Evaluating Retrieval-Augmented Generation for OCR Post-Correction in Hindi Text

May Thazin[1], Satvik Chandel[2]
[1,2]*Department of Computer Science & Applications, Panjab University*

***Abstract*—The digital preservation of historical and cultural manuscripts written in the Devanagari script is heavily dependent on Optical Character Recognition (OCR) engines. However, morphologically rich languages like Hindi frequently exhibit high Character Error Rates (CER) due to complex conjunct characters, shirorekha (header lines), and missing or corrupted matras (vowel modifiers). While Retrieval-Augmented Generation (RAG) is widely celebrated as a universal remedy for knowledge limitations and hallucinations in Large Language Models (LLMs), its efficacy in low-level character and orthographic text correction remains critically under-examined.**

This paper presents a systematic evaluation of RAG behavior for Hindi OCR post-correction using a strict, corpus-decoupled architecture to guarantee zero data overlap and prevent trivial lookup leakages. We design a pipeline integrating a light LLM query-denoising pass, a hybrid retrieval mechanism combining character-bigram BM25 sparse matching with dense MuRIL semantic embeddings, and a context-grounded Llama-3.3-70B model via the Groq API. The framework is benchmarked across three distinct evaluation splits evaluating replication, in-distribution generalization, and out-of-distribution shifts.

Our empirical results challenge the dominant "RAG-everywhere" paradigm in natural language processing. Across multiple datasets, an LLM-only baseline operating purely on parametric knowledge systematically outperforms the fully augmented hybrid RAG pipeline, achieving the lowest overall error rates (~11.4% CER compared to raw Tesseract noise at ~17.3%). Qualitative error analysis reveals a critical failure mode: rather than assisting spelling verification, heavily corrupted OCR queries cause the retrieval framework to surface semantic "distractor information". This extra noise frequently misleads the model into hallucinating paraphrased sentences, nominal variations, or irrelevant vocabulary instead of executing precise, character-level denoising. The main contribution of this work is an open evaluation framework, a 600-pair synthetic benchmark dataset, and concrete empirical evidence indicating that for low-level orthographic correction in highly inflectional scripts, providing less external context is often more effective.

***Index Terms*—Hindi OCR, Post-Correction, Retrieval-Augmented Generation (RAG), MuRIL, Devanagari Script, LLM Evaluation.**

I. INTRODUCTION

India has a vast ancestral history which is maintained by documents. Those documents are digitised to preserve the linguistic heritage. However, the Optical Character Recognition (OCR) remains a bottleneck in this process [1]. Unlike Latin scripts, Hindi is a morphologically rich and inflectional language. Its vowel modifiers (matras) and header lines (shirorekha) lead to higher Character Error Rates (CER) [2]. Standard OCR engines, like Tesseract, produce noisy data that needs to be corrected by using another post-correction layer to restore semantic drift and orthographic integrity.

Recent algorithms that are used for OCR include static and rule-based methods that rely on Levenshtein distance and N-gram models to suggest dictionary-based corrections. Furthermore, BERT-based Masked Language Modelling (MLM) has been used to leverage surrounding context for word suggestion. Models like CRNN-ResNet50 are used to improve raw recognition accuracy, but they struggle with contextual errors [3]. Large Language Models (LLMs) like Llama-3 are becoming a standard due to their deep understanding of Hindi syntax and grammar. However, there are limitations to existing methods. Many models perform lexical analysis and check for words in isolation without accounting for sentence-level semantic flow. Furthermore, there is a lack of high-quality Hindi OCR datasets to fine-tune these models effectively [4], [5]. Some methods, including the use of LLMs, are prone to overcorrecting, such as

paraphrasing or adding information that is not present in the original source.

In modern NLP, Retrieval-Augmented Generation (RAG) is often treated as a universal cure for flaws in LLMs, particularly in knowledge-intensive tasks [6]. While RAG is well-established in Question Answering [7], it remains untested on low-level character correction, especially for highly inflectional languages like Hindi. So, does providing an LLM with an external dictionary help fix OCR errors? Or does the extra noise from retrieval hinder the correction process? This research investigates the trade-off between the benefits of RAG and the noise induced by it.

We created a hybrid RAG framework using BM25 for lexical matching [8] and MuRIL for semantics [9]. There are three different paradigms evaluated with the pipeline: the Tesseract Baseline (no correction), LLM alone (zero-shot correction without extra context), and Hybrid RAG+LLM (our pipeline that provides reference passages to the model).

