AI-Based Hindu Shloka Understanding and Context-Aware Meaning Generation Using Natural Language Processing and Large Language Models

Pooja Yagnik
Department of Software Engineering
Gandhinagar University, Kalol, Gujarat, India

Abstract — Sanskrit shlokas (verses) are compact, metrically constrained texts whose meaning depends on word-level morphology, compound structure, the work in which they occur, the speaker and situation, and the interpretive tradition of the reader. General-purpose large language models (LLMs) can paraphrase well-known verses, yet they frequently flatten polysemy, ignore commentarial context, and produce fluent but unfaithful explanations. This paper presents the design of a hybrid, context-aware system that takes a Sanskrit shloka as input and produces word-level glosses, a literal translation, and an explanatory contextual meaning in English and Hindi. The architecture combines (i) Devanagari/IAST normalisation and metre identification, (ii) a linguistic analysis layer for sandhi splitting, compound analysis and morphological tagging, (iii) retrieval-augmented generation (RAG) over an indexed corpus of verses and traditional commentaries, and (iv) a prompted or parameter-efficiently fine-tuned LLM with source-grounded output constraints. An evaluation protocol is specified that pairs automatic metrics (BLEU, chrF, BERTScore, COMET) with expert human assessment of faithfulness, fluency, contextual appropriateness and hallucination, together with ablations that isolate the contribution of each component. The paper contributes a reproducible methodology, an error taxonomy for shloka interpretation, and a discussion of the responsible handling of sacred texts, in which multiple legitimate interpretations are surfaced rather than collapsed into a single answer.

Keywords: Sanskrit NLP, Shloka Interpretation, Large Language Models, Retrieval-Augmented Generation, Machine Translation, Low-Resource Languages, Hallucination, Indian Knowledge Systems

I. Introduction

A. Motivation

Hindu scriptural and literary traditions preserve an enormous body of verse in Sanskrit, from the Vedas and Upanishads to the Mahabharata, the Ramayana, the Puranas and the stotra literature. For most contemporary readers, including many native Hindi speakers, these verses are not directly accessible. Readers depend on printed translations and commentaries, which vary in quality and are rarely searchable at the level of an individual verse and its context. An automated system that explains a shloka accurately, in the reader's own language, and with clear grounding in tradition would support education, digital humanities and cultural preservation.

B. Problem Statement

Understanding a shloka is not equivalent to translating a sentence. Three properties make the task difficult:

1) Linguistic complexity. Sanskrit is highly inflected, relies on sandhi (phonological fusion at word boundaries) and samasa (compounding), and permits flexible word order because grammatical relations are marked morphologically.
2) Semantic and contextual dependence. A single term such as dharma, yoga or brahman carries different senses in different works and schools. The correct reading often depends on the chapter, the interlocutor and the commentator followed.
3) Data scarcity. Sanskrit is a low-resource language for neural methods. Parallel corpora are small compared with those of modern languages, and expert-annotated resources are limited.

General large language models (LLMs), though exposed to some Sanskrit in pre-training, tend to produce confident paraphrases that may omit grammatical detail, misattribute verses, or blend interpretations from different traditions.

C. Research Questions

- RQ1. To what extent does explicit morphological and compound analysis improve the faithfulness of LLM-generated shloka translations?
- RQ2. Does retrieval of source context and traditional commentary reduce hallucination and improve contextual appropriateness?
- RQ3. How does output quality differ between English and Hindi, and between zero-shot prompting and lightweight fine-tuning?
- RQ4. What categories of error remain, and how should a system present interpretive plurality?

D. Contributions

- A modular architecture for context-aware shloka understanding with Sanskrit input and English and Hindi output.
- A multi-dimensional evaluation protocol combining automatic metrics with expert judgement.
- An ablation design that isolates linguistic analysis, retrieval and fine-tuning.
- An error taxonomy and guidelines for responsible deployment on religious texts.

The rest of the paper is organised as follows. Section II reviews related work, Section III describes the data, Section IV presents the proposed methodology, Section V details the experimental design, Section VI gives the analysis framework, Section VII discusses limitations and ethics, and Section VIII concludes.

II. Literature Review

A. Computational Analysis of Sanskrit

Sanskrit computational linguistics has a long history of rule-based and lexicon-driven tools. The Sanskrit Heritage Platform [1] provides a finite-state morphological analyser and a sandhi-aware segmenter that returns the set of possible

All rights reserved by www.ijsrd.com

analyses of an input string. The Digital Corpus of Sanskrit (DCS), developed by Oliver Hellwig, supplies lemmatised and morphologically annotated text widely used to train taggers and segmenters. The Classical Language Toolkit [2] offers pipelines for several historical languages, including Sanskrit. Neural approaches to word segmentation and dependency parsing have improved coverage, but ambiguity in sandhi resolution remains an open issue.

