A Case Study of Cross-Lingual Zero-Shot Generalization for Classical Languages in LLMs

V.S.D.S. Mahesh Akavarapu[α], Hrishikesh Terdalkar[β], Pramit Bhattacharyya[γ], Shubhangi Agarwal[β], Vishakha Deulgaonkar[γ], Pralay Manna[γ], Chaitali Dangarikar[γ], Arnab Bhattacharya[γ]

[α]University of Tübingen, [β]University of Lyon 1, [γ]Indian Institute of Technology Kanpur

mahesh.akavarapu@uni-tuebingen.de, hrishikesh.terdalkar@liris.cnrs.fr, arnabb@cse.iitk.ac.in

Abstract

Large Language Models (LLMs) have demonstrated remarkable generalization capabilities across diverse tasks and languages. In this study, we focus on natural language understanding in three classical languages—Sanskrit, Ancient Greek and Latin—to investigate the factors affecting cross-lingual zero-shot generalization. First, we explore named entity recognition and machine translation into English. While LLMs perform equal to or better than fine-tuned baselines on out-of-domain data, smaller models often struggle, especially with niche or abstract entity types. In addition, we concentrate on Sanskrit by presenting a factoid question–answering (QA) dataset and show that incorporating context via retrieval-augmented generation approach significantly boosts performance. In contrast, we observe pronounced performance drops for smaller LLMs across these QA tasks. These results suggest model scale as an important factor influencing cross-lingual generalization. Assuming that models used such as GPT-4o and Llama-3.1 are not instruction fine-tuned on classical languages, our findings provide insights into how LLMs may generalize on these languages and their consequent utility in classical studies.

1 Introduction

Large Language Models (LLMs) (Brown, 2020; Ouyang et al., 2022; Touvron et al., 2023) are known to generalize across various tasks using data from languages present in their pre-training phase, even when not present in instruction tuning (Wang et al., 2022; Muennighoff et al., 2023; Han et al., 2024). Previous work has demonstrated generalization to several low-resource languages via few-shot in-context learning (Cahyawijaya et al., 2024). In this study, we focus on zero-shot generalization to natural language understanding (NLU) tasks for three *classical languages*—Sanskrit (`san`), Ancient Greek (`grc`), and Latin (`lat`)—with a primary focus on Sanskrit. These languages represent a unique category of low-resource languages – despite scarce data for downstream NLU tasks, they have rich ancient literature available in digitized formats (Goyal et al., 2012; Berti, 2019), and they exert significant influence on the vocabulary and narrative structures of better-resourced languages (e.g., Latin contributes approximately 28% of English vocabulary (Finkenstaedt and Wolff, 1973)). Moreover, the high inflection of these languages presents a challenge.

To investigate these issues, we conduct two sets of *zero-shot* experiments using `gpt-4o` (OpenAI, 2024; OpenAI et al., 2023), `llama-3.1-405b-instruct` (Dubey et al., 2024), and their smaller variants. First, we assess performance on two NLU tasks with available datasets for all three languages, namely, named entity recognition (NER) and machine translation to English (MT). We observe that larger models perform comparably or even better than previously reported fine-tuned models on out-of-domain data. Second, we focus on Sanskrit by introducing a factoid closed-book QA dataset and show that question-answering performance improves with retrieval augmented generation (RAG) (Lewis et al., 2020) when the model is provided with relevant texts. The tasks are illustrated in Figure 1.

Given the recent nature of these datasets relative to the models' knowledge cut-off dates, and the likelihood that instruction-tuning on these languages is limited, the robust performance observed can be attributed to cross-lingual generalization. We refer to our prompting strategy as zero-shot, as it includes no task-specific examples, and it is unlikely that such examples in these languages were present in the models' training or instruction-tuning data. In both experimental setups, we find that smaller models struggle, particularly with niche entity types in NER, and in effectively leveraging contextual information via RAG, thereby suggesting model scale as an important factor.

Findings of the Association for Computational Linguistics: ACL 2025, pages 2745–2761
July 27 - August 1, 2025

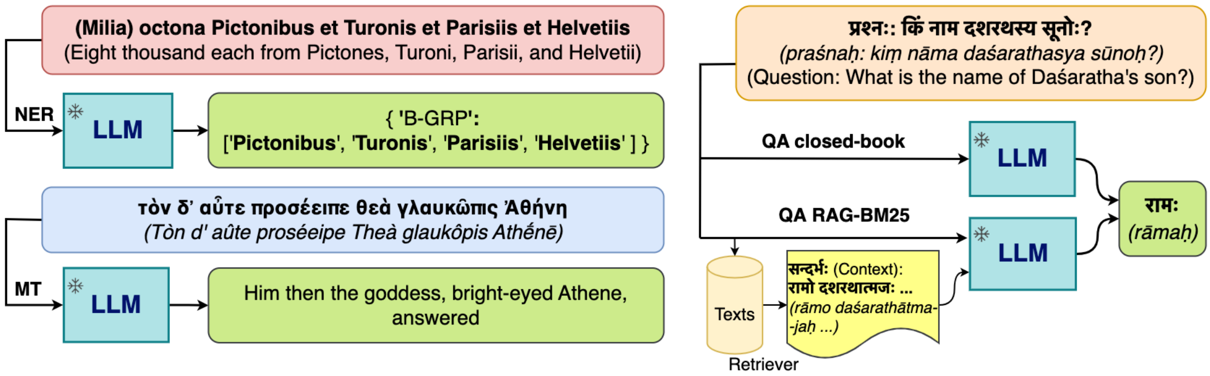

Figure 1: The three NLU tasks evaluated on the classical languages: Named-Entity Recognition (top-left), Machine Translation (bottom-left) and Question-Answering (right).

Task	Language	Source	Size
NER	san	Terdalkar (2023)	139
	lat	Erdmann et al. (2019)	3410
	grc	Myerston (2025)	4957
MT-en	san	Maheshwari et al. (2024)	6464
	lat	Rosenthal (2023)	1014
	grc	Palladino et al. (2023)	274
QA	san	Ours	1501

Table 1: An overview of NLU datasets used in this study for Sanskrit (`san`), Latin (`lat`) and Ancient Greek (`grc`). QA dataset for `san` is contributed in this work. Size represents the number of sentences of test sets (wherever train-test splits exist).

2 Datasets and Methods

The datasets used in our experiments are summarized in Table 1. Our aim is to evaluate zero-shot capabilities where evaluation is done directly on test data without fine-tuning on the training data. Thus, we only consider the test sets of these datasets. Notably, the Sanskrit NER dataset (`san`) is the smallest, comprising 139 sentences (1558 tokens) (Terdalkar, 2023). In addition to these publicly available datasets, we contribute a new factoid closed-domain QA dataset in Sanskrit, described in detail in Section 2.1.

We evaluate the zero-shot capabilities of both large and small variants of our models: proprietary (`gpt-4o` and `gpt-4o-mini` (OpenAI, 2024)) and open-source (`llama-3.1-405b-instruct` and `llama-3.1-8b-instruct` (Dubey et al., 2024)). According to official documentation, these models have knowledge cut-off dates at the end of 2023. Many datasets considered in this work (Table 1) are released beyond these dates, in other words, they are unlikely to be seen in the pre-training data of these models. Given that none of the documentation indicates explicit instruction tuning on Sanskrit, Ancient Greek, or Latin, any observed performance in these languages is likely attributable to cross-lingual generalization. Previous zero-shot applications of LLMs to classical languages have been limited to Latin machine translation and summarization (Volk et al., 2024), although several works have been dedicated to building language models for these languages (Riemenschneider and Frank, 2023; Nehrdich et al., 2024), however, with fine-tuning restricted to morphological parsing-related tasks like dependency parsing (Nehrdich and Hellwig, 2022; Hellwig et al., 2023; Sandhan et al., 2023).

All prompts used for these tasks are provided in Appendix A. The prompts are tested in both English and the respective languages.