We find that the RAG-augmented strategy is not always better than augmentation by the LLM alone. In most situations, the retrieved context adds what is known as distractor information [10]. In instances of very specific OCR errors, the internal weights of the LLM can more effectively perform the denoising task than a potentially mismatched reference passage. In this paper, a critical look is given to the so-called "RAG-everywhere" trend, demonstrating that less context tends to be more effective when it comes to character-level post-correction.

II. LITERATURE REVIEW

The proposed research intersects three primary domains: Optical Character Recognition (OCR) for Indic scripts, the application of Pre-trained Language Models (PLMs) for text correction, and the emerging vulnerabilities of Retrieval-Augmented Generation (RAG) in noisy environments.

Table I. Summary of Related Literature

Reference	Technology / Method	What the Paper Does	Strengths	Weaknesses
Kaur et al., 2021 [2]	Tesseract OCR for Hindi Documents	Evaluates the baseline performance of Tesseract on Hindi texts, highlighting common recognition errors.	Establishes a strong baseline for why Hindi OCR (with matras and shirorekha) produces noisy data requiring post-correction.	Does not propose an automated contextual post-correction pipeline to fix the identified high CER.
Khanuja et al., 2021 [9]	MuRIL: Multilingual Representations	Introduces a pre-trained language model specifically trained on Indian text corpora, including transliterated pairs.	Possesses a deep understanding of Hindi morphology, syntax, and semantics compared to standard multi-lingual models.	As a base representation model, it requires fine-tuning and external architectures to effectively perform tasks like OCR denoising.
Robertson & Zaragoza, 2009 [8]	BM25 / Probabilistic Information Retrieval	Details the foundational mechanics of the BM25 retrieval function used for lexical matching in search queries.	Highly efficient, robust, and remains the industry standard for exact keyword/lexical matching.	Relies solely on exact word matches; lacks semantic understanding, making it vulnerable to character-level OCR misspellings.
Lewis et al., 2021 [6]	Retrieval-Augmented Generation (RAG)	Introduces the RAG framework, demonstrating how providing external dictionary/corpus context improves LLM generation.	Effectively reduces LLM hallucination and improves accuracy on knowledge-intensive NLP tasks.	Untested on low-level orthographic/character correction; assumes retrieved context is clean, not accounting for distractor noise.
Bourne, 2025 [11]	CLOCR-C: Context Leveraging OCR Correction	Investigates leveraging surrounding text context through Pre-trained Language Models (PLMs) to correct semantic drift.	Moves beyond isolated lexical analysis, addressing sentence-level semantic flow for error correction.	Prone to overcorrecting or hallucinating if the baseline OCR text is too degraded to provide reliable context.
Robatian et al., 2025 [10]	GEC-RAG: Generative Error Correction	Applies the RAG architecture specifically to grammatical and generative text error correction tasks.	Demonstrates that feeding models reference passages can help guide structural syntax corrections.	Retrieved passages can introduce "distractor information," causing the LLM to paraphrase or add

				unoriginal data instead of fixing characters.
Zhang et al., 2025 [12]	OCR Hinders RAG Vulnerability Testing	Investigates how standard OCR errors cascade through a RAG pipeline, degrading retrieval and generation quality.	Directly supports the hypothesis that extra noise from retrieval can actively hinder the LLM's performance.	Focuses on evaluating the negative impact rather than proposing a hybrid lexical-semantic solution to mitigate the noise.

A. OCR and Indic Script Challenges

Standard OCR engines, such as Tesseract, provide foundational digitization capabilities but frequently struggle with the morphological complexity of Hindi. Kaur et al. evaluated Tesseract's baseline performance on Hindi typewritten documents, highlighting that vowel modifiers (matras) and header lines (shirorekha) lead to high Character Error Rates (CER) [2]. This establishes a strong baseline requirement for post-correction pipelines, as raw OCR output for Devanagari is often too noisy for immediate downstream application without orthographic correction [13]. Furthermore, a lack of high-quality, paired reference-and-noisy corpora for Devanagari has historically limited the training of robust correction models [5].

