DR-RAG: Addressing Retrieval Misalignment in Low-Resource Urdu Question Answering

Saad Ahmad, Muhammad Hammad, Muhammad Zeeshan, Faizad Ullah, Asim Karim
Department of Computer Science
Lahore University of Management Sciences (LUMS), Lahore, Pakistan
{23030014, 23030013, 23030011, faizadullah, akarim}@lums.edu.pk

Abstract

Retrieval-Augmented Generation performs well on English QA benchmarks, but degrades considerably in morphologically rich, low-resource languages. Urdu presents a particularly challenging case: heavy inflectional morphology, Nastaliq script inconsistencies, and limited training data produce a systematic mismatch between query representations and indexed document content that standard retrieval architectures cannot bridge. We propose **DR-RAG** (Dual-Representation Retrieval-Augmented Generation), which addresses this through dual indexing. Each document is represented as overlapping text chunks and as automatically generated question-answer pairs. Queries are first matched against the QA index, which aligns more reliably with natural query phrasing than declarative document chunks. When retrieval confidence falls below $\tau = 0.80$, the system falls back to chunk-based retrieval, maintaining coverage without sacrificing precision. Evaluated on Urdu UQA and English SQuAD 2.0, DR-RAG improves Urdu METEOR by $38\times$, ROUGE-1 by 140%, and reduces generation latency by 43%. LLM-as-judge scores show higher faithfulness (3.03 vs 1.93) and overall quality (2.99 vs 2.21) over MultiVector. English performance remains competitive throughout. These results indicate that representation-level alignment between queries and indexed content, rather than increased model complexity, is the critical factor for reliable retrieval in underserved South Asian languages. Code: https://github.com/saadahmad17/DR_RAG.

Keywords: Low-resource NLP, Urdu question answering, Retrieval-augmented generation, South Asian languages, Morphological complexity, Dual-representation retrieval

1. Introduction

Retrieval-Augmented Generation has become a standard approach for grounding language model outputs in external knowledge, yet its development has remained disproportionately centred on English. The assumptions underlying most RAG pipelines, well-calibrated dense embeddings, consistent tokenization, and morphologically stable text, hold reasonably well for English but break down in languages where these conditions are not met. Urdu is precisely such a language, and building a functional Urdu QA system exposes these assumptions in concrete terms.

Large Language Models have achieved strong results across summarization, question answering, and dialogue (Brown et al., 2020), but retain well-documented limitations: knowledge is fixed at training time, factual hallucination remains a persistent problem, and domain-specific or time-sensitive information is inaccessible without external retrieval (Petroni et al., 2019; Shuster et al., 2021). RAG addresses part of this by grounding generation in retrieved evidence (Lewis et al., 2020; Izacard and Grave, 2020). In low-resource settings, however, the retrieval component itself becomes the bottleneck. For Urdu specifically, tokenization errors and weaker semantic alignment substantially depress retrieval precision (Khanuja et al., 2021). A query using one inflected form of a word will not reliably match a document passage using a different inflected form of the same word, even when both refer to identical content.

Multi-vector retrieval, as implemented in LangChain's MultiVector retriever, attempts to address this through richer document representation. Our experiments show that while this approach improves retrieval in English, it degrades to near-zero precision on Urdu. The morphological and orthographic gap between query and document is sufficiently large that English-tuned retrieval strategies cannot bridge it reliably.

We introduce **DR-RAG** (Dual-Representation Retrieval-Augmented Generation) as a linguistically motivated response to this failure mode. The framework indexes each document in two complementary forms: overlapping text chunks and automatically generated question-answer (QA) pairs. The QA index functions as a query-language resource, storing document content in a syntactic form that more closely mirrors user query structure, thereby reducing the representational distance that causes retrieval to fail in Urdu. A confidence-aware fallback mechanism routes queries to the QA index first, switching to chunk retrieval only when the top similarity score falls below a set threshold. Language-specific choices in chunk sizing and embedding

Proceedings of the Second workshop on Challenges in Processing South Asian Languages (CHiPSAL2026) @ LREC, pages 49–58
16 May 2026.

models additionally address the tokenization noise characteristic of low-resource Urdu text.

Evaluated on Urdu UQA (Arif et al., 2024) and English SQuAD 2.0 (Rajpurkar et al., 2018), DR-RAG achieves a $38\times$ METEOR improvement, 140% ROUGE-1 gain, and 43% latency reduction on Urdu, while remaining competitive on English. Our contributions are as follows:

1. We provide a systematic analysis of retrieval misalignment in Urdu QA, attributing it to morphological inflection, orthographic variability, and inadequate low-resource embedding coverage.
2. We propose DR-RAG, a dual-representation retrieval framework with a confidence-aware fallback policy specifically designed to address retrieval misalignment in morphologically rich, low-resource languages.
3. We demonstrate through evaluation on UQA and SQuAD 2.0 that linguistically motivated design choices, rather than increased architectural complexity, are the primary driver of reliable performance gains in South Asian language question answering.

