scheme before returning the final top- k . If fewer than two matching chunks pass filtering which occurs for narrow queries about under-documented scheme aspects the filter is relaxed and the unfiltered top- k set is used as fallback. This approach is necessary because MuRIL's within-domain saturation makes raw similarity scores unreliable across the full four-scheme corpus, without filtering, section-boosted scoring operates over a mixed candidate pool where scheme-irrelevant chunks compete on near-identical similarity values, producing misleading rank distributions rather than meaningful scheme-relevant ordering.

A distinguishing aspect of the retrieval pipeline is the section-aware score boosting, where raw retrieval scores are adjusted based on the originating chunk's section. The final retrieval score for each candidate chunk is computed as:

$$f(c) = s(q, c) + \beta \cdot 1[\text{sec}(c) \in \mathcal{S}]$$

where $s(q, c)$ is the MuRIL cosine similarity between query q and chunk c , $\beta = 0.15$ is the section boost value determined through hyperparameter tuning, $\text{sec}(c)$ is the section label

of chunk c , $\mathcal{S} = \{\text{eligibility, benefits, overview, process, implementation}\}$ is the set of boosted sections, and $1[\cdot]$ is the indicator function returning 1 if the condition holds and 0 otherwise. Section boost = 0.15 for chunks which are from eligibility, benefits, overview, process, and implementation sections, and 0 otherwise. The intuition is pretty straightforward, that citizens asking about these government schemes are more likely to ask about these sections, and a small but consistent boost will help surface relevant chunks which might otherwise be ranked lower than the less directly related content from the less critical sections. The boost value of 0.15 was determined by hyperparameter tuning on the 47-pair development set across 20 configurations, using a combined optimization objective of

$$\text{Score} = 0.6 \times \text{F1} + 0.2 \times \text{Correct Refusal Rate} + 0.2 \times (1 - \text{Hallucination Rate})$$

This weighting explicitly penalizes hallucination alongside standard QA quality metrics as a reflection of our broader design objective as verified responses matter more than marginal fluency gains.

4.3 Tiered Web Search

Tier 1 queries official government domains (gov.in, nic.in, pmkisan.gov.in, nha.gov.in, pmjdy.gov.in), Tier 2 covers government-adjacent portals (myscheme.gov.in, jansamarth.in, vikaspedia.in) combined with a broad gov.in search and Tier 3 falls back to the open web when higher tiers return no helpful response. The system halts at the first tier that yields a valid answer, keeping source credibility as high as the query permits.

Retrieved page content is cleaned using BeautifulSoup4 before chunking. Elements such as script, style, noscript, and head are removed with their full subtrees prior to text extraction via `get_text()`. The prior implementation used regex-based tag stripping, which left JavaScript function bodies and CSS rules in the extracted text, these incidentally matched query terms and received inflated CrossEncoder scores, causing incorrect chunk selection. After cleaning, chunks are

scored using a cross-encoder (ms-marco-MiniLM-L-6-v2) rather than positional selection. Since this model is English-only, the Kannada query is translated before scoring, reusing the translation which is already produced for Tavily search and incurring no additional API cost. Fetched URLs are deduplicated by MD5 hash of page content before processing, as several government portals - including nsiindia.gov.in return identical content across different URL parameters without deduplication, the CrossEncoder runs redundantly over identical chunks. Domains that consistently fail via urllib timeout, HTTP 403, or Selenium returning fewer than 200 characters are added to a runtime blacklist after two failures and skipped for the remainder of the session, eliminating repeated 120-second Selenium timeouts that otherwise cause multi-minute delays on discovery queries.

Page fetching uses a four-layer fallback per URL, Layer 1 checks a Drive-backed government scrape cache for a network-free response, Layer 2 attempts urllib with BeautifulSoup4 for static HTML and PDF content, Layer 3 invokes Selenium for JavaScript-rendered pages, only when the urllib response fails a minimum quality check, and Layer 4 returns None, causing the upstream handler to fall back to Tavily snippets directly. Each tier uses Tavily's advanced depth setting with a maximum of three results.

4.4 Answer Generation

Answer generation is handled by llama-3.3-70b versatile (Groq), with llama-3.1-8b-instant serving as a fallback in the event of API failure or timeout. Temperature is set to 0.1 during evaluation — favouring determinism and faithfulness over creative variation — and relaxed slightly to 0.2 in demo mode. The prompting strategy differs meaningfully between the document and web paths. Document-path prompts explicitly ground the model in the retrieved chunks, instructing it to answer only from the provided context and to decline if the answer is not present. Web-path prompts are constructed without reference to "our documents," since the source is external and the appropriate framing is different — the model is

told to synthesise from the provided web snippets rather than a curated corpus.

Conversation history is injected into the generation prompt for multi-turn interactions: the last six turns are maintained in memory, and the system detects follow-up queries through a combination of Kannada trigger words (ಅದರ, ಅದಕ್ಕೆ, ಹೇಗೆ, ಎಷ್ಟು and others), English continuation phrases (what about, tell me more, explain more), and a heuristic that flags short queries — fewer than six words — that mention no scheme name when conversation memory already holds a scheme context. Detected follow-ups have the relevant scheme context appended to the query before retrieval, which substantially reduces the frequency of context-loss errors in multi-turn sessions.

We have set the generation temperature to 0.1 for all the experiments to balance the response stability with answer generation while maintaining reevaluation conditions. Llama-3.3-70B-Versatile [19] is a 70-billion parameter instruction-tuned model accessed via the Groq API, it was selected for its strong multilingual instruction-following capability and its low-latency inference profile, both of which are relevant to the multi-step pipeline architecture described here. Generation is performed via API call with a maximum output length of 512 tokens.

4.5 Cross-Lingual Hallucination Detection

This is, arguably, the most technically distinctive component of the system, and the one whose contribution is most clearly supported by the ablation results. The core problem is this: reliable NLI models for Kannada do not currently exist at the scale required for production use. One could wait for them to be developed, or one could work around the gap — we chose the latter, via a translate-then-verify pipeline.

For document-grounded answers, the generated Kannada response is first translated to English using llama-3.1-8b instant. The translated answer is then passed to mDeBERTa-v3-base-xnli[8], a cross-lingual NLI model with strong multilingual coverage which checks entailment against the retrieved source chunks. Rather than checking

against a single chunk, the system uses a multi-window strategy: the top three retrieved chunks are each checked independently, and the maximum entailment score across all three is taken as the final signal. This reduces false positives that arise when a correct answer is grounded in a chunk that narrowly missed the top retrieval position.

NLI entailment verification is computed against scheme-matched chunks only, those whose metadata field corresponds to the scheme detected from the query rather than against the top-3 retrieved chunks irrespective of scheme membership. This distinction matters because wrong-scheme chunks sharing semantically adjacent vocabulary can satisfy entailment conditions superficially. In one recurring failure case, Jan Dhan chunks referencing guardian and account terminology passed NLI verification for a Sukanya Samriddhi answer, producing a low hallucination rating on a factually incorrect response, the entailment model had no mechanism to distinguish adjacency from genuine factual support. Therefore, restricting verification to scheme-matched chunks eliminates this class of false negatives. When no scheme-matched chunks exist in the retrieved set, which occurs when the scheme mismatch guard has not rerouted the query, the top-3 retrieved chunks are used as fallback to preserve verification coverage rather than leaving the answer unverified.

