Evaluating User Intent Classification and Hybrid Retrieval in a RAG-based Conversational Tourism Recommender System

Akshat Tandon[†], Ashmi Banerjee[*,†]

Technical University of Munich

Abstract

Traditional tourism recommender systems struggle with cold-start problems and lack the natural interaction capabilities of conversational agents. This paper introduces a modular Hybrid Retrieval-Augmented Generation (RAG)-based Conversational Tourism Recommender System (C-TRS) for European cities. Our architecture combines LLMs with a hybrid retrieval pipeline (dense and sparse vector search) and a structured dialogue state tracker. We use a curated knowledge base from Wikivoyage and Tripadvisor, encompassing over 100 European cities. User utterances are parsed for multi-label intent classification, triggering the retrieval of relevant city-level knowledge chunks to augment LLM prompts for actions like answering queries or providing recommendations. Our evaluation focuses on user intent classification (comparing traditional models with few-shot LLM prompting) and retrieval quality (using the RAGAS framework). Findings demonstrate the efficacy of our hybrid retrieval approach and the power of few-shot learning with LLMs in handling the complexities of conversational recommendation. Overall, our hybrid retrieval strategy balances recall and precision by combining dense (semantic) and sparse (lexical) embeddings, conditioned on conversational intent. This dynamic selection addresses limitations of static or purely lexical/semantic retrieval in tourism RAG systems. It represents a novel integration of intent-driven hybrid retrieval in a conversational tourism framework. Our code and artifacts are available at https://github.com/Akshat125/conversational-trs.

Keywords

City Tourism Recommendations, Conversational Recommender Systems, Retrieval-Augmented Generation, Hybrid Vector Search, Large Language Models

1. Introduction

Recommender systems are pivotal in helping users discover new destinations, attractions, and points of interest (POIs). In tourism, this assistance is crucial for enabling users to explore and plan enriching travel experiences. However, traditional recommendation approaches often grapple with the "cold-start" problem, which stems from insufficient user information and limited interaction data. This problem hinders their ability to provide personalized suggestions, especially for new users or in novel contexts, as the system lacks the necessary data to make accurate recommendations [30, 33]. Furthermore, as the desire for personalized and contextually relevant recommendations grows, conversational interfaces have emerged as a natural and intuitive way for users to express their needs and preferences [8].

The emergence of LLMs pretrained on extensive textual corpora introduces a paradigm shift by leveraging implicit world knowledge, advanced natural language understanding, and few-shot in-context learning. This term refers to the ability of the model to learn from a small amount of data in a specific context, allowing it to generate recommendations with limited explicit training data [6, 14]. Nonetheless, LLMs are prone to hallucinations and factual inaccuracies when generating responses without grounded external knowledge, necessitating retrieval-augmented approaches. Retrieval-Augmented Generation (RAG) frameworks combine information retrieval with conditional text generation, allowing grounding

Workshop on Recommenders in Tourism (RecTour 2025), September 22, 2025, co-located with the 19th ACM Conference on Recommender Systems, Prague, Czech Republic.

[*] Corresponding author.

[†] These authors contributed equally.

akshat.tandon@tum.de (A. Tandon); ashmi.banerjee@tum.de (A. Banerjee)
https://ashmibanerjee.com (A. Banerjee)

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

of LLM outputs in external, domain-specific corpora. This mitigates hallucination while reducing fine-tuning overhead by decoupling retrieval and generation stages [38, 31]. Existing RAG-based tourism RS typically utilize single retrieval strategies and static pipelines, which limit adaptability and incur high computational costs [5].

This paper presents a modular Hybrid Retrieval-Augmented Generation (RAG)-based Conversational Tourism Recommender System (C-TRS) tailored to recommend European city trips. Our architecture integrates LLMs with a hybrid retrieval pipeline — combining dense and sparse vector search and a structured dialogue state tracker. This design allows the system to detect user intent, maintain context across turns, and retrieve and generate grounded, contextually appropriate recommendations or answers.

To support this system, we use a knowledge base combining structured and unstructured travel information from Wikivoyage and Tripadvisor, covering over 100 European cities [4]. Each user utterance, such as *"Recommend relaxing destinations in Spain for early spring"*, is parsed for multi-label intent classification, and depending on the system action (e.g., answer, recommend, and explain), relevant city-level knowledge chunks are retrieved from a vector database and used to augment the LLM prompt.

We focus our evaluation on two key components: (1) user intent classification using both traditional models (BERT and BART) and few-shot prompting with LLMs, and (2) the retrieval quality of different vector search strategies using the RAGAS evaluation framework [16]. Our experiments show that LLM-based few-shot classification outperforms traditional, nuanced, multi-intent classification methods. Furthermore, the hybrid retrieval strategy balances recall and precision by switching between dense embeddings (capturing semantic similarity) and sparse retrieval (capturing exact term matches), addressing the limitations of purely dense or sparse retrieval in domain-specific recommendation. This intent-driven hybrid retrieval mitigates LLM hallucination while reducing computational overhead — advancing prior RAG-based tourism systems that lacked dynamic retrieval or conversational modeling [5]. To our knowledge, this is the first C-TRS to embed hybrid retrieval selection directly within a conversational framework.

This paper is structured as follows: Section 2 reviews the relevant literature and identifies the research gaps. Section 3 outlines our proposed system architecture and core components. In Section 4, we evaluate two critical aspects of the C-TRS: user intent classification and the RAG pipeline, with a key focus on the retrieval strategies. Finally, Section 5 concludes the paper by summarizing our contributions and findings, and discusses directions for future work.

2. Related Work

This section surveys the existing research in CRS, user intent classification, and RAG for recommender systems in the tourism domain.

2.1. Conversational Tourism Recommender Systems (C-TRS) using LLMs

Conversational recommender systems (CRS) have emerged as a promising approach to address challenges such as cold-start problems and information asymmetry in recommendation scenarios [19]. By enabling users to express preferences, ask questions, and provide feedback in natural language, CRS offers a more interactive and personalized recommendation experience. The recent surge in LLM capabilities has significantly enhanced the reasoning and dialog management capacities of CRS. This has led to the development of various LLM-enhanced architectures that incorporate fine-tuning, RAG, and hybrid frameworks integrating LLMs with traditional recommender systems [22, 23, 18, 21].