2.1 Sanskrit QA

To further evaluate comprehension, we focus on question-answering (QA) in Sanskrit – a domain with very limited datasets. The only existing dataset by Terdalkar and Bhattacharya (2019) comprises 80 kinship-related questions. To address this gap, we created a new dataset containing 1501 factoid QA pairs covering distinct domains in Sanskrit: epics and healthcare. Specifically, we selected two key texts: (1) Rāmāyaṇa, an ancient Indian epic, and (2) Bhāvaprakāśanighaṇṭu, a foundational text on Āyurveda. Details of the dataset are provided in Appendix B.

For QA evaluation, we employ a closed-book setting using prompts detailed in Appendix A.3. To emulate extractive QA, we implement a Retrieval-Augmented Generation (RAG) approach by retrieving the top-k relevant passages (k tuned to

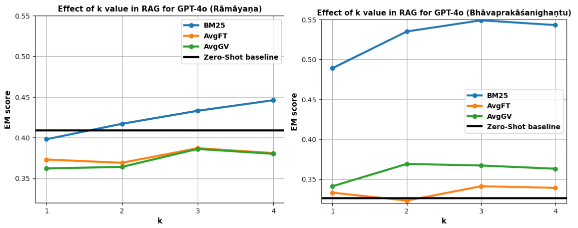

Figure 2: Effect of k on RAG, denoting the number of top best matching text chunks retrieved, on the performances of GPT-4o with retrievers based on BM25, averaged FastText (AvgFT) and GloVe (AvgGV) embeddings respectively of datasets Rāmāyaṇa (left) and Bhāvaprakāśanighaṇṭu (right).

4) from the original Sanskrit texts using BM25 (Sparck Jones, 1972; Robertson et al., 2009). We also compare BM25 with embedding-based retrievers—FastText (Bojanowski et al., 2017) and GloVe (Pennington et al., 2014)—and vary k to assess performance using `gpt-4o` with Sanskrit prompts. As shown in Fig. 2, BM25 consistently outperforms the embedding-based methods, and $k = 4$ emerges as an optimal choice across metrics.

To examine whether the inclusion of answer-bearing contexts benefits model performance, we manually annotated the relevance of retrieved passages. Since BM25 relies on lexical similarity, typically favoring lemmas over inflected forms, we introduce a lemmatization step using a transformer-based Seq2Seq Sanskrit lemmatizer trained on the DCS corpus (Hellwig, 2010-2024), achieving a mean F1 score of 0.94 on a held-out test set. Further details on RAG and lemmatization are provided in Appendix C, and implementation details in Appendix D. Code and data are available at `https://github.com/mahesh-ak/SktQA`.

3 Results

Figure 3 presents our zero-shot evaluation results, demonstrating that larger LLMs exhibit robust cross-lingual generalization across four NLU tasks—named entity recognition (NER), machine translation (MT), closed-book QA, and extractive QA (simulated via RAG-BM25)—in three classical languages (with QA evaluated solely on Sanskrit). To assess variability, each test set is segmented into 10 chunks during evaluation resulting in a box-plot. Larger models perform better than previous fine-tuned models on out-of-domain data as reported in Appendix E. Notably, when answer-bearing contexts are provided (Fig. 3d) versus when they are absent (Fig. 3e), the models show significant performance gains, suggesting comprehension abilities in Sanskrit, a language characterized by high inflection. This behavior is however, not exhibited by smaller models when prompted in Sanskrit.

3.1 Prompt Language: English versus Native

During evaluation, we prompted models both in English and in each target language. As shown in Figure 3, English prompts generally outperform Sanskrit prompts, particularly with smaller models, providing partial evidence that these models have not been instruction-tuned on Sanskrit (Muennighoff et al., 2023). For Latin and Ancient Greek, this English-prompt advantage holds mainly for smaller models; larger models perform equally well, or even better, with native-language prompts (e.g., in Latin NER). This does not imply instruction tuning in these languages, since larger and smaller models likely saw comparable amounts of tuning data. Rather, it suggests that cross-lingual transfer is especially strong for Latin and Ancient Greek in larger models, reflecting their substantial influence on high-resource languages such as English.

3.2 Misclassified Entities in NER

Figure 4 displays confusion matrices for the NER task. Across all three languages, the smaller models exhibit more confusion among semantically related classes (see G for descriptions of entity types), while the larger models show fewer off-diagonal errors. In `san`, mythological entities like Deva, Asura, and Rakshasa or semantically closed entities like Kingdom versus City (e.g., Kośala vs Ayodhyā) or Forest (e.g., Daṇḍaka) versus Garden (e.g., Nandana) often get misclassified with each other in the smaller models. For `lat`, entity type GRP proves troublesome for the smaller models, suggesting struggles in separating individual entities from collective ones. In `grc`, categories such as LOC and ORG show higher confusion counts akin to GRP in `lat` while confusion between God and Persons seems similar to what was discussed for Sanskrit. In contrast, much clearer boundaries emerge in the larger models' confusion matrices. In many of these cases, the domain or style of the texts, especially if they involve mythological or archaic terms typical of classical texts, can be understood to influence performance, as models with less exposure to specialized terminology may conflate related entity types.

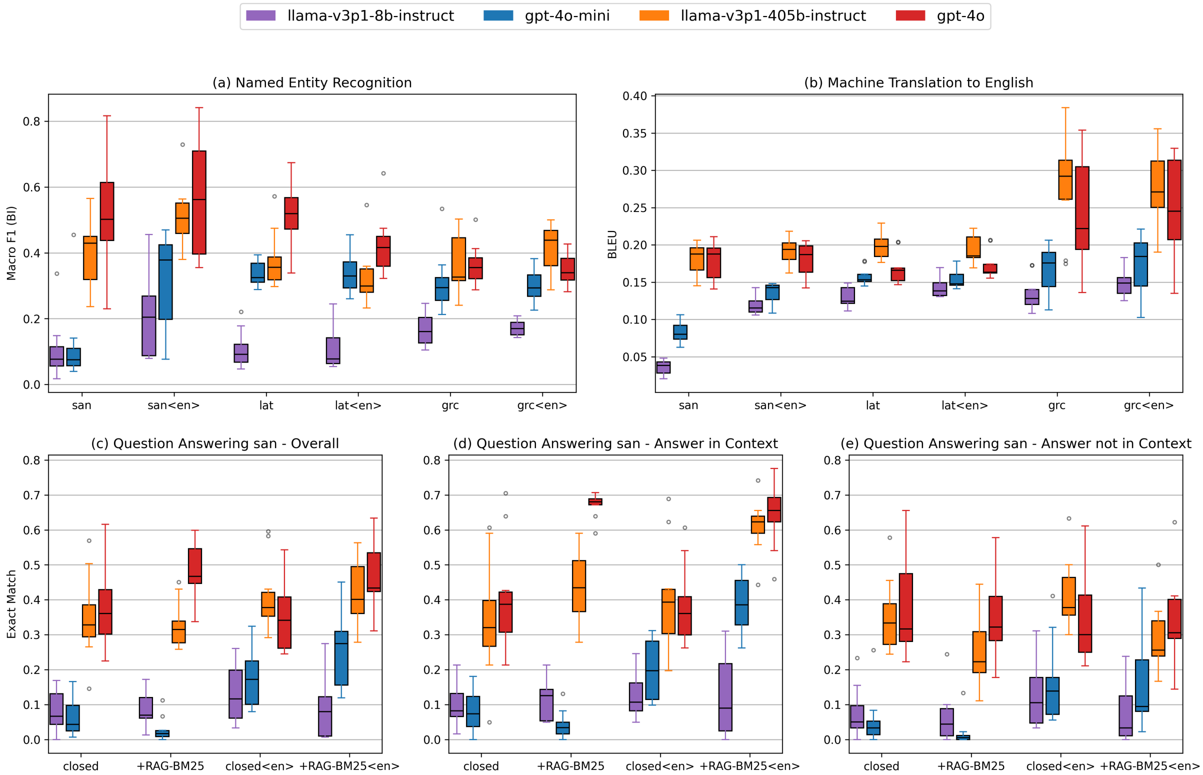

Figure 3: Zero-shot evaluation of LLMs on three NLU tasks for classical languages (language codes in ISO 639-2). Prompts when in English are denoted by <en>, otherwise are in respective languages. Larger LLMs are represented in red and orange, while smaller LLMs in blue and purple. First row displays the performances on NER (a) and MT (to en) (b) for all three languages. Second row displays the performances on QA for Sanskrit alone. Out of 1501 QA pairs considered (c), 607 QA pairs are with answer present in at least one of the $k = 4$ contexts extracted by BM25 and 894 QA pairs with answer not inferable from contexts, which are respectively considered in (d) and (e).