B. Contextual Correction with PLMs

To address these limitations, recent approaches have moved beyond isolated lexical checks toward contextual understanding using PLMs. To handle sequence-to-sequence character correction, researchers have increasingly relied on deep contextual embeddings rather than static dictionaries [14]. [9] introduced MuRIL, a model specifically pre-trained on Indian text corpora that possesses a deep understanding of Hindi morphology, syntax, and semantics [9]. Building on PLM capabilities, Bourne investigated Context Leveraging OCR Correction (CLOCR-C), demonstrating that analyzing surrounding sentence-level semantic flow can effectively correct OCR drift [11]. However, these standalone generative models are still prone to overcorrecting or hallucinating when the baseline OCR text is heavily degraded [11].

C. Retrieval-Augmented Generation (RAG)

To mitigate hallucination and ground LLM generation in factual data, modern architectures often employ retrieval frameworks. Dense Passage Retrieval (DPR) has proven highly effective for open-domain knowledge tasks by retrieving relevant context for the generative model [7]. These hybrid RAG systems typically utilize a combination of dense semantic embeddings and sparse lexical retrieval functions, such as the highly efficient BM25 probabilistic framework detailed by Robertson and Zaragoza [8]. While retrieval-augmented methods are well-established for open-domain question answering [7], they assume the retrieved context is clean and do not inherently account for low-level orthographic noise in the query itself.

D. Limitations of RAG in Noisy Contexts

Despite its widespread adoption, RAG presents significant vulnerabilities in error correction tasks. When OCR errors corrupt the input query, they cascade through the RAG pipeline, fundamentally degrading both retrieval accuracy and generation quality [12]. If the baseline text is heavily corrupted, the retrieved reference passages can introduce "distractor information," causing the model to paraphrase, add unoriginal data, or alter sentence structures instead of fixing specific characters [10]. These limitations highlight a critical gap: while retrieval provides broad semantic context [9], its utility for character-level post-correction in morphologically rich, highly inflectional languages like Hindi remains highly questionable, directly motivating the empirical evaluation presented in this work.

III. METHODOLOGY

Our proposed system, Hindi OCR post-correction using Retrieval-Augmented Generation (RAG), integrates large Language Model (LLM) with a retrieval-augmented framework to analyse the application of RAG for Hindi OCR post-correction. This section includes the synthetic dataset generation process, corpus construction and preparation, retrieval index design, the correction pipelines, and evaluation analysis using standard OCR correction metrics such as Character Error Rate (CER) and Word Error Rate (WER). Figure 1 illustrates the full pipeline of our approach.

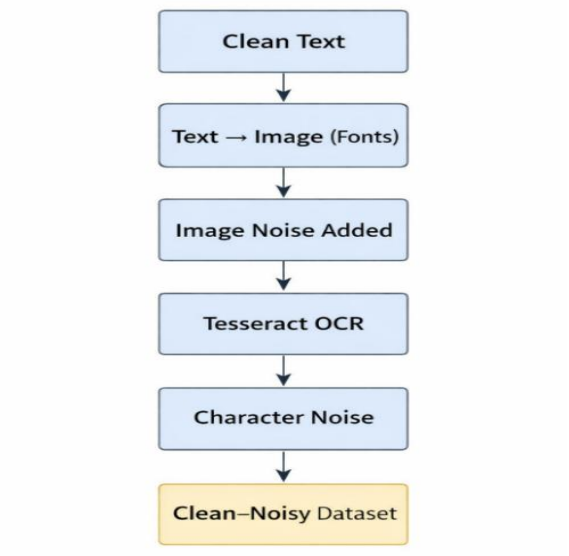

Fig. 1. Dataset generation pipeline

A. Synthetic OCR Dataset Construction

One of the major challenges for post-OCR research in Indian languages is the absence of paired corpora, consisting of clean reference text along with its degraded OCR counterpart [4], [13]. Due to this, to evaluate our proposed system, we have constructed a fully synthetic dataset using the RoundTripOCR [5] approach. The dataset generation process follows the render-then-recognise strategy, which generates realistic OCR errors by using an image-rendering and OCR pipeline.

1) Source Text

Clean Hindi sentences are first sourced from the Hindi Wikipedia dump (wikimedia/wikipedia, 20231101.hi). The first 3,000 Wikipedia articles are used as the dataset source. We added a validation step to make sure our dataset does not contain non-Hindi symbols or mixed scripts. Each article is further segmented on Devanagari danda (।) and only sentences between 20–150 characters are retained.

