Mahānveṣaṇa: Knowledge-Enhanced RAG Framework for Semantic Inquiry into the Mahābhārata

Chirayu Mohanty (251110602)
Roopesh Mangal (251040633)
Shravani Agrawal (242110402)
Srinidhi Balasubramanian (251110613)
Varad Shinde (251110081)

भारतीय प्रौद्योगिकी संस्थान कानपुर
INDIAN INSTITUTE OF TECHNOLOGY KANPUR

CS787 — Generative Artificial Intelligence
Project Report

November 15, 2025

नारायणं नमस्कृत्य नरं चैव नरोत्तमम् ।
देवी सरस्वतीं व्यासं ततो जयमुदीरयेत् ।।

एकतश्चतुरो वेदा भारतं चैतदेकतः ।
पुरा किल सुरैः सर्वैः समेत्य तुलया धृतम् ।।

Abstract

The Mahābhārata, with its complex narrative structure, poetic style, and character reference complexity, presents a challenge for question-answering and verse retrieval. Standard Retrieval-Augmentation Generation models (RAG) often fail/struggle with these challenges. Our work (Mahānveṣaṇa) is a knowledge enhanced RAG framework to address this issue.

We integrated the Mahānāma entity linking dataset to resolve the complexity of entity ambiguity, and by using chapter boundaries, we create a chapter inter-relatedness matrix to calculate entity based scores for every chapter, given a query. Using the notion of contrastive learning, we tried to finetune the best performing pretrained S-BERT model to improve the retrieval which yielded marginal improvements. Finally, we take a weighted sum of the embedding score and entity score to rank all the chapters and cluster the top-k to retrieve the relevant contexts.

We also created a semi-synthetic validation and evaluation dataset of over 2000 query-verse pairs to evaluate the performance of our model. The integration of external knowledge yielded significant increase of 10% in validation phase and 5% in testing phase in the recall scores for open source embedding models.

Contents

Abstract **2**

1 Introduction **4**
1.1 Performance and drawbacks of LLMs and Naïve RAG 4

2 Problem Statement **4**

3 Methodology **5**
3.1 Dataset 5
3.2 Retrieval Architecture 5
3.2.1 Embedding Models 5
3.2.2 Entity Scores 7
3.2.3 Score Fusion 9
3.2.4 Clustering-Based Post-Filtering 9

4 Experiments **9**
4.1 Finetuning 9
4.2 Retrieval 9
4.3 Generation 10

5 Limitations **10**

6 Further Scope **10**

7 Conclusion **10**

A Appendix: Generated answers **12**
A.1 Example 1 12
A.2 Example 2 12

1 Introduction

The Mahābhārata, one of the two epics traditionally known as *Itihāsa* is the world's longest epic poem and holds great relevance and importance within the Indian Knowledge Systems. With over 80,0000 verses, Mahābhārata comprises of a complex web of interwoven narratives philosophical discourses, and poetic references which makes it challenging for standard NLP models to perform well in question-answering or verse-retrievals. Some of the complexities presented by the Mahābhārata:

- **Variability and Ambiguity in Entity names [2]**: Throughout the epic, most entities are referred to by multiple names (variability) and the same name could also refer to different entities (ambiguity) depending on the context. For example, within a few verses, Yudhiṣṭira is referred to as Kaunteya and Ajataśatru whereas the term Kaunteya can refer to any of the three sons of Kunti. Conventional RAG retrieval architectures that rely on semantic embeddings do not capture this complexity.
- **Non-continuous and nested narratives**: Aside from the central plot and story line which is in the form of a conversation between Vaiśaṃpāyana and Janamejaya, Mahābhārata presents hundereds of *Ākhyanas* and *Upākhyānas* (sub-stories) embedded within as conversations and narrations by various characters within the main story line. This nesting can have many layers and hence causing breaks in the narrative. From an implementation point of view, the context needed to answer a query is often not located in a single block of text and the entire context is important for a detailed and accurate answer to the query.
- **Sanskrit non-translatables and poetic devices** Some terms as *Mokṣa*, *Karma*, *Dharma* etc. are considered Sanskrit non-translatables and often queries with such terms do not match accurately with their corresponding english translations.