The entailment threshold is set at 0.25 for most question types, answers with a maximum entailment score below this value are flagged as high hallucination risk. For generative question types, where answers synthesise across multiple document sections rather than extracting a single fact, the threshold is relaxed to 0.20 to account for the inherently lower per-chunk entailment scores that synthesis produces. Answers flagged as high risk are either refused (in evaluation mode) or accompanied by a warning and SACS downgrade (in demo mode).

The web-path answers undergo a little different verification process. Three sequential checks are applied to it. The first check is a domain related

check which confirms whether the answer generated contains scheme related keywords or not, the second is a NLI entailment performed on the retrieved web snippets, and the last one is source credibility assessment which applies a stricter entailment threshold (0.20) for the answers which are generated from .gov.in domains because misrepresenting official government content will have a higher consequence.

4.6 Source-Aware Confidence Score (SACS)

The standard metrics related to confidence for QA systems mostly rely on model's internal signals such as token probabilities, consistency scores, or the computed SoftMax outputs. These are the common measures which are used, but they have a significant limitation that is they treat all sources as equivalent. For an advisory system, this is not an acceptable behaviour. An answer generated from an official scheme PDF should be evaluated with different trust levels than the answer generated from a news article or an unknown webpage, regardless of how confidently the language model generates it. The Source-Aware Confidence Score (SACS) is computed as:

$$\text{SACS}(a, s, q) = 0.5 \cdot w_e + 0.3 \cdot w_s + 0.2 \cdot w_c$$

where:

- $w_e \in [0,1]$ is the entailment weight, derived from the maximum NLI entailment score across the top-3 retrieved scheme-matched chunks
- $w_s \in \{0.95, 0.85, 0.65, 0.45, 0.35\}$ is the source credibility weight assigned by tier: official scheme PDF (0.95), government website (0.85), educational source (0.65), news source (0.45), and open web (0.35)
- $w_c \in [0,1]$ is the model's confidence.
- a represents generated answer, s represents source document, and q represents input query.

The SACS score is related to three trust levels shown to the user: High ($\text{SACS} > 0.75$), Medium ($0.50 \leq \text{SACS} \leq 0.75$), and Low ($\text{SACS} < 0.50$).

The weights in SACS score that is, 50% of entailment, 30% of source, and 20% of confidence, favors evidence-based grounding over the source for the generated answers. Here, the idea is that a highly entailed answer generated from a fairly reliable source should outrank a weakly entailed answer generated from an official document, which conveys that verification evidence should dominate.

4.7 Mode-Aware Unanswerable Handling

To handle unanswerable queries, the system processes in 2 distinct ways, for evaluation and deployment purpose. In evaluation mode, for the queries where no proper answer can be retrieved will result in a clear refusal. This is important to have a fair comparison against the baselines as a system which falls back to do a web search for every unanswerable query will make its metrics appear to be higher at the cost of hallucination and making like-to-like comparison impossible.

In demo mode, the same queries trigger a tiered web search fallback, with the returned answer accompanied by a source credibility warning and a correspondingly lower SACS score. This design acknowledges that a real tension in deployment: a system that simply refuses a citizen's question without any fallback is frustrating and potentially unhelpful, but a system that silently substitutes an unverified web answer for a document-grounded one is potentially dangerous. Making the fallback explicit that is surfacing the source type and downgraded confidence score, attempts to thread this needle, though we acknowledge it places some burden of interpretation on the user.

5. Experimental Setup

5.1 Baselines

We compare SchemeQA-Kannada against five baseline configurations, selected to isolate the contribution of individual system components and to situate our approach within the broader landscape of retrieval and generation strategies applicable to this task.

Zero-Shot LLM. The simplest possible baseline: llama-3.3-70b-versatile is prompted directly with the user query, with no retrieved context, no

document grounding, and no hallucination verification. This configuration represents what a practitioner might deploy with minimal engineering effort, and its results, 100% hallucination rate, 0% unanswerable detection establish a useful lower bound that motivates the rest of the system design. It is worth being explicit about what 100% hallucination rate means here: the model generates a response to every query, including those that have no valid answer in the document corpus, and none of those responses are verifiably grounded. Fluent, confident, and almost entirely unverified.

BM25 + Reader. A sparse retrieval baseline using BM25 [13] for chunk ranking, with the same LLM reader used in our system for generation. BM25 is a strong baseline for English-language retrieval tasks, but its term-matching approach is poorly suited to Kannada's agglutinative morphology, a single Kannada root can surface in dozens of inflected forms, and BM25 has no mechanism to treat these as semantically equivalent. The reader component is held constant across this and subsequent baselines to ensure that retrieval quality, rather than generation quality, is what differentiates the systems.

Naive RAG (MuRIL). Dense retrieval using MuRIL embeddings with FAISS, identical to our system's retrieval component, but without section-aware score boosting, without NLI-based hallucination detection, and without the intent routing layer. This baseline is particularly important for interpreting the ablation results: it uses the same retrieval backbone as our system, so performance differences isolate the contribution of the verification and routing components.

mBERT RAG. Dense retrieval using mBERT embeddings in place of MuRIL, with the same FAISS index structure and LLM reader. This baseline tests whether MuRIL's Indic-language specialization translates into meaningful retrieval quality gains over a generic multilingual model. The results, as discussed in Section 6, suggest it does and fairly dramatically so in terms of false refusal rate.

Naive RAG + Web. The Naive RAG configuration extended with tiered web search fallback, but again without NLI verification. This baseline is important because it controls for web access: comparing it against our full doc+web configuration isolates the contribution of hallucination detection independently of whether the system has access to external sources.

All baselines were evaluated on the same 48-pair test split under identical conditions. Baselines that lack an unanswerable detection mechanism which is all of them, generate responses for every query regardless of answerability, which is reflected in their 0% unanswerable detection rates. The nominally high refusal rates reported for BM25 and Naive RAG (100% and 100% respectively) are, as noted in Section 3.3, artifacts of retrieval failure rather than genuine abstention behavior; the correspondingly high false refusal rates (50% and 39.5%) confirm this interpretation.

5.2 Evaluation Metrics

Selecting appropriate metrics for a Kannada QA system requires some care. Standard extractive QA metrics such as exact match and token F1, were designed with English in mind, and several properties of Kannada text make them unreliable to be considered as primary quality indicators. Numeral representation is one issue where the gold answer may use Kannada numerals (೨೦೧೪) while the system generates numerals (2014), producing a token mismatch despite semantic equivalence. More broadly, LLM-generated answers tend to be more verbose than concise like human-written gold answers, which systematically decreases token F1 even when the generated response is factually correct and informative. We report token F1 for completeness and comparability with prior work, but treat BERTScore as the primary quality metric.