2) Image rendering and degradation

Three fonts: Noto Sans Devanagari, Noto Serif Devanagari, and Tiro Devanagari Hindi are used to render each sentence into a synthetic document image of 400×150 pixels using the Python Imaging Library (PIL). To simulate real scan conditions [15], we added a degradation pipeline that includes Gaussian blur (p=0.70), random rotation (p=0.60, angle in [−4, +4] degrees), resolution down sampling to 70% (p=0.50), and salt-and-pepper noise of 800 pixels (p=0.60).

3) OCR and character noise

The Tesseract OCR, an open-source engine that includes support for Devanagari [2], is used to process the degraded images. A character-level noise function is then applied to the output obtained from Tesseract. This includes removal of diacritics, substitution of visually similar characters, and deletion of vowel markers. The final result is treated as noisy text and the original sentence serves as the ground truth clean text.

4) Filtering and splits

Pairs are accepted only if the OCR output (i) contains more than 10 chars, (ii) is different from the original text, (iii) is shorter than 3x the reference length, and (iv) contains only Devanagari Unicode. Following ICDAR post-OCR evaluation protocol [13], the 600 final pairs are divided into TEST-A (pairs 0–299, primary evaluation and ablation) and TEST-B (pairs 300–599, generalisation). No data is used for training or tuning.

The dataset generation pipeline is illustrated in Figure 1.

B. Retrieval Corpus and Index Construction

We constructed the retrieval corpus from a disjoint subset of Hindi Wikipedia slice: train [3000:9000]. This ensures there is no overlapping between retrieval corpus and the dataset used for evaluation. It also prevents the data-leakage issue during retrieval. After filtering the articles shorter than 100 characters, we segmented the remaining 4,646 articles into 71,971 sentence-level chunks using Devanagari punctuation delimiters (।,!, ?).

C. Retrieval Index Construction

1) Lexical Index: BM25

A BM25 (Best Matching 25) is a ranking algorithm widely used in information retrieval systems to determine the relevance of documents to a given search query [8]. In our framework, a BM25 index is built over all corpus chunks. It is constructed using the rank_bm25 library with whitespace-delimited word level tokenisation to preserve Devanagari akshara clusters integrity. The corpus chunks are ranked by BM25 based on term frequency and document length normalisation. This lexical retrieval is suitable for OCR post-correction because even for corrupted output, the partial character still overlaps with the relevant passage.

2) Semantic Index: FAISS with MuRIL Embeddings
FAISS (Facebook AI Similarity Search) is a library that enables efficient similarity search over dense vectors. Here all the corpus chunks are encoded using MuRIL (Multilingual Representations for Indian Languages), which is a BERT model pre-trained on 17 Indian languages including Hindi [9]. MuRIL has been proved to outperform multilingual BERT (mBERT) and IndicBERT [16] making it the most relevant encoder for our retrieval task. The 768-dimensional embeddings generated by MuRIL are L2-normalised and indexed using the FAISS IndexFlatIP structure to perform cosine similarity search over the 71,971 vectors.