LLM	Closed-book		+ RAG-BM25	
	Inflected	Lemmatized	Inflected	Lemmatized
gpt-4o	0.36	0.37	0.46	0.48
llama-3.1-405b-instruct	0.41	0.40	0.42	0.44
gpt-4o-mini	0.18	0.20	0.25	0.28
llama-3.1-8b-instruct	0.13	0.15	0.09	0.10

Table 2: Comparison of EM scores in san QA (English prompts) when predicted and gold answers are considered with inflection or lemmatized.

LLM	MT (BLEU)		NER (Macro F1-BI)	
	Devanagari	IAST	Devanagari	IAST
gpt-4o	0.179	0.165	0.637	0.599
llama-v3p1-405b-instruct	0.193	0.148	0.561	0.556
gpt-4o-mini	0.135	0.099	0.359	0.318
llama-v3p1-8b-instruct	0.120	0.063	0.164	0.149

Table 3: Comparison of performances in san MT and NER (English prompts) when the input sentences are given Devanagari script or in IAST script.

3.3 Inflection in Answers in Sanskrit QA

In the Sanskrit question-answering task, models are expected to generate single-word answers with the correct inflection. For computing exact match (EM) scores, we manually identified all acceptable answers, excluding those with incorrect inflection (e.g., wrong case or gender endings). To quantify inflection errors, we also calculated EM scores on lemmatized versions of the gold standard and predicted answers, as shown in Table 2. Most models show only a slight increase in EM scores on lemmatized answers, suggesting that inflection errors are relatively minor, a finding corroborated by manual inspection. Future work could extend this analysis to investigate inflection accuracy in full sentence generation within broader natural language generation scenarios.

3.4 Sanskrit Orthography: Devanagari versus IAST

So far, we have shown robust cross-lingual generalization in the models. We now turn to one possible underlying mechanism—orthographic transfer—where models benefit from shared scripts across languages. Prior work has identified orthography as a key factor in cross-lingual transfer for LLMs

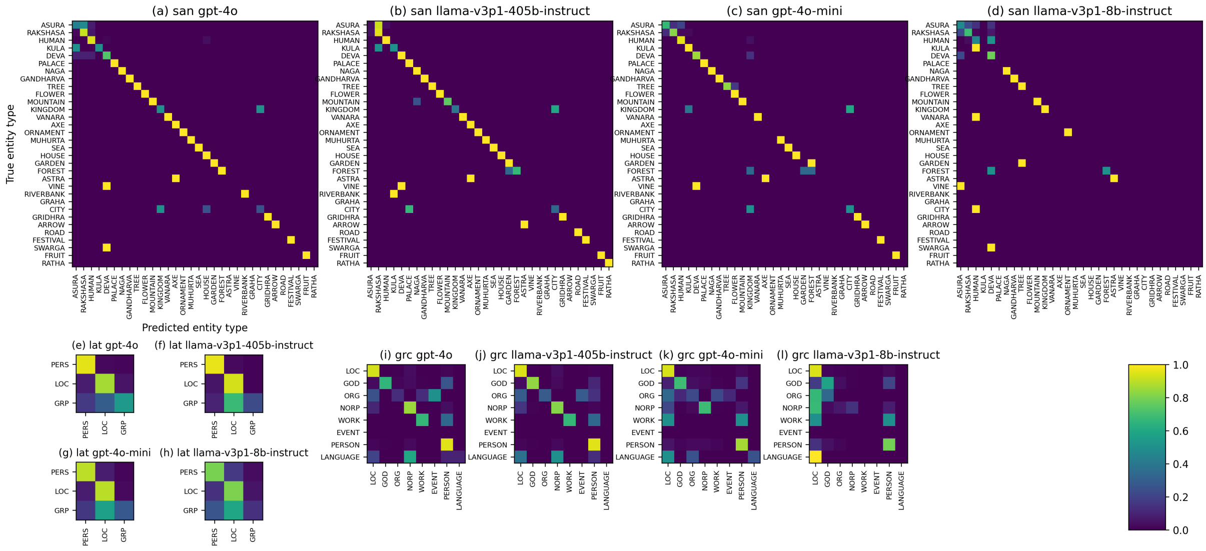

Figure 4: Confusion matrices from the NER task in san (a-d), lat (e-h) and grc (i-l), all with <en> prompts, normalized across rows.

(Muller et al., 2021; Fujinuma et al., 2022). To isolate this effect, we re-ran our Sanskrit NER and MT experiments (using English prompts) in Roman-based IAST transliteration instead of Devanagari. Table 3 compares performance in both scripts. Models perform better with the Devanagari script, which is shared by higher-resource relatives like Hindi and Marathi, reinforcing the importance of script sharing. However, results in IAST are only slightly lower, suggesting that Roman-based transliterations also feature prominently in the pre-training data. In future, we will investigate whether model outputs are consistent across both scripts, that is, whether these LLMs are effectively *digraphic* in Sanskrit.

3.5 Knowledge-Graph Question-Answering

Additionally, we explore the use of knowledge graphs (KGs) for Sanskrit QA. We evaluated a KG derived from the Bhāvaprakāśanighaṇṭu text (Terdalkar et al., 2023) and constructed a small KG for Rāmāyaṇa (details in Appendix F). Using the Think-On-Graph (ToG) paradigm (Sun et al., 2024), which iteratively explores the KG paths for answer retrieval in a training-free zero-shot manner (Xu et al., 2024), we observed that gpt-4o could effectively execute this method. Although it occasionally extracted correct answers, its performance did not significantly exceed that of the closed-book setting, most likely due to the incompleteness of the KGs (see §F.3). Future work may focus on developing more comprehensive KGs to enhance this retrieval method.

4 Conclusions

In summary, our zero-shot evaluations demonstrate that larger language models exhibit robust cross-lingual generalization across diverse natural language understanding tasks in classical languages, including NER, machine translation, and QA. Notably, the significant performance gains achieved when answer-bearing contexts are provided, particularly in Sanskrit QA, suggest comprehension abilities in highly inflected languages. Moreover, our contribution of a novel Sanskrit QA dataset provides a valuable resource for evaluating and benchmarking LLM performance on classical language tasks. Importantly, these models have not been explicitly instruction tuned on Sanskrit, Latin, or Ancient Greek—evidenced by the superior performance achieved when using English prompts for Sanskrit—which indicates that their zero-shot performance is attributable solely to cross-lingual generalization.

Future work will focus on expanding dataset coverage, knowledge graphs and exploring additional classical languages and tasks, further advancing our understanding of cross-lingual generalization in LLMs and its applications in digital humanities and multilingual NLP research.

Acknowledgements

This research is financially supported by the Indian Knowledge Systems (IKS) Division of Ministry of Education, Govt. of India (project number AICTE/IKS/RFP1/2021-22/12). Mahesh Akavarapu received funding from Volkswagen Foundation under the Phylomilia project within the Pioneering Projects funding line. We also thank anonymous reviewers and the Area Chairs for their comments that have helped improve the paper.