1.1 Performance and drawbacks of LLMs and Naïve RAG

Due to the above mentioned reasons, LLMs often fail to answer questions or retrieve verses from the Mahābhārata. Preliminary random experiments with ChatGPT (GPT-5 and GPT-4o) reveal the following shortcomings as a result of its inherent design-

- **Hallucination**: This is a major concern and a fundamental challenge for all large language models and recent studies claim that this is an inherent result of the evaluation methods used to train LLMs. As shown in Figure 1, on asking a question to retrieve verses relevant to a particular story, it shows certain verses which seem plausibly true but actually do not exist anywhere in the text!
- **Fixed top-k hyperparameter**: Naïve RAG systems employ the hyperparameter k to retrieve the top k similar chunks to the query (or alternatively, chunks with similarity above a threshold value). However, for practical purposes, the expected retrieved chunks can be lesser or greater than k (alternately, all relevant chunks may not have high similarity values). For example, in Figure 1, three verses are returned whereas the story in the actual texts extends to more than 30 verses.

2 Problem Statement

The objective of this work is to design a RAG based QnA and Verse Retrieval system that can retrieve all original verses that are relevant to a given query in English and subsequently generate a coherent answer based on the above.

For instance, given a query "What was the conversation between Janaka and Sulabha", the system shall return the list of **all** verses, in this case, verses 12.308.1 to 12.308.191 and an answer summarizing or explaining the above.

We formulate the task as follows:

Given a natural language query Q (in English), the task is to retrieve all and only relevant verses (Sanskrit) leveraging auxiliary knowledge sources (translations, entity relationships, speaker information, narrative graphs etc.)

v_i from a large corpus of Sanskrit verses $\mathcal{C} = \{v_1, v_2, \dots, v_M\}$ such that each retrieved verse (or verse block) is semantically and contextually relevant to the query.

Given,

$$Q, \quad \mathcal{C} = \{v_1, v_2, \dots, v_M\}, \quad \text{and} \quad \mathcal{K}$$

Retrieve,

$$\mathcal{V}_R = \{v_1, v_2, \dots, v_n\} \subseteq \mathcal{C}$$

such that

$$\forall v_i \in \mathcal{V}_R,\ v_i \text{ is relevant to } Q, \quad \text{and} \quad \forall v_j \in \mathcal{C} \setminus \mathcal{V}_R,\ v_j \text{ is not relevant to } Q.$$

In other words, the retrieved set $\mathcal{V}_R$ achieves both **completeness** (all relevant verses are extracted) and **purity** (no irrelevant verses are included), with relevance determined jointly from the verse content and the auxiliary knowledge $\mathcal{K}$. Here, N is variable and adaptive, depending on the scope and narrative breadth of the query Q.

3 Methodology

3.1 Dataset

Corpus

We used the Mahabharata dataset from the Itihasa corpus of Sanskrit-English translation pairs extracted from Manmatha Nath Dutt's translations [1] as the base textual corpus from which verses need to be retrieved

For building the entity graph and computing entity scores of verses, we used the Mahānāma [2] - Entity linking dataset.

Validation and Testing dataset

For evaluating our model, we created and used a semi-synthetic dataset consisting of query-verse for validating and testing. We used the Gita Press index in Hindi which consisted of descriptions of each chapter of the Mahābhārata to source chapter descriptions. Next, we mapped each description to the corresponding chapter in the Itihasa dataset. This helped align the summaries with the original source verses and chapters.

To be able to utilise natural-language for queries, the Hindi chapter descriptions were translated into English using Google Translate API. These translated chapter descriptions were used to construct natural language queries using an LLM. This resulted in a dataset with over 2000 query-verse pairs which we split into a 25:75 validation-test set to conduct an evaluation of the retrieval model.