In the tourism domain, several LLM-based conversational systems have been proposed to assist users with trip planning and itinerary generation. For instance, zIA [9] is a persona-driven assistant that offers localized destination suggestions and supports itinerary planning. TravelAgent [10] combines recommendation, planning, memory, and tool-use components to deliver personalized travel itineraries through conversational interaction. Similarly, Vaiage [26] introduces a graph-structured multi-agent architecture that recommends points of interest (POIs) and dynamically builds adaptive itineraries based on user preferences and contextual factors.

While these systems demonstrate the potential of LLMs for conversational tourism assistance, they primarily focus on POI recommendation and detailed itinerary planning, often assuming that a destination or city has already been selected. In contrast, our work targets an earlier and less explored stage in the travel planning pipeline: recommending sustainable cities or destinations. We design a conversational framework that supports natural language interaction while prioritizing sustainability, a critical but underrepresented objective in existing LLM-based tourism CRS.

2.2. User Intent Classification in CRS

Several studies have addressed user intent classification in conversational recommender systems (CRS). Early work by Cai et al. [8] proposed a taxonomy of user intents in movie recommendation. It evaluated traditional machine learning models such as logistic regression, XGBoost, and SVM, highlighting the benefit of contextual features. With the rise of transformers, Moradizeyveh [32] developed a pipeline using a fine-tuned BERT model for intent recognition, while Kemper et al. [23] applied few-shot prompt-based classification in restaurant CRS. Other advances include Sauer et al. [39], who leveraged knowledge distillation for few-shot intent classification, and H. Liu et al. [27], who utilized label-enhanced graph neural networks to capture relationships among intent classes. Techniques for dynamic label refinement were proposed by Park et al. [36] to improve semantic separability in few-shot settings, while Hou et al. [20] addressed multi-label intent detection with adaptive thresholding. Complementary approaches that jointly model user preferences and intents in CRS have been introduced by Li et al. [25] and Park et al. [36], focusing on multi-aspect preference modeling and explainability. Despite these contributions, intent classification specifically tailored to tourism CRS remains underexplored, motivating our evaluation of both supervised and LLM-based zero- and few-shot methods in this domain.

2.3. RAG for Tourism Recommender Systems

Retrieval-Augmented Generation (RAG) techniques have recently gained traction in recommender systems for their ability to combine external knowledge retrieval with powerful language models, improving both accuracy and contextual relevance while reducing hallucination [2, 24]. In the tourism domain, Qi et al. [37] demonstrates the effectiveness of RAG by optimizing a Tibet-focused conversational tourism recommender system grounded in a vector database of tourist viewpoints, which reduces hallucinations and enhances personalization. Similarly, Song et al. [41] propose TravelRAG, a framework that augments RAG with knowledge graphs for tourist attractions, yielding improvements in both efficiency and accuracy.

More recently, Banerjee et al. [5] introduced a single-shot RAG-driven recommendation pipeline incorporating sustainability-aware reranking to generate eco-friendly city recommendations. However, their approach is not conversational and focuses on a single retrieval step. We extend this work by developing a fully conversational system that integrates hybrid retrieval strategies, improves computational efficiency, and incorporates user intent classification to better understand and respond to user preferences. This enables iterative and personalized interactions tailored to sustainable tourism.

3. Approach: RAG-driven System Design

To enable context-aware and dynamically adaptive recommendations within our CRS, we propose an architecture centered around a RAG-pipeline. This design enhances the traditional CRS interaction loop by integrating vector-based semantic search and conditional prompt augmentation, triggered based on the conversational context. The RAG pipeline primarily supports two types of recommender actions: *Answer* and *Recommend and Explain*, where external knowledge is required to generate informative and grounded responses. A high-level overview of the system architecture is depicted in Figure 1.

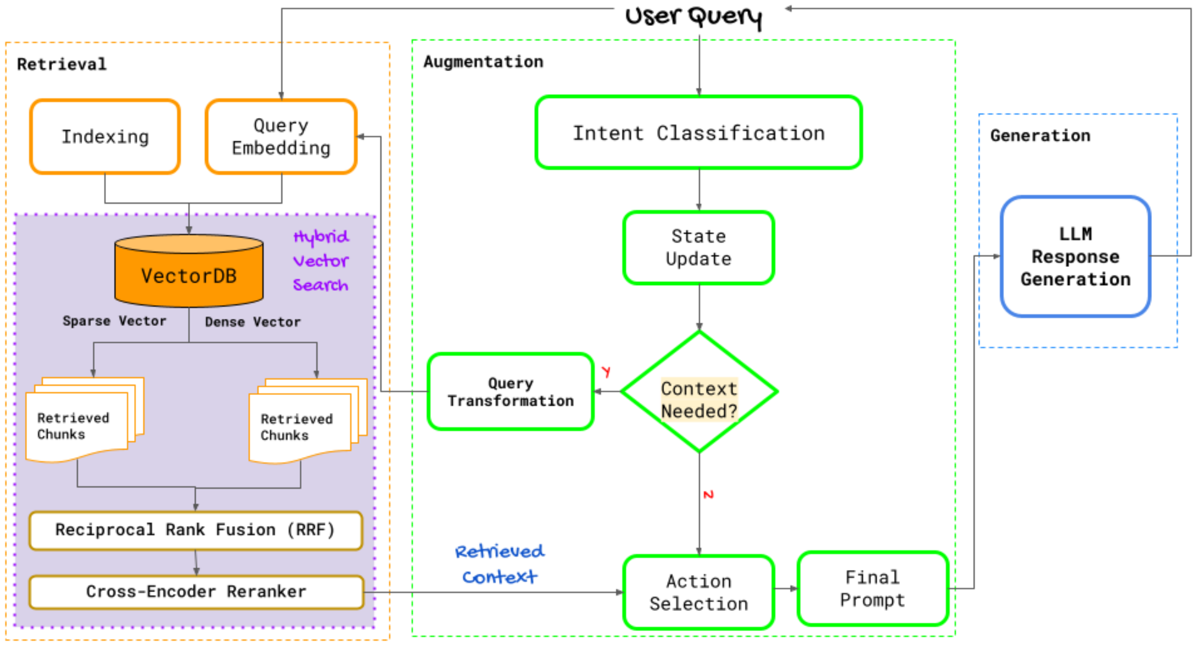

Figure 1: Overview of our Hybrid RAG architecture for a C-TRS. The pipeline operates in a multi-turn setting, where user utterances are parsed to extract context and intent. Based on the dialogue state, relevant city-level chunks are retrieved from a hybrid semantic index (combining dense and sparse retrieval), optionally reranked, and used to augment the prompt for the LLM. The generation module produces grounded, context-aware responses for recommending European cities.