Limitations

While our study demonstrates robust cross-lingual generalization in large language models for classical languages, several limitations warrant discussion. First, our newly contributed Sanskrit QA dataset, although valuable, is limited in size. Our evaluation relies exclusively on zero-shot performance, as the models have not been explicitly instruction tuned on these languages; this design choice may obscure potential benefits achievable through targeted fine-tuning. Further, a few datasets we experimented were released within the models' knowledge cut-off dates raising the issue of data contamination. Among these, only Ancient Greek MT exhibits anomalously high performance, suggesting possible exposure. In general, NER, owing to its structural data should be less susceptible to contamination than MT. Furthermore, the effectiveness of our BM25-based retrieval approach depends heavily on preprocessing steps such as lemmatization, which might not optimally address all linguistic variations in highly inflected languages. Finally, our comparisons are based on a limited set of proprietary and open-source models, and future work should extend this analysis to a broader range of models and tasks to fully understand the nuances of cross-lingual generalization in classical languages.

Ethics Statement

Classical Sanskrit epics hold deep cultural and religious significance in Indian traditions, and similarly, Āyurveda represents a revered tradition-bound area within healthcare. We acknowledge that any research involving these subjects must be conducted with particular care. It is essential to note that, as with conventional treatment, Āyurvedic practices require professional consultation and should not be substituted by automated responses. Although our experiments indicate that paradigms like RAG produce more grounded and, hence, potentially safer outputs, there is no assurance that the responses from current LLMs in these domains meet clinical or religious safety standards. Consequently, the authors do not endorse using the datasets beyond the scope of linguistic research. These datasets are released for open-source, non-commercial use, and all annotators have been compensated at fair, standard rates.

References

V.S.D.S.Mahesh Akavarapu and Arnab Bhattacharya. 2023. Creation of a digital rig Vedic index (anukramani) for computational linguistic tasks. In *Proceedings of the Computational Sanskrit & Digital Humanities: Selected papers presented at the 18th World Sanskrit Conference*, pages 89–96, Canberra, Australia (Online mode). Association for Computational Linguistics.

AnthropicAI. 2024. Claude-3.5-sonnet.

Marijke Beersmans, Evelien de Graaf, Tim Van de Cruys, and Margherita Fantoli. 2023. Training and evaluation of named entity recognition models for classical Latin. In *Proceedings of the Ancient Language Processing Workshop*, pages 1–12, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Monica Berti. 2019. *Digital classical philology: Ancient Greek and Latin in the digital revolution*, volume 10. Walter de Gruyter GmbH & Co KG.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146.

Tom B Brown. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Samuel Cahyawijaya, Holy Lovenia, and Pascale Fung. 2024. LLMs are few-shot in-context low-resource language learners. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 405–433, Mexico City, Mexico. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Alexander Erdmann, David Joseph Wrisley, Benjamin Allen, Christopher Brown, Sophie Cohen-Bodénès, Micha Elsner, Yukun Feng, Brian Joseph, Béatrice Joyeux-Prunel, and Marie-Catherine de Marneffe. 2019. Practical, efficient, and customizable active learning for named entity recognition in the digital humanities. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2223–2234, Minneapolis, Minnesota. Association for Computational Linguistics.

Thomas Finkenstaedt and Dieter Wolff. 1973. *Ordered profusion: Studies in dictionaries and the English lexicon*. C. Winter.

Yoshinari Fujinuma, Jordan Boyd-Graber, and Katharina Kann. 2022. Match the script, adapt if multilingual: Analyzing the effect of multilingual pretraining on cross-lingual transferability. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1500–1512, Dublin, Ireland. Association for Computational Linguistics.

Google. 2024. Gemini-1.5-pro.

Pawan Goyal, Gérard Huet, Amba Kulkarni, Peter Scharf, and Ralph Bunker. 2012. A distributed platform for Sanskrit processing. In *Proceedings of COLING 2012*, pages 1011–1028, Mumbai, India. The COLING 2012 Organizing Committee.

Janghoon Han, Changho Lee, Joongbo Shin, Stanley Jungkyu Choi, Honglak Lee, and Kyunghoon Bae. 2024. Deep exploration of cross-lingual zero-shot generalization in instruction tuning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 15436–15452, Bangkok, Thailand. Association for Computational Linguistics.

Oliver Hellwig. 2010-2024. Dcs - the digitial corpus of sanskrit.

Oliver Hellwig, Sebastian Nehrdich, and Sven Sellmer. 2023. Data-driven dependency parsing of vedic sanskrit. *Language Resources and Evaluation*, 57(3):1173–1206.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. 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. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Ayush Maheshwari, Ashim Gupta, Amrith Krishna, Atul Kumar Singh, Ganesh Ramakrishnan, Anil Kumar Gourishetty, and Jitin Singla. 2024. Samayik: A benchmark and dataset for English-Sanskrit translation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 14298–14304, Torino, Italia. ELRA and ICCL.

Christopher D Manning. 2008. Introduction to information retrieval.

I. Dan Melamed, Ryan Green, and Joseph P. Turian. 2003. Precision and recall of machine translation. In *Companion Volume of the Proceedings of HLT-NAACL 2003 - Short Papers*, pages 61–63.

MistralAI. 2024. Mistra-large-2.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15991–16111, Toronto, Canada. Association for Computational Linguistics.

Benjamin Muller, Antonios Anastasopoulos, Benoît Sagot, and Djamé Seddah. 2021. When being unseen from mBERT is just the beginning: Handling new languages with multilingual language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 448–462, Online. Association for Computational Linguistics.

Jacobo Myerston. 2025. NEReus: A named entity corpus of ancient greek. `https://github.com/jmyerston/NEReus`. [Online; accessed 01-Feb-2025].

Sebastian Nehrdich and Oliver Hellwig. 2022. Accurate dependency parsing and tagging of Latin. In *Proceedings of the Second Workshop on Language Technologies for Historical and Ancient Languages*, pages 20–25, Marseille, France. European Language Resources Association.

Sebastian Nehrdich, Oliver Hellwig, and Kurt Keutzer. 2024. One model is all you need: ByT5-Sanskrit, a unified model for Sanskrit NLP tasks. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 13742–13751, Miami, Florida, USA. Association for Computational Linguistics.

OpenAI. 2024. Gpt-4o.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Chiara Palladino, Farnoosh Shamsian, Tariq Yousef, David J. Wright, Anise d'Orange Ferreira, and Michel Ferreira dos Reis. 2023. Translation alignment for ancient greek: Annotation guidelines and gold standards. *Journal of Open Humanities Data*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Rekha Phull and Gaurav Phull. 2017. *Ayurveda Amrtam: MCQs on Laghutrayi & Medical Research in Ayurveda*. Chaukhamba Surabharati Prakashana.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Ramkumar Rai. 1965. *Valmiki-Ramayana Kosha: Descriptive Index to the Names and Subjects of Ramayana*. Chowkhamba Sanskrit Series Office.

Manmatha Natha Ray. 1984. *An Index to the Proper Names Occuring in Valmiki's Ramayana*. The Princess of Wales Sarasvati Bhavana studies: Reprint series. Sampurnanand Sanskrit University.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Frederick Riemenschneider and Anette Frank. 2023. Exploring large language models for classical philology. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15181–15199, Toronto, Canada. Association for Computational Linguistics.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Gill Rosenthal. 2023. Machina cognoscens: Neural machine translation for latin, a case-marked free-order language. Master's thesis, University of Chicago.

Siba Sankar Sahu and Sukomal Pal. 2023. Building a text retrieval system for the sanskrit language: Exploring indexing, stemming, and searching issues. *Computer Speech & Language*, 81:101518.

Jivnesh Sandhan, Laxmidhar Behera, and Pawan Goyal. 2023. Systematic investigation of strategies tailored for low-resource settings for low-resource dependency parsing. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2164–2171, Dubrovnik, Croatia. Association for Computational Linguistics.

Rajendra Pratap Singh. 2009. *1000 Ramayana Prashnottari*. Prabhat Prakashan.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel Ni, Heung-Yeung Shum, and Jian Guo. 2024. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *The Twelfth International Conference on Learning Representations*.

Hrishikesh Terdalkar. 2023. *Sanskrit Knowledge-based Systems: Annotation and Computational Tools*. Ph.D. thesis, Indian Institute of Technology Kanpur.

