KathaYatra- Retrieval Augmented Generation for Personalised Cultural Narratives in Tourism

Srushti Srikanth[1], Priyadarshini V[2], Ananya Sripuram[3] and P Kokila[4]

[1] Department of Computer Science, PES University, Bengaluru, India
srushti.srikanth@gmail.com
[2] Department of Computer Science, PES University, Bengaluru, India
vpriyasamac05@gmail.com
[3] Department of Computer Science, PES University, Bengaluru, India
ansripuram@gmail.com
[4] Department of Computer Science, PES University, Bengaluru, India
pkokila@pes.edu

Abstract. Storytelling has been the basis of tourist experience, ever since the inception of tourism itself. Stories and legends bridge the gap between the user and the destination, allowing the tourist to connect with the history and fantasy of what once was or could have been. An itinerary, if combined with the local stories, would provide for a richer experience. This work discusses the proposal for a storytelling based itinerary generator, Katha Yatra, which uses generative AI techniques like retrieval augmented generation and prompt engineering to weave a powerful and entertaining narrative for every destination intended for pleasure tourism.

Keywords: storytelling, tourism, itinerary generation, retrieval augmented generation.

1 INTRODUCTION

Tourism is the act of temporary relocation for the pursuit of relaxation, pleasure, recreation, and the provision for such activities. Tourism has long been recognized as a sector that creates and maintains high economic value while providing both vocation and profit for the locals, while at the same time inviting inflow of revenue from foreign sources. Tourists often hire local guides who reveal to them the history, legends, mythology- summarily, the stories, associated with the location, enriching tourist experience and allowing them to connect with the destination and the artifacts on a more personal level. The role of tourist guides is, as is well known, as an important one, since the locality of their experiences in the destination brings much needed value.

The importance of storytelling has been detailed in multiple studies [1-5]. Moscardo [6] identifies three categories of stories- those that exist before the experience, the pre-experience stories, those that emerge or unfold during the experience, or the emerging experience stories, and those that are told after the experience, which are told to others by the tourists, or the post-experience stories. The pre-experience stories create a sense

of excitement for the tourist to want to come, because they define why the destination should be visited.
Local legends are often contradictory, sometimes even negatory towards each other, but combined, they all create a compelling experience. With folktales, the concept of a ground truth is often discarded in favor of an entertaining experience, or strictly held beliefs, thus no story is truer than any other. Gramsci as interpreted by Crehan, defines folklore similarly as a collective, contradictory system of beliefs, superstitions and beliefs bundled together [7].
In this work, the use of Retrieval Augmented Generation is combined with standard itinerary generation for the creation of a compelling and exciting pre-story to entice prospective visitors, ignite wanderlust, or inspire travel, by drawing information from blog posts, history pages, and mentions of local legends across the internet.

2 RETRIEVAL AUGMENTED GENERATION (RAG) PIPELINES

2.1 What is a RAG Pipeline?

Retrieval Augmented Generation (RAG) is a method that enhances the strength of Large Language Models (LLMs) by leveraging external, up-to-date sources of information in the generation. Unlike relying on the pre-trained information in an LLM, RAG pipelines utilize contextually pertinent documents or information from external knowledge bases, which are generally unstructured, to generate contextually accurate and domain-specific text without the need for retraining the model [8, 9].

Here, two methods are proposed.

3 THE RETRIEVAL, SUMMARISATION, AND GENERATION OF STORIES

Retrieval Augmented Generation is particularly relevant in narrative building, where coherence, creativity, and factuality are important.

Traditional RAG pipelines draw the right information from an unchangeable database and use it to inform guide generation. But all too frequently, we need something more dynamic-a process that can handle shifting narrative context and draw not only curated data sets but live data sources. Two broad solutions are therefore proposed:

1. Static Retrieval – From an existing body of stories, perhaps historical records or ancient mythologies.

2. Dynamic Retrieval – Ongoing collection of context data through web scraping or user input. Irrespective of source, the pipeline generally starts with gathering and categorizing narrative content. Narrative content is obtained from diverse sources such as story libraries, history documents, folklore, and user feedback. The content is used as fact sources or concepts.