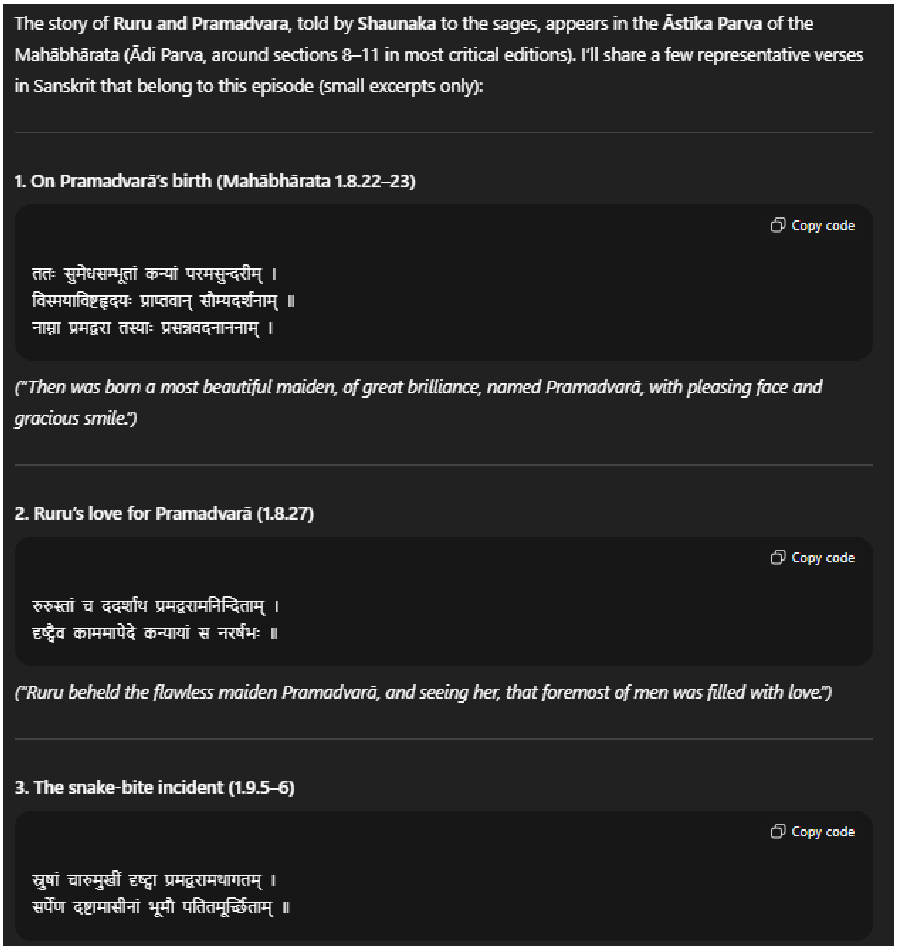
3.2 Retrieval Architecture

The complete retrieval pipeline consists of:

1. Dense retrieval using sentence embeddings.
2. Entity extraction and TF–IDF relevance.
3. Graph-like propagation using chapter similarity.
4. Fusion of semantic and entity scores.
5. Clustering and filtering to refine final context for generation.

3.2.1 Embedding Models

We experimented with open source SBERT models for semantic retrieval of chapters. Initial exploration resulted in MPNET giving the best results. We employed the `multi-qa-mpnet-base-dot-v1` model from the SentenceTransformers library. MPNet (Masked



The story of **Ruru and Pramadvara**, told by **Shaunaka** to the sages, appears in the **Āstīka Parva** of the Mahābhārata (Ādi Parva, around sections 8–11 in most critical editions). I'll share a few representative verses in Sanskrit that belong to this episode (small excerpts only):

1. On Pramadvarā's birth (Mahābhārata 1.8.22–23)

```
ततः सुमेधसम्भूतां कन्यां परमसुन्दरीम् ।
विस्मयाविष्टहृदयः प्राप्तवान् सौम्यदर्शनाम् ॥
नाम्ना प्रमद्वरा तस्याः प्रसन्नवदनाननाम् ।
```

("Then was born a most beautiful maiden, of great brilliance, named Pramadvarā, with pleasing face and gracious smile.")