The full metric suite is as follows:

- **Token F1** - Defines word-level overlap between predicted and gold answer. It underrepresents quality for generative systems

that produce full sentences rather than extractive spans.

- **BERTScore**[12]- Semantic similarity between predicted and gold answer using multilingual contextual embeddings. Its considered as primary quality metric for our system. It does not penalize factual incorrectness, and hallucinated answers semantically close to gold can still score highly. False refusals are penalized as 0.0 and true refusals excluded.
- **Hallucination Rate (HAL%)** - Proportion of answered questions where the prediction is not entailed by retrieved source documents. Short factoid answers (length < 60, yes/no, numeric) bypass NLI and are assigned LOW risk while remaining answers are translated to English and checked via mDeBERTa-v3 sentence-split entailment. Its applied identically to all systems. Zero-shot LLM's value is considered as 100% by definition as it performs no retrieval.
- **False Refusal Rate (FRef%)** - Proportion of answerable questions where the system incorrectly refuses to answer. Reflects overly conservative retrieval.
- **Correct Refusal Rate (Unans%)** - Proportion of unanswerable questions correctly refused. Its critical for government QA where incorrect answers are more harmful than refusals.
- **Source-Aware Confidence Score (SACS)** - Reported for the proposed system configurations only, as baselines lack the source credibility infrastructure required to compute it meaningfully.

5.3 Hyperparameter Tuning

The hyperparameter optimisation objective is defined as:

$$\mathcal{L}(\theta) = 0.6 \cdot F_1(\theta) + 0.2 \cdot R(\theta) + 0.2 \cdot (1 - H(\theta))$$

where:

- $\theta = \{\beta, \tau\}$ is the set of tunable hyperparameters, comprising the section boost value β and the NLI entailment threshold τ

- $F_1(\theta)$ is the token F1 score on the development set
- $R(\theta)$ is the correct refusal rate on unanswerable questions
- $H(\theta)$ is the hallucination rate, so $(1 - H(\theta))$ rewards lower hallucination
- The weights penalize hallucinations as per the design priority as verified responses matter more than the fluency gains

This objective was evaluated on the 47-pair development set, yielding optimal values of $\beta = 0.15$ and $\tau = 0.25$.

5.4 Implementation Details

All experiments were conducted on Google Colab using a T4 GPU. The FAISS index contains 151 chunks across the four scheme PDFs, with chunk boundaries determined by section-aware parsing, and MuRIL embeddings are 768-dimensional.

Answer generation uses Llama-3.3-70B Versatile via the Groq API at temperature 0.1; the lighter fallback model Llama-3.1-8B Instant is invoked only on API failure. Generation is capped at 512 output tokens. NLI verification uses mDeBERTa-v3-base-xnli, with Kannada-to-English translation handled by Llama-3.1-8B Instant prior to verification. The same translated query is reused for CrossEncoder scoring in the web path, avoiding additional API calls. Web chunk selection uses ms-marco-MiniLM-L-6-v2 as a cross-encoder rather than positional selection. HTML content from fetched pages is cleaned via BeautifulSoup4, script, style, noscript, and head elements are removed using `decompose()` before text extraction with `get_text()` replacing prior regex-based stripping that allowed

JavaScript and CSS content to enter the chunking pipeline. Fetched URLs are deduplicated by MD5 content hash before CrossEncoder scoring, and domains exceeding two consecutive fetch failures are added to a runtime blocklist for the remainder of the session.

Scheme detection follows the three-stage pipeline described in System Architecture: exact alias matching against SCHEME_ALIASES, fuzzy matching via difflib.SequenceMatcher at threshold ≥ 0.75 , and MuRIL semantic similarity against pre-encoded alias clusters at threshold ≥ 0.55 . NLI entailment is computed against scheme-matched chunks only, with the top-3 retrieved chunks used as fallback when no scheme-matched chunks exist. Discovery-intent queries do not update the last_scheme field in conversation memory. Web search uses the Tavily API at advanced depth with a maximum of three results per tier. BERTScore evaluation uses bert-base-multilingual-cased with language set to Kannada. The demo configuration additionally incorporates OpenAI Whisper (medium)[20] for Kannada speech recognition and gTTS for text-to-speech output, though neither component is included in the reported experiments.

6. Results and Analysis

6.1 Main Results

Table 3 presents the full evaluation results across all system configurations and baselines. We discuss the headline findings first, then turn to the component-level analysis in subsequent subsections. Figure 3 illustrates a response generated using information which is retrieved from an external government website. Figure 4 illustrates the performance comparison across all system configurations and baselines.

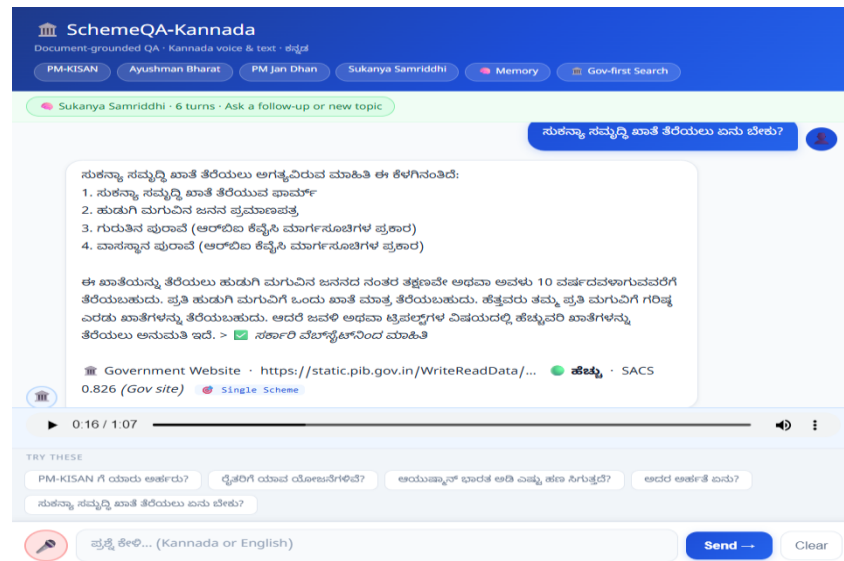


Figure 3. Sample system output from the SchemeQA-Kannada demo interface.