Our architecture follows a modular design and is composed of three primary stages: **Retrieval**, **Augmentation**, and **Generation**. Each user query is processed through the augmented pipeline and, depending on intent and dialogue state, may trigger the retrieval mechanism.

3.1. User Interaction Scenario

The system operates in a multi-turn conversational setting where users engage in natural dialogue to discover travel destinations. A typical session starts with an open-ended query such as *"Can you suggest a relaxing destination in Europe for early spring?"* or a factual question like *"What is the best time to visit Ljubljana?"*. As the conversation evolves, users may provide preferences (e.g., *"I prefer less crowded places with good public transport"*) or respond to system recommendations (e.g., *"That sounds interesting—tell me more about local cuisine there"*). These utterances are parsed to infer intents, preferences, and context cues, which are stored in a structured dialogue state and used to guide retrieval and generation.

3.2. Dataset

The underlying knowledge base consists of structured and unstructured city-level travel information for 160 European cities from Wikivoyage, parsed and processed using the method described in Banerjee et al. [4]. Each article is hierarchically chunked by section headings (e.g., `Get Around`, `Do`, `Eat`) to preserve semantic organization similar to Banerjee et al. [5]. The resulting corpus covers over 100 European cities, enriched with metadata such as month-wise seasonality, sustainability indicators, and geolocation. Additionally, for the *Recommend and Explain* action, this knowledge base is augmented with structured metadata from Tripadvisor, including Points of Interest (POIs), green accommodations flagged under Tripadvisor's sustainability program [1], and user ratings. The evaluation dataset used in this study contains 50 single-hop queries, where answers are explicitly derivable from one or more

[1] https://www.tripadvisor.com/GreenLeaders

knowledge chunks. These queries cover a diverse range of intents, destinations, and seasons, and are used to benchmark the retrieval component of the system.

3.3. Retrieval Stage

The retrieval stage is responsible for identifying relevant external knowledge chunks to support system responses. This stage is invoked only when deemed necessary by a lightweight routing mechanism (described in the augmentation stage). This selective invocation ensures computational efficiency, avoiding retrieval during trivial turns such as acknowledgments. This stage includes document indexing, query embedding, vector similarity search, optional reranking, and strategy-specific chunk filtering.

3.3.1. Embedding and Hybrid Vector Search

We construct a vector database using city-level travel data from Wikivoyage using the knowledge-base developed in Banerjee et al. [4]. Each document is parsed and chunked hierarchically using markdown headers to preserve semantic structure. Large chunks are recursively split by subheaders, subsubsections, and sentences, resulting in an average size of 600 characters, similar to Banerjee et al. [5].

Two types of embeddings are generated:

- **Dense embeddings:** using the `all-MiniLM-L6-v2` model [44] to capture semantic similarity.
- **Sparse embeddings:** using the `splade-cocondenser-selfdistil` model [17] to capture contextual keyword-based relevance.

Embeddings are stored in a Milvus Lite instance [2], using `FLAT` indexing for dense vectors with cosine similarity and `SPARSE_INVERTED_INDEX` for sparse vectors with inner product similarity.

At runtime, the user query is embedded using the same model(s) as during indexing. Three retrieval modes are supported:

- **Dense Retrieval**: Semantic similarity using cosine distance.
- **Sparse Retrieval**: Lexical matching based on token overlap.
- **Hybrid Retrieval**: Combines both using Reciprocal Rank Fusion (RRF) [12], as shown in Figure 1.

3.3.2. Retrieval Strategies

We employ intent-aware retrieval workflows that utilize both the dialogue state and metadata for more targeted filtering. The underlying intent classification approach is detailed in Section 4.1.

For *Answer* Action When the user issues a factual query about a known destination:

1. **Metadata filtering** narrows search to chunks matching the current city and (optionally) relevant subheadings.
2. **Vector search** retrieves top-k matching chunks using dense, sparse, or hybrid methods.

This enables precise, city-specific responses, even when city references are implicit, by using an entity extractor agent to update the dialogue state.

For *Recommend and Explain* Action This strategy involves a multi-stage retrieval with sustainability-aware reranking to ensure better explanation:

1. **Query generation:** An LLM transforms dialogue constraints into a pseudo-natural language query.
2. **Initial search:** Top-k chunks are retrieved using dense, sparse, or hybrid methods.

[2] https://milvus.io/

3. **SFI reranking:** Candidate cities are reranked for sustainability using the Societal Fairness Indicator (SFI) [3, 5], which combines CO_2e emission trade-offs, destination popularity (ratings and reviews), and seasonality indices (monthly footfall by destination and travel month). Although sustainability is not the primary focus of this work, incorporating SFI helps elevate eco-friendly destinations.
4. **City-specific search:** The top-ranked city is re-searched in the vector database to extract context-specific chunks.
5. **Tripadvisor augmentation:** Retrieved chunks are enriched with structured POI and green hotel metadata from Tripadvisor.

This hierarchical approach balances user constraints and contextual relevance while prioritizing sustainability to generate high-quality recommendations with meaningful explanations, improving the transparency and interpretability of our system.

3.3.3. Reranking and Candidate Selection

To improve context quality, we apply a cross-encoder-based reranker [3] after initial search. This model embeds chunks and queries together to obtain a similarity score for each pair, making it slower yet more accurate than a dense retriever [13]. While RRF is an effective algorithm for aggregating ranked results from dense and sparse retrieval, it does not take the semantic alignment into account. Applying a cross-encoder reranker refines the results with semantic relevance scoring, a strategy shown to improve performance in hybrid retrieval [40].

Algorithm 1 abstracts the retrieval workflow used in our RAG pipeline, unifying the retrieval strategies discussed earlier. `SearchWikivoyageDocs` retrieves the top-k chunks for a given query, retriever (dense, sparse, or hybrid), and optional filter. When the `is_recommendation` flag is set, as in the *Recommend and Explain* action, the candidate pool undergoes SFI reranking and a city-specific search to yield sustainable, context-rich recommendations. Building on N. F. Liu et al. [28], which highlights the role of reranking in mitigating position bias, we further enlarge the candidate pool (e.g., top-10) before reranking to ensure broader coverage and select the final top-5 chunks.