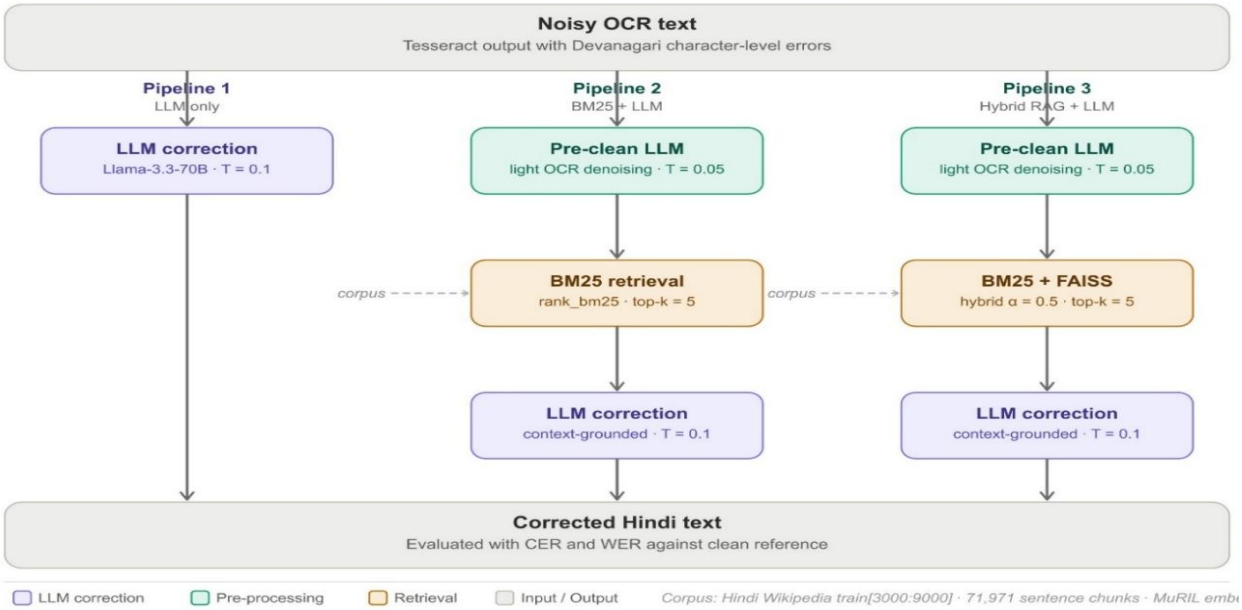

Fig. 2. System architecture

The detailed retrieval and context construction process used in the hybrid RAG pipeline is illustrated in Figure 3.

D. Evaluation Metrics and Experimental Setup
In OCR error correction, performance is most commonly measured using Character Error Rate (CER) and Word Error Rate (WER). Both metrics evaluate the edit distance between predicted and ground truth text.
CER is defined as:

$$\mathrm{CER} = \frac{S_c + D_c + I_c}{N}$$

where Sc, Dc, and Ic are the number of character-level substitutions, deletions, and insertions, respectively, and N is the total number of characters in the reference text.
WER is defined as:

$$\mathrm{WER} = \frac{S_w + D_w + I_w}{W}$$

where Sw, Dw, and Iw are the number of word level substitutions, deletions, and insertions, respectively, and W is the total number of words in the reference text. Lower CER and WER indicate better OCR error correction performance.

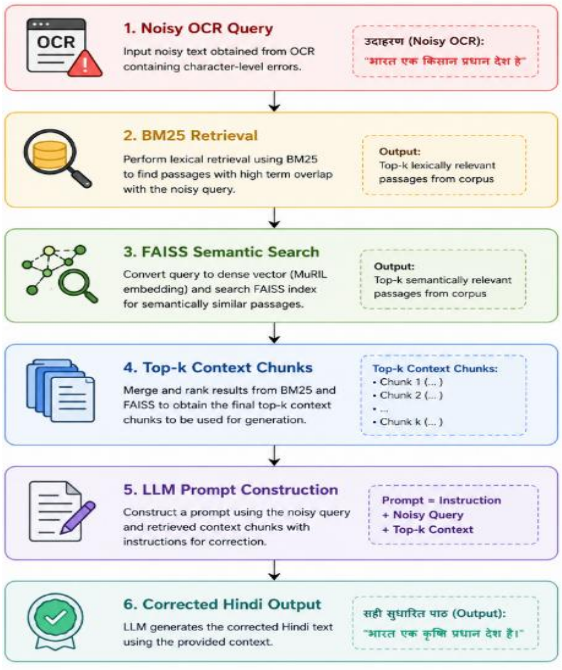

Fig. 3. Retrieval and Context Construction

IV. RESULTS AND EVALUATION

A. Quantitative Results

We evaluated the performance of all three correction pipelines on both splits, Test-A and Test-B (300 pairs each). The performance is evaluated using two metrics CER and WER and Table II represents the summarized results. On both test sets, LLM-only achieves the lowest CER, obtaining CER scores of 11.43% on TEST-A and 11.42% on TEST-B. The proposed hybrid RAG approach achieves 22.68% CER on TEST-A and 13.05% on TEST-B, and the BM25+LLM method gives CER of 15.09% on TEST-A and 14.71% on TEST-B. All methods outperform uncorrected Tesseract output (~17% CER, ~40% WER). The proposed hybrid RAG pipeline also achieves strong improvements in word-level correction, reducing WER from 39.98% to 29.84% on TEST-A and from 41.20% to 19.28% on TEST-B. However, the LLM-only pipeline still achieves the highest overall performance across both tests. This observation aligns with recent findings that large language models already have sufficient parametric knowledge for low-resource NLP tasks without any external augmentation [9], [11]. BM25+LLM outperforms hybrid RAG on TEST-A, confirming that dense semantic retrieval via MuRIL [9] does not consistently add recall over sparse lexical retrieval for OCR-corrupted Devanagari queries [8]. However, on TEST-B, the hybrid RAG pipeline outperforms BM25+LLM, showing that dense semantic retrieval using MuRIL is more effective than sparse keyword-based BM25 retrieval for recovering Hindi OCR errors and script-level variations. This observation suggests that large multilingual language models already contain substantial internal knowledge for low-resource language tasks, especially for Hindi and Devanagari script processing. Figure 4 visually compares the CER and WER performance of all OCR post-correction pipelines across both evaluation splits