Table 3. Evaluation Results for SchemeQA-Kannada Test Set (48 pairs)

System	F1	BERT	HAL%	FRef%	Unans%	SACS
Zero-Shot LLM	0.2977	0.6830	100.0	0.0	0.0	—
BM25 + Reader	0.3187	0.3971	31.6	50.0	100.0	—
mBERT RAG	0.2222	0.2859	33.3	60.5	90.0	—
Naive RAG (MuRIL)	0.3200	0.4625	34.8	39.5	100.0	—
Naive RAG + Web	0.3958	0.7024	41.7	5.3	50.0	—
Ours (doc only)	0.4270	0.7468	16.1	13.9	100.0	0.745
Ours (doc + web)	0.4195	0.7453	18.2	13.2	100.0	0.735

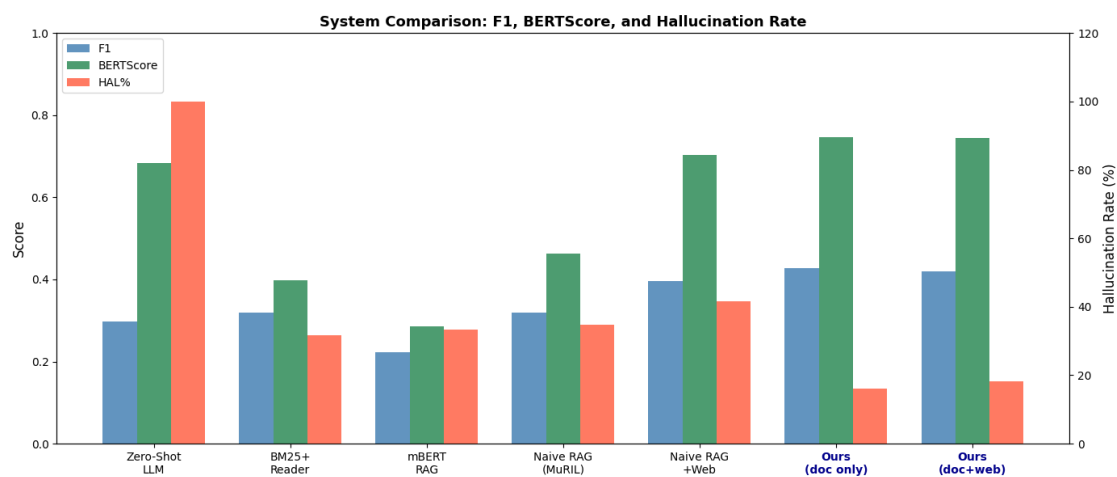


Figure 4. Represents a comparison of all systems across F1, BERTScore, and Hallucination Rate on the SchemeQA-Kannada test set (48 pairs). Our configurations achieve the lowest hallucination rate while maintaining the highest BERTScore.

Perhaps the most consequential finding concerns is about the hallucination. The document-only configuration achieves a hallucination rate of 16.1%, compared to 41.7% for the strongest baseline (Naive RAG + Web) and 100% for the Zero-Shot LLM condition. These are not marginal gains attributed to the better retrieval or a stronger language model, but they reflect a structurally different approach to answer generation. This equally represents the behavior of the baselines as a group, that is, all configurations lacking explicit NLI-based verification cluster between 31.6% and 100% hallucination rate, irrespective of whether retrieval was present. This consistency makes it difficult to attribute the difference to retrieval quality alone. Even when well retrieved context is given to the language model without verification, it appears to be insufficient to prevent the model from generating hallucinated responses. In civic information, where a user's knowledge about their rights and entitlements may depend on the accuracy of the given response, so in such cases, failure is not something which is tolerable here.

The document-only configuration achieves a BERTScore of 0.7468 against the best-performing baseline of 0.7024, while all the other baselines score lower than this. NLI-verified responses most of the time are similar to the source text because verification penalizes elaboration which cannot be grounded, and the system is less likely to produce hallucinated extensions of the retrieved content in this case. The sharp BERTScore collapse in the Naive RAG (MuRIL) condition demonstrates that retrieval without any verification can produce responses which are different from the reference answers at the semantic level, which shows that the surface-level evaluation metrics might miss this entirely.

The doc+web configuration produces results which are similar to doc-only config. The hallucination rate is 18.2% and 16.1%, BERTScore is 0.7453 and 0.7468 and SACS is 0.735 and 0.745. The 2.1% difference in hallucination is within the range of noise and doesn't not impact on the performance. The

distinction in false refusal rate is 13.2% for doc+web and 13.9% for doc-only which suggests that web fallback is useful for queries that fall outside the document index, but it introduces slight uncertainty on borderline cases such as queries where the system has to decide if the retrieved content is reliable or not. For practical scenarios, where the document corpus provides sufficient information, the doc-only configuration is preferable due to verification precision, but doc+web variant, can be used in situations where out-of-index queries are expected to be very frequent, maybe due to the need of latest information, and therefore, we need to accept a small increase in false refusals in exchange for the boarder coverage in such situations.

6.2 Unanswerable Detection

The unanswerable queries are correctly refused in most of the baselines as shown in the table 3. The Unanswerable detection accuracy of 0.0% means that the system has no mechanism to verify and decide if the answer is correct or not. The 100% correct refusal reflects indiscriminate refusal rather than uncertainty, and treating it as a hallucinated response.

In this context, both our configurations achieve a Correct Refusal Rate of 100% and an Unanswerable Detection Accuracy of 100%, correctly handling all 10 unanswerable queries in the evaluation set. This is the only system in the comparison to do so without incurring excessive false refusals on answerable questions, our false refusal rate remains at 13.9% (doc-only) and 13.2% (doc+web), compared to 50.0% and 39.5% for the baselines that also achieve 100% correct refusal. The distinction matters: a system that refuses everything trivially achieves perfect unanswerable detection at the cost of being useless for answerable queries. Our system achieves the same correct refusal rate while answering 86.1 % of answerable questions.

6.3 Ablation Study

Table 4 and figure 5 presents ablation results obtained by systematically removing each novel component from the full configuration, holding

the MuRIL-based FAISS index, Llama 3.3-70B generation model, and 48-pair test set constant across all conditions.

Table 4. Ablation Study Results for SchemeQA-Kannada

Configuration	F1	BERTScore	HAL%	FRef%	SACS
Full System (Doc + Web)	0.4195	0.7453	18.2%	13.2%	0.735
w/o Section Boost	0.2417	0.6184	26.7%	21.1%	0.760
w/o Scheme Guard	0.2342	0.5621	18.5%	28.9%	0.644
w/o NLI Detection	0.2286	0.6340	N/A	18.4%	N/A

HAL% and SACS are reported only when NLI verification is enabled, as both metrics depend on entailment-based validation.

A question answering system that always generates a response is, in certain contexts, less trustworthy than one that sometimes declines. In government advisory applications, an incorrect answer about eligibility or document requirements could lead a user to make decision on false responses, whereas an honest refusal would atleast leave the question open. Therefore, evaluating abstention behaviour here is not just for answer quality but is an important component.

The dataset includes 63 unanswerable questions, which are designed to simulate the real queries for which there will not be any valid answer present in the indexed document collection. This includes queries about schemes which are not present in the corpus, eligibility conditions which are not covered by official document, and queries that are expected but have no supported content. Inclusion of these questions will help assess the system's refusal behaviour.

The proposed system achieves a Correct Refusal Rate of 100% which reflects the combined effect of confidence thresholding and NLI verification step, which together prevent unsupported content in grounded answer. The False Refusal Rate is an informative metric, which represents the proportion of answerable questions the system refuses to answer. The false refusal rate values of document-only and document+web configurations score are 13.9% and 13.2% respectively which are lower than several baseline systems. This difference matters here because some baselines have high refusal rates which is due to retrieval failure as they couldn't find relevant passages and hence produced no answer even when it was available. The results convey that the system maintains a meaningful balance between answering and abstaining.

