

Comparative Analysis of Rule Based Symbolic Logic and a Large Language Model in Sanskrit Morphological Analysis

Khyati Chaurasia

khyatichaur8909@gmail.com

St. Norbert S. S. School

Research Article

Keywords: Sanskrit morphology, rule-based systems, large language models, word-order robustness, morphological ambiguity

Posted Date: October 14th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11348447/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: The authors declare no competing interests.

Abstract

Sanskrit morphology provides a setting for comparing explicit suffix rules and language-model predictions under changes in word order. This exploratory study evaluates a Python suffix classifier and supplied Gemini prediction files for 100 Bhagavad Gita entries representing 98 distinct verse identifiers. The corpus contains 614 split-token occurrences and 280 selective reference annotations. Natural and scrambled inputs preserve the same token multisets. A scoring audit identified repeated matching and unconditional credit for ambiguous outputs in the original symbolic evaluator. Using one-to-one annotation matching and candidate-set membership, the unchanged symbolic classifier achieved compatibility on 228/280 annotations (81.43%) in each order. The supplied natural and scrambled Gemini CSVs each achieved 242/280 compatible predictions (86.43%), after resolving one scrambled-token transcription mismatch by its input position. Only two of 614 paired Gemini classifications changed; neither affected the annotated score. The comparison shows no decrease in compatibility under scrambling and documents stable compatibility in this recorded comparison, alongside limitations from selective annotations, candidate ambiguity, and incomplete model metadata. The results describe robustness of these outputs under the supplied permutations rather than general word-order invariance or full morphological parsing accuracy.

1 Introduction

Sanskrit is morphologically rich and permits flexible word order. Case endings can provide grammatical information when token positions change, but morphological case, syntactic function, and semantic role are distinct. Syncretic forms can correspond to more than one case, and sentence-level interpretation may require agreement and context. Krishna et al. (2018) demonstrated the importance of structured analysis for Sanskrit segmentation and morphological tagging. The present study operates on supplied split tokens; it does not perform full segmentation or dependency parsing.

Transformer language models process sequences using learned representations (Vaswani et al., 2017). Gemini 1.5 includes multimodal language models with long-context capabilities (Gemini Team, 2024), but its technical report does not establish performance on this study's Sanskrit task. Prompting a model to use suffixes alone is an instruction, not evidence that its internal processing excludes context or previously learned information.

The study compares an explicit suffix classifier with supplied Gemini outputs under natural and scrambled token order. The symbolic heuristic rules are not a complete implementation of Panini grammar. The research questions are whether token classifications change under permutation and whether reference compatibility differs across the paired conditions. Earlier qualitative examples are retained separately from the full-corpus CSV evaluation.

2 Materials and methods

2.1 Corpus and annotations

The repository provides `sanskrit_gold.json` and `scrambled_verses.json`, each with 100 entries. Entries are identified as Bhagavad Gita verses or excerpts and contain `text_raw`, `text_split`, and selected word-label annotations. The files contain 98 distinct verse identifiers: BG_18.61 appears twice with the same raw text, and BG_2.20 appears twice with different excerpt lengths. These entries were retained and reported explicitly rather than silently treated as 100 independent verses.

There are 614 split-token occurrences and 280 annotations distributed across 99 entries. One entry has no annotations. Annotations are selective, so the evaluated subset does not represent complete analysis of every word. The reference labels include case categories with some number information, two Verb labels, and six Predicate labels. The latter categories are not morphological cases. No independent annotation review or inter-annotator agreement is documented in the available repository; these labels are therefore treated as the supplied reference annotations rather than an externally validated gold standard.

2.2 Symbolic classifier

The classifier in `scripts/test.py` assigns labels using an ordered sequence of suffix checks and a small set of lexical exceptions. Verb-like endings are checked first, followed by selected genitive, ablative, plural, instrumental, dative, locative, and nominative patterns. Some rules return candidate sets, including Accusative/Nominative and Dative/Ablative Plural. Unmatched forms return Unknown.

The classifier receives a token string and does not consult adjacent tokens, verse position, or the full sentence. Its output for an unchanged token is consequently invariant to permutation. This design supports a test of structural consistency, not a claim that word order is irrelevant to Sanskrit interpretation or that the classifier performs full grammatical-role assignment.

The existing classifier was not modified for the reported evaluation. In particular, rule priority, lexical exceptions, and suffix handling were preserved so that scoring corrections could be distinguished from model changes.

2.3 Word-order manipulation

For every aligned entry, the natural and scrambled files contain identical token multisets and unchanged reference annotations. Only the order of `text_split` tokens differs. Existing stored permutations were used for the analysis. The manuscript's original procedure describes `Python random.shuffle`, but `scripts/scramble.py` is empty in the inspected repository, and no original random seed is recorded. The available evidence therefore verifies the supplied permutations without establishing reproducibility of their original generation.