Hrishikesh Terdalkar and Arnab Bhattacharya. 2019. Framework for question-answering in Sanskrit through automated construction of knowledge graphs. In *Proceedings of the 6th International Sanskrit Computational Linguistics Symposium*, pages 97–116, IIT Kharagpur, India. Association for Computational Linguistics.

Hrishikesh Terdalkar and Arnab Bhattacharya. 2021. Sangrahaka: A tool for annotating and querying knowledge graphs. In *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ESEC/FSE 2021, pages 1520–1524, New York, NY, USA. Association for Computing Machinery.

Hrishikesh Terdalkar, Arnab Bhattacharya, Madhulika Dubey, S Ramamurthy, and Bhavna Naneria Singh. 2023. Semantic annotation and querying framework based on semi-structured ayurvedic text. In *Proceedings of the Computational Sanskrit & Digital Humanities: Selected papers presented at the 18th World Sanskrit Conference*, pages 155–173, Canberra, Australia (Online mode). Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Martin Volk, Dominic Philipp Fischer, Lukas Fischer, Patricia Scheurer, and Phillip Benjamin Ströbel. 2024. LLM-based machine translation and summarization for Latin. In *Proceedings of the Third Workshop on Language Technologies for Historical and Ancient Languages (LT4HALA) @ LREC-COLING-2024*, pages 122–128, Torino, Italia. ELRA and ICCL.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. 2022. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yao Xu, Shizhu He, Jiabei Chen, Zihao Wang, Yangqiu Song, Hanghang Tong, Guang Liu, Jun Zhao, and Kang Liu. 2024. Generate-on-graph: Treat LLM as both agent and KG for incomplete knowledge graph question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18410–18430, Miami, Florida, USA. Association for Computational Linguistics.

Appendix

A Prompts

The Sanskrit prompts are in Devanagari script. In this appendix, we provide these prompts transliterated in IAST scheme.

A.1 Prompts for Named Entity Recognition

Prompt in English

Recognize the named entities from the following sentence in {LANGUAGE}. The valid tags are {ENTITY TYPES}. Do not provide explanation and do not list out entries of ‘O’. Example:
Sentence: <word_1> <word_2> <word_3> <word_4> <word_5>
Output: {{‘B-<entity1>’: [‘<word_1>’, ‘<word_4>’], ‘B-<entity2>’:[‘<word_5>’]}}
Sentence: {INPUT}
Output:

(The example is never a real sentence and is only provided to specify the output structure. Hence, the evaluations are strictly zero-shot.)

Prompt in Sanskrit

adho datta vākye nāmakṛtāḥ sattvāḥ (named entities) abhijānīhi. tadapi vivṛtam mā kuru, kevalam pṛṣṭa viṣayasya uttaram dehi. api ca ‘O’-sambandhitāni na deyāni.
sattvāḥ eteṣu vargeṣu vartante - {ENTITY TYPES}. udāharaṇāya -
vākyam: <padam_1> <padam_2> <padam_3> <padam_4> <padam_5>
phalitam: {{ ‘B-<sattvaḥ1>’: [‘<padam_1>’, ‘<padam_4>’], ‘B-<sattvaḥ2>’: [‘<padam_5>’]}}
vākyam: {INPUT}
phalitam:

Prompt in Latin

Agnosce nomina propria (named entities) ex hac sententia Latina. Notae validae sunt {ENTITY TYPES}. Explanationem ne praebeas nec elementa ‘O’ elenca. Exemplar:
Sententia: <verbum_1> <verbum_2> <verbum_3> <verbum_4> <verbum_5>
Productus: {{‘B-<entitatem1>’: [‘<verbum_1>’, ‘<verbum_4>’], ‘I-<entitatem1>’: [‘<verbum_2>’], ‘B-<entitatem3>’:[‘<verbum_5>’]}}
Sententia: {INPUT}
Productus:

Prompt in Ancient Greek

’Αναγνώρισον τὰ ’ονόματα (named entities) *’εκ τῆςδε τῆς ‘Ελληνικῆς περιόδου. τὰ ’έγκυρα ε’ίδη ’εστιν* {ENTITY TYPES}.
NORP *σημαίνει ’έθνη* (*οἷον ‘Έλληνες, πέρσαι*), *’εθνωνύμια, καὶ ’άλλας κοινωνικὰς ‘ομάδας* (*οἷον θρησκευτικὰς ‘οργανώσεις*).
Μὴ παρέχου ’εξήγησιν ’εν τῇ ’αποκρίσει μηδὲ τὰ εὶς ‘O’ *’εγγεγραμμένα παρατίθεσο. παράδειγμα*:
πρότασις: <*λέξις*_1> <*λέξις*_2> <*λέξις*_3> <*λέξις*_4> <*λέξις*_5>
παραγωγή: {{‘B-<’*Οντότης*1>’: [‘<*λέξις*_1>’, ‘<*λέξις*_4>’], ‘B-<’*Οντότης*2>’:[’<*λέξις*_5>’]}}
πρότασις: {INPUT}
παραγωγή:

A.2 Prompts for Machine Translation

Prompt in English

Translate the following sentence in {LANGUAGE} into English. Do not give any explanations.

Prompt in Sanskrit

adho datta-saṃskṛta-vākyam āṅgle anuvādaya, tad api vivṛtam mā kuru -

Prompt in Latin

Verte hanc sententiam Latinam in Anglicam. Nullam explicationem praebe.

Prompt in Ancient Greek

Μετάφρασον τὴνδε τὴν 'Ελληνικὴν πρότασιν εἰς τὴν 'Αγγλικήν. Μηδεμίαν 'εξήγησιν παρέχου.

(Sanskrit QA Prompts)

In the following prompts, `TOPIC` is either 'Rāmāyaṇa' or 'Āyurveda'.

A.3 Prompts for Closed-book QA

Prompt in English

Answer the question related to {TOPIC} in the Sanskrit only. Give a single word answer if reasoning is not demanded in the answer. With regards to how-questions, answer in a short phrase, there is no single word restriction.
{QUESTION} {CHOICES}

Prompt in Sanskrit

tvayā saṃskṛta-bhāṣāyām eva vaktavyam. na tu anyāsu bhāṣāsu. adhaḥ {TOPIC}-sambandhe pṛṣṭa-praśnasya pratyuttaraṃ dehi. tadapi ekenaiva padena yadi uttare kāraṇam nāpekṣitam. katham kimartham ityādiṣu ekena laghu vākyena uttaraṃ dehi atra tu eka-pada-niyamaḥ nāsti.
{QUESTION} {CHOICES}

A.4 Prompts for RAG-QA

Prompt in English

Answer the following question related to {TOPIC} in Sanskrit only. Give a single word answer if reasoning is not demanded in the answer. With regards to how-questions, answer in a short phrase. Also take help from the contexts provided. The contexts may not always be relevant."
contexts: {CONTEXTS}
question:{QUESTION} {CHOICES}

Prompt in Sanskrit

tvayā saṃskṛta-bhāṣāyām eva vaktavyam. na tu anyāsu bhāṣāsu. adhaḥ {TOPIC}-sambandhe pṛṣṭa-praśnasya pratyuttaraṃ dehi. tadapi ekenaiva padena, yāvad laghu śakyaṃ tāvad, taṃ punaḥ vivṛtam mā kuru. api ca yathā'vaśyam adhaḥ datta-sandarbhebhyaḥ ekatamāt sahāyyaṃ gṛhāṇa. tattu sarvadā sādhu iti nā'sti pratītiḥ.
sandarbhāḥ: {CONTEXTS}
praśnaḥ: {QUESTION} {CHOICES}

B Question Answering Dataset

In this appendix, we describe the creation of Sanskrit QA dataset.

We referred to two books that contain multiple-choice questions (MCQs) with answers: one comprising 1000 MCQs on Rāmāyaṇa (Singh, 2009), and another featuring a collection of 2600 questions from three prominent texts of Āyurveda (Phull and Phull, 2017). The questions and options in these books are in Hindi language.