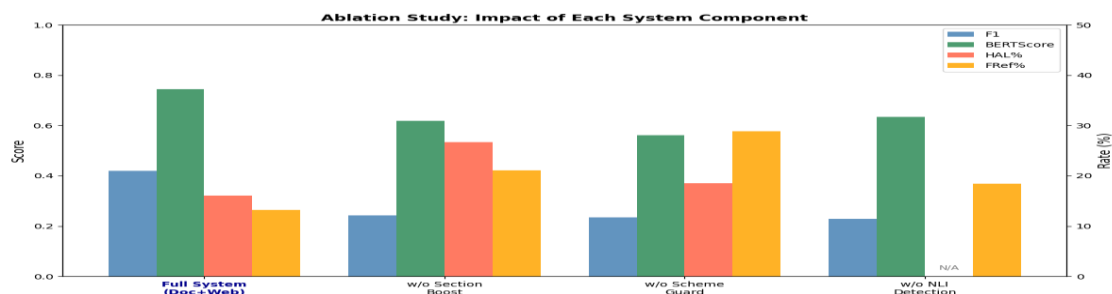


Figure 5. Above figure represents Ablation study results showing the impact of removing each system component on F1, BERTScore, Hallucination Rate, and False Refusal Rate. HAL% is not reported for the configuration without NLI Detection as it depends on entailment-based validation.

6.4 Error Analysis

An analysis of the responses was conducted to categorize the primary failure and edge cases which were observed in evaluation set. Table 5 represents the examples from the schemes.

6.4.1 Correct Answers with Surface Mismatch

One of the patterns in the error analysis, answers that are factually correct but score low on Token F1 due to surface-level mismatches with the reference data. LLaMA-3.3-70b model generates complete, fluent Kannada sentences rather than answering directly from extracted source, it means that even when the answer is accurate, the surrounding sentence structure reduces token overlap with a short gold answer for the same question. For example, in the annual cashless hospital cover under Ayushman Bharat, the gold answer is in Kannada numerals (ರೂ.೫,೦೦,೦೦೦/-), but the system produces Arabic numeral equivalent (₹5,00,000), which results in a Token F1 value of 0.603, even though the 2 answers are factually identical. Here, the BERTScore is 0.817 and SACS is 0.967, which tells us that these metrics capture the semantic agreement which the token matching cannot. This kind of numeral script mismatch is not a single such case, it occurs across multiple scheme-related answers involving amounts and dates, which results in a gap between Token F1 and BERTScore values. Due to this issue which we discussed, we consider BERTScore as the primary quality metric for the system, while still retaining Token F1 as a secondary metric.

6.4.2 False Refusals

Five of the answerable questions in the test set where refused that is false refusal by the application. Here, it's not like hallucinations where the model generates wrong answers confidently, but false refusal withholds correct information, this has a usability consequence for a user seeking welfare guidance.

For example, the question about which authority is responsible for implementing Ayushman Bharat was asked, the correct answer is National Health Authority (NHA) which is present in the

documents, but the system responded with a refusal string (ಈ ಮಾಹಿತಿ ದಾಖಲೆಗಳಲ್ಲಿ ಲಭ್ಯವಿಲ್ಲ). Analysis of the retrieved chunks indicate that FAISS returned passages from peripheral document sections rather than the overview sections where information about NHA is mentioned. The confidence was below the threshold, therefore the system decided to refuse over a low-confidence answer, it's a decision which is consistent with the system's design, but in this case its incorrect. The SACS score is 0.58 for such answers which shows low confidence, and it represents that the scoring functionality is working as expected, but the issue lies in retrieval precision rather than the verification logic.

3 out of the 5 false refusals have similar pattern, where it has questions related to administrative or procedural details which appear in less frequently indexed document sections. To solve this issue, we can improve the index coverage for such sections.

6.4.3 Genuine Hallucinations

Six records are mentioned as high hallucination with BERTScore below 0.70, representing that the generated answer diverts from the reference at the semantic level. Examining their structure reveals that the underlying cause is a retrieval failure than a generation one.

A representative example involves the ineligibility criteria for Ayushman Bharat. The gold answer lists specific exclusion categories such as vehicle owners, holders of Kisan cards above Rs. 50,000, and government employees, but the system responded that no such categories are mentioned in the documents (ಆಯುಷ್ಮಾನ್ ಭಾರತ ಯೋಜನೆಗೆ ಅರ್ಹರಾಗದ ವರ್ಗಗಳು ವಿಷಯದಲ್ಲಿ ನಮೂದಿಸಲ್ಪಟ್ಟಿಲ್ಲ). The exclusion criteria exist in the corpus but appear in a separate subsection from the eligibility content the retriever located. The model generated a response consistent with the incomplete context it received, producing a factual error through omission rather than fabrication. The SACS value of 0.365 correctly identifies this as low-

confidence. A second pattern involves procedural questions where the model introduces possible but unsupported process details, again traceable to incomplete retrieval rather than

unconstrained generation. Both failure modes point to the same intervention: improved section coverage in the FAISS index, rather than changes to the generation or verification components.