Algorithm 1 Multi-stage retrieval function with cross-encoder reranking

```
function RetrieveWikivoyageContext(query, top_k, rerank, filter, retriever, is_recommendation)
    if rerank then
        top_k ← 2 × top_k
    end if
    docs ← SearchWikivoyageDocs(query, top_k, retriever, filter)
    if is_recommendation then
        docs ← RerankBySFI(docs)
        new_filter ← docs[0].city
        docs ← SearchWikivoyageDocs(query, top_k, retriever, new_filter)
    end if
    if rerank then
        docs ← RerankWithCrossEncoder(query, docs, top_k / 2)
    end if
    return docs
end function
```

3.4. Augmentation Stage

The augmentation stage governs intent interpretation, dialogue state tracking, context routing, and action selection. It is the core logic engine of the system and builds upon previous works in conversational

[3] https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-6-v2

recommendation [29, 7, 23], introducing optimizations for modularity, efficiency, and interpretability.

3.4.1. Intent Classification

Each user utterance is passed through a multi-label intent classifier to determine one or more conversational intents. We adopt a taxonomy comprising five categories [23]:

- **Ask Recommendation**: Direct request for a travel recommendation.
- **Provide Preference**: Statement of preferences or constraints (e.g., travel style, destination type).
- **Inquire**: Factual or exploratory questions about destinations.
- **Accept Recommendation**: Positive feedback confirming interest in a suggested destination.
- **Reject Recommendation**: Negative feedback dismissing a prior suggestion.

Multiple intents can co-occur in a single turn, provided they are semantically compatible. For instance, the utterance *'Amsterdam looks great to me, can you tell me more about the local cuisines there?"* would be classified with the intents: *Accept Recommendation*, *Inquire*, and *Provide Preference*—a valid combination. In contrast, a contradictory statement such as *Looks great. I don't like it."* would be flagged for intent conflict (Accept + Reject), and corrective heuristics would be applied. The corrective heuristics, such as constraint validation and priority-based resolution, act as guardrails to handle diverse user queries. For example, when processing feedback on a recommendation, we first verify that a recommendation was issued in the previous turn.

Intent classification is performed using autoregressive LLMs, with few-shot prompting for robustness. Additionally, when dealing with accept/reject intents, we verify that a recommendation was indeed provided in the previous turn to ensure contextual coherence.

3.4.2. Dialogue State Update

Following intent classification, the system updates the dialogue state, which is maintained as a semi-structured JSON-like object. Inspired by Kemper et al. [23], this state representation balances transparency and flexibility, capturing:

- Recent user and system exchanges (`conversation_history`).
- Required preferences (`hard_constraints`) and optional ones (`soft_constraints`).
- Accepted and rejected destinations (`recommendation_feedback`).
- Metadata-driven query focus keys (`current_destination_of_interest`, `current_subheadings_of_interest`).
- Current user intents and selected system action.

This structure supports dynamic adaptation of the CRS over multiple dialogue turns and prevents premature recommendations when essential constraints are missing—unless overridden by a strong *Ask Recommendation* intent.

3.4.3. Context Router and Query Transformation

A lightweight routing mechanism determines whether external context retrieval is necessary for the current turn. This decision is based on the identified recommender action: only *Answer* and *Recommend and Explain* actions trigger the retrieval stage. For simpler actions (e.g., *Acknowledge Acceptance*), a pre-defined response is returned. This selective routing reduces system latency and computational cost.

When external context is needed, the system may reformulate the query using relevant elements from the dialogue state (e.g., inserting a specific destination name or subtopic). This improves semantic alignment between the user query and vector database content, enhancing retrieval precision.

3.4.4. Action Selection and Final Prompt Construction

Based on the classified intents and current dialogue state, the system selects a recommender action from a predefined set:

- **Recommend and Explain**: Generate a recommendation and justify it using user preferences.
- **Answer**: Provide information in response to a user query.
- **Request Information**: Ask for missing hard constraints.
- **Acknowledge Acceptance / Rejection**: Respond to user feedback.

Unlike scoring-based approaches [23], we adopt a rule-based mapping strategy, prioritizing explicit intents (*Inquire*, *Ask Recommendation*) and using dialogue completeness to guide fallback actions. This approach enhances responsiveness and reduces interpretive ambiguity.

The final LLM prompt is constructed by integrating four key components: (1) the transformed user query, (2) any retrieved context passages (if applicable), (3) the serialized dialogue state, and (4) instructional framing tailored to the selected recommender action. This structured composition ensures that the LLM is guided by both the ongoing dialogue context and relevant external information, resulting in coherent, context-aware, and grounded responses. Listing 1 illustrates the combined system-user prompt template used for the *Recommend and Explain* action.

```
You are a sustainable tourism recommendation system. A city has been pre-selected for the user
after considering both user preferences and sustainability factors. Using the provided context for
 the selected city, your task is to:

1. Summarize the key highlights and explain why the city is recommended, focusing on
sustainability factors.
2. Explain how the city matches the user’s preferences based on the query and context.
3. Highlight the top 3 attractions and green hotels that align with the user’s preferences.
4. Recommend the most sustainable mode of travel from the user’s starting location, and encourage
off-peak/shoulder season travel.

**Instructions**:
1. Begin your response with a bold heading for the city and country (**City, Country**).
2. Follow with: "I recommend [city_name]" and explain why this destination is recommended.
3. Use only the provided context to craft your response. If insufficient, reply: "Sorry, I am
unable to provide a recommendation for the given preferences."
4. Ensure your response is accurate and professional.

**Query:**
{{ query }}. Which city do you recommend and why?

**Context:**
{{ context }}

**Response:**
```

Listing 1: Prompt template for *Recommend and Explain* action

3.5. Response Generation

In the final stage, a response is generated using an LLM:

- For *Answer* and *Recommend and Explain* actions, a context-augmented prompt is passed to the LLM for free-form response generation.
- For simpler actions (e.g., acknowledgments or information requests), a lightweight template or hard-coded message is returned to the user.

This selective generation mechanism ensures that expensive model inference is performed only when warranted, optimizing both user experience and system efficiency. We use *Llama-3.1-8B-Instruct* as our LLM for response generation.

In summary, our RAG-driven architecture tightly integrates semantic retrieval, structured dialogue modeling, and LLM-based generation. By decoupling the intent classification, context routing, and generation processes, the system maintains high modularity, interpretability, and extensibility. This design facilitates seamless adaptation to evolving use cases, including the integration of new intents and retrieval strategies.

4. Evaluation

Due to the multifaceted nature of our CRS, evaluating every component is challenging. For the scope of this paper, we evaluate the performance of our CRS through offline evaluation of two critical components: user intent classification and the RAG pipeline for European city tourism recommendations. To gain a meaningful understanding of the system's performance, we use a combination of current state-of-the-art metrics, including standard classification metrics[4] for user intent classification, and metrics from a model-based evaluation framework RAGAS[5] to evaluate the RAG pipeline.