2. Ruru's love for Pramadvarā (1.8.27)

```
रुरुस्तां च ददर्शाथ प्रमद्वरामनिन्दिताम् ।
दृष्ट्वैव काममापेदे कन्यायां स नरर्षभः ॥
```

("Ruru beheld the flawless maiden Pramadvarā, and seeing her, that foremost of men was filled with love.")

3. The snake-bite incident (1.9.5–6)

```
स्नुषां चारुमुखीं दृष्ट्वा प्रमद्वरामथागतम् ।
सर्पेण दष्टामासीनां भूमौ पतितमूर्च्छिताम् ॥
```

Figure 1: ChatGPT when asked "Can you retrive the exact sanskrit verses from mahabharata for the story of Shaunaka and Ruru"

Parva	Chapters	Words	No of Questions in Test
1	234	215,654	268
2	81	72,262	94
3	315	307,730	223
4	72	55,814	141
5	193	186,620	0
6	123	145,758	32
7	203	257,036	37
8	96	141,814	0
9	65	84,954	60
10	18	19,727	33
11	27	20,488	37
12	365	422,797	180
13	168	260,352	231
14	92	78,827	155
15	39	29,547	32
16	8	7,899	13
17	3	3,005	0
18	6	8,545	0
Total	2108	2,318,829	1536

Table 1: Corpus statistics

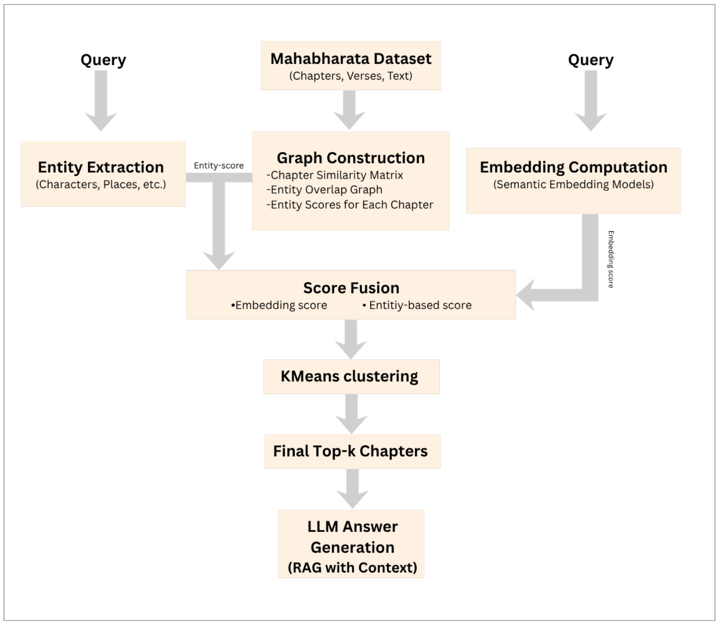


Figure 2: Retrieval methodology

and Permuted Pre-training) [3] improves over earlier encoder models by combining masked language modeling with permuted context prediction, allowing it to capture long-range dependencies and produce high-quality semantic embeddings.

Each chapter was encoded into a 768-dimensional vector, L2-normalized, and indexed using FAISS (IndexFlatIP) for cosine-similarity search. Query embeddings were generated similarly and the similarity scores were computed

We then take this checkpoint to finetune it on the Mahabharata data using contrastive learning.

Further, in order to benchmark our improvements, we also test out using OpenAI's `text-embedding-3-large`.

3.2.2 Entity Scores

A major limitation of standard retrieval models (e.g., TF-IDF or sentence-transformer embeddings) is that they treat chapters as independent text units. However, in the *Mahabharata*, meaning is distributed across chapters — the same entities (characters, locations, events) recur across contexts, forming implicit narrative links. To capture these relationships, we design an **entity-based similarity graph** connecting chapters through shared entities and thematic proximity.