Table 5. Qualitative Examples from SchemeQA-Kannada Evaluation

#	Category	Query	Gold Answer	Predicted Answer	F1	BERT	HAL	SACS
1	Correct Answer	PM-KISAN ಭೂಮಿ ದಾಖಲೆಗಳ ಮ್ಯುಟೇಷನ್ ಕೈಗೊಳ್ಳುವ ಅಧಿಕಾರಿಗಳಿಗೆ ಯಾವ ಜವಾಬ್ದಾರಿಯನ್ನು ವಹಿಸಬೇಕು?	ಸ್ಪಷ್ಟವಾದ ಜವಾಬ್ದಾರಿಯನ್ನು ವಹಿಸಬೇಕು	ಸ್ಪಷ್ಟವಾದ ಜವಾಬ್ದಾರಿಯನ್ನು ವಹಿಸಬೇಕು	1.00	1.00	LOW	0.935
2	Surface Mismatch	ಆಯುಷ್ಫಾನ್ ಭಾರತ ಯೋಜನೆಯಡಿ ಪ್ರತಿ ಕುಟುಂಬಕ್ಕೆ ವರ್ಷಕ್ಕೆ ಎಷ್ಟು ನಗದುರಹಿತ ಚಿಕಿತ್ಸೆ ಕವರ್ ಸಿಗುತ್ತದೆ?	ರೂ.30,00,000/- ವರೆಗೆ ನಗದುರಹಿತ ಚಿಕಿತ್ಸೆ	5,00,000 ನಗದುರಹಿತ ಆಸ್ಪತ್ರೆ, ಚಿಕಿತ್ಸೆ ಕವರ್ ಸಿಗುತ್ತದೆ	0.60	0.817	LOW	0.967
3	False Refusal	ಆಯುಷ್ಫಾನ್ ಭಾರತ ಯೋಜನೆಯ ಅನುಷ್ಠಾನಕ್ಕೆ ಜವಾಬ್ದಾರವಾದ ಪ್ರಾಧಿಕಾರ ಯಾವುದು?	ರಾಷ್ಟ್ರೀಯ ಆರೋಗ್ಯ ಪ್ರಾಧಿಕಾರ (ಎನ್‌ಎಚ್‌ಎ)	ಈ ಮಾಹಿತಿ ದಾಖಲೆಗಳಲ್ಲಿ ಲಭ್ಯವಿಲ್ಲ	0.03	0.591	—	0.580
4	Correct Refusal	ಆಯುಷ್ಫಾನ್ ಭಾರತ ಯೋಜನೆಗೆ ಸಂಪರ್ಕ ವಿವರಗಳ ಬಗ್ಗೆ ಮಾಹಿತಿ ಇದೆಯೇ?	ಈ ಮಾಹಿತಿ ದಾಖಲೆಗಳಲ್ಲಿ ಲಭ್ಯವಿಲ್ಲ	ಈ ಮಾಹಿತಿ ದಾಖಲೆಗಳಲ್ಲಿ ಲಭ್ಯವಿಲ್ಲ	1.00	—	NONE	0.580
5	Hallucination	ಆಯುಷ್ಫಾನ್ ಭಾರತ ಯೋಜನೆಗೆ ಅರ್ಹರಾಗದ ವರ್ಗಗಳು ಯಾವುವು?	ಎರಡು, ಮೂರು ಅಥವಾ ನಾಲ್ಕು ಚಕ್ರ ವಾಹನ ಹೊಂದಿರುವವರು, ಸರ್ಕಾರಿ ಉದ್ಯೋಗಿಗಳು...	ಆಯುಷ್ಫಾನ್ ಭಾರತ ಯೋಜನೆಗೆ ಅರ್ಹರಾಗದ ವರ್ಗಗಳು ವಿಷಯದಲ್ಲಿ ನಮೂದಿಸಲ್ಪಟ್ಟಿಲ್ಲ	0.32	0.642	HIGH	0.365
6	Paraphrase Penalization	ಸುಕನ್ಯಾ ಸಮೃದ್ಧಿ ಯೋಜನೆಯ ಯಶಸ್ವಿ ಅನುಷ್ಠಾನಕ್ಕಾಗಿ ಸರ್ಕಾರ ಕ್ರಮಗಳನ್ನು ತೆಗೆದುಕೊಂಡಿದೆಯೇ?	ಸರ್ಕಾರ ಪರಿಣಾಮಕಾರಿ ಹಂತಗಳನ್ನು ತೆಗೆದುಕೊಂಡಿದೆ	ಅನೇಕ ಹೌದು, ಸರ್ಕಾರ ಅನೇಕ ಪರಿಣಾಮಕಾರಿ ಕ್ರಮಗಳನ್ನು ತೆಗೆದುಕೊಂಡಿದೆ	0.62	0.803	HIGH [†]	0.485

[†]HAL=HIGH due to NLI entailment=0.0 on semantically correct answer, paraphrase penalization. BERT not computed for correct refusals (row 4). HAL not applicable for false refusals (row 3).

Note: Unanswerable questions are defined relative to the indexed document corpus.

6.4.4 Paraphrase Penalization

Two records present an instructive edge case: HAL=HIGH flags despite BERTScore above 0.80, indicating that the NLI verification module is penalizing semantically correct answers on account of surface paraphrasing. A representative instance involves a Sukanya Samriddhi question about government implementation steps. The gold answer reads ಸರ್ಕಾರ ಅನೇಕ ಪರಿಣಾಮಕಾರಿ ಹಂತಗಳನ್ನು ತೆಗೆದುಕೊಂಡಿದೆ, while the prediction reads ಹೌದು, ಸರ್ಕಾರ ಸುಕನ್ಯಾ ಸಮೃದ್ಧಿ ಯೋಜನೆಯ ಯಶಸ್ವಿ ಅನುಷ್ಠಾನಕ್ಕಾಗಿ ಅನೇಕ ಪರಿಣಾಮಕಾರಿ ಕ್ರಮಗಳನ್ನು ತೆಗೆದುಕೊಂಡಿದೆ, a longer but semantically equivalent restatement. The NLI model returns an entailment score of 0.0, likely because the retrieved chunk enumerates specific implementation steps while the prediction makes

a general claim about them. The generalization is a valid abstraction, but NLI treats it as neutral rather than entailed, producing a false hallucination flag.

This is a known limitation of post-hoc NLI verification applied to generative outputs and is not specific to Kannada. The translate-then-verify approach compounds it by introducing an additional failure point, a paraphrase that survives translation may still fail entailment against the original document chunk if the translation flattens meaningful semantic distinctions. Replacing the translation step with a native Kannada NLI model would likely reduce this failure directly, and represents one of the more concrete directions for future work.

6.5 SACS Analysis

The SACS scores reported in Table 3, 0.745 for doc-only and 0.735 for doc+web, follows the

expected pattern. Answers grounded in official scheme PDFs carry higher source weights, and the NLI entailment scores for document-retrieved responses tend to be stronger than those from web-sourced or fallback paths, which together push the doc-only average higher. The difference between configurations is small (0.010), but this is itself informative rather than a limitation of the metric. When two configurations produce comparable hallucination rates, assigning them similar SACS values is the correct behavior, artificially separating them on the basis of source type alone would suggest a precision the metric does not actually have.

The more interesting signal lies within the doc-only configuration. Answers backed by retrieved document passages receive an average SACS of 0.745, while queries that resulted in system refusals where source is recorded as none, pull the average down to 0.580. This spread reflects the metric functioning as intended: refusals carry no source weight and, by definition, they lack verified entailment evidence, so they receive lower trust scores. The practical consequence is that the system's self-assessed confidence is broadly consistent with its actual output reliability. Higher scores correspond to cases where document grounding was confirmed, lower scores correspond to cases where the system declined to answer. For a citizen-facing advisory system, this kind of internal calibration has real value, in principle, use SACS thresholds to decide when to surface an answer directly versus when to flag uncertainty to the user. This kind of calibrated self-assessment is a meaningful property for a citizen-facing advisory system to have, even if the SACS metric itself is novel and its calibration has not yet been validated against external ground truth.

6.6. Discussion

6.6.1 The Hallucination Problem is Not Solved by Better Retrieval Alone

The Naive RAG (MuRIL) baseline shares the same embedding model, FAISS index, and LLM reader as our full system. Its hallucination rate is 34.8%, ours is 16.1% (for document only). The

difference is over 18.7 percentage and is attributable entirely to post-hoc NLI verification, section boosting, intent routing, and the scheme mismatch guard, the components that do not alter what the retriever returns, only what happens to the retrieved content afterward.

This finding sits somewhat alongside the prevailing assumption in RAG research that retrieval quality is the primary step for improving factual grounding. That assumption may hold in higher-resource settings where both the retrieval model and the verification component are well-adapted to the target language. For Kannada and most low-resource Indic languages, the generation model's tendency to produce fluent, plausible-sounding output that extends beyond or contradicts the retrieved context is a problem that better retrieval alone cannot address adequately. The results here suggest that post-hoc verification is not a supplementary safeguard but a structurally necessary component in this setting, and that its role in low-resource language RAG pipelines may be systematically underestimated in the existing literature.