Section 4.1 evaluates the system's ability to classify diverse user intents and sentiments. Section 4.2 evaluates the RAG pipeline, specifically the information retrieval and generation stages for the *Answer* action. In both sections, we first discuss the experimental setup and metrics, followed by the results.

4.1. User Intent Classification Evaluation

User intent classification is a multi-label classification problem in which the user intents can be classified into multiple categories. Classifying the user sentiment is one of the most critical components of the system because of its direct impact on the recommender action selected and the response returned by the system. Moreover, the task's lower complexity provides a good opportunity to benchmark the results against smaller models. In order to effectively evaluate our user intent classifier, we compare the performance of four different approaches:

1. **BERT (Fine-Tuned Sequence Classification)**: A fine-tuned BERT model trained on a synthetically generated dataset [15].
2. **BART-large-MNLI (Zero-Shot)**: BART model pre-trained on MultiNLI (MNLI) for zero-shot sequence classification [46].
3. **LLMs (Zero-Shot)**: Classification using LLMs with the prompt containing only the user query and label names, providing no descriptions or examples.
4. **LLMs (Few-Shot)**: Classification using LLMs, where each user intent is classified individually with a clear description and few-shot examples. The results are then aggregated to form the final prediction.

We use *GPT-4o-mini* [34] as the primary LLM for evaluation. For the supervised baseline, BERT is fine-tuned on a synthetically generated dataset, while BART-large-MNLI is used out of the box.

Due to the lack of labeled training data for our use case, we manually created user examples covering over ten intent combinations. To augment this dataset, we leveraged *Gemini-1.5-pro-001* [42], following a prompt-based intent description strategy inspired by Parikh et al. [35].

Our prompt template includes: (1) natural language descriptions of each intent as described in Section 3.4.1, (2) a few-shot set of manually crafted examples, and (3) specific instructions to handle edge cases and encourage diverse phrasing. For instance, to generate utterances for the `Accept_Recommendation` intent, the model is prompted to act as an expert sustainable travel consultant and produce 20 distinct examples that clearly accept a recommendation. The instructions emphasize

[4] https://scikit-learn.org/stable/modules/model_evaluation.html
[5] https://www.ragas.io/

clarity, intent exclusivity (avoiding preferences or inquiries), and variation in tone and persona. Each output is structured in CSV format with binary labels across five intent categories: `Ask_Recommendation`, `Provide_Preference`, `Inquire`, `Accept_Recommendation`, and `Reject_Recommendation`. For example, the utterance "*Looks perfect to me!*" is labeled with `Accept_Recommendation=1` and all others=0. This structured generation approach ensures label consistency and seamless integration with supervised intent classification models. The same template structure is adapted for other intents, with corresponding examples and tailored instructions.

This results in a final dataset with 330+ labeled user inputs, where the label is a binary vector over the intent classes. The distribution of label combinations is varied, with between 20 and 54 examples for over 10 unique intent combinations. The dataset is then split into 80% for training and 10% for validation and testing each. We fine-tune BERT on a MacBook with Apple M2 and 16GB RAM, freezing all but the last two layers to reduce overfitting given the limited dataset size. During inference, we apply a decision threshold of 0.5 for BERT and BART-large-MNLI, while LLMs are not provided an explicit threshold.

4.1.1. Metrics

Four key metrics are used to evaluate the performance of multi-label classification models: Accuracy, Precision, Recall, and F1-score [11, 1]. Accuracy (subset accuracy) measures the proportion of samples where the predicted label set matches the ground truth label set. Precision reflects the proportion of correctly predicted labels among all predicted labels, while recall measures the proportion of actual labels that are correctly predicted. The F1-score combines precision and recall using the harmonic mean, offering a balanced view of the model's performance [1]. To account for multiple labels, micro-averaging is applied when calculating precision, recall, and F1-score.

4.1.2. Results

Table 1
Evaluation results for user intent classification using various models. The best overall performance is achieved by GPT-4o-mini in the few-shot setting, outperforming both the fine-tuned BERT and zero-shot baselines across all metrics. **Bold** is used to denote the best-performing results across all configurations.

Model	Accuracy	Precision	Recall	F1-score
Fine-tuned BERT sequence classifier	0.68	**0.91**	0.85	0.88
BART-large-MNLI (zero-shot)	0.03	0.32	0.67	0.43
GPT-4o-mini (zero-shot)	0.35	0.67	0.69	0.68
GPT-4o-mini (few-shot)	**0.74**	0.87	**0.96**	**0.91**

Our findings in Table 1 reveal that few-shot classification with LLMs outperforms zero-shot classification, with *GPT-4o-mini* notably achieving the highest score across most metrics. For the F1-score particularly, we observed an improvement of 34% for *GPT-4o-mini*, reinforcing the effectiveness of having clear intent descriptions and few-shot examples to guide the models. This aligns with our expectations, as few-shot examples cover several edge cases present in our dataset, which a pre-trained model may not capture. Sequence classification using BERT also emerges as an effective approach for a smaller model size, with the highest precision of 91% and an F1-score of 88%, trailing just behind *GPT-4o-mini* (few-shot). This shows that, with a sufficiently large and diverse training dataset, smaller models like BERT can compete with, or potentially outperform, few-shot classification using LLMs. One of the common sources of inaccuracies we observed with LLMs is the misinterpretation of ambiguous user utterances as preference elicitation.

4.2. RAG Pipeline Evaluation

The RAG pipeline plays a central role in our C-TRS, both when responding to user inquiries and when generating recommendations. In this section, we focus on evaluating the former, as recommendations often lack a clear reference context for objective comparison. Since our primary contribution lies in the retrieval stage, we restrict our evaluation primarily to this component rather than the entire pipeline. Furthermore, the open-ended nature of the *Recommend and Explain* action is better suited to a comprehensive user study, which we leave for future work.

We design an offline experiment to compare different vector search approaches, evaluating both the retrieval and generation quality. LLM-judge-based frameworks have emerged as a promising tool for evaluating RAG pipelines, with research by Zheng et al. [47] reporting over 80% agreement between human and LLM judgements for models like *GPT-4* [47]. This approach enables a scalable and cost-effective evaluation. Therefore, we utilize the state-of-the-art RAGAS framework proposed by Es et al. [16]. RAGAS employs a stronger LLM-judge to assess the response generated by a weaker model, while taking the question, ground truth, and retrieved context into account.