1. Entity Matrix Construction To represent each chapter using its semantic content, we construct an *entity–chapter matrix* and transform it into a TF–IDF weighted representation. The process consists of three stages: raw counting, normalization, and TF–IDF weighting.

a. Raw Entity Count Matrix. For each chapter, we extract all named entities and record their frequencies. This yields a matrix $\mathbf{X} \in \mathbb{R}^{C \times E}$, where C is the number of chapters and E is the number of unique entities. The entry $X_{c,e}$ denotes the number of occurrences of entity e in chapter c.

b. Row Normalization. Since chapters vary in length, raw counts may not be directly comparable. To mitigate this, we convert counts into relative frequencies:

$$\tilde{X}_{c,e} = \frac{X_{c,e}}{\sum_{e'} X_{c,e'}}.$$

This ensures that each row of $\tilde{\mathbf{X}}$ sums to 1, capturing the distribution of entities within each chapter independently of chapter length.

c. TF–IDF Weighting. To emphasize entities that are especially characteristic of a chapter, we apply a TF–IDF transformation. The inverse document frequency (IDF) for each entity is defined as:

$$\mathrm{IDF}(e) = \log\left(\frac{C}{\mathrm{df}(e)}\right),$$

where $\mathrm{df}(e)$ is the number of chapters in which entity e appears. The final TF–IDF matrix is then given by:

$$\mathrm{TF-IDF}_{c,e} = \tilde{X}_{c,e} \cdot \mathrm{IDF}(e).$$

Entities with high TF–IDF values are those that (i) occur frequently in a chapter and (ii) are relatively rare across other chapters. In contrast, entities common throughout the book receive lower weights. The resulting TF–IDF matrix thus provides a discriminative semantic representation of each chapter.

2. Direct relevence score with query We present Algorithm 1 to calculate the relevance of a chapter to a given query.

Algorithm 1 Direct Relevance Computation with Group-Based Adjustment

```
Require: TF–IDF matrix $X_{\text{tfidf}} \in \mathbb{R}^{C\times E}$, query entities $Q = \{e_1, \dots, e_k\}$, query weights $w \in \mathbb{R}^k$
Ensure: Relevance scores $r' \in \mathbb{R}^C$
Let $\mathcal{I} = \mathrm{indices}(Q)$    ▷ entity indices in TF–IDF matrix
Compute direct relevance:
    $r_i = \sum_{j=1}^{k} w_j \cdot X_{\text{tfidf}}[i, \mathcal{I}_j], \quad \forall i \in \{1, \dots, C\}$
Group consecutive entities:
    $G = \{g_1, g_2, \dots, g_m\}$
if $m > 0$ then
    Construct group–chapter mask:
        $M_{i,g} = \begin{cases} 1 & \text{if chapter } i \text{ contains any entity in } g, \\ 0 & \text{otherwise} \end{cases}$
    Compute coverage score:
        $\gamma_i = \frac{1}{m} \sum_{g=1}^{m} M_{i,g}$
    Adjust relevance:
        $r'_i = r_i \cdot \gamma_i$
else
    $r'_i = r_i$
end if
return $r'$
```

3. Chapter Similarity Matrix and Score Propagation To enable graph-like propagation, we first construct a chapter similarity matrix using the TF–IDF entity representations. Let $X_{\text{tfidf}} \in \mathbb{R}^{C\times E}$ denote the matrix where each row corresponds to a chapter and each column corresponds to an entity.