2. Related Work

2.1. Retrieval-Augmented Generation

RAG was originally proposed to mitigate factual hallucination in large language models by conditioning generation on retrieved evidence (Lewis et al., 2020). Subsequent work has extended this foundation in several directions. Hybrid retrieval combines sparse BM25 with dense vector search to improve robustness across domains (Mandikal and Mooney, 2024), while modular architectures incorporate query rewriting and reranking as task-dependent components (Yu et al., 2023). Adaptive retrieval methods introduce selectivity: DRAGIN conditions retrieval on model uncertainty (Su et al., 2024), reinforcement learning policies optimise the accuracy-efficiency trade-off (Shuster et al., 2024), and MemoRAG caches prior retrievals to support multi-step reasoning (Qian et al., 2025). For complex multi-document reasoning, layered retrieval architectures decompose queries into tractable subproblems (Li et al., 2025; Zhang et al., 2025), while passage integration models such as FiDO (De Jong et al., 2023) and ColBERT-RAG (Santhanam et al., 2022) improve cross-passage comparison during decoding. Structured evaluation frameworks like RAGAS (Rajagopal et al., 2023) now enable systematic assessment of faithfulness and contextual relevance.

Despite this considerable progress, virtually all RAG research has been conducted on English benchmarks. The extent to which these advances transfer to morphologically rich, low-resource languages, where embedding quality is lower and linguistic preprocessing tools are limited, remains largely uninvestigated. This work addresses that gap directly, using Urdu as the primary test case.

2.2. Urdu NLP and RAG

Urdu NLP has historically been constrained by the absence of large-scale annotated resources. The release of the UQA dataset (Arif et al., 2024) represented a significant step forward, providing a benchmark for both extractive and generative question answering in Urdu. Tahir et al. (Tahir et al., 2024) subsequently evaluated a range of multilingual models on UQA, documenting consistent and substantial performance gaps relative to English, gaps that motivate the present work directly.

Recent model development has made further progress possible. UrduLLaMA 1.0 (Fiaz et al., 2025) demonstrates that instruction tuning on native Urdu corpora yields meaningful improvements over general multilingual models such as mBERT and XLM-R. Alif (AI, 2025) extends this further through large-scale Urdu-centric pretraining. However, even these dedicated models exhibit weaknesses in factual consistency and multi-hop reasoning, indicating that stronger generation capacity alone cannot compensate for retrieval-level failures. Linguistic preprocessing resources including UrduHack (Contributors, 2021) and LughaatNLP (Farooq and Khan, 2023) provide morphological normalisation support that underpins this work. Recent findings confirm that adaptation gaps for South Asian languages persist even with modern instruction-tuned LLMs (Khade et al., 2025).

To our knowledge, no prior work has examined the specific mechanisms by which standard RAG retrieval fails in Urdu, nor proposed an architecture explicitly designed to address those mechanisms.

2.3. Urdu Linguistic Challenges for RAG

Three structural properties of Urdu create specific and compounding difficulties for retrieval-based systems.

Morphological Complexity. Urdu is highly inflectional, marking gender, number, and case through suffixation. A single lexical root can yield multiple orthographically distinct surface forms across grammatical contexts: the verb meaning “to write”, for instance, surfaces as *likha*, *likhi*, *likhta*, or *likhti* depending on subject agreement. Both BM25 and dense embedding models operate on surface forms, meaning that a query using one inflected variant will not reliably retrieve a document passage using a different variant of the same root. In Urdu,

this is not an edge case but a pervasive retrieval failure mode.

Script and Orthographic Variation. Urdu is written in Nastaliq, a connected cursive script in which word boundaries are often ambiguous and optional spacing alters tokenization in unpredictable ways. Diacritical marks, zabar, zer, and pesh, which disambiguate phonemic and occasionally semantic distinctions, are routinely omitted in digital text.

Multilingual tokenizers, optimised predominantly for high-resource languages, handle this variation inconsistently, producing embeddings that fail to generalise across orthographic forms of the same token (Khanuja et al., 2021). Specifically, this ambiguity manifests at the tokenization level in digital text processing: tokenizers not designed with Nastaliq's cursive connectivity in mind frequently mis-segment word boundaries, producing inconsistent token sequences for the same lexical item across different orthographic renderings. We characterise this as *embedding brittleness*: the tendency of Urdu token representations to fragment across surface variants rather than converging on stable semantic representations.

Low-Resource Embedding Weakness. Urdu is substantially underrepresented in the pretraining corpora of most multilingual embedding models. Dense retrieval requires a well-calibrated semantic space, and that calibration is data-dependent. BM25 faces a parallel problem: vocabulary coverage is limited and reliable morphological normalisation tools are scarce, leaving both retrieval paradigms less effective than they would be for a high-resource language (Tahir et al., 2024).