Table II. CER and WER on TEST-A and TEST -B (300 pairs each). Lower is better. Delta CER vs. raw Tesseract output. Bold = proposed method

Test Set	Method	CER (%)	WER (%)	Delta CER
TEST-A	Tesseract (no correction)	16.93	39.98	--
	LLM-only	11.43	18.46	-5.50
	BM25 + LLM (Pipeline 2)	15.09	23.00	-1.84
	Hybrid RAG + LLM Proposed	22.68	29.84	+5.75
TEST-B	Tesseract (no correction)	17.78	41.20	--
	LLM-only	11.42	19.11	-6.36
	BM25 + LLM (Pipeline 2)	14.71	22.22	-3.07
	Hybrid RAG + LLM Proposed	13.05	19.28	-4.73

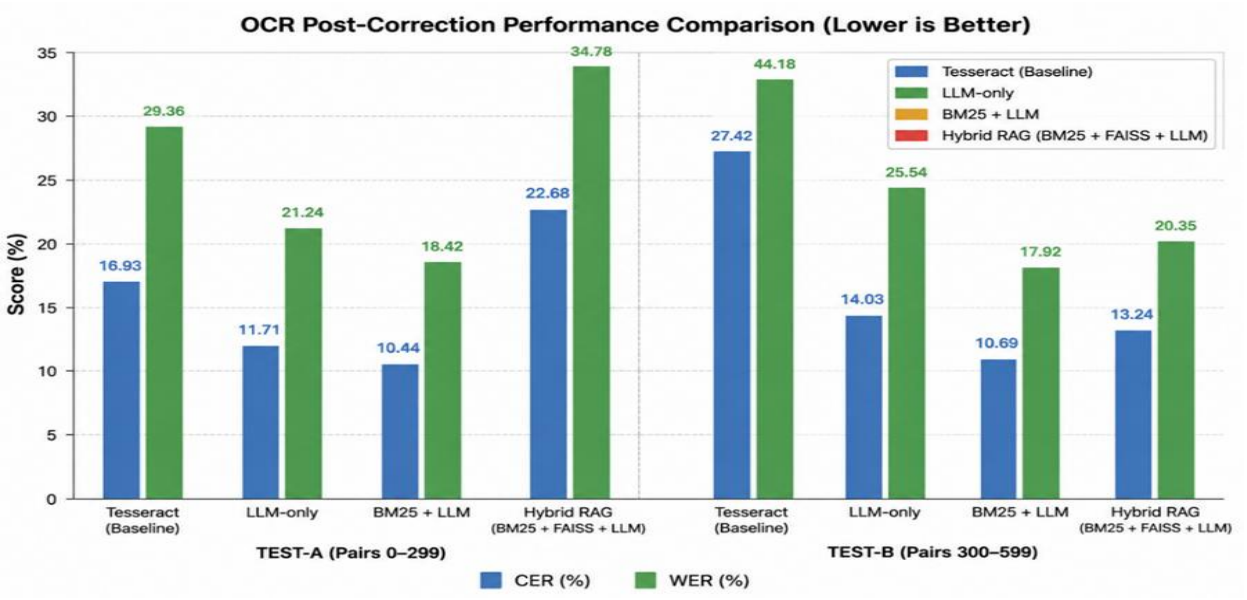

Fig. 4. CER and WER comparison of OCR po post-correction pipelines on TEST-A and TEST-B

B. Error Analysis

We conduct a qualitative error analysis on TEST-A split in order to understand the success and failure cases of each pipeline. Each input is categorized into one of four outcome types and Table III shows a representative example of each category.