3.1 The Static Pipeline

```
Algorithm: StoryRAGPipeline: Retrieval-Augmented Story
Generation
Input: API key for OpenRouter, embedding model name
Output: Generated story

Class StoryRAGPipeline(openrouter_api_key, embed-
ding_model)
    Initialize SentenceTransformer with embedding model
    Initialize ChromaDB client and get or create 'sto-
ries' collection
    Initialize TF-IDF vectorizer and story state varia-
bles

Function add_story_fragments(fragments)
    foreach fragment in fragments do
        Encode text with embedding model
        Create metadata including characters, location,
emotions
        Store embedding, text, and metadata in ChromaDB
        Update internal documents and TF-IDF matrix
        Update story state per location

Function hybrid_retrieval(query, location, top_k)
    Perform semantic search using ChromaDB
    Transform query using TF-IDF vectorizer
    Calculate similarity scores for retrieved docs
    Combine semantic and TF-IDF scores for hybrid ranking
    Return top-k relevant story fragments

Function update_story_state(generated_text, location)
    Tokenize into sentences
    Generate episode summary via summarization API
    Update location's last visit time and append summary
    If enough summaries, update global story summary

Function generate_story(location, user_prompt)
    Retrieve context using hybrid_retrieval
```

```
    Compose prompt using fragments, story state, and user
prompt
    Send request to OpenRouter API to generate story
    Update story state with generated story
    return generated story

Function Main()
    Instantiate pipeline with API key
    Define story fragments (location, characters, etc.)
    Add fragments to pipeline
    Call generate_story with location and prompt
    Output the generated story
```

Long units are divided into semantically coherent pieces, like scenes, paragraphs, or significant plot points. This segmentation strategy favours complex retrieval at the cost of overall consistency of the narrative. Each of these segments is associated with pre-trained models like `all-MiniLM-L6-v2`, where textual information is embedded as highdimensional vectors to capture semantic meaning. These embeddings provide the foundation needed to perform similarity-based queries.

Embeddings are also stored in a vector database like ChromADB, along with other metadata like characters, locations, objects, emotions, and timestamps. Such metadata enhances retrieval quality and narrative constraints.

This model employs a hybrid retrieval method with semantic similarity combined with vector search and keyword relevance established based on TFIDF scoring. This method ensures contextual coherence and accurate term matching.

The form of these items can be diversified based on thematic relevance analyses, narrative cohesion analyses, or affective strength analyses to produce a higher quality of the resulting story output. The highest-ranked narrative content and user prompt are input into a generative model like GPT-4o (through OpenRouter API). The model generates a coherent, affective story that adheres to the restored context. The other tasks supply narrative constraints such as tone, genre, coherence among characters, and global memory management. Some of them include episodic summarization, location tracking, and generation of a global story path.

This multi-faceted system provides fact and theme consistency, as well as emotional coherence. Through the combination of symbolic metadata, deep embeddings, and transformer generation, the system effectively closes the gap between fact retrieval and the art of storytelling—making the system a solid framework for applications such as historical fiction, interactive fiction, and story generation for learning.

3.2 The Dynamic Pipeline

```
Algorithm: Dynamic Story Generation Using Live Web Data
Input: User-specified location
Output: Generated myth or folklore story
```

```
main()
    Prompt user for location input
    search_web(location + "myths and folklore")
    Retrieve top links excluding Wikipedia
    foreach url in links do
        content ← get_web_content(url)
        Append to web.text
    filtered_web ← filter_relevant_text(web.text)
    if filtered_web is empty then
        Print "No relevant content found."
        return
    summary ← summarize_text(filtered_web)
    if summary is empty then
        Print "Generated summary is empty."
        return
    story ← generate_story(summary)
    Print final story

search_web(query)
    Use DuckDuckGo API to get top 5 results
    Exclude Wikipedia links
    return list of URLs

get_web_content(url)
    Fetch HTML via HTTP GET with user-agent
    Parse with BeautifulSoup to extract paragraphs
    Concatenate text and return

filter_relevant_text(text)
    Define folklore-related keywords
    Tokenize text into sentences
    return sentences containing at least one keyword

summarize_text(text)
    Use BART summarization model (transformers)
    return summary

generate_story(summary)
    Use GPT-2 to generate story text
    return generated story
```

This model takes a real-time, data-driven approach to generating stories by retrieving, processing, summarizing, and rewriting information into a coherent narrative. The goal

is to create storytelling experiences that are timely and contextually relevant, reflecting what a user might discover through their own online exploration.

The process begins with a query-specific search using the DuckDuckGo Search API, which is used for privacy preserving, accurate search result retrieval. The search process performs keyword based lexical lookup across culturally or thematically relevent terms.

Once the URLs are retrieved, the model scrapes their contents using the BeautifulSoup library. It extracts the main body text and applies lightweight natural language processing techniques to clean the data—removing extraneous elements such as hyperlinks, embedded media references, advertisements, author bios, and navigation text. This step is crucial in isolating only the semantically relevant content that can serve as the foundation of a compelling narrative.

The cleaned text is then passed through a keyword selective retention layer, which selects sentences that contain culturally and thematically important terms (e.g., "ghost", "legend", "cursed", "divine"). These filtered segments are concatenated and fed into a summarization model—specifically, the facebook/bart-large-cnn model. The facebook/bart-large-cnn model was selected for summarization due to its architecture, which combines a denoising autoencoder with fine-tuning on news-style content. This makes it highly effective at generating coherent multi-sentence summaries from long, noisy input without losing narrative flow. BART condenses the collected data into a structured summary that retains core elements such as characters, settings, and key events.