Because segmentation is supplied beforehand, this manipulation preserves the written split-token strings. It does not test raw sandhi resolution, missing words, spelling corruption, or damaged

manuscripts.

2.4 Evaluation and scoring audit

The original evaluator normalises word strings by removing selected endings, then compares every matching predicted token with every reference annotation. This can count an annotation more than once when multiple tokens match. It also accepts any of three ambiguous prediction strings without requiring the reference label to belong to the predicted set. Re-executing that logic yielded 256 credited matches out of 290 comparisons (88.28%) in both orders.

The corrected evaluation matches each annotation to one unused, exactly matching token occurrence within its entry. Each reference case is reduced to its first word. Symbolic prediction candidates are separated at slashes; CSV candidates are separated at commas. Number information is not evaluated. The same 280 annotations are used for both systems and both conditions. An annotation is scored once, even if its surface form occurs more than once. Repeated identifiers are distinguished by corpus row position.

A prediction is reference-compatible only when its candidate set contains the reference label. A separate singleton-correct count records predictions with exactly one candidate that matches the reference. Unknown outputs are counted as incompatible. These measures distinguish inclusion of a correct candidate from successful unique disambiguation; neither is full morphological parsing accuracy.

Results are descriptive. No significance claim is made for the unchanged compatibility scores. The numerical difference between systems is reported on the same annotations, but does not establish superior generalisation. Counts retain duplicate corpus entries and mixed reference-label categories.

2.5 Gemini prediction records

The author supplied natural-order and scrambled-order CSVs, each containing 614 predictions across 100 ordered records, with fields `verse_id`, `token_position`, `token`, and `predicted_case`. Natural-order rows match the stored input tokens exactly. The scrambled file matches the stored permutation except for `ra` at BG_14.7, record 96, position 3, where the input token is `ra`. This occurrence is aligned by record and position for scoring; its prediction is unchanged. The original CSV is preserved. Excluding the mismatched annotation gives 241/279 (86.38%); treating it as an unmatched error gives 241/280 (86.07%). The position-aligned primary analysis gives 242/280 (86.43%).

The prediction records are attributed by the author to Gemini; the earlier study description identifies Gemini 1.5 Flash and morphology-focused prompts discouraging translation, context, and prior knowledge. Exact model-version metadata, generation dates, decoding settings, and a complete fixed-prompt session log are unavailable for the supplied CSVs. They are therefore evaluated as recorded predictions rather than a fully specified, repeatable model-generation protocol. Historical eight-example responses include a retrieved prompt and are treated separately as qualitative observations.

2.6 Artificial intelligence assistance

ChatGPT supported code preparation, analysis checks, and editing. Results were computed from saved data, and the author takes responsibility for the manuscript.

3 Results

3.1 Symbolic predictions and coverage

The classifier produced identical token-aligned outputs for the natural and scrambled inputs. Reference-compatible predictions occurred for 228 of the 280 annotated occurrences (81.43%) in each condition. Correct singleton labels accounted for 145 occurrences (51.79% of all annotations), while compatible ambiguous sets accounted for 83 (29.64%). The reference label was absent from the candidate set for 52 occurrences (18.57%).

Of the annotated occurrences, 102 received an ambiguous output and three received Unknown. Thus, 277/280 annotated occurrences received a non-Unknown result (98.93%), but this does not imply that all were correct. Across all 614 split-token occurrences, including unannotated words, 112 returned Unknown and 502 returned another label (81.76%). The original claim of 100% grammatical-label coverage is therefore not supported as a non-Unknown coverage measure.

Table 1
Compatibility on the same 280 reference annotations. The scrambled CSV includes one position-aligned transcription mismatch.

Measure	Natural	Scrambled
Symbolic compatibility	228/280 (81.43%)	228/280 (81.43%)
Gemini compatibility	242/280 (86.43%)	242/280 (86.43%)
Symbolic correct singletons	145/280 (51.79%)	145/280 (51.79%)
Gemini correct singletons	171/280 (61.07%)	171/280 (61.07%)
Gemini compatible ambiguous sets	71/280 (25.36%)	71/280 (25.36%)

3.2 Error patterns

Rule ordering produces concrete errors. For example, the early verb-ending check classifies *vāsāṃsi* as Verb even though the supplied reference is accusative plural. *Manah* is assigned Genitive by an ending rule although its supplied reference is nominative. *Yuktaḥ* receives Ablative because the earlier *-taḥ* check takes precedence over later nominative handling. These examples demonstrate rule-priority conflicts rather than inaccessible internal reasoning.

The supplied references also include Predicate labels that the classifier is not designed to predict. Such disagreements remain counted in the primary 280-item evaluation rather than being excluded retrospectively. Their presence limits interpretation of the score as case-identification performance.