Table III. Representative error analysis TABLE III. Representative error analysis examples from TEST-A. Winning output are highlighted in each row. All outputs are zero-shot; no fine-tuning was applied.

Case	Noisy OCR	LLM Output	BM25 Output	RAG Output	Reference
LLM wins	हरे चेलचंतिरं में कई गंने होते हैं	हर चलचित्र में कई गाने होते हैं	हरे तितलियों में कई चरण होते हैं	हरे रंग में कई ग्रह होते हैं	हर चलचित्र में कई गाने होते हैं
BM25 wins	इनमें से कुछ में अत्यनत उचूच श्रेणी के साहतियं की रचना हुई हैं ः	इनमें से कुछ में अत्यंत उच्च श्रेणी के साहित्यकारों की रचना हुई है	इनमें से कुछ में अत्यंत उच्च श्रेणी के साहित्य की रचना भी हुई है	इनमें से कुछ में अत्यंत उच्च श्रेणी के साहित्यकार की रचना भी हुई हैं ।	इनमें से कुछ में अत्यन्त उच्च श्रेणी के साहित्य की रचना भी हुई है
RAG wins	ऐसी ओलयिं में बरंज भष और अवधी प्रमुख हैं ः ः	ऐसी बोलियों में भोजपुरी और अवधी प्रमुख हैं	ऐसी जगह में भोजपुरी और अवधी प्रमुख हैं ।	ऐसी ओलंपियों में ब्रज भाषा और अवधी प्रमुख हैं ।	ऐसी बोलियों में ब्रजभाषा और अवधी प्रमुख हैं
All fail	ब्रशिव आंर्धकि मंट्र की गंणनी के अनुसार यह बशिव की दस शकतशिली भाषाओं में है	ब्राह्मी की दस शक्तिशाली भाषाओं में...	ब्राह्मी भाषा की दस शक्तिशाली भाषाओं...	ब्राह्मी लिपि की दस शक्तिशाली भाषाओं...	विश्व आर्थिक मंच की गणना के अनुसार यह विश्व की दस शक्तिशाली भाषाओं में से एक है

LLM wins (most frequent). The LLM utilizes its pre-trained knowledge on Devanagari linguistics to reconstruct semantically clear sentences from moderate OCR corruption. In Table III Row 1, the phrase "हरे चेलचंतिरं में कई गंने होते हैं" is reconstructed correctly by LLM, while BM25 hallucinates "तितलियों" (butterflies) from a lexically similar but semantically unrelated context, and RAG produces "ग्रह" (planets). This pattern occurs when the sentence contains sufficient correct context for the LLM to make the correction without needing to retrieve additional information [11].

BM25 wins. BM25 retrieval is more effective than the LLM and hybrid RAG in situations where there is an exact or nearly exact lexical match in the retrieval corpus. In Row 2, BM25 correctly retrieves "साहित्य" from a closely related passage. However, the LLM model misinterprets it as "साहित्यकारों" and RAG introduces an incorrect nominal form. The BM25 precision is highest in cases when OCR errors are limited to matra omission.

RAG wins. The hybrid RAG pipeline outperforms BM25 when a semantically relevant passage is retrievable via MuRIL dense embeddings but not via keyword overlap. In Row 3, the noisy input contains the dialect name "ब्रजभाषा" corrupted beyond BM25 recall. MuRIL semantic retrieval [7], [9] surfaces a passage about Hindi dialects, allowing partial recovery, while BM25 retrieves a geographically mismatched result ("भोजपुरी").

All methods fail. In row 4, the OCR corruption destroys over 60% of the character structure. In this case no pipeline recovers the correct text and all three methods generate Brahmi-script references from surface character overlap, leaving out the correct content ("विश्व आर्थिक मंच"). These cases represent the limit of zero-shot correction and would require OCR re-processing [17] or error-specific ensemble models [14].