Finally, this summarized content is used as the seed input for a generative model—in this case, GPT-2. The model expands the summary into a full-length story, weaving together the retrieved facts and legends into a creative, readable narrative. The final output is then returned to the user as a localized story, rooted in cultural lore but reshaped through modern natural language models.

Importantly, factual accuracy is not a primary concern in this context. Since folklore and mythology are naturally multiform and often contradictory, the model embraces their fluidity. By grounding the story in what is most visible or resonant online, the system produces versions of tales that are not definitive, but rather representative—offering a reflection of how these stories live and evolve in public discourse today.

The use of compact models (facebook/bart-large-cnn, GPT-2) ensures sequence processing remains under 700 tokens throughout. This choice enables low-latency generation, suitable for real-time interaction on edge devices or low-resource deployments.

While the pipeline uses strict string matching, fuzzy string matching and token frewuency thresholds can be implemented to ensure that words semantically similar to keywords, like haunting or haunts instead of haunted, can be used.

KathaYatra employs a live, unsupervised Retrieval-Augmented Generation (RAG) architecture composed of lightweight, pretrained large language models (LLMs), each designated for a specific subtask—facebook/bart-large-cnn for summarization and GPT-2 for narrative generation. This configuration avoids common limitations associated with large pretrained models, including reliance on fine-tuning, excessive n-shot

prompting, memorization of training corpora, and poor generalization to geographically or culturally underrepresented domains.

The unsupervised pipeline enables broad adaptability across regions, cultures, and content domains. It is designed to accommodate heterogeneous, unstructured narrative data—including travel blogs, local folklore, and informal storytelling formats—without requiring annotated corpora. Wikipedia is deliberately excluded to reduce lexical anchoring and narrative flattening, introducing what is referred to as folk noise: conflicting, culturally embedded, and unverifiable narrative elements that mirror oral traditions and increase user engagement through ambiguity and contradiction.

Finally, an image-generation model like stable diffusion can be used to generate an image based on the given generated story to not only represent the actual location, whose images will be retrieved by the gemini itinerary generator, but an image that depicts the overarching story or theme of the location.

Results with the dynamic pipeline

Previously proposed RAG evaluation methods are deemed inappropriate for this application due to the use case proposed in this work. KILT [10] compares generated material to Wikipedia, which is filtered out to create robust and diverse narratives in the pipeline. RAGAs [11] and ARES [12] evaluate the final product on faithfulness ro source material, answer relevance and context relevance [13], which while applicable, are not sufficient evaluation metrics for the live model.

To quantify the amount of content retained in the selective retention process, Content Retention Rate (CRR) is used. CRR is defined as the proportion of sentence-level units that are preserved after the filtering and summarisation process. Each senstence is a discrete unit of content. CRR is the ratio of non-empty filtered sentences to original sentences retrieved via web scraping multiplied by 100.

So, for each step, scraping, summarisation, and generation, the text after the step process is compared to the scraped text.

CRR here does not account for paraphrasing or semantic similarity but does provide an estimate of how much of the content is directly used in the final story.

It must be noted that this live pipeline fails if scraped content corpus is small and unvaried.

Table 1. Content Retention Rate (CRR) obtained at each step of the dynamic RAG pipeline

Step	Location	CRR
Filtering web text	Mysuru	0.29%
Summarisation	Mysuru	2.9%
Story generation	Mysuru	6.38%
Filtering web text	Rajasthan	0.62%
Summarisation	Rajasthan	4.38%

Story generation	Rajasthan	10.0%
Filtering web text	Bijapur	0.07%
Summarisation	Bijapur	0.55%
Story generation	Bijapur	0.82%

Results for 'Mysuru' as location

Fig. 1. Image generated for location 'Mysuru'

4 THE APPLICATION AND USER EXPERIENCE

A web application is proposed, which will use Reach, CSS, HTML, and use a python backend. Two sections exist- KathaYatra, the itinerary generator, and Find the

Monkey, a game which allows the user to uncover the mysteries and folklore associated with the destination.

KathaYatra will allow user input through a simple form, which takes entries regarding the location, duration of vacation, intensity of activities (high, medium, low), etc. A Gemini API is then called to generate a suitable itinerary, through prompt engineering.