Implications for Retrieval Design. These properties combine to produce a query-document mismatch that operates at the level of surface form rather than semantic content. A user query and the relevant document passage may refer to identical information while appearing sufficiently different to current retrieval systems to go unmatched. DR-RAG addresses this by indexing content in question form, which preserves the syntactic structure of natural queries and substantially reduces the representational distance that causes retrieval to fail in Urdu.

3. Methodology

DR-RAG (Dual-Representation Retrieval-Augmented Generation) indexes each document twice: as overlapping text chunks and as automatically generated question-answer pairs. At query time, the QA index is searched first, since questions align more naturally with other questions than with raw document text. If no confident match is found, the system falls back to chunk-based retrieval. Figure 1 shows the complete three-phase pipeline.

3.1. Document Processing and Indexing

Each document is split into overlapping chunks of 200-500 tokens using a recursive splitter that preserves continuity across boundaries. Overlapping chunks matter especially for Urdu, where morphological context often spans clause boundaries and losing that carryover weakens semantic understanding.

From each chunk, multiple QA pairs are generated to create a semantically dense secondary index. For English, QA pairs are produced using LLaMA 2 7B (Touvron et al., 2023). For Urdu, GPT-4o is used instead. Pilot tests showed that Urdu-specific models such as Alif (AI, 2025), GLM-4, and AYA produced inconsistent outputs with weaker factual alignment, consistent with findings in (Arif et al., 2024) that general-purpose models outperform Urdu-specific ones on instruction-following tasks.

Generated QA pairs were subsequently reviewed by professional Urdu speakers to verify factual accuracy and filter hallucinated or culturally inconsistent content prior to indexing, ensuring the quality of the dual-representation index.

Embedding and Vectorization. Chunks and QA pairs are embedded using language-appropriate encoders. For English, `all-MiniLM-L6-v2` (Wang et al., 2020) provides lightweight but reliable semantic representations. For Urdu, `intfloat/multilingual-e5-large` (Wang et al., 2024b) performs better given its stronger multilingual alignment for low-resource settings. All vectors are normalised before indexing.

Vector Indexing. Embeddings are stored in `Qdrant` (Team, 2023), a vector database built for efficient approximate nearest neighbour search. Separate indices for QA pairs and chunks allow the system to switch between representations at query time.

3.2. Retrieval and Answer Generation

Primary Retrieval. Incoming queries are embedded and matched against the QA index using cosine similarity. Question-form representations produce better matches for user queries than declarative document chunks do, particularly in Urdu where the morphological gap between query and passage can be wide.

Confidence-Aware Retrieval Policy. When the top QA match falls below $\tau = 0.80$, the system switches to chunk-based retrieval. This threshold