6.6.2 Cross-Lingual Verification as a Practical Workaround

The translate-then-verify approach we adopt is, admittedly, a workaround. In an ideal world, a Kannada-native NLI model of sufficient quality would be available, and the translation step, which introduces its own error surface, would be unnecessary. We are not in that world. mDeBERTa-v3-base-xnli[8] does not natively support Kannada, and the Kannada-capable NLI resources that do exist are either too small, too noisy, or too narrowly scoped for production use in a QA verification pipeline.

The translate-then-verify approach works better than one might expect, and the ablation results suggest it works well enough to be genuinely useful. One reason, we think, is that translation errors that matter for NLI mistranslations that change the entailment relationship between premise and hypothesis are relatively rare for the kind of factual, domain-constrained content involved in government scheme QA. The

vocabulary is specialized but not ambiguous. Scheme names, eligibility conditions, and benefit amounts tend to translate consistently. This may not generalize to more open-domain or colloquial Kannada text, where idiomatic expressions and dialectal variation would create more translation uncertainty.

It is also worth acknowledging the latency cost. The full pipeline has retrieval, generation, translation, and NLI verification stages which involves multiple sequential API calls, and in a production environment this would create response time challenges. Our current implementation does not optimize for latency, and is not measured systematically. This is a limitation for real-time application which needs to be addressed in future, possibly through model distillation, caching of common query patterns, or asynchronous verification with provisional responses.

6.6.3 On the Design of SACS

The Source-Aware Confidence Score is, to our knowledge, the first confidence metric for QA systems that explicitly incorporates source credibility tier as a weighted component. The design rationale is straightforward — official government documents are more trustworthy than open-web sources for government scheme information, and a confidence score that ignores this conflates meaningfully different epistemic situations. But we want to be honest about the limitations of the current formulation.

The SACS weights — 0.5 for entailment, 0.3 for source credibility, and 0.2 for model confidence — were set through principled reasoning rather than empirical optimisation. A different weighting scheme might yield better-calibrated confidence estimates, and without a sufficiently large validation set to fit weights against external ground truth, we cannot claim that 0.5/0.3/0.2 is optimal. What the current weights do produce is sensible behaviour: scores are consistently higher for document-grounded answers than web-sourced ones, they correlate with measured hallucination rate in the expected direction, and the three-level trust categorisation maps onto

meaningful differences in answer reliability. The narrow gap between the two system configurations (0.745 doc-only versus 0.735 doc+web) reflects their comparable hallucination rates — when two configurations perform similarly on verification, SACS correctly assigns them similar trust levels rather than differentiating artificially on source tier alone. Whether these specific weights generalise to a larger or more varied evaluation set remains an open question that future work with expanded annotation resources could address.

There is also a design question we set aside but that deserves future attention: should SACS be user-facing at all? Our current demo surfaces the trust level in Kannada (ಹೆಚ್ಚು / ಮಧ್ಯಮ / ಕಡಿಮೆ), on the assumption that citizens benefit from knowing how much confidence to place in a given answer. This assumption seems reasonable but has not been empirically tested. It is at least plausible that displaying a "Low confidence" warning alongside an otherwise helpful answer reduces user trust more than is warranted — or, conversely, that "High confidence" labels lead users to over-rely on answers that are verified but not infallible. User studies examining these questions would be valuable, and they are entirely absent from the current work.

6.6.4 Scheme Coverage and Generalisation

The four schemes covered in SchemeQA-Kannada were chosen for their reach and documentation availability, but they represent a small fraction of the welfare landscape. Karnataka alone has dozens of state-level schemes — Raita Siri, Gruha Lakshmi, Yuva Nidhi, and many others — none of which are covered here. Central schemes beyond the four we include — MGNREGS, PM Awas Yojana, Atal Pension Yojana — are similarly absent. Extending coverage raises several practical challenges beyond simply indexing more documents. Scheme boundaries are not always crisp: a question about housing eligibility might span PM Awas Yojana and a state-level housing scheme simultaneously, requiring multi-scheme

retrieval and answer synthesis in a more complex way than our current comparative query handling supports.

Generalisation beyond government schemes is a more open question. The section-boosted retrieval design is fairly domain-specific — the section taxonomy (eligibility, benefits, overview, process, implementation) maps naturally onto government scheme documents but would need rethinking for, say, legal texts or agricultural advisories. The cross-lingual NLI pipeline is more portable; in principle, the same translate-then-verify approach could be applied to any Kannada QA task where a capable multilingual NLI model is available and translation quality is sufficient. Whether the system's hallucination control properties generalise to more open-domain Kannada QA is something we cannot claim on the basis of the current evaluation.

6.6.5 Broader Implications for Low-Resource Civic AI

We want to close this section with a somewhat broader observation, because we think the challenges encountered in this work are symptomatic of a pattern that deserves more explicit attention in the NLP community. Building reliable AI systems for low-resource languages in high-stakes domains is hard in a specific way: the problems compound. Weak retrieval models degrade generation quality. Poor generation quality makes post-hoc verification more necessary. The absence of native NLI resources forces workarounds that introduce additional error surfaces. Thin evaluation datasets make it difficult to know whether the workarounds are actually working. Each individual gap is tractable; the combination is not.

The response to this, we think, is not to wait for low-resource languages to catch up to English along every infrastructure dimension before deploying systems — that could be a very long wait, and the citizens who would benefit from these systems do not have the luxury of waiting. It is, rather, to be explicit about the failure modes, build verification and confidence signalling into the system from the start rather than as an

afterthought, and treat honest evaluation — including of unanswerable detection and hallucination rate — as a first-class research contribution rather than a secondary concern. SchemeQA-Kannada is an attempt to do this, imperfectly and in a narrow domain, but we hope the design decisions and the evaluation methodology are useful to others building in similar conditions.

6.7 Limitations

Dataset construction: The dataset is researcher-constructed rather than derived from actual citizen query logs, which means the test distribution may not accurately reflect the range of questions a deployed system would encounter in practice. The development set used for hyperparameter tuning consists of 47 pairs which are sufficient for directional tuning but not for high-confidence parameter selection. Future work could enhance this by incorporating query logs from actual citizen interactions and expand coverage to state-level schemes and additional Indian languages.

Evaluation scope: Latency was not measured so, the pipeline was not optimized for real-time use. The system uses multiple components such as MuRIL embedding, FAISS retrieval, LLaMA generation via Groq, and mDeBERTa NLI verification, each introducing latency. Real-time performance optimization, including model quantization and batched inference.

Hallucination detection: The cross-lingual NLI verification pipeline has not been tested on any out-of-domain or colloquial Kannada, where translation and entailment accuracy may decrease. The translate-then-verify approach can also introduce errors, due to mistranslations that may affect the final entailment score. Paraphrase penalization accounts for around two of the six HAL=HIGH cases, potentially increasing the reported hallucination rate by about three to five percentage points. Future work could explore native Kannada NLI models to reduce reliance on translation.