3.3 Gemini predictions and paired comparison

The natural-order CSV contains compatible candidate sets for 242/280 annotations (86.43%), including 171 correct singleton predictions and 71 compatible ambiguous sets. The scrambled CSV yields the same counts after position alignment of the documented transcription mismatch. Both conditions have 38 incompatible predictions. Gemini compatibility exceeds the symbolic result by 14 annotations, or 5.00 percentage points, in this corpus. This is a descriptive difference under candidate-membership scoring.

After matching token occurrences within corresponding records, 612/614 Gemini classifications are unchanged (99.67%). The two changes are `ॐॐॐॐॐॐ` and `ॐॐॐॐॐॐ`, each changing from Nominative, Accusative to Indeclinable in the scrambled CSV. Neither occurrence is among the scored reference annotations, so the paired compatibility difference is zero. The supplied files therefore show no measured decline in compatibility under scrambling. Earlier qualitative records illustrate contextual interpretation, a meta-response, and prompt refinement, but cannot establish the causes of the recorded predictions.

4 Discussion

The symbolic system's principal supported property is token-local order invariance. The unchanged tokens necessarily produce the same outputs under its implemented rules, and the paired files confirm this behaviour. Order invariance is therefore a structural property of the baseline, not evidence that it has learned or reproduced the full grammatical system of Sanskrit.

The scoring audit separates reproducibility from validity. The original approximately 88% result is reproducible under the original evaluator, but its denominator includes repeated matching and its ambiguity rule grants overly broad credit. One-to-one, reference-membership scoring yields 81.43% compatibility without changing the classifier. Reporting both the scoring correction and candidate-set behaviour is more informative than presenting a single accuracy value without its definition.

Ambiguous outputs can be useful when morphology alone does not identify a unique case. However, candidate inclusion must not be confused with disambiguation. Morphological tags also cannot automatically be converted into subjects, objects, agents, or predicates. Future work should use a consistent case inventory, separate number and grammatical-role labels, and independent linguistic review of the reference annotations.

The supplied Gemini CSVs make a full-record comparison possible, but the scored reference subset remains selective. Equal compatibility and nearly identical predictions provide evidence of stability in these files under the stored permutations. They do not demonstrate that Gemini ignores word order, relies exclusively on suffixes, or behaves identically across repeated generations, other prompts, models, or Sanskrit corpora. A recorded null difference is an informative observation, even when it contradicts the initial expectation of a decline.

Publicly available scriptural text may overlap with language-model training material, but no contamination test was performed here. Scrambling is not sufficient to establish or exclude memorisation; contamination requires a separate evaluation design (Cheng et al., 2025). A hybrid approach combining neural analysis with explicit grammatical constraints is a direction for future testing, consistent with neural-symbolic research (Besold et al., 2017). The present study does not demonstrate a hybrid system, improved translation, or reliable restoration of damaged manuscripts.

The corpus is small, selectively annotated, and includes repeated identifiers. Segmentation is supplied, rules may have been developed on the same examples used for evaluation, and no held-out development/test split is documented. The measurements therefore describe this stored corpus and implementation. They should not be interpreted as broad performance estimates for Sanskrit or for all large language models.

5 Conclusion

The symbolic classifier achieved 81.43% reference compatibility and the supplied Gemini CSVs achieved 86.43% in both natural and scrambled order on the same 280 annotations. Symbolic predictions were identical by design; 99.67% of paired Gemini token classifications were unchanged. No decrease in compatibility was observed under scrambling. The study instead documents stable compatibility in one recorded comparison and demonstrates how scoring rules affect interpretation. Broader conclusions require reviewed annotations, confirmed model metadata, fixed prompts, and repeated evaluation on held-out examples.

Declarations

Data and code availability

The corpus and original symbolic classifier are available at <https://github.com/khyatichaur8909-ctrl/sanskrit-logic-project>. Supplementary File 1 (Sanskrit_Supplementary_Materials.zip) contains the corpus snapshot, original classifier, natural-order and scrambled-order prediction CSVs, corrected evaluation script, annotation-level scores, and reproduction instructions. The evaluation script reproduces the reported counts without external Python dependencies. The original prediction contents are preserved; the documented transcription mismatch is aligned by input position during evaluation.