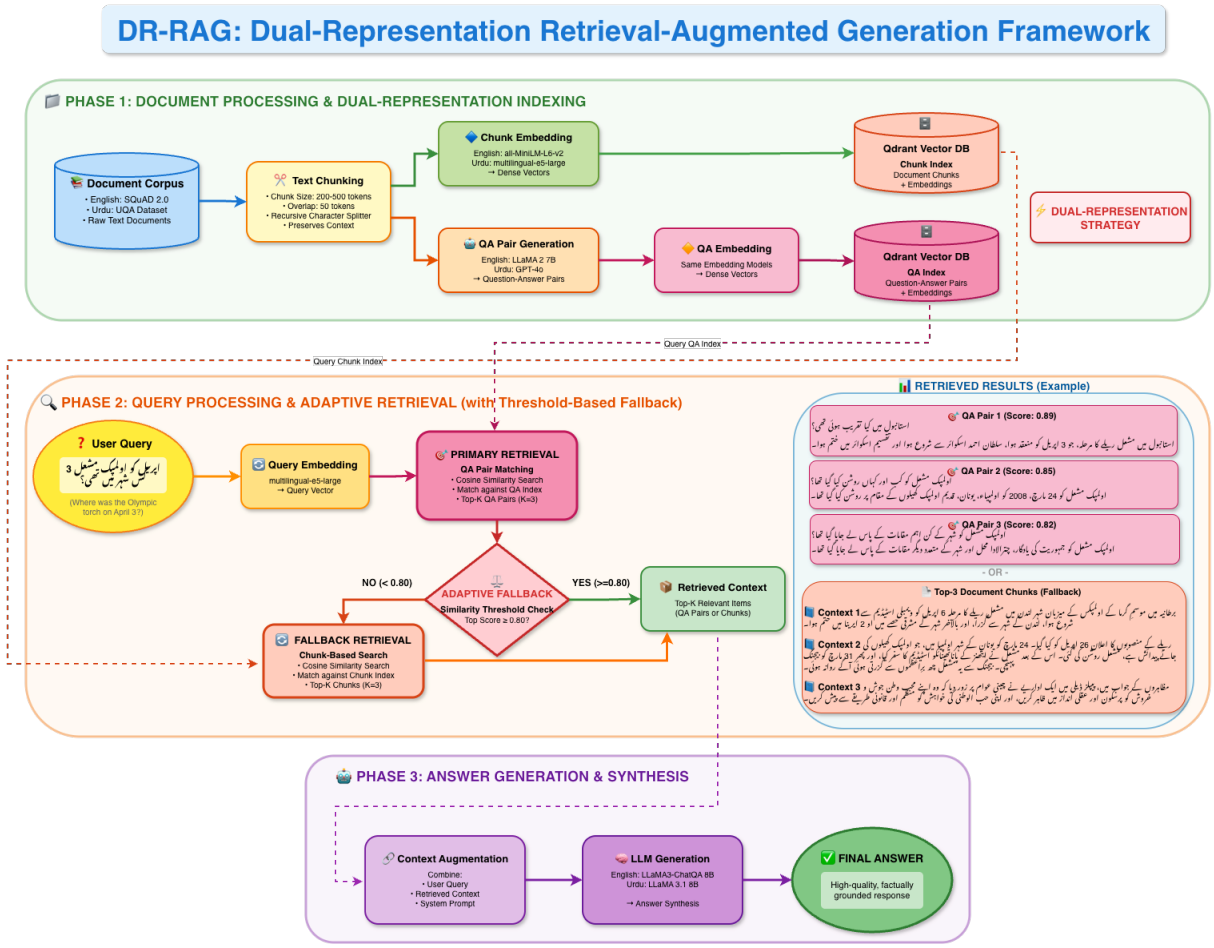

Figure 1: DR-RAG system architecture showing the dual-representation retrieval pipeline. **Phase 1: Document Processing**—Documents are split into overlapping chunks while an LLM generates question–answer pairs from the same content. Both are embedded (`all-MiniLM-L6-v2` for English, `multilingual-e5-large` for Urdu) and indexed separately. **Phase 2: Confidence-Aware Retrieval**—Queries first search the QA index. If the best match scores above $\tau = 0.80$, that context is used; otherwise the system falls back to document chunks. **Phase 3: Answer Generation**—Retrieved context feeds into LLaMA3-ChatQA 8B (English) or LLaMA 3.1 8B (Urdu) to produce the final answer. This dual-index design with confidence-aware fallback improves both precision and coverage, especially for morphologically complex low-resource languages such as Urdu.

was selected empirically based on observed confidence score distributions across the evaluation set, where scores above 0.80 consistently corresponded to semantically relevant matches. This keeps the system functional even when QA pairs are sparse or poorly matched, something that static multi-vector systems like LangChain's default implementation (Chase and Contributors, 2023) cannot do.

Answer Synthesis. Retrieved context is passed to a generative model for final answer production. LLaMA3-ChatQA 8B handles English queries and LLaMA 3.1 8B handles Urdu, as it better accommodates Urdu morphology.

3.3. Comparative Baselines

Traditional RAG. A standard chunk-only retriever that finds the top-3 most similar chunks via cosine similarity and passes them to the generator. This represents the common deployment baseline for low-resource QA.

LangChain MultiVector. Uses hierarchical parent-child indexing, where document chunks are parents and derived representations such as summaries or QA pairs are children. This gives richer semantic coverage than flat chunk retrieval. However, the retrieval policy is static with no confidence-based switching, which our experiments show causes it to collapse on Urdu (Chase and Contributors, 2023).

3.4. Evaluation Framework

Performance is measured across four dimensions: lexical accuracy, semantic alignment, retrieval effectiveness, and computational efficiency.

Automatic Metrics. BLEU, ROUGE, METEOR, and SacreBLEU measure lexical overlap with reference answers. We acknowledge these metrics have known limitations for morphologically rich languages and complement them with BERTScore for semantic alignment and two factuality indicators: Faithfulness (consistency with retrieved evidence) and Answer Relevancy (how directly the response addresses the query). Retrieval quality is assessed via Context Precision, Context Recall, and Entity Recall.

LLM-as-Judge. GPT-3.5 Turbo rates each response on a 1-5 scale across Faithfulness, Correctness, Relevance, Conciseness, and Overall Quality. This gives a human-aligned quality signal that automatic metrics alone cannot capture.

Latency. Retrieval and generation times are recorded separately. For low-resource deployment contexts, where hardware is often constrained, latency is not just a performance metric but a practical deployment consideration.

3.5. Evaluation Setup

Hardware and Software. Experiments ran on a local workstation with an NVIDIA GTX 1080 (8 GB VRAM) and 24 GB RAM, using `PyTorch`, `Hugging Face Transformers`, `LangChain`, and `Qdrant`. Model quantization was applied to reduce memory load and inference latency.