The evaluation dataset consists of 50 synthetically generated Q&A pairs from the Wikivoyage corpus using the RAGAS `TestsetGenerator`[6]. The questions span a diverse range of user personas and question types, covering articles for the following 5 cities with 10 questions for each: Amsterdam, Munich, Istanbul, Madrid, and Zurich. We restrict the query category to `single-hop specific queries`, which can be answered in one retrieval step from a single document, as the metadata filtering step may exclude relevant context for cross-city comparison questions [43].

For our evaluation, we employ *GPT-4o-mini* as the evaluator LLM and set the corresponding city names as metadata filters prior to vector search. *Llama-3.1-8B-Instruct* is used as the generator LLM throughout the evaluation. Additionally, we fix the top_k value to 5 and apply a cross-encoder as the reranker.

4.2.1. Metrics

We use the following RAGAS metrics to evaluate the retrieval (context recall and precision) and generation (faithfulness and answer relevancy) stages of the RAG pipeline [16]:

- **Context Recall**: Measures the proportion of relevant chunks needed to arrive at the ground truth that are actually retrieved.
- **Context Precision**: Measures the proportion of the retrieved chunks that are relevant for arriving at the ground truth.
- **Faithfulness**: Measures the extent to which statements generated by the LLM can be inferred from the retrieved chunks. A lower score indicates hallucination in the generated response.
- **Answer Relevancy**: Measures how relevant the response generated by LLM is to the user query.

4.2.2. Results

Table 2 presents the evaluation of different retrieval strategies using four core metrics: **Context Recall**, **Context Precision**, **Faithfulness**, and **Answer Relevancy**. Among all configurations, **Sparse Search + Rerank** achieves the highest Context Recall (0.77) and Context Precision (0.83), indicating its strong ability to retrieve chunks that are both comprehensive and relevant to the ground truth. This makes it particularly well-suited for scenarios where accurate and complete contextual grounding is essential.

In contrast, **Hybrid Search** without reranking yields the highest Faithfulness score (0.81), suggesting that this method is most effective in reducing hallucinations by ensuring that generated responses are well-supported by the retrieved content. When it comes to user-centric evaluation, **Hybrid Search + Rerank** achieves the best performance in Answer Relevancy (0.90), implying that it delivers the most query-relevant responses, even though it shows a slight drop in recall. Notably, reranking improves

[6] https://docs.ragas.io/en/stable/getstarted/rag_testset_generation/

Table 2
Evaluation of different vector search strategies for RAG retrieval using RAGAS metrics. Results are reported for context recall, context precision, faithfulness, and answer relevancy. **Bold** indicates the best-performing score for each metric across all configurations.

Vector Search Type	Context Recall	Context Precision	Faithfulness	Answer Relevancy
Dense Search	0.62	0.66	0.79	0.83
Dense Search + Rerank	0.62	0.68	0.75	0.81
Sparse Search	0.76	0.82	0.77	0.83
Sparse Search + Rerank	**0.77**	**0.83**	0.76	0.82
Hybrid Search	0.73	0.73	**0.81**	0.89
Hybrid Search + Rerank	0.68	0.75	0.79	**0.90**

precision across all methods by filtering out less relevant chunks, though this sometimes comes at the cost of reduced recall.

Finally, **Dense Search** underperforms relative to sparse and hybrid methods across all metrics, highlighting its limitations in this task. Overall, we observe that sparse retrieval excels in key retrieval metrics, while hybrid retrieval performs best for generation quality. Hybrid approaches offer a balanced trade-off, making them particularly promising as we extend support for more complex, multi-hop, or abstract queries in future work.

5. Conclusion and Future Work

In this paper, we introduced a modular Hybrid RAG-based Conversational Tourism Recommender System (C-TRS) for recommending European cities. Our system integrates LLMs with hybrid dense-sparse retrieval and a structured dialogue state tracker to support real-time, multi-turn interactions. By combining intent classification, dynamic retrieval strategies, and grounded response generation, the system delivers contextually relevant recommendations and factual answers tailored to user preferences.

Through offline evaluation, we demonstrated that few-shot prompting with LLMs significantly improves multi-label user intent classification over traditional models. Moreover, our hybrid retrieval strategy—conditioned on conversational intent—balances recall and precision, improving answer relevance and reducing hallucinations compared to static or single-strategy retrieval methods.

Despite promising initial results, several aspects of our approach can be further improved. First, our evaluation revealed that few-shot prompting with LLMs for intent extractions can struggle with ambiguous or complex user utterances. Future work could explore more robust prompting strategies, such as chain-of-thought (CoT) prompting [45], or fine-tuning smaller models for domain-specific task [35]. Second, our current evaluation relies solely on model-based metrics. Conducting a comprehensive user study would provide deeper insights into user satisfaction and recommendation quality, while also helping refine the taxonomy of user intents and recommender actions. Moreover, our current evaluation is limited by the use of a relatively small synthetic intent dataset and single-hop queries, which may not fully reflect the complexity of real-world conversational recommendation. Future work could expand evaluation to larger datasets and multi-hop queries, and consider ablation study to better understand the contribution of intent-driven retrieval. Lastly, the current system lacks guardrails for moderating input and output, which are essential for ensuring safety, reliability, and responsible behavior in real-world deployment. Addressing these limitations will be key to advancing this framework toward scalable, trustworthy, and user-aligned conversational recommender systems.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT (OpenAI) and Grammarly to correct grammar and spelling inconsistencies and to improve the clarity of the text. ChatGPT was also used for code snippet suggestions during system development. We have critically reviewed and revised all GenAI outputs to ensure that accuracy and originality are maintained, and we accept full responsibility for the content presented in this draft.

References

[1] *3.4. Metrics and Scoring: Quantifying the Quality of Predictions.* scikit-learn. URL: https://scikit-learn/stable/modules/model_evaluation.html (visited on 01/28/2025).

[2] M. Arslan, H. Ghanem, S. Munawar, and C. Cruz. “A Survey on RAG with LLMs”. In: *Procedia computer science* 246 (2024), pp. 3781–3790.

[3] A. Banerjee, T. Mahmudov, E. Adler, F. N. Aisyah, and W. Wörndl. *Modeling Sustainable City Trips: Integrating CO2e Emissions, Popularity, and Seasonality into Tourism Recommender Systems.* Sept. 17, 2024. arXiv: 2403.18604 [cs]. URL: http://arxiv.org/abs/2403.18604 (visited on 10/01/2024). Pre-published.