References

1. Besold TR, d'Avila Garcez A, Bader S, Bowman H, Domingos P, Hitzler P, Kuehnberger K-U, Lamb LC, Lowd D, Lima PMV, de Penning L, Pinkas G, Poon H, Zaverucha G (2017) Neural-symbolic learning and reasoning: A survey and interpretation. arXiv. <https://doi.org/10.48550/arXiv.1711.03902>
2. Cheng Y, Chang Y, Wu Y (2025) A survey on data contamination for large language models. arXiv. <https://doi.org/10.48550/arXiv.2502.14425>

3. Gemini Team (2024) Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. <https://doi.org/10.48550/arXiv.2403.05530>. arXiv
4. Krishna A, Santra B, Bandaru SP, Sahu G, Sharma VD, Satuluri P, Goyal P (2018) Free as in free word order: An energy based model for word segmentation and morphological tagging in Sanskrit. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2550–2561. <https://doi.org/10.18653/v1/D18-1276>
5. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. Advances in Neural Information Processing Systems, 30, 5998–6008. <https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html>

Appendix

Appendix A Evaluation specification

Visit corpus entries in stored order and distinguish duplicate verse identifiers by entry position. Match each annotation to one unused exact token occurrence. Obtain symbolic predictions from unchanged detect_case; obtain CSV predictions by entry and token position. Split comma-separated or slash-separated candidates and reduce each case label to its first word. Credit compatibility only when the reference case belongs to the candidate set; credit singleton success only when the matching set has one member. Match the single documented scrambled transcription mismatch by its recorded input position. Retain other unmatched items as errors. Score each of the 280 annotations once. For paired prediction agreement, match identical token strings within the corresponding entry, consuming repeated occurrences in order; use position alignment for the documented mismatch.

Figures

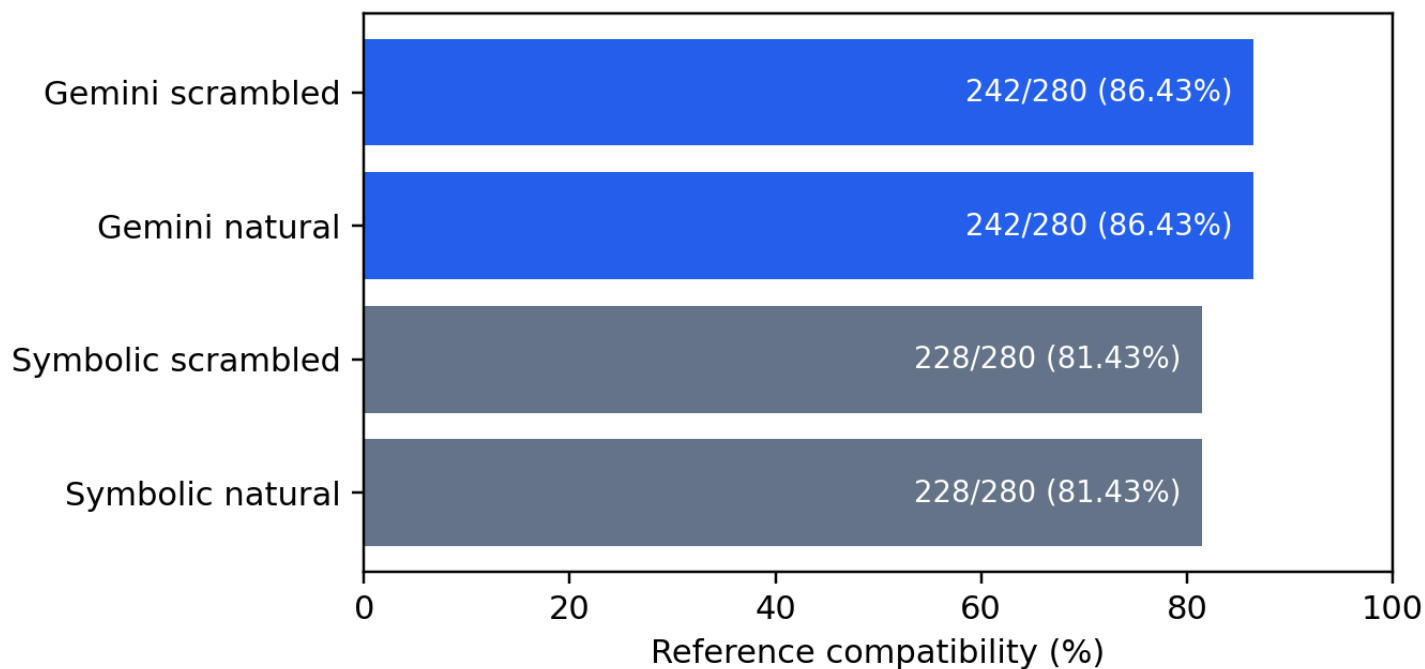


Figure 1

Reference-label compatibility under natural and scrambled Sanskrit token order. The symbolic classifier achieved 228/280 compatible predictions (81.43%) and Gemini achieved 242/280 (86.43%) in each condition. Predictions were credited when their candidate set contained the reference label. One scrambled-token transcription mismatch was aligned by input position. Number and full grammatical-role assignment were not evaluated.