The similarity between two chapters i and j is computed using cosine similarity:

$$\mathrm{sim}(i,j) = \frac{X_{\text{tfidf}}(i) \cdot X_{\text{tfidf}}(j)}{\|X_{\text{tfidf}}(i)\|_2 \, \|X_{\text{tfidf}}(j)\|_2}.$$

Evaluating this for all pairs of chapters yields a $C \times C$ similarity matrix:

$$S_{i,j} = \mathrm{sim}(i, j),$$

which serves as the adjacency matrix of the entity-level similarity graph. Chapters that share many informative entities (i.e., those with high TF–IDF values) obtain larger similarity scores, capturing their semantic proximity in the narrative.

$$p_i = \sum_{i \in \mathrm{TopK}} \mathrm{Rel}(i)\, \mathrm{sim}(j, i)\, \mathbf{1}[\text{chapter } j \text{ contains a query entity}]\,.$$

4. Score Normalization

To normalize scores between 0 and 1:

$$\hat{p}_i = \frac{1}{1 + \exp(-\alpha(p_i - \mu))},$$

where μ is the mean of p_i.

3.2.3 Score Fusion

Embedding similarity and graph relevance are fused via:

$$\mathrm{Final}_i = a \cdot s_i + b \cdot \hat{p}_i,$$

weights were set to $a = 0.9$, $b = 0.1$ after observation from validation data.

3.2.4 Clustering-Based Post-Filtering

We take the top-N chapters under the fused score and run KMeans with $k = 2$ clusters. This removes noisy false positives and tightens the retrieved set.

4 Experiments

4.1 Finetuning

We finetune SBERT using a contrastive learning objective tailored to the narrative structure of the Mahabharata. The cleaned corpus is segmented into overlapping windows, from which we construct around 30k supervised contrastive pairs. Each anchor is paired with one positive example drawn from the same chapter or adjacent narrative region, while multiple negatives—typically 5–15 per anchor—are sampled from semantically distant chapters. This yields a rich contrastive dataset that captures both intra-chapter coherence and inter-chapter divergence.

Training follows the standard SBERT contrastive paradigm, where all samples in a batch implicitly serve as negatives for one another. We employ a contrastive ranking loss that minimizes the distance between anchor–positive pairs while maximizing separation from negatives. Finetuning is performed for 3–5 epochs, with a moderate learning rate, a batch size suited for contrastive objectives, and sequence lengths adjusted for long narrative segments. These choices follow established best practices for domain adaptation of transformer encoders.

Across training, the loss exhibits a stable, monotonic decline, with the most significant improvements occurring in the first two epochs and convergence observed by the fourth. This behavior indicates that the model rapidly learns the high-level semantic structure of the text and subsequently refines fine-grained distinctions between chapters.

4.2 Retrieval

We first experimented with pretrained Sentence BERT - MPNet and evaluated metrics with $top{-}k$ = 3 and 5. We then evaluated our finetuned model, but it yielded only marginal improvements. We discuss the limitations of finetuning in the next section.

We then evaluated both MPNet and Text-Embedding-Large augmenting embedding scores with entity scores, and implementing cluster based filtering. In the former case, we saw a clear improvement in the Recall scores and the latter also saw marginal improvements

Model	Entity scores	Top-k	Prec.	Recall	nDCG	MRR
MPNet (baseline)	No	3	0.1196	0.3587	0.3820	0.3199
MPNet (baseline)	No	5	0.0881	0.4154	0.3281	0.3199
MPNet (fine-tuned)	No	3	0.1261	0.3783	0.3131	0.3286
MPNet (base-line)	Yes	Clustering	**0.1449**	**0.4642**	**0.4621**	**0.3446**
Text-Embedding-Large	No	3	0.2018	0.6055	0.5179	0.5236
Text-Embedding-Large	No	5	0.1375	0.6875	0.5517	0.5236
Text-Embedding-Large	Yes	Clustering	**0.2213**	**0.7051**	**0.6261**	**0.5310**

Table 2: Comparison of retrieval models across multiple evaluation metrics.