B. Neural Machine Translation for Indic and Low-Resource Languages

Transformer architectures [3] underpin current translation systems. Multilingual pre-trained models such as mT5 [4] and the byte-level ByT5 [5] offer transfer to languages with little data, and IndicTrans2 [6] targets translation among Indian languages. For Sanskrit specifically, the Itihasa corpus [7] provides roughly 93,000 Sanskrit-English pairs from the Ramayana and the Mahabharata and established baselines for translating epic verse. Translating verse remains considerably harder than prose, in part because of metrical compression and archaic vocabulary.

C. Large Language Models and Retrieval-Augmented Generation

Large pre-trained language models [8], [9] show strong few-shot ability but are prone to hallucination, broadly defined as fluent output not supported by the source or by reliable knowledge [10]. Retrieval-augmented generation [11] conditions generation on passages retrieved from an external index, which can improve factual grounding and allow provenance to be shown. Low-rank adaptation (LoRA) [12] enables adaptation of large models at modest computational cost, which is attractive for domain specialisation with limited data.

D. Evaluation of Generated Text

BLEU [13] and chrF [14] measure n-gram and character n-gram overlap with references. Embedding-based metrics such as BERTScore [15] and learned metrics such as COMET [16] correlate better with human judgement in many settings. For interpretive text with several acceptable renderings, overlap metrics under-credit valid paraphrase, which motivates expert human evaluation.

E. Research Gap

Existing work addresses Sanskrit analysis, Sanskrit-English translation and general retrieval-augmented generation largely in isolation. Few studies combine explicit grammatical analysis with commentary-grounded retrieval to generate explanatory, context-aware meaning in both English and Hindi, and few evaluate faithfulness and hallucination for sacred-text interpretation with domain experts. This paper addresses that gap.

III. DATA AND RESOURCES

A. Text Sources

The study uses openly licensed, digitised texts. Candidate sources include the Bhagavad Gita (as part of the Mahabharata), the Ramayana, selected principal Upanishads, and well-known subhashita and stotra verses. Each verse is stored with metadata: work, chapter and verse number, speaker, and metre where known.

B. Parallel and Annotated Corpora

- Translation pairs: Itihasa (Sanskrit-English) for training and benchmarking, and public-domain or openly licensed Hindi renderings where available.
- Linguistic annotation: DCS for morphology and lemmas, used to train or validate the analysis layer.
- Commentary corpus: machine-readable, rights-cleared commentaries segmented by verse, used as the retrieval index.

C. Gold Evaluation Set

A held-out set of 300 to 500 verses is selected to span genres (narrative, philosophical, devotional, ethical), metres and difficulty. For each verse, at least two qualified annotators produce reference English and Hindi renderings and mark the sense of key polysemous terms. Disagreements are recorded rather than forced into consensus, so that interpretive plurality is preserved.

D. Licensing and Preprocessing

All sources are checked for licence compatibility. Text is Unicode-normalised, transliterated between Devanagari and IAST with a standard scheme, and de-duplicated. Verses in the evaluation set are excluded from any training or retrieval index to prevent leakage.

IV. PROPOSED METHODOLOGY

A. Overview

The system is a four-stage pipeline: (1) input processing, (2) linguistic analysis, (3) context retrieval and (4) grounded generation, followed by a presentation layer that returns structured output.

B. Stage 1: Input Processing

Input may be Devanagari or IAST. The text is normalised, verse boundaries are detected, and a metre classifier identifies common metres such as anushtubh and trishtubh using syllable-weight (laghu/guru) patterns. Metre serves both as a validation signal for the transcription and as context for later stages.

C. Stage 2: Linguistic Analysis

The analysis layer produces, for each verse: (i) sandhi splitting into candidate word sequences; (ii) morphological tagging of case, number, gender, person, tense, mood and voice; (iii) compound analysis, including classification (for example tatpurusha, dvandva, bahuvrihi) and bracketing; and (iv) lemmatisation with lexicon lookup giving candidate senses per lemma.

Because sandhi resolution is ambiguous, the system retains the top-k analyses with scores and passes them to the generator rather than committing early. This design choice reduces error propagation.