False refusals: False refusals on answerable questions (13.2%) represent the most significant

failure. Three of the five false refusal cases involve administrative or procedural details that appear in less-indexed document sections. This could be solved by improving section coverage in the FAISS index, particularly for overview and contact sections would address a portion of this failure.

User validation: The real user experience is yet to be evaluated. The unanswerable handling, that is the decision to show a refusal string rather than a low-confidence answer with a warning and the SACS based trust display, all these remain unvalidated with respect to the real time users.

These limitations direct to specific directions of future work which would make this work even better and solve the user queries.

7. Conclusion

This paper addresses a concrete and underserved problem: the absence of any document-grounded, hallucination-aware question answering system for Indian government schemes in Kannada. The population that would benefit are rural and semi-urban Kannada speakers who navigate welfare programs that remain quite inaccessible, large, and the consequences of getting answers wrong in this domain are not trivial.

The contributions are the SchemeQA-Kannada dataset, section-boosted MuRIL retrieval with three-stage scheme detection, a scheme mismatch guard for cross-scheme rerouting, cross-lingual NLI-based hallucination detection, and the Source-Aware Confidence Score. Each targets a specific gap in the existing landscape. The contribution lies in their composition: assembling these components into a pipeline designed specifically for low-resource Indic language QA, and evaluating it against baselines that go beyond standard QA metrics to include hallucination rate, false refusal rate, and unanswerable question detection.

The results are: hallucination rate of 16.1% against baselines ranging from 31.6% to 100%, a BERTScore of 0.7468, and 100%, correct unanswerable detection represent a meaningful improvement over what naive retrieval or retrieval-free generation achieves on this task.

The ablation analysis shows that cross-lingual NLI verification accounts for the largest share of the hallucination reduction. This finding has implications beyond this specific system that is for low-resource language QA deployments where native NLI infrastructure is unavailable, a translate-then-verify approach appears to be a practically viable and underexplored alternative.

7.1 Future Work

Several directions merit pursuit. Most immediately, scheme coverage needs to expand — to additional central programs. The dataset should grow beyond 315 pairs, ideally incorporating real citizen query logs. Pipeline latency requires serious attention before any production deployment; the current sequential API architecture is not compatible with a real-time user experience, and distillation, caching, or asynchronous verification strategies all warrant investigation. SACS weights, currently set through design reasoning, should be empirically validated against human reliability judgements on a larger output sample. User studies are conspicuously absent and arguably the most important next step — how citizens interpret confidence signals, respond to refusals, and engage with the Kannada trust labels are empirical questions that system evaluation alone cannot answer. Most structurally, the translate-then-verify workaround points to a deeper gap: the absence of robust Kannada-native NLI resources is a community-level problem, and annotated entailment corpora, fine-tuned verification models, and higher-quality parallel data for Kannada would collectively eliminate the translation noise floor that currently constitutes an irreducible limitation of our verification pipeline — and would benefit Kannada NLP far beyond the narrow scope of this work.

Acknowledgement

Funding Information: The authors received no external funding.

Conflict of Interest: The authors declare no conflict of interest.

Author Contributions: Harshita R, developed the methodology, implemented the system,

conducted experiments, and prepared the manuscript. Asha G R supervised the research, methodology review, validation and reviewed the manuscript.

Ethics Statement: Not applicable.

References

- J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, Proceedings of NAACL-HLT (2019), Minneapolis, MN, USA, pp. 4171–4186. <https://doi.org/10.18653/v1/N19-1423>
- A. Conneau et al., Unsupervised cross-lingual representation learning at scale, Proceedings of ACL (2020), Online, pp. 8440–8451. <https://doi.org/10.18653/v1/2020.acl-main.747>
- P. Lewis et al., Retrieval-augmented generation for knowledge-intensive NLP tasks, Proceedings of NeurIPS (2020), Online, pp. 9459–9474.
- S. Khanuja et al., MuRIL: Multilingual representations for Indian languages, arXiv preprint arXiv:2103.10730 (2021). <https://doi.org/10.48550/arXiv.2103.10730>
- D. Kakwani et al., IndicNLP Suite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages, Findings of EMNLP (2020), Online, pp. 4948–4961. <https://doi.org/10.18653/v1/2020.findings-emnlp.445>
- A. Kunchukuttan et al., AI4Bharat-IndicNLP corpus: Monolingual corpora and word embeddings for Indic languages, arXiv preprint arXiv:2005.00085 (2020). <https://doi.org/10.48550/arXiv.2005.00085>
- Z. Ji et al., Survey of hallucination in natural language generation, ACM Computing Surveys, 55(12) (2023) 1-38. <https://doi.org/10.1145/3571730>
- P. He, J. Gao and W. Chen, DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing, Proceedings of ICLR (2023), Virtual.
- J. Clark et al., TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages, Transactions of the Association for Computational Linguistics, 8 (2020) 454-470. https://doi.org/10.1162/tacl_a_00317
- P. Rajpurkar, R. Jia and P. Liang, Know what you don't know: Unanswerable questions for SQuAD, Proceedings of ACL (2018), Melbourne, Australia, pp. 784–789. <https://doi.org/10.18653/v1/P18-2124>
- P. Lewis et al., MLQA: Evaluating cross-lingual extractive question answering, Proceedings of ACL (2020), Online, pp. 7315–7330. <https://doi.org/10.18653/v1/2020.acl-main.653>
- T. Zhang et al., BERTScore: Evaluating text generation with BERT, Proceedings of ICLR (2020), Addis Ababa, Ethiopia.
- S. Robertson and H. Zaragoza, The probabilistic relevance framework: BM25 and beyond, Foundations and Trends in Information Retrieval, 3(4) (2009) 333-389. <https://doi.org/10.1561/15000000019>
- AI4Bharat, Jugalbandi: Multilingual question answering for government schemes, Technical Report (2023). <https://ai4bharat.iitm.ac.in>
- Ministry of Agriculture and Farmers Welfare, Government of India, Pradhan Mantri Kisan Samman Nidhi (PM-KISAN) Operational Guidelines, New Delhi, India (2019). <https://pmkisan.gov.in>
- National Health Authority, Government of India, Ayushman Bharat Pradhan Mantri Jan Arogya Yojana: Scheme Guidelines, New Delhi, India (2018). <https://nha.gov.in>
- Ministry of Finance, Government of India, Pradhan Mantri Jan Dhan Yojana: Scheme Document, New Delhi, India (2014). <https://pmjdy.gov.in>
- Ministry of Finance, Government of India, Sukanya Samriddhi Yojana: Scheme Guidelines, New Delhi, India (2015).
- A. Dubey et al., The Llama 3 herd of models, arXiv preprint arXiv:2407.21783 (2024). <https://doi.org/10.48550/arXiv.2407.21783>
- A. Radford et al., Robust speech recognition via large-scale weak supervision, Proceedings of ICML (2023), Honolulu, HI, USA, pp. 28492–28518.