We carefully selected a relevant subset of questions from these books, including all 1000 questions from Rāmāyaṇa dataset and 431 from that of Āyurveda. These questions are then translated into Sanskrit with the help of experts in the language who are also familiar with the original Sanskrit texts. Further, we consulted with a specialist in Āyurveda to review and discard incorrect question-answer pairs, as well as to generate 70 new questions based on Bhāvaprakāśanighaṇṭu. Ultimately, the question-answering dataset consists of 1501 questions.

The answers typically agree in grammatical case with the corresponding interrogative of the question. The following is a question-answer pair as an illustration[1]:

Q: *śītala-jalasya*	*pānaṃ*	*kasmin*	*roge*	*niṣiddham*	*asti*?	**A:** *gala-grahe*
Q: cold-water.**gen**	drinking	what.**loc**	disease.**loc**	forbidden	is	**A:** pharyngitis.**loc**

Question: During which condition is the drinking of cold water forbidden? Answer: During pharyngitis.

Most questions in the datasets have a single-word answer except a few including those in the Rāmāyaṇa that fall under the category 'Origins' (Table 4). An example question-answer pair under this category that demands reasoning in the answer:

Q: *rājā-sagareṇa sagaraḥ iti nāma kutaḥ prāptam?*

"How did King Sagara obtain such a name?"

A: *saha tena gareṇaiva jātaḥ sa sagaro 'bhavat*

"He was indeed born along with (*sa-*) the poison (*gara*), thus he became *Sagara*."

For such questions (only about 50), the answers can be paraphrased variously, thereby requiring manual evaluation.

The broad semantic and domain-specific categories of the questions are detailed in Tables 4 and 5.

C Retrieval Augmented Generation

In the RAG paradigm, the LLM is provided with additional context that consists of top-k passages retrieved from the original texts. The texts of Rāmāyaṇa and Bhāvaprakāśanighaṇṭu are obtained from GRETIL[2] and Sanskrit Wikisource[3] respectively. The texts are pre-processed following standard procedures (Manning, 2008), namely, dividing the texts into chunks, followed by lemmatization, and then building a document store. Lemmatization would not have been necessary if retrieval frameworks such as Dense Passage Retrieval (Karpukhin et al., 2020) or a vector space retrieval framework with SentenceBERT embeddings (Reimers and Gurevych, 2019) could be used. However, due to insufficient data in Sanskrit, such models cannot be trained now. Hence, we used BM25 retrieval and vector space retrieval with averaged FastText (AvgFT) (Bojanowski et al., 2017) and GloVe (Pennington et al., 2014) (AvgGV) embeddings, which are employed on lemmatized documents and queries. To achieve this, a lemmatizer for Sanskrit was built as described below.

Sanskrit Lemmatizer

Seq2Seq transformer-based Sanskrit lemmatizer was trained from the words and their respective lemmas present in the DCS corpus (Hellwig, 2010-2024)[4]. During lemmatization, if a word in an input sentence is a compound word or involves Sandhi, the lemmatizer is expected to break the word into sub-words and generate their respective lemmas in the output. For example, if the input sentence is '*haridrāmalakaṃ gṛhṇāti*', the corresponding lemmatized output should be '*haridrā āmalaka gṛh*'. Our lemmatizer achieves a mean F1-score of 0.94 across the sentences from the held-out test set (Appx. D) calculated according to Melamed et al. (2003), however with a significant standard deviation of 0.11. While the accuracy is high, future attempts for improvements should focus on minimizing the variance, which is rarely ever reported although important.

The information retrieval pipelines thus formulated can be considered novel concerning Classical Sanskrit. A known earlier attempt towards building retrieval systems in Sanskrit (Sahu and Pal, 2023) focused on news corpora with much terminology consisting of borrowings from Hindi and even English. As a result, the lemmatizer trained on Classical Sanskrit and thereby, our entire retrieval pipeline may not be appropriate on such corpora and hence are not comparable.

The prompts for RAG are detailed in Appx. A.4.

[1] **gen** - genitive, **loc** - locative

[2] https://gretil.sub.uni-goettingen.de/

[3] https://sa.wikisource.org/wiki/

[4] http://www.sanskrit-linguistics.org/dcs/

Category	Description	#Q
Names	Names of various characters	97
Actions	Who performed certain actions?	47
Origins	Origin of various names	49
Numeric	Questions with numerical answers	79
Quotes	Who said to whom?	31
Boons and Curses	Who endowed boons / curses on whom	31
Weapons	Questions related to various types of weapons	59
Locations	Locations of important events or characters	71
Kinship	Questions pertaining to human kinship relationships	133
Slay	Who slayed whom	49
Kingdoms	Which king ruled which kingdom	27
Incarnations	Who were incarnations of which deities	27
MCQ	Multiple choice questions	140
Miscellaneous	Other questions	196

Table 4: Question Categories for Rāmāyaṇa QA Dataset

Category	Description	#Q
Synonym	Synonyms of substances	174
Type	Variants or types of substances	30
Property	Properties of substances	20
Comparison	Comparison between properties of various substances or their variants	24
Consumption	Related to consumption of medicine including suitability, method, accompaniments etc.	23
Count	Counting types or properties of substances	59
Quantity	Quantity of substances in various procedures or methods	21
Time-Location	Time or location in the context of substances or methods	17
Effect	Effect of substances	15
Treatment	Diseases and treatments	23
Method	Methods of preparation of substances	21
Meta	Related to the verbatim source text, the structure of the text and external references	38
Multi-Concept	About more than one aforementioned concepts	11
Miscellaneous	Miscellaneous concepts	24

Table 5: Question Categories for Āyurveda QA Dataset

Model	**BLEU**
Google Trans (Maheshwari et al., 2024)	13.9
IndicTrans (Maheshwari et al., 2024)	13.1
gpt-4o	16.5
llama-3.1-405b-instruct	**17.1**

MT (san-eng) on *Mann ki Baat* dataset

Model	**Macro F1 (BI)**
LatinBERT1 (Beersmans et al., 2023)	0.54
LatinBERT2 (Beersmans et al., 2023)	0.50
gpt-4o	**0.55**
llama-3.1-405b-instruct	0.36

NER (lat) on *Ars Amatoria* dataset

Table 6: Comparison of out of domain performances of LLMs against previously reported fine-tuned models.

D Implementation

This appendix outlines the implementation details. All LLMs are operated through API calls using LangChain[5]. In case of Llama-3.1, we used API provided by Fireworks AI[6].

The lemmatizer was implemented using HuggingFace transformers (Wolf et al., 2020) upon base model T5 (Raffel et al., 2020) initiated with the model configuration of 4 layers per each encoder and decoder, 4 attention-heads, embedding of size 256, and hidden size of 1024, totaling about 100M parameters. The tokenizer trained by Akavarapu and Bhattacharya (2023) was used[7]. The lemmatizer was trained for 15 epochs on DCS (Hellwig, 2010-2024) data with batch size of 32, that took about 15 hours on NVIDIA RTX 2080 with 11GB graphics memory. There are total 1.04M sentences in the data, that are randomly divided into proportions $0.675 : 0.075 : 0.15$ respectively for training, validation and testing. FastText and GloVe embeddings are trained on lemmas obtained from DCS (Hellwig, 2010-2024) with embedding size 100.

E Supplementary Results

In Table 6, we compare the out-of-domain performance of our evaluated models against previously reported fine-tuned models. For MT (san-eng) on *Mann ki Baat* dataset (Maheshwari et al., 2024), open-source model llama-3.1-405b-instruct outperforms both Google Trans and IndicTrans, while for NER (lat) on Ovid's *Ars Amatoria* dataset (Beersmans et al., 2023), the performance of gpt-4o is better than that of fine-tuned LatinBERT variants. Although fine-tuned models yield superior results on in-domain data, our findings indicate that multilingual LLMs are superior in their zero-shot generalization.