D. Stage 3: Context Retrieval

The retrieval index contains verse-level entries, each linked to its work, chapter, neighbouring verses and available commentaries. A hybrid retriever combines lexical matching

All rights reserved by www.ijsrd.com

on lemmas with dense embeddings from a multilingual encoder. Given the input verse, it returns the exact verse record if present (to recover source and speaker), neighbouring verses for discourse context, and the most relevant commentary passages. Where several commentarial traditions disagree, passages from each are retained and labelled.

E. Stage 4: Grounded Generation

An LLM receives a structured prompt containing the verse, the ranked linguistic analyses, the retrieved context and explicit instructions. It returns output in a fixed schema: a word-by-word gloss (anvaya) aligned to the analysed tokens; a literal translation in English and Hindi; a contextual meaning that situates the verse in its passage and notes distinct traditional interpretations with attribution; and a confidence indicator with the identifiers of the retrieved passages used.

Two configurations are studied: prompting a general-purpose LLM with few-shot examples, and LoRA fine-tuning of an open-weight model on verse-gloss-translation triples. Requiring the output to cite retrieved passages, and instructing the model to state uncertainty when context is insufficient, are used as hallucination mitigations.

F. Presentation Layer

The output shows the Sanskrit text, transliteration, gloss, translations and contextual explanation with source references. Where interpretations diverge, they are shown side by side with attribution rather than merged.

V. Experimental Design

A. Systems Compared

- B1, zero-shot LLM: a general-purpose LLM with a simple translation prompt.
- B2, few-shot LLM: the same model with curated examples.
- B3, sequence-to-sequence baseline: a fine-tuned mT5/ByT5 or IndicTrans2-based model.
- S1, LLM with linguistic analysis: adds the Stage 2 output to the prompt.
- S2, LLM with retrieval: adds the Stage 3 context without Stage 2.
- S3, full pipeline (prompted): Stages 1 to 4 with a prompted LLM.
- S4, full pipeline (LoRA): Stages 1 to 4 with a fine-tuned open-weight model.

B. Ablation Plan

Comparing B2 with S1 isolates the effect of linguistic analysis (RQ1). Comparing B2 with S2 isolates retrieval (RQ2). Comparing S3 with S4 measures the effect of fine-tuning (RQ3). Removing commentary passages while keeping neighbouring verses separates discourse context from commentarial context. Each comparison is run for both English and Hindi outputs.

C. Automatic Metrics

BLEU, chrF, BERTScore and COMET are computed against the reference renderings. Because multiple valid renderings exist, scores are computed against each annotator's reference and the maximum and mean are reported. Significance is assessed with paired bootstrap resampling.

D. Human Evaluation

Qualified evaluators (Sanskrit scholars and bilingual domain experts) rate a stratified random sample of outputs, blind to system identity, on five-point scales for the following dimensions:

- Faithfulness: does the output convey what the verse says?
- Fluency: is the English or Hindi natural and grammatical?
- Contextual appropriateness: is the explanation consistent with the passage and tradition?
- Gloss accuracy: are the word-level meanings correct?
- Hallucination: are there claims unsupported by the verse or retrieved sources (binary, with span marking)?

Inter-annotator agreement is reported using Krippendorff's alpha or weighted Cohen's kappa. Evaluator qualifications and instructions are documented.

E. Error Taxonomy

Errors are annotated into eight categories: incorrect sandhi or compound resolution; wrong lexical sense; wrong grammatical relation (for example subject-object confusion); omission; unsupported addition; misattributed source or speaker; conflation of distinct interpretive traditions; and fluency-only errors.

F. Implementation and Reproducibility

The implementation uses Python with standard NLP libraries. Prompts, random seeds, model versions, retrieval settings and hyperparameters are recorded, and code and non-restricted data splits are to be released. Hardware and compute budget are reported, with LoRA used to keep training feasible on a single GPU.

VI. Results and Analysis

Note to the author: this section describes how results will be reported. Insert only figures from experiments you have actually run.

A. Reporting of Results

For each system (B1 to B3 and S1 to S4), the automatic metric scores (BLEU, chrF, BERTScore and COMET) will be reported separately for English and Hindi output on the gold set. Mean human ratings for faithfulness, fluency, contextual appropriateness and gloss accuracy, and the hallucination rate as a percentage of outputs, will be reported for the principal systems. Results will be presented in running text or figures, with confidence intervals from bootstrap resampling.

B. Hypotheses to Test

- H1. Adding linguistic analysis improves gloss accuracy and faithfulness, particularly for verses with long compounds.
- H2. Retrieval of commentary reduces unsupported additions and improves contextual appropriateness, with the largest gains on philosophical verses.