[4] A. Banerjee, A. Satish, F. N. Aisyah, W. Wörndl, and Y. Deldjoo. “SynthTRIPs: A Knowledge-Grounded Framework for Benchmark Data Generation for Personalized Tourism Recommenders”. In: *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval.* 2025, pp. 3743–3752.

[5] A. Banerjee, A. Satish, and W. Wörndl. “Enhancing Tourism Recommender Systems for Sustainable City Trips Using Retrieval-Augmented Generation”. In: *Recommender Systems for Sustainability and Social Good.* Ed. by L. Boratto, A. De Filippo, E. Lex, and F. Ricci. Cham: Springer Nature Switzerland, 2025, pp. 19–34. ISBN: 978-3-031-87654-7.

[6] K. Bhardwaj, R. S. Shah, and S. Varma. “Pre-training LLMs using human-like development data corpus”. In: *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*. Ed. by A. Warstadt, A. Mueller, L. Choshen, E. Wilcox, C. Zhuang, J. Ciro, R. Mosquera, B. Paranjabe, A. Williams, T. Linzen, and R. Cotterell. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 339–345. DOI: 10.18653/v1/2023.conll-babylm.30.

[7] W. Cai and L. Chen. “Predicting User Intents and Satisfaction with Dialogue-based Conversational Recommendations”. In: *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization.* UMAP ’20: 28th ACM Conference on User Modeling, Adaptation and Personalization. Genoa Italy: ACM, July 7, 2020, pp. 33–42. ISBN: 978-1-4503-6861-2. DOI: 10.1145/3340631.3394856.

[8] W. Cai and L. Chen. “Predicting user intents and satisfaction with dialogue-based conversational recommendations”. In: *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization.* 2020, pp. 33–42.

[9] A. Cassani, M. Ruberl, A. Salis, G. Giannese, and G. Boanelli. *zIA: a GenAI-powered local auntie assists tourists in Italy.* 2024. arXiv: 2407.11830 [cs.DC].

[10] A. Chen, X. Ge, Z. Fu, Y. Xiao, and J. Chen. *TravelAgent: An AI Assistant for Personalized Travel Planning.* 2024. arXiv: 2409.08069 [cs.AI].

[11] *Classification: Accuracy, Recall, Precision, and Related Metrics | Machine Learning.* Google for Developers. 2025. URL: https://developers.google.com/machine-learning/crash-course/classification/accuracy-precision-recall (visited on 01/28/2025).

[12] G. V. Cormack, C. L. A. Clarke, and S. Buettcher. “Reciprocal Rank Fusion Outperforms Condorcet and Individual Rank Learning Methods”. In: *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval.* SIGIR ’09: The 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval. Boston MA USA: ACM, July 19, 2009, pp. 758–759. ISBN: 978-1-60558-483-6. DOI: 10.1145/1571941.1572114.

[13] Y. Deldjoo, Z. He, J. Mcauley, A. Korikov, S. Sanner, A. Ramisa, R. Vidal, M. Sathiamoorthy, A. Kasrizadeh, S. Milano, and F. Ricci. “Recommendation with Generative Models”. In: *Foundations and Trends® in Information Retrieval* (Sept. 2024), pp. 1–120.

[14] Y. Deldjoo, Z. He, J. McAuley, A. Korikov, S. Sanner, A. Ramisa, R. Vidal, M. Sathiamoorthy, A. Kasirzadeh, and S. Milano. “A Review of Modern Recommender Systems Using Generative Models (Gen-RecSys)”. In: *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. KDD ’24. Barcelona, Spain: Association for Computing Machinery, 2024, pp. 6448–6458. isbn: 9798400704901. doi: 10.1145/3637528.3671474.

[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. 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)*. Ed. by J. Burstein, C. Doran, and T. Solorio. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. doi: 10.18653/v1/N19-1423.

[16] S. Es, J. James, L. Espinosa Anke, and S. Schockaert. “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*. Ed. by N. Aletras and O. De Clercq. St. Julians, Malta: Association for Computational Linguistics, Mar. 2024, pp. 150–158. doi: 10.18653/v1/2024.eacl-demo.16.

[17] T. Formal, C. Lassance, B. Piwowarski, and S. Clinchant. “From Distillation to Hard Negative Sampling: Making Sparse Neural IR Models More Effective”. In: *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’22. Madrid, Spain: Association for Computing Machinery, 2022, pp. 2353–2359. isbn: 9781450387323. doi: 10.1145/3477495.3531857.

[18] L. Friedman, S. Ahuja, D. Allen, Z. Tan, H. Sidahmed, C. Long, J. Xie, G. Schubiner, A. Patel, H. Lara, B. Chu, Z. Chen, and M. Tiwari. *Leveraging Large Language Models in Conversational Recommender Systems*. 2023. arXiv: 2305.07961 [cs.IR].

[19] C. Gao, W. Lei, X. He, M. de Rijke, and T.-S. Chua. “Advances and challenges in conversational recommender systems: A survey”. In: *AI Open* 2 (2021), pp. 100–126. issn: 2666-6510. doi: 10.1016/j.aiopen.2021.06.002.

[20] Y. Hou, Y. Lai, Y. Wu, W. Che, and T. Liu. “Few-shot learning for multi-label intent detection”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 14. 2021, pp. 13036–13044.

[21] J. Jin, X. Chen, F. Ye, M. Yang, Y. Feng, W. Zhang, Y. Yu, and J. Wang. “Lending Interaction Wings to Recommender Systems with Conversational Agents”. In: *Advances in Neural Information Processing Systems*. Ed. by A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine. Vol. 36. Curran Associates, Inc., 2023, pp. 27951–27979.

[22] H. Joko, S. Chatterjee, A. Ramsay, A. P. de Vries, J. Dalton, and F. Hasibi. “Doing Personal LAPS: LLM-Augmented Dialogue Construction for Personalized Multi-Session Conversational Search”. In: *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. July 10, 2024, pp. 796–806. doi: 10.1145/3626772.3657815. arXiv: 2405.03480 [cs].

[23] S. Kemper, J. Cui, K. Dicarlantonio, K. Lin, D. Tang, A. Korikov, and S. Sanner. “Retrieval-Augmented Conversational Recommendation with Prompt-based Semi-Structured Natural Language State Tracking”. In: *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR 2024: The 47th International ACM SIGIR Conference on Research and Development in Information Retrieval. Washington DC USA: ACM, July 10, 2024, pp. 2786–2790. isbn: 9798400704314. doi: 10.1145/3626772.3657670.

[24] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, et al. “Retrieval-augmented generation for knowledge-intensive nlp tasks”. In: *Advances in neural information processing systems* 33 (2020), pp. 9459–9474.