[5] https://www.langchain.com/
[6] https://fireworks.ai/
[7] https://huggingface.co/mahesh27/vedicberta-base

F LLMs with Knowledge Graphs

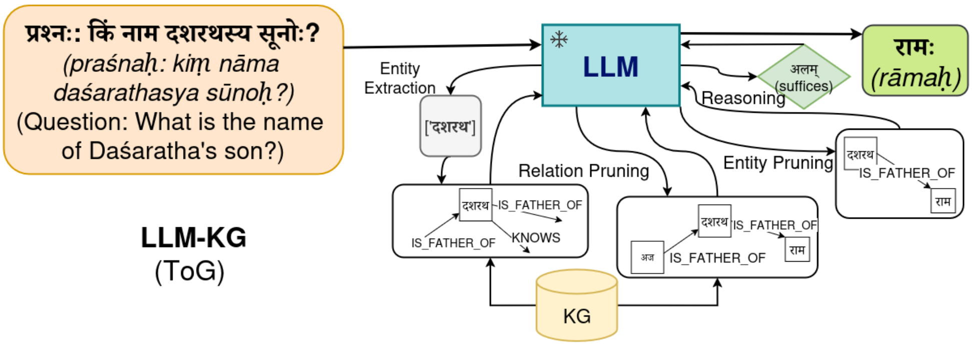

Figure 5: Overview of augmenting a LLM with a knowledge graph (KG) through Think-on-Graph (ToG) paradigm.

Arriving at an answer by an LLM integrated with a knowledge graph (KG) through Think-on-Graph (ToG) (Sun et al., 2024) paradigm involves several prompting steps for each hop from starting entity nodes as illustrated in Fig. 5. Firstly, the LLM lists entities from the input questions further lemmatized by our lemmatizer previously described. The relationships from and to these entities are then extracted by traversing the KG. The LLM then lists relationships with relevance scores, which are further used to prune the relationships, retaining only the best three. Unexplored entities connected by these relationships are then known from the KG, which are similarly pruned to retain the three most relevant ones. The LLM then reasons whether these extracted paths suffice to answer the given question. If no, the cycle is repeated, i.e., it traverses a hop further up to a depth d. Otherwise, the LLM answers using the context from the extracted paths.

The prompts for each step and an outline pseudo-code can be found respectively in Appx. F.2, Alg. 1. Technical terminology such as 'entity', 'knowledge graph', and so forth are mostly retained in English in these prompts resulting in minimal and unavoidable code-mixing. Further, the output of these prompts is often a list of elements and, hence, has to abide by a structured format.

F.1 Knowledge Graphs

A knowledge graph (KG) was constructed for Rāmāyaṇa using two key references, (Ray, 1984) and (Rai, 1965). The graph was annotated with the help of two experts proficient in both Sanskrit and Rāmāyaṇa. For annotation, we used a custom deployment of *Sangrahaka* (Terdalkar and Bhattacharya, 2021). The resulting knowledge graph contains 867 nodes and 944 relations, encompassing entities like characters of the story including humans and divine beings, places (cities, rivers, kingdoms), and animals, and relationships such as kinship, actions, locations, and others, highlighting associations between the characters, natural features, and other elements from the text.

Additionally, a work-in-progress knowledge graph for Bhāvaprakāśanighaṇṭu obtained from the authors of (Terdalkar et al., 2023) was referenced. The KG currently includes 4685 nodes and 10596 relations from 12 out of 23 chapters covering substances such as grains, vegetables, meats, metals, poisons, dairy products, prepared substances and other miscellaneous medicinal substances.

The knowledge graphs were loaded and accessed through Neo4j[8]. Python packaage, `indic-transliteration`[9] is used to move among transliteration schemes of Sanskrit. The pseudo-code for our implementation of ToG (Sun et al., 2024) is given in Algorithm 1. The sample limit S is set to 15, depth limit D to 1 and width limit W to 3.

[8] https://neo4j.com/
[9] https://github.com/indic-transliteration/indic_transliteration_py

Algorithm 1 Outline of LLM-KG i.e., ToG (Sun et al., 2024)

Require: Input: x
LLM prompt-chains: ExtractEntities, RelationPrune, EntityExtractPrune, Reason, Answer
Interface to KG: FetchRelations, FetchEntities; Depth limit: D; Sample limit for KG: N; Width limit for LLM: W
Current Entities $E \leftarrow$ ExtractEntities(x)
Current depth $d \leftarrow 0$
Stored Paths $P \leftarrow$ []
while $d < D$ **do**
  $R \leftarrow$ FetchRelations(E, N)
  $R \leftarrow$ RelationPrune(R, W)
  $E, P \leftarrow$ FetchEntities(E, R, P, N)
  $E, P \leftarrow$ EntityExtractPrune(E, R, P, W)
  if Reason(x, E, P) **then**
    Answer(x, E, P)
    break
  end if
  $d \leftarrow d + 1$
end while
if $d = D$ **then** Answer(x, E, P)
end if

F.2 LLM-KG Prompts

ExtractEntities

system tvam *knowledge-graph*-taḥ uttarāṇi niṣkarṣyituṃ praśnāt *entities* vindasi ca tāni saha *relevance-score* (0-1 madhye) samarpayasi.
output udāharaṇam (‘rāmaḥ’, 0.8), (‘sītā’, 0.7). tato vivṛtaṃ mā kuru.
human praśnaḥ: `{QUESTION}` `{CHOICES}`

RelationPrune

system tvam datta-praśnasya uttarāṇi *knowledge-graph*-taḥ niṣkarṣituṃ *knowledge-graph*-taḥ idānīṃ paryantaṃ niṣkarṣita-sambandhebhyaḥ avaśyāni saha *relevance-score* (0-1 madhye) samarpayasi.
output udāharaṇam (‘IS_FATHER_OF’, 0.8), (‘IS_CROSSED_BY’, 0.7), ... tato vivṛtaṃ mā kuru.
human praśnaḥ: `{QUESTION}` `{CHOICES}`
niṣkarṣitāni sambandhāni: `{RELATIONS}`

EntityExtractPrune

system tvam datta-praśnasya uttarāṇi *knowledge-graph*-taḥ niṣkarṣituṃ *knowledge-graph*-taḥ idānīṃ paryantaṃ niṣkarṣita-sambandhebhyaḥ avaśyāni *nodes (lemmas)* saha *relevance-score* (0-1 madhye) samarpayasi.
output udāharaṇam (‘rāmaḥ’, 0.8), (‘sītā’, 0.7). tato vivṛtaṃ mā kuru.
human praśnaḥ: `{QUESTION}` `{CHOICES}`
niṣkarṣitāni sambandhāni: `{RELATIONS, ENTITIES}`

Reason

system tvam datta-praśnasya uttarāṇi *knowledge-graph*-taḥ niṣkarṣituṃ *knowledge-graph*-taḥ idānīṃ paryantaṃ niṣkarṣitaṃ yat-kiñcid praśnasya uttaraṃ dātuṃ alam (1) vā nālam (0) iti vaktavyam.
output 1 athavā 0. na anyat vadasi
human praśnaḥ: `{QUESTION}` `{CHOICES}`
niṣkarṣitam: `{PATHS}`

Method	gpt-4o	claude-3.5-sonnet	gemini-1.5-pro	mistral-large-2	llama-3.1-405b-instruct
Closed-book	0.381	0.242	0.148	0.333	0.346
RAG-BM25	0.478	**0.521**	0.459	0.434	0.323
LLM-KG	0.381	0.254	-	0.341	-

Table 7: Exact Match (Scores) of various models (including those not part of main experiments) in Sanskrit Question-Answering task (Sanskrit Prompts) with LLM-KG paradigm compared against zero-shot and RAG-BM25 paradigms.

Method	gpt-4o	claude-3.5-sonnet	mistral-large-2
closed-book	0.32	0.21	0.25
LLM-KG	0.34	0.34	0.35

(a)

Method	gpt-4o	claude-3.5-sonnet	mistral-large-2
closed-book	0.40	0.25	0.36
LLM-KG	0.39	0.23	0.34

(b)

Table 8: Comparison of Exact Match (EM) scores between closed-book and LLM-KG paradigms for selected questions when the answer (a) can likely be inferred from KG and (b) cannot be inferred from KG.