All rights reserved by www.ijsrd.com

– H3. Overlap metrics under-represent improvements that human evaluators detect, because explanatory output differs lexically from short references.
– H4. Fine-tuning improves format adherence and terminology consistency but may not exceed a strong prompted model on rare constructions.

These are hypotheses derived from the literature and the design, not findings. They should be accepted or rejected only on the basis of measured results.

C. Analysis Plan

Results are broken down by genre, metre, verse length, compound density and degree of polysemy. Qualitative case studies examine representative successes and failures, including verses where traditions legitimately diverge. Error-taxonomy counts are reported per system, and statistical tests are applied to key differences.

VII. Discussion, Limitations and Ethics

A. Interpretation of Findings

The discussion should connect observed gains to specific components, state which research questions were answered, and explain surprising results. If retrieval helps while morphological analysis does not, for instance, consider whether analyser errors propagated or whether the LLM already handles the sampled constructions.

B. Limitations

– Coverage: the corpus covers a subset of works and genres, so results may not generalise to Vedic Sanskrit or technical treatises.
– Reference subjectivity: multiple valid renderings exist, so overlap metrics are imperfect and gold sets reflect annotators' choices.
– Tool errors: sandhi and morphological analysers are imperfect and can mislead the generator.
– Model opacity and drift: proprietary LLMs may change over time, affecting reproducibility; open-weight models mitigate this.
– Scale of human evaluation: expert time limits sample size.

C. Ethical and Cultural Considerations

Sacred texts demand care. The system should present its output as an aid to study rather than authoritative religious doctrine, attribute interpretations to traditions rather than asserting a single reading, avoid generating content that distorts or fabricates scripture, and respect the licensing and cultural ownership of commentaries and translations. Evaluators from relevant traditions should be consulted. To limit misuse, such as fabricating verses to support a claim, input verses should be verified against the index and unmatched text flagged.

VIII. Conclusion and Future Scope

This paper proposed a hybrid architecture that couples classical Sanskrit linguistic analysis with retrieval over source context and commentary and with a grounded LLM generator, producing word-level, literal and contextual meanings in English and Hindi. It also specified an evaluation methodology combining automatic metrics, expert judgement and a fine-grained error taxonomy, with ablations that attribute performance to individual components. Future work includes extending coverage to Vedic and technical Sanskrit, adding other Indian languages as output, incorporating audio recitation and chanting recognition, learning to rank among commentarial interpretations, and building interactive tools that allow scholars to correct and extend the knowledge base.

References

[1] P. Goyal and G. Huet, "Design and analysis of a lean interface for Sanskrit corpus annotation," Journal of Language Modelling, vol. 4, no. 2, 2016.
[2] K. P. Johnson et al., "The Classical Language Toolkit: An NLP framework for pre-modern languages," in Proc. ACL-IJCNLP System Demonstrations, 2021.
[3] A. Vaswani et al., "Attention is all you need," in Advances in Neural Information Processing Systems 30, 2017.
[4] L. Xue et al., "mT5: A massively multilingual pre-trained text-to-text transformer," in Proc. NAACL-HLT, 2021.
[5] L. Xue et al., "ByT5: Towards a token-free future with pre-trained byte-to-byte models," Trans. Assoc. Comput. Linguistics, vol. 10, 2022.
[6] J. Gala et al., "IndicTrans2: Towards high-quality and accessible machine translation models for all 22 scheduled Indian languages," Trans. Machine Learning Research, 2023.
[7] R. Aralikatte, M. de Lhoneux, A. Kunchukuttan, and A. Søgaard, "Itihasa: A large-scale corpus for Sanskrit to English translation," in Proc. 8th Workshop on Asian Translation (WAT), 2021.
[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. NAACL-HLT, 2019.
[9] T. Brown et al., "Language models are few-shot learners," in Advances in Neural Information Processing Systems 33, 2020.
[10] Z. Ji et al., "Survey of hallucination in natural language generation," ACM Computing Surveys, vol. 55, no. 12, 2023.
[11] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in Advances in Neural Information Processing Systems 33, 2020.
[12] E. Hu et al., "LoRA: Low-rank adaptation of large language models," arXiv:2106.09685, 2021.
[13] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in Proc. ACL, 2002.
[14] M. Popović, "chrF: Character n-gram F-score for automatic MT evaluation," in Proc. Tenth Workshop on Statistical Machine Translation, 2015.
[15] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating text generation with BERT," in Proc. ICLR, 2020.
[16] R. Rei, C. Stewart, A. C. Farinha, and A. Lavie, "COMET: A neural framework for MT evaluation," in Proc. EMNLP, 2020.

All rights reserved by www.ijsrd.com