[25] S. Li, R. Xie, Y. Zhu, X. Ao, F. Zhuang, and Q. He. “User-centric conversational recommendation with multi-aspect user modeling”. In: *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2022, pp. 223–233.

[26] B. Liu, J. Ge, and J. Wang. *Vaiage: A Multi-Agent Solution to Personalized Travel Planning*. 2025. arXiv: 2505.10922 [cs.MA].

[27] H. Liu, S. Zhao, and X. Zhang. “Few-Shot Intent Detection with Label-Enhanced Hierarchical Feature Learning and Graph Neural Networks”. In: *Proceedings of the ACM Turing Award Celebration Conference-China 2024*. 2024, pp. 226–227.

[28] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, and P. Liang. “Lost in the Middle: How Language Models Use Long Contexts”. In: *Transactions of the Association for Computational Linguistics* 12 (2024), pp. 157–173. DOI: 10.1162/tacl_a_00638.

[29] S. Lyu, A. Rana, S. Sanner, and M. R. Bouadjenek. “A Workflow Analysis of Context-driven Conversational Recommendation”. In: *Proceedings of the Web Conference 2021*. WWW ’21: The Web Conference 2021. Ljubljana Slovenia: ACM, Apr. 19, 2021, pp. 866–877. ISBN: 978-1-4503-8312-7. DOI: 10.1145/3442381.3450123.

[30] D. Massimo and F. Ricci. “Building Effective Recommender Systems for Tourists”. In: *AI Magazine* 43.2 (June 2022), pp. 209–224. ISSN: 0738-4602, 2371-9621. DOI: 10.1002/aaai.12057.

[31] S. Meyer, S. Singh, B. Tam, C. Ton, and A. Ren. *A Comparison of LLM Finetuning Methods & Evaluation Metrics with Travel Chatbot Use Case*. 2024. arXiv: 2408.03562 [cs.CL].

[32] S. Moradizeyveh. *Intent Recognition in Conversational Recommender Systems*. 2022. arXiv: 2212.03721 [cs.CL].

[33] F. Nagy, A. Haroun, H. Abdel-Kader, and A. Keshk. “A Review for Recommender System Models and Deep Learning”. In: *IJCI. International Journal of Computers and Information* 8.2 (Dec. 1, 2021), pp. 170–176. ISSN: 2735-3257. DOI: 10.21608/ijci.2021.207864.

[34] OpenAI et al. *GPT-4 Technical Report*. 2024. arXiv: 2303.08774 [cs.CL].

[35] S. Parikh, M. Tiwari, P. Tumbade, and Q. Vohra. “Exploring Zero and Few-shot Techniques for Intent Classification”. In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*. Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track). Toronto, Canada: Association for Computational Linguistics, 2023, pp. 744–751. DOI: 10.18653/v1/2023.acl-industry.71.

[36] J. Park, S. Kim, and S. Lee. “A user preference and intent extraction framework for explainable conversational recommender systems”. In: *Companion Proceedings of the 2023 ACM SIGCHI Symposium on Engineering Interactive Computing Systems*. 2023, pp. 16–23.

[37] J. Qi, S. Yan, Y. Zhang, W. Zhang, R. Jin, Y. Hu, and K. Wang. “RAG-Optimized Tibetan Tourism LLMs: Enhancing Accuracy and Personalization”. In: *Proceedings of the 2024 7th International Conference on Artificial Intelligence and Pattern Recognition*. AIPR ’24. Association for Computing Machinery, 2025, pp. 1185–1192. ISBN: 9798400717178. DOI: 10.1145/3703935.3704112.

[38] J. Rao and J. Lin. “RAMO: Retrieval-Augmented Generation for Enhancing MOOCs Recommendations”. In: *Joint Proceedings of the Human-Centric eXplainable AI in Education and the Leveraging Large Language Models for Next Generation Educational Technologies Workshops (HEXED-L3MNGET 2024) co-located with 17th International Conference on Educational Data Mining (EDM 2024)*. Vol. 3840. CEUR Workshop Proceedings. CEUR-WS.org, 2024.

[39] A. Sauer, S. Asaadi, and F. Küch. “Knowledge distillation meets few-shot learning: An approach for few-shot intent classification within and across domains”. In: *Proceedings of the 4th Workshop on NLP for Conversational AI*. 2022, pp. 108–119.

[40] O. Şerbetçi, X. D. Wang, and U. Leser. "HU-WBI at BioASQ12B Phase A: Exploring Rank Fusion of Dense Retrievers and Re-rankers". In: *Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2024).* Vol. 3740. CEUR Workshop Proceedings. 2024, pp. 269–275.

[41] S. Song, C. Yang, L. Xu, H. Shang, Z. Li, and Y. Chang. "TravelRAG: A Tourist Attraction Retrieval Framework Based on Multi-Layer Knowledge Graph". In: *ISPRS International Journal of Geo-Information* 13.11 (2024). ISSN: 2220-9964. DOI: 10.3390/ijgi13110414.

[42] G. Team et al. *Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context.* 2024. arXiv: 2403.05530 [cs.CL].

[43] *Testset Generation for RAG - Ragas.* URL: https://docs.ragas.io/en/stable/concepts/test_data_generation/rag/ (visited on 07/21/2025).

[44] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou. "MINILM: deep self-attention distillation for task-agnostic compression of pre-trained transformers". In: *Proceedings of the 34th International Conference on Neural Information Processing Systems.* NIPS '20. Vancouver, BC, Canada: Curran Associates Inc., 2020. ISBN: 9781713829546.

[45] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. H. Chi, Q. V. Le, and D. Zhou. "Chain-of-thought prompting elicits reasoning in large language models". In: *Proceedings of the 36th International Conference on Neural Information Processing Systems.* NIPS '22. New Orleans, LA, USA: Curran Associates Inc., 2022. ISBN: 9781713871088.

[46] W. Yin, J. Hay, and D. Roth. "Benchmarking Zero-shot Text Classification: Datasets, Evaluation and Entailment Approach". 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).* Ed. by K. Inui, J. Jiang, V. Ng, and X. Wan. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 3914–3923. DOI: 10.18653/v1/D19-1404.

[47] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. P. Xing, H. Zhang, J. E. Gonzalez, and I. Stoica. "Judging LLM-as-a-judge with MT-bench and Chatbot Arena". In: *Proceedings of the 37th International Conference on Neural Information Processing Systems.* NIPS '23. New Orleans, LA, USA: Curran Associates Inc., 2023.