Datasets. We used SQuAD 2.0 (Rajpurkar et al., 2018) for English and UQA (Arif et al., 2024) for Urdu. From a corpus of 440 topic-aligned PDF documents, five were indexed for English and two for Urdu, reflecting the genuine data scarcity in the Urdu setting. Models were tested on 1,200 English and 900 Urdu queries.

Compared Systems.

- **Traditional RAG:** Top-3 chunk retrieval via cosine similarity.
- **LangChain MultiVector:** Parent-child indexing without adaptive fallback.
- **DR-RAG:** QA-first retrieval with confidence-aware fallback at $\tau = 0.80$.

3.6. Retrieval Quality

LangChain MultiVector is competitive in English but falls apart on Urdu, recording precision of 0.001 and recall of 0.004. This is not a minor gap but a near-total failure, confirming that English-optimised retrieval does not transfer to morphologically complex languages (Tahir et al., 2024).

Metric	Trad. RAG	MultiVector	DR-RAG
Context Precision			
English	0.0402	**0.0710**	0.0495
Urdu	0.0311	0.0010	**0.0672**
Context Recall			
English	**0.4555**	0.3340	0.3697
Urdu	0.2880	0.0040	**0.4023**
Entity Recall			
English	**0.3386**	0.2750	0.2968
Urdu	0.2042	0.0450	**0.3331**

Table 1: Retrieval metrics for English and Urdu. DR-RAG leads on Urdu across all three metrics. LangChain MultiVector collapses to near-zero on Urdu, confirming it is not suited for low-resource settings.

Metric	Trad. RAG	MultiVector	DR-RAG
BERTScore F1	0.7955	0.8057	**0.8134**
BLEU	1.5424	4.7063	**4.8860**
ROUGE-1	0.0660	0.1569	**0.1584**
ROUGE-2	0.0211	**0.0890**	0.0817
ROUGE-L	0.0652	0.1549	**0.1577**
Faithfulness	0.8150	0.8480	**0.8760**
Answer Relevancy	**0.8650**	0.8960	0.8470
METEOR	0.0056	**0.2555**	0.2134
SacreBLEU	2.5278	5.8257	**7.2140**

Table 2: Urdu QA results. DR-RAG leads on most metrics. MultiVector scores higher on METEOR (0.2555 vs 0.2134) but DR-RAG leads on BLEU, SacreBLEU, BERTScore, ROUGE-1, ROUGE-L, and Faithfulness.

DR-RAG reaches Urdu precision of 0.067 and recall of 0.402, a more than sixtyfold precision improvement over MultiVector. Storing content as QA pairs reduces the inflectional variability that breaks chunk-only retrieval, giving the system a stable semantic target even when embeddings are trained on limited data.

3.7. Generation Quality

3.7.1. Urdu Generation Performance

Compared to Traditional RAG, DR-RAG improves ROUGE-1 by 140%, METEOR by nearly $38\times$, and BERTScore by around 2%. The METEOR gain is the most telling: that metric rewards systems that handle morphological variation and synonymy, which are the two core difficulties in Urdu. The QA-pair index appears to normalise inflectional diversity

Metric	Trad. RAG	MultiVector	DR-RAG
BERTScore F1	0.8632	0.1903	**0.8681**
BLEU	0.0694	**0.1134**	0.0730
ROUGE-1	0.2807	**0.3262**	0.3003
ROUGE-2	0.1759	**0.2115**	0.1857
ROUGE-L	0.2771	**0.3249**	0.2960
METEOR	0.4210	**0.4747**	0.4350
Faithfulness	**0.5377**	0.5570	0.5292
Answer Relevancy	**0.6555**	0.7150	0.6033
SacreBLEU	0.0950	**0.1504**	0.0980

Table 3: English QA results. MultiVector leads on lexical overlap. DR-RAG achieves the highest BERTScore, confirming no regression on high-resource data.

System	Retrieval (s)	Generation (s)	Total (s)
English			
Traditional RAG	2.851	4.933	7.784
LangChain MultiVector	**0.074**	1.038	1.112
DR-RAG	2.837	**4.799**	**7.637**
Urdu			
Traditional RAG	**6.793**	14.461	21.255
LangChain MultiVector	0.066	1.465	1.532
DR-RAG	6.846	**8.303**	**15.149**

Table 4: Latency results. DR-RAG cuts Urdu generation time by 43% over Traditional RAG while also improving output quality.

in a way that raw chunk retrieval simply cannot.

Faithfulness rises from 0.815 to 0.876, meaning DR-RAG's answers stay closer to the retrieved evidence. MultiVector scores higher on METEOR (0.2555 vs 0.2134), but DR-RAG leads on the majority of metrics including the factual grounding ones that matter most for QA reliability.