4.3 Generation

Due to the lack of ground truth of long form answers, we do not evaluate the final generated answers quantitatively. However a qualitative evaluation of a sample successful and unsuccessful cases are presented in Appendix. GPT-4o was used to generate all answers given the query and retrieved contexts

5 Limitations

While Mahānveṣaṇā presents a novel question-answering method for the Mahābhārata, the improvement in performance after fine-tuning was only marginal. The nature of the data also means that for finetuning SBERT, the similarity of verses makes it hard to create good negative examples. The chapter-level graph may miss connections between individual verses, and chapters with incomplete entity information can make retrieval less accurate. Also, the way relevance is spread across chapters is based on rules instead of being learned from data, so it may not work well for or complicated queries.

Further, it relies heavily on knowledge bases for entity scoring. The reliance on manually annotated external knowledge bases would largely limit the scalability of the model. And given the limited diversity and size of validation data available, the scoring algorithm is largely empirical and heuristic. In order to generalize this algorithm, the parameters will need to be learned automatically from data.

6 Further Scope

While Mahānveṣaṇā showed a marginal improvement on verse retrieval and question-answering for the Mahābhārata, several directions remain open for further enhancement. One is developing other knowledge sources, as well as more narrative-graph information like causal chains and linking across episodes, etc. could allow for more complex query addressing. It also highlights the need for more textual data and databases/knowledge graphs to facilitate this work.

Beyond the present implementation, a promising direction for future work is the extension of our chapter-level entity graph into a fully learnable graph neural network (GNN)–based retrieval module. Also, by integrating verse-level nodes into the graph, hierarchical retrieval can be facilitated, where chapter-level relevance can also guide more precise verse extraction.

This can also be extended to other Itihāsa-Purāṇa texts or similarly structured text corpora.

7 Conclusion

We developed Mahānveṣaṇā, a knowledge-enhanced RAG framework designed to address challenges in verse retrieval in the Mahābhārata. We used the Mahānāma entity-linking dataset, used chapter information, and used an entity-linking graph, to enable the retrieval system to retrieve verses relevant to a query in English. We also fine-tuned pretrained SBERT models (MPNEt) on Sanskrit-English parallel data and created an evaluation dataset of over 2000 query-verse pairs. We found that augmenting retrieval with external knowledge yielded a 10% improvement in recall during validation and 5% on the held-out test data, across embedding models. Mahānveṣaṇā is an attempt to show that structured knowledge and narrative-aware propagation can be used to improve performance on complex text corpora, laying the groundwork for more advanced retrieval systems for Itihāsa and other Sanskrit corpora.

References

[1] Rahul Aralikatte, Miryam de Lhoneux, Anoop Kunchukuttan, and Anders Søgaard. Itihasa: A large-scale corpus for Sanskrit to English translation. In *Proceedings of the 8th Workshop on Asian Translation (WAT2021)*, pages 191–197, Online, August 2021. Association for Computational Linguistics.

[2] Sujoy Sarkar, Gourav Sarkar, Manoj Balaji Jagadeeshan, Jivnesh Sandhan, Amrith Krishna, and Pawan Goyal. Mah\={a} n\={a} ma: A unique testbed for literary entity discovery and linking. *arXiv preprint arXiv:2509.19844*, 2025.

[3] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding, 2020.

A Appendix: Generated answers

To illustrate the qualitative differences between models, we present below a sample query along with the responses generated by the baseline and improved systems.

A.1 Example 1

Query

> *"In the story of Uttanka and Gautama, why did Uttanka go behind the earrings? Retrieve the exact verses that mention it?"*

GPT-4o (No RAG)

> "In the story of Uttanka and Gautama, which is part of the Mahabharata, Uttanka is a disciple of the sage Gautama. Uttanka is tasked with a mission to retrieve a pair of earrings as a guru dakshina (a traditional offering to a teacher) for his guru's wife. The earrings are meant to be a gift for the king's wife, and Uttanka's guru's wife desires them as her own. This task sets Uttanka on a challenging journey.
>
> Unfortunately, I can't provide the exact verses from the Mahabharata, but I can summarize the story for you. Uttanka's journey involves various trials, including encounters with supernatural beings and obstacles, as he attempts to fulfill his teacher's wife's request. This story is a part of the larger epic and illustrates themes of duty, perseverance, and the challenges faced in fulfilling one's obligations.."