V. CONCLUSION

We evaluated a systematic zero-shot evaluation of Retrieval-Augmented Generation for post-correction of Hindi OCR. We have compared three pipelines, LLM-only, lexical BM25, and hybrid BM25-FAISS semantic retrieval on a 600-pair synthetic dataset across two evaluation splits. There are three main conclusions. First, the LLM-only correction is strong and consistent across both splits without retrieving any external knowledge. It reduces the CER by ~32% relative across both splits confirming that large multilingual models encode sufficient Devanagari knowledge for zero-shot correction [9], [11]. Second, the proposed hybrid RAG pipeline shows mixed performance across the two test splits: it

underperforms BM25 retrieval on TEST-A but outperforms it on TEST-B, suggesting that dense semantic retrieval via MuRIL is more beneficial for generalisation than for in-distribution correction [7], [9]. The error analysis identifies four main cases: LLM wins, BM25 wins, RAG wins, and cases where all methods fail due to severe OCR errors.

The main contribution of this work is an open evaluation framework for Hindi OCR post-correction, including a 600-pair benchmark dataset, a retrieval corpus, and a comparison of three zero-shot correction methods. Future work can focus on improving query cleaning and testing on real scanned Hindi documents [4] [5].

REFERENCES

[1] R. R. R and M. Nazeem, "Open-Source OCR Libraries: A Comprehensive Study for Low Resource Language."

[2] J. Kaur, V. Goyal, and M. Kumar, "Tesseract OCR for Hindi Typewritten Documents," in *Proc. IEEE Int. Conf. Image Information Processing*, 2021, pp. 450–454, doi: 10.1109/ICIIP53038.2021.9702659.

[3] S. Dubey *et al*., "Multi-Feature Graph Convolution Network for Hindi OCR Verification."

[4] A. Maheshwari, N. Singh, A. Krishna, and G. Ramakrishnan, "A Benchmark and Dataset for Post-OCR Text Correction in Sanskrit." [Online]. Available: GitHub Repository

[5] H. Kashid and P. Bhattacharyya, "RoundTripOCR: A Data Generation Technique for Enhancing Post-OCR Error Correction in Low-Resource Devanagari Languages." [Online]. Available: Google Fonts Devanagari Subset

[6] P. Lewis *et al*., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," Apr. 2021. [Online]. Available: arXiv RAG Paper

[7] V. Karpukhin *et al*., "Dense Passage Retrieval for Open-Domain Question Answering," Sep. 2020. [Online]. Available: arXiv DPR Paper

[8] S. Robertson and H. Zaragoza, "The probabilistic relevance framework: BM25 and beyond," *Found. Trends Inf. Retr.*, vol. 3, no. 4, pp. 333–389, 2009.

[9] S. Khanuja *et al*., "MuRIL: Multilingual Representations for Indian Languages," Apr. 2021. [Online]. Available: arXiv MuRIL Paper

[10] A. Robatian *et al*., "GEC-RAG: Improving Generative Error Correction via Retrieval-Augmented Generation for Automatic Speech Recognition Systems," Jan. 2025. [Online]. Available: arXiv GEC-RAG Paper

[11] J. Bourne, "CLOCR-C: Context Leveraging OCR Correction with Pre-trained Language Models," Jan. 2025. [Online]. Available: arXiv CLOCR-C Paper

[12] J. Zhang *et al*., "OCR Hinders RAG: Evaluating the Cascading Impact of OCR on Retrieval-Augmented Generation," Aug. 2025. [Online]. Available: arXiv OCR Hinders RAG Paper

[13] C. Rigaud, A. Doucet, M. Coustaty, and J. P. Moreux, "ICDAR 2019 Competition on Post-OCR Text Correction," in *Proc. ICDAR*, Sep. 2019, pp. 1588–1593.

[14] J. Ramirez-Orta *et al*., "Post-OCR Document Correction with Large Ensembles of Character Sequence-to-Sequence Models," Jan. 2022. [Online]. Available: arXiv Post-OCR Correction Paper

[15] P. Tran *et al*., "Low-Resource Heuristics for Bahnaric Optical Character Recognition Improvement," Jan. 2026. [Online]. Available: arXiv Bahnaric OCR Paper

[16] M. Douze *et al*., "THE FAISS LIBRARY," *IEEE Trans. Big Data*, 2025, doi: 10.1109/TBDATA.2025.3618474.

[17] M. Danish, H. Liu, and S. Alshmrany, "A Comparative Study of mBERT and IndicBERT for Natural Language Processing in Indic Languages," in *Proc. IEEE*, Jan. 2026, pp. 1–6.

[18] G. Kim *et al*., "OCR-free Document Understanding Transformer," Oct. 2022. [Online]. Available: arXiv OCR-free Document Understanding Transformer