3.7.2. English Generation Performance

On English SQuAD, MultiVector leads on lexical metrics (ROUGE-1: 0.326, BLEU: 0.113), which reflects its English-tuned retrieval favouring surface token overlap. DR-RAG achieves the best BERTScore (0.868), showing stronger semantic alignment even when exact wording differs. Faithfulness scores are close across all systems (0.54-0.56), which makes sense: English has mature embeddings and tokenization tools that already handle factual grounding well. These results confirm DR-RAG is competitive on high-resource data while delivering its main gains in Urdu.

3.8. Latency and Efficiency

MultiVector is the fastest overall, but this appears to reflect implementation optimisation rather than any architectural advantage. DR-RAG's main efficiency gain is in Urdu generation: 8.3s versus 14.5s for Traditional RAG, a 43% reduction. Better retrieval produces tighter context, and the generator processes fewer irrelevant tokens as a result. For real-world South Asian language deployments, where hardware is often constrained, this kind of efficiency gain is as important as accuracy.

System	Faith.	Correct.	Relev.	Concise.	Overall
Urdu					
Traditional RAG	2.208	**2.617**	2.925	2.508	2.565
LangChain MultiVector	1.925	2.158	2.500	2.267	2.212
DR-RAG	**3.033**	2.608	**3.400**	**2.917**	**2.990**
English					
Traditional RAG	3.033	2.267	2.525	2.808	2.658
DR-RAG	**3.067**	2.242	2.867	2.992	2.792
LangChain MultiVector	3.167	**2.717**	**2.917**	**3.267**	**3.017**

Table 5: LLM-as-Judge scores (1-5). DR-RAG leads on all Urdu dimensions and maintains the highest Faithfulness in English.

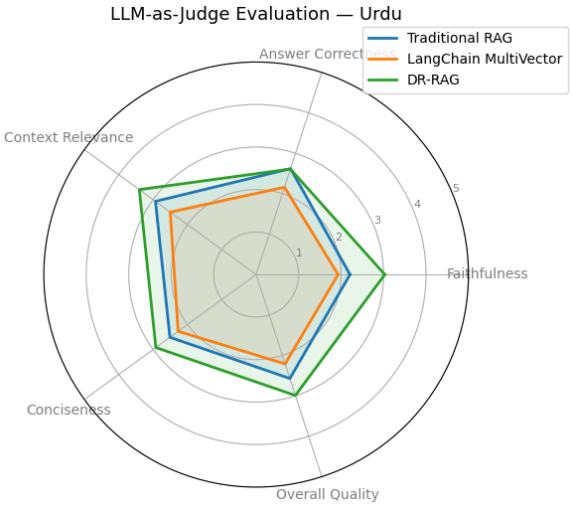

Figure 2: LLM-as-Judge radar chart for Urdu QA. DR-RAG leads across all qualitative dimensions.

3.9. LLM-as-Judge Evaluation

GPT-3.5 Turbo rated each response on Faithfulness, Correctness, Relevance, Conciseness, and Overall Quality (1-5 scale). For Urdu, DR-RAG leads on every dimension. The 57% Faithfulness advantage over MultiVector (3.03 vs 1.93) is the most important result here: it shows the system's answers stay grounded in retrieved evidence rather than drifting into hallucination.

In English, MultiVector scores higher on Overall Quality (3.02 vs 2.79) and Conciseness, consistent with its English-tuned pipeline. DR-RAG holds the highest Faithfulness (3.07). For applications where evidence traceability matters more than brevity, that is the right trade-off to make.

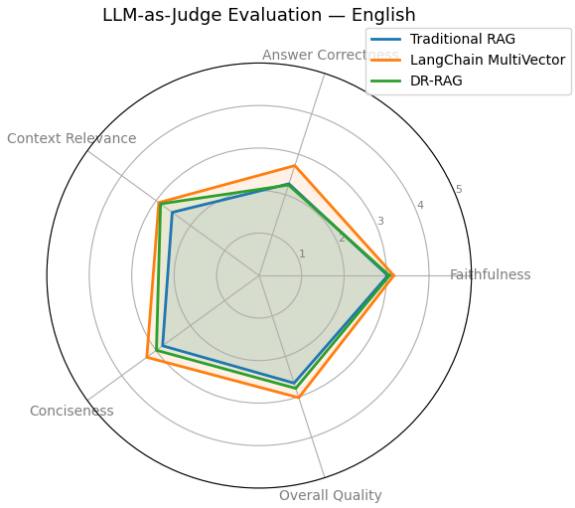

Figure 3: LLM-as-Judge radar chart for English QA. MultiVector leads on correctness; DR-RAG leads on faithfulness.

3.10. Comparative Analysis

The results tell a consistent story across all evaluation dimensions.

Where DR-RAG helps most. Urdu shows the biggest gains. QA pairs reduce the inflectional and paraphrastic variation that trips up chunk-only retrieval, which is exactly the morphological mismatch described in Section 2.3. The effect shows up in both retrieval (precision 60x higher than MultiVector) and generation (METEOR 38x above Traditional RAG). These are not narrow metric wins but broad quality improvements that hold across lexical, semantic, and factual dimensions.