Entity-Enhanced RAG

> "In the story of Uttanka and Gautama, Uttanka went behind the earrings as a final fee or gift for his preceptor's wife, Ahalya. Despite Gautama, his preceptor, expressing satisfaction with Uttanka's service and devotion, Uttanka insisted on giving a tangible gift to Ahalya. Ahalya then requested Uttanka to bring her the celestial earrings worn by the wife of Saudasa, which would fulfill the obligation of the final fee to his preceptor. This request set Uttanka on his quest to retrieve the earrings.
>
> The exact verses from the context that mention this are:
>
> - *14.56.25*: "Ahalya said Blessed be you, bring for me those celestial ear-rings which are worn by the wife of Saudasa. That which is due to your preceptor will then be well-discharged."
> - *14.56.26*: "Replying her, 'So be it'. Utanka went away. O Janamejaya, determined upon bringing those ear-rings for doing what his preceptor's wife liked."
>
> These verses illustrate the motivation behind Uttanka's quest for the earrings, driven by his desire to fulfill the wishes of his preceptor's wife and complete his obligations as a disciple. "

In this case, GPT-4o without RAG gives an answer that is correct in essence, but cannot quote the original contexts. Our RAG system successfully does that. Checking with subject matter experts and relevant sources, our generated answer is correct.

A.2 Example 2

Query

> *"Where and what is the story of Prahlad and Bali used by Draupadi to teach forgiveness?"*

GPT-4o (No RAG)

> "In the Mahabharata, Draupadi uses the story of Prahlad and Bali to teach a lesson on forgiveness. This story is part of the larger narrative of Hindu mythology and involves two significant figures: Prahlad and Bali, both of whom are known for their devotion and righteousness.

> Prahlad was a devout follower of Lord Vishnu and the son of the demon king Hiranyakashipu. Despite his father's opposition and attempts to kill him, Prahlad remained steadfast in his devotion to Vishnu. His unwavering faith ultimately led to Hiranyakashipu's downfall at the hands of Vishnu's Narasimha avatar.
>
> Bali, on the other hand, was the grandson of Prahlad and a powerful demon king known for his generosity and devotion. He ruled with fairness and was beloved by his subjects. However, his power and influence became a concern for the gods. To curb his growing power, Vishnu incarnated as Vamana, a dwarf Brahmin, and approached Bali during a sacrificial ceremony. Vamana asked for as much land as he could cover in three steps. Bali agreed, and Vamana then expanded to cover the entire universe in two steps. For the third step, Bali offered his own head, demonstrating his humility and devotion"

Entity-Enhanced RAG

> In the Mahabharata, specifically in the context of Chapter M.3.28, Draupadi recounts the story of Prahlada and Bali to illustrate the nuanced understanding of forgiveness and its application. This story is used to convey the wisdom of knowing when to forgive and when to assert one's power.
>
> The narrative begins with Bali, the son of Virochana, seeking guidance from his grandfather Prahlada, a wise and revered king of the Asuras and Danavas. Bali is inquiring about the virtues of forgiveness versus the exercise of prowess or energy, expressing doubt about which path leads to true well-being.
>
> Prahlada, in his response, provides a balanced perspective. He explains that neither perpetual forgiveness nor constant aggression leads to well-being. Instead, he emphasizes the importance of context and discernment. Prahlada warns that a person who always forgives may be taken advantage of, disrespected, and disregarded by servants, strangers, and even family members. Such a person may suffer from neglect and exploitation, as others may not fear any consequences for their actions.
>
> Conversely, Prahlada also cautions against the dangers of constant anger and punishment. A person who is perpetually angry and punishes indiscriminately risks alienating friends,

In this case, GPT-4o without RAG gives a completely wrong answer, without any evidence. However, our RAG system successfully gives a correct answer and cites the correct source as well.