Find the monkey utilizes the RAG models proposed to generate a story. This story is then chunked into three portions. An API is called to retrieve location pictures. The first portion of the story is presented to the user. Each subsequent portion of the story is 'unlocked' after the user scores a certain number of points, five and ten respectively to unlock story sections three and four. This ensures continued engagement as the user reads the story. The two components of the application, KathaYatra and Find the Monkey, are designed to be deeply intertwined, creating a synergistic experience that enhances the user's engagement with their chosen destination. This design ensures that the user's interest in the destination will be maximized, since the story will contain tales about the destinations in the itinerary as well as other interesting locations which the user may want to visit. This contextual connection ensures that the stories resonate with the user's immediate surroundings during the travel and planned activities, fostering a deeper appreciation for the destination's cultural and historical fabric.

5 Conclusion

KathaYatra introduces a lightweight, culturally adaptive storytelling system built on a Retrieval-Augmented Generation (RAG) architecture. It illustrates that generative models, when guided by semantically informed preprocessing, can do more than create—they can recover, restructure, and elevate the relevance of real-world content. The system's introduction of the Content Retention Rate (CRR) as a novel evaluation metric reveals a surprising trend: as content moves through each stage—summarization and story generation—semantic alignment with the original scraped corpus actually improves. This reinforces the system's ability to enhance narrative coherence through layered abstraction, rather than diminish it.

Although the keyword-driven filtering stage occasionally allows non-narrative artifacts—like promotional text ("visit our website")—to pass through, their impact on narrative quality is minimal. If anything, their presence slightly inflates sentence-level CRR values, making the reported scores a conservative estimate of true semantic fidelity.

By embedding this RAG-based architecture into a dual-module tourism platform—KathaYatra for itinerary generation and Find the Monkey for interactive narrative exploration—the project reimagines the role of AI in cultural tourism. What begins as fragmented, noisy folklore scraped from the web is transformed into a personalized, compelling, and interactive pre-experience. This work not only delivers a working

prototype but also lays the groundwork for scalable, culturally sensitive, real-time narrative systems.

References

1. Bassano C., Barile S., Piciocchi P., Spohrer J. C., Iandolo F., Fisk R. (2019). Storytelling about places: Tourism marketing in the digital age. Cities, 87, 10–20. https://doi.org/10.1016/j.cities.2018.12.025
2. Ben Youssef K., Leicht T., Marongiu L. (2019). Storytelling in the context of destination marketing: An analysis of conceptualisations and impact measurement. Journal of Strategic Marketing, 27(8), 696–713. https://doi.org/10.1080/0965254X.2018.1464498
3. Bessiere J., Ahn Y.-J. (2021). Components of DMZ storytelling for international tourists: A tour guide perspective. Sustainability, 13(24), Article 13725. https://doi.org/10.3390/su132413725
4. Hartman S., Parra C., de Roo G. (2019). Framing strategic storytelling in the context of transition management to stimulate tourism destination development. Tourism Management, 75, 90–98. https://doi.org/10.1016/j.tourman.2019.04.014
5. Moreira, A. C., Costa, R. A. da, & de Sousa, M. J. N. (2024). Is a Good Story Enough? A Critical Analysis of Storyteller Roles in Tourism. Journal of Hospitality & Tourism Research, 49(4), 688-704. https://doi.org/10.1177/10963480241251450 (Original work published 2025)
6. Moscardo, G. (2021), "The story turn in tourism: forces and futures", Journal of Tourism Futures, Vol. 7 No. 2, pp. 168-173. https://doi.org/10.1108/JTF-11-2019-0131
7. Crehan, K., 2022. Gramsci's folklore bundle. *Archivio Anuac*, *11*(1), pp.55-64.
8. LakeFS. (n.d.). RAG Pipeline: Example, Tools & How to Build It https://lakefs.io/blog/what-is-rag-pipeline/
9. Klesel, M. and Wittmann, H.F., 2025. Retrieval-Augmented Generation (RAG). *Business & Information Systems Engineering*, pp.1-11.
10. Petroni, F., Piktus, A., Fan, A., Lewis, P., Yazdani, M., De Cao, N., Thorne, J., Jernite, Y., Karpukhin, V., Maillard, J. and Plachouras, V., 2020. KILT: a benchmark for knowledge intensive language tasks. arXiv preprint arXiv:2009.02252.
11. Es, S., James, J., Anke, L.E. and Schockaert, S., 2024, March. Ragas: Automated evaluation of retrieval augmented generation. In Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations (pp. 150-158).
12. Saad-Falcon, J., Khattab, O., Potts, C. and Zaharia, M., 2023. Ares: An automated evaluation framework for retrieval-augmented generation systems. arXiv preprint arXiv:2311.09476.
13. Zhao, P., Zhang, H., Yu, Q., Wang, Z., Geng, Y., Fu, F., Yang, L., Zhang, W., Jiang, J. and Cui, B., 2024. Retrieval-augmented generation for ai-generated content: A survey. *arXiv preprint arXiv:2402.19473*.