English trade-offs. In English, strengths split across systems. MultiVector leads on lexical overlap and speed; DR-RAG leads on semantic alignment and faithfulness. The contrast suggests the dual index matters most when resources are scarce. When mature embeddings already bridge query-document gaps, the two approaches converge.

Speed and quality together. For Urdu, DR-RAG cuts generation time by 43% while improving quality. This happens because better retrieval produces shorter, tighter context for the generator. Getting both speed and accuracy improvements from the same architectural change is the clearest sign that the design is working as intended.

What the judge scores add. The qualitative evaluation reinforces the quantitative findings. DR-RAG is more faithful and more relevant in Urdu. In English it trades some conciseness for stronger grounding. Given that the primary use case is low-resource Urdu QA, that is a reasonable trade-off.

4. Conclusion

Retrieval failure in Urdu is not a general NLP problem but a specific structural one. The inflectional distance between query and document surface forms is wide enough that standard retrieval architectures, whether sparse or dense, consistently fail to establish reliable matches. DR-RAG addresses this by indexing document content in question form, creating representations that are syntactically and semantically closer to natural user queries. A confidence-aware fallback ensures the system remains robust when QA matching is uncertain, routing queries to chunk-based retrieval without significant overhead.

The results on Urdu UQA are substantial. METEOR improves by $38\times$, ROUGE-1 by 140%, and generation latency decreases by 43% relative to a traditional RAG pipeline. LLM-as-judge scores confirm stronger faithfulness and contextual relevance across all evaluated dimensions. Crucially, performance on English SQuAD 2.0 remains competitive with both baselines, demonstrating that these gains are additive rather than the result of optimising narrowly for Urdu at the expense of high-resource performance.

The central contribution of this work is not the architecture itself but the principle underlying it: that retrieval quality in low-resource languages depends on representational alignment between queries and indexed content, and that this alignment must be designed for rather than assumed. The improvements documented here emerge from linguistic reasoning about Urdu's specific properties rather than from increased model capacity or architectural complexity. This suggests the approach generalises naturally to other morphologically rich, low-resource languages that share similar retrieval failure modes, several of which are prominent across South Asia.

5. Limitations and Future Work

The current implementation has three notable limitations. The Urdu evaluation relies on only two indexed documents, a constraint imposed by the availability of suitable Urdu corpora at the time of experimentation. While the results are encouraging, validating whether these gains hold across larger and more topically diverse collections remains an important open question. Second, system output quality is evaluated using LLM-as-judge scoring via GPT-3.5 Turbo, without direct human evaluation of generated answers. While LLM-as-judge provides a scalable quality signal, human evaluation by native Urdu speakers remains an important next step to fully validate that these metrics reflect real-world perceived quality. Finally, the fallback threshold $\tau = 0.80$ was selected empirically

based on observed confidence score distributions, and has not been subjected to full sensitivity analysis. Retrieval confidence distributions in Urdu and other South Asian languages may differ substantially from English, suggesting that a learned, query-adaptive threshold policy would generalise more reliably across domains and linguistic settings.

Future work will extend evaluation to larger and more topically diverse Urdu document collections to validate that the observed gains generalise beyond the current small-scale index. Broader evaluation across additional low-resource South Asian languages sharing similar morphological and orthographic properties is also planned. The fallback threshold $\tau = 0.80$, currently selected empirically, will be subjected to systematic sensitivity analysis, with the goal of developing a learned, query-adaptive thresholding policy suited to latency-constrained deployment environments. Finally, DR-RAG will be benchmarked against recent RAG systems developed for Persian and Arabic, languages that share the same script family with Urdu, to assess cross-lingual transferability and position DR-RAG as a generalised solution across this script community.

Ethical Considerations

This study raises no significant ethical concerns. The datasets used, UQA and SQuAD 2.0, are publicly available, widely used in the research community, and contain no personally identifiable information. All experiments were conducted on locally hosted hardware, avoiding third-party data transmission and limiting environmental overhead. While DR-RAG improves factual grounding by anchoring generation in retrieved evidence, it does not eliminate the risk of bias inherited from pretrained models. This is a particular concern for low-resource South Asian languages, where the nature and extent of such biases are less thoroughly documented than for English, and warrants attention in any downstream deployment.

6. Bibliographical References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, et al. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 1877–1901.

Harrison Chase and LangChain Contributors. 2023. Langchain: Building applications with large language models. https://github.com/langchain-ai/langchain. GitHub Repository.

Michiel De Jong, Joris van Es, and Malvina Nissim. 2023. Fido: Fusion-in-decoder optimized for stronger retrieval-augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4571–4585, Singapore. Association for Computational Linguistics. Retrieval-Augmented Generation Optimization.