Answer

system adhaḥ {TOPIC}-sambandhe pṛṣṭa-praśnasya pratyuttaraṃ dehi. tadapi praśnocitavibhaktau bhavet na tu prātipadika rūpe. tadapi ekenaiva padena yadi uttare kāraṇam nāpekṣitam. katham kimartham ityādiṣu ekena laghu vākyena uttaraṃ dehi atra tu eka-pada-niyamaḥ nāsti.
api ca yathā'vaśyam adhaḥ dattaiḥ *knowledge-graph*-taḥ niṣkarṣita-viṣayaiḥ sahāyyaṃ gṛhāṇa. tattu sarvadā sādhu iti nā'sti pratītiḥ. uttaram yāvad laghu śakyam tāvat laghu bhavet.
human praśnaḥ: {QUESTION} {CHOICES}
niṣkarṣitam: {PATHS}
uttaram:

F.3 LLM-KG Results

The LLM-KG paradigm was evaluated exclusively using Sanskrit prompts on the two QA datasets and included additional models not part of the main experiments—namely, `claude-3.5-sonnet` (AnthropicAI, 2024), `gemini-1.5-pro` (Google, 2024), and `mistral-large-2` (MistralAI, 2024). Table 7 presents the results in comparison with the closed-book and RAG-BM25 paradigms. Overall, performance gains from closed-book to LLM-KG are modest and fall short of the improvements observed with RAG. This may be partly attributed to the complexity of the LLM-KG setup, which requires multi-step prompting and adherence to a structured output format. Notably, models like `gemini-1.5-pro` and `llama-3.1` frequently fail to follow this structured format, rendering them ineffective for running ToG. The strict formatting requirements may also pose challenges for other models, particularly those less adapted to Sanskrit. Interestingly, while `claude-3.5-sonnet` achieves the best results with RAG-BM25, it lags behind `gpt-4o` and `mistral-large-2` in both the closed-book and LLM-KG paradigms.

Table 8 presents a breakdown of performance based on whether the question topics are covered in the current KG—specifically, the *kingdoms* category (27 questions) in the Rāmāyaṇa dataset and the annotated chapters (299 questions) in Bhāvaprakāśanighaṇṭu. For these subsets, which are likely answerable from the KG, LLM-KG shows clear improvements over the closed-book setting, indicating that access to a near-complete KG can significantly enhance performance. In contrast, for questions outside these categories or chapters, no such improvement is observed, reinforcing the hypothesis that KG completeness is crucial for the effectiveness of LLM-KG. Determining domains where knowledge graphs may outperform or be more appropriate than RAG remains an open question for future research.

G Categories for Named Entity Recognition

The categories for NER in Sanskrit, Ancient Greek, and Latin, along with their rough translation and brief explanations, wherever applicable, are provided here.

Entity Type	Translation	Description
Manuṣya	Human	A mortal human being
Deva	Deity	Divine celestial being; god or goddess
Gandharva	∼	Heavenly musician in the service of the gods
Apsaras	∼	Beautiful female spirits known for dance and charm
Yakṣa	∼	Guardian spirit of natural treasures.
Kinnara	∼	Certain Semi-divine beings
Rākṣasa	∼	Malevolent being
Asura	Anti-god	Powerful beings opposed to the gods
Vānara	Monkey-being	Monkey-like humanoid
Bhallūka	Bear-being	Bear or Bear-like humanoid
Gṛdhra	Vulture-being	Vulture-like being
Ṛkṣa	Bear-being	Bear-like humanoid
Garuḍa	Eagle-being	Eagle-like being
Nāga	Serpent-being	Semi-divine serpent race
Svarga	Heaven	Abode of the gods
Naraka	Hell	Realm of punishment after death
Nadī	River	Flowing body of freshwater
Sāgara	Sea	Vast saltwater body
Sarovara	Lake	Large inland water body
Kūpa	Well	Man-made water source
Tīra	Riverbank	Edge or shore of a river
Dvīpa	Island	Land surrounded by water
Parvata	Mountain	Large natural elevation of earth
Nagara	City	Urban settlement or metropolis
Tīrtha	Sacred Place	Holy pilgrimage spot, often near water
Grāma	Village	Small rural settlement
Rājya	Kingdom	Territory ruled by a king
Vana	Forest	Dense growth of trees; wilderness
Udyāna	Garden	Cultivated green space
Marubhūmi	Desert	Dry, arid region
Prāsāda	Palace	Royal residence
Mandira	Temple	Sacred structure for worship
Āśrama	Hermitage	Secluded place for spiritual practice
Gṛha	House	Dwelling or home
Kuṭīra	Hut	Small and simple shelter
Guhā	Cave	Natural underground chamber
Mārga	Road	Pathway or route
Ratha	Chariot	Two- or four-wheeled ancient vehicle
Vimāna	Airborne Vehicle	Flying chariot or aircraft
Khadga	Sword	Bladed weapon
Dhanus	Bow	Weapon for shooting arrows
Bāṇa	Arrow	Projectile shot from a bow
Cakra	Discus	Spinning circular weapon
Gadā	Mace	Blunt weapon, often spiked
Tomara	Javelin	Thrown spear or missile
Śūla	Spear	Long-shafted piercing weapon
Kavaca	Shield	Defensive armor piece
Kañcuka	Armor	Protective body gear
Paraśu	Axe	Bladed tool/weapon
Astra	Divine Weapon	Supernatural weapon, often invoked
Ābharaṇa	Ornament	Decorative jewelry
Śaṅkha	Conch	Sacred spiral shell
Vādya	Musical Instrument	Instrument used in music
Nāṇa	Currency	Form of money or coin
Kula	Clan	Extended family or lineage
Jāti	Species	Species/Socio-economical Group
Gaṇa	Tribe / Group	Assembly or community
Ṛtu	Season	Climatic period of the year
Saṃvatsara	Year	Vedic year cycle
Māsa	Month	Lunar or solar month
Tithi	Lunar Day	Phase in the moon's waxing/waning
Pakṣa	Fortnight	Half of a lunar month
Ayana	Solstice Cycle	Six-month movement of the sun
Yuga	Epoch	Cosmic age or era
Yoga	Astronomical Combination	Planetary conjunction
Karaṇa	Half of Tithi	Subdivision of a lunar day
Muhūrta	Moment / Auspicious Time	Small unit of time (about 48 minutes)
Lagna	Ascendant	Zodiac rising at time of birth
Graha	Planet	Celestial influencer
Nakṣatra	Lunar Mansion	One of 27 lunar constellations
Rāśi	Zodiac Sign	Segment of the zodiac
Dhuma-ketu	Comet	Celestial object with a tail
Utsava	Festival	Celebratory event
Pūjā	Worship	Ritual offering and prayer
Yajña	Vedic Sacrifice	Sacred fire ritual
Upacāra	Ritual Offering	Ceremonial gesture or item
Saṃskāra	Life-Cycle Rite	Hindu ritual of life transition
Aniścita	Undecided	Something that is not yet determined
Vṛkṣa	Tree	Large woody plant
Guccha	Shrub	Small bushy plant
Lata	Vine	Climbing or trailing plant
Puṣpa	Flower	Blossom of a plant
Phala	Fruit	Edible plant product
Patra	Leaf	Green foliage part
Stambha	Stem	Main structural plant part
Tvak	Bark	Outer layer of tree
Mūla	Root	Underground part of plant
Pakṣī	Bird	Feathered flying animal
Sarpa	Snake	Legless reptile

Table 9: Entity types occuring in Sanskrit NER

Entity Type	Description
NORP	Ethnic groups, demonyms, schools
ORG	Organizations
GOD	Supernatural beings
LANGUAGE	Languages and dialects
LOC	Cities, empires, rivers, mountains, and so forth.
PERSON	Individual persons

Table 10: Entity types occuring in Ancient Greek NER (Myerston, 2025). The types without descriptions—EVENT and WORK—have very few occurances in the dataset.

Entity Type	Description
PER	Person
LOC	Locations, places
GRP	Other groups such as tribes

Table 11: Entity types occuring in Latin NER are quite standard types.