Layba Fiaz, Munief Hassan Tahir, Sana Shams, and Sarmad Hussain. 2025. Urdullama 1.0: Dataset curation, preprocessing, and evaluation in low-resource settings. *arXiv preprint arXiv:2502.16961*.

Gautier Izacard and Edouard Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2619–2630.

Omkar Khade, Shruti Jagdale, Abhishek Phaltankar, Gauri Takalikar, and Raviraj Joshi. 2025. Challenges in adapting multilingual LLMs to low-resource languages using LoRA PEFT tuning. In *Proceedings of the First Workshop on Challenges in Processing South Asian Languages (CHiPSAL)*.

Simran Khanuja, Sandipan Dandapat, Anirudh Srinivasan, Sunayana Sitaram, and Monojit Choudhury. 2021. Muril: Multilingual representations for indian languages. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4549–4564, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 9459–9474.

Ziyang Li, Chen Liu, and Rui Zhang. 2025. Multihop-rag: Layered reasoning and evidence aggregation in retrieval-augmented generation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Priyanka Mandikal and Raymond J. Mooney. 2024. Hybridir: A hybrid sparse and dense retrieval

framework for robust open-domain qa. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, et al. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2463–2473.

Hongjin Qian, Zheng Liu, Peitian Zhang, Kelong Mao, Defu Lian, Zhicheng Dou, and Tiejun Huang. 2025. Memorag: Boosting long context processing with global memory-enhanced retrieval augmentation. In *Proceedings of the ACM Web Conference 2025 (TheWebConf 2025)*, Sydney, Australia. ACM.

Divyanshu Rajagopal, Aman Madaan, Sahil Gupta, Rajkumar Krishnan, Jiatong Chen, Alexander Zhang, Chris Callison-Burch, and Sameer Singh. 2023. Ragas: An evaluation framework for retrieval-augmented generation. arXiv preprint arXiv:2311.09514.

Keshav Santhanam, Omar Khattab, Christopher Potts, and Matei Zaharia. 2022. Colbert-x: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2762–2773.

Kurt Shuster, Jing Xu, Stephen Roller, and Jason Weston. 2024. Adaptive-rag: Reinforcement learning for adaptive retrieval in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024)*, Bangkok, Thailand. Association for Computational Linguistics. Adaptive Retrieval Optimization for RAG.

Chi Su, Yu Wang, Zhi-Hong Deng, and Minlie Huang. 2024. Dragin: Dynamic retrieval-augmented generation via iterative neural querying. In *Proceedings of The 12th International Conference on Learning Representations (ICLR)*.

Hanna Tahir, Bameesha Bahim, and Asim Karim. 2024. Benchmarking multilingual models on urdu nlp tasks. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, et al. 2023. Llama: Open and efficient foundation language models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Pengcheng Yu, Zhengliang Liu, Qi Zhang, Yizhe Xie, and Yongchao Zhu. 2023. Generate-retrieve-read: Large language models are strong zero-shot retriever-readers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 14318–14333, Singapore. Association for Computational Linguistics. Retrieval-Augmented Generation Framework.

Minghao Zhang, Yizhou Wang, and Jimmy Lin. 2025. Layered query retrieval: Structuring multi-hop reasoning in retrieval-augmented generation. In *Proceedings of the 2025 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

7. Language Resource References

Traversaal AI. 2025. *Alif 1.0: Urdu-English Large Language Model*. Traversaal AI. Hugging Face, 1.0. PID https://huggingface.co/large-traversaal/Alif-1.0-8B-Instruct.

Samee Arif and Sualeha Farid and Awais Athar and Agha Ali Raza. 2024. *UQA: Corpus for Urdu Question Answering*. Urdu NLP Research Community. arXiv, 1.0. PID https://arxiv.org/abs/2405.01458.

UrduHack Contributors. 2021. *UrduHack: An NLP Toolkit for Urdu Language*. UrduHack. GitHub, 2.1. PID https://github.com/urduhack/urduhack.

Taha Farooq and Zunaira Khan. 2023. *LughaatNLP: Linguistic Toolkit for Low-Resource Urdu Text Processing*. Lughaat. GitHub, 1.0. PID https://github.com/lughaat/lughaatnlp.

Pranav Rajpurkar and Robin Jia and Percy Liang. 2018. *Know What You Don't Know: Unanswerable Questions for SQuAD*. Stanford NLP Group. arXiv. PID https://arxiv.org/abs/1806.03822.

Qdrant Team. 2023. *Qdrant: Vector Search Engine for the Next Generation of AI Applications*. Qdrant. Qdrant Technologies, 1.8. PID https://qdrant.tech.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2024a. Improving multilingual text embedding with large language models. arXiv preprint arXiv:2402.05640.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024b. Multilingual E5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05640*.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 5776–5788.