SynthTRIPs: A Knowledge-Grounded Framework for Benchmark Query Generation for Personalized Tourism Recommenders

Ashmi Banerjee
ashmi.banerjee@tum.de
Technical University of Munich
Munich, Germany

Adithi Satish
adithi.satish@tum.de
Technical University of Munich
Munich, Germany

Fitri Nur Aisyah
fitri.aisyah@tum.de
Technical University of Munich
Munich, Germany

Wolfgang Wörndl
woerndl@in.tum.de
Technical University of Munich
Munich, Germany

Yashar Deldjoo
yashar.deldjoo@poliba.it
Polytechnic University of Bari
Bari, Italy

Abstract

Tourism Recommender Systems (TRS) are crucial in personalizing travel experiences by tailoring recommendations to users' preferences, constraints, and contextual factors. However, publicly available travel datasets often lack sufficient breadth and depth, limiting their ability to support advanced personalization strategies — particularly for sustainable travel and off-peak tourism. In this work, we explore using Large Language Models (LLMs) to generate synthetic travel queries that emulate diverse user personas and incorporate structured filters such as budget constraints and sustainability preferences.

This paper introduces a novel SynthTRIPs framework for generating synthetic travel queries using LLMs grounded in a curated knowledge base (KB). Our approach combines persona-based preferences (e.g., budget, travel style) with explicit sustainability filters (e.g., walkability, air quality) to produce realistic and diverse queries. We mitigate hallucination and ensure factual correctness by grounding the LLM responses in the KB. We formalize the query generation process and introduce evaluation metrics for assessing realism and alignment. Both human expert evaluations and automatic LLM-based assessments demonstrate the effectiveness of our synthetic dataset in capturing complex personalization aspects underrepresented in existing datasets. While our framework was developed and tested for personalized city trip recommendations, the methodology applies to other recommender system domains.

Code and dataset are made public at https://bit.ly/synthTRIPs

CCS Concepts

• **Information systems** → **Information retrieval**; • **Computing methodologies** → **Artificial intelligence**.

Keywords

Large Language Models, Synthetic Data Generation, Personalization, Tourism Recommender Systems

SIGIR '25, Padua, Italy

ACM ISBN 979-8-4007-1592-1/2025/07
https://doi.org/10.1145/3726302.3730321

ACM Reference Format:
Ashmi Banerjee, Adithi Satish, Fitri Nur Aisyah, Wolfgang Wörndl, and Yashar Deldjoo. 2025. SynthTRIPs: A Knowledge-Grounded Framework for Benchmark Query Generation for Personalized Tourism Recommenders . In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '25), July 13–18, 2025, Padua, Italy.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3726302.3730321

1 Introduction

Recommender Systems have become indispensable in helping users sift through massive amounts of information in domains such as e-commerce, media streaming, and travel [16]. In the travel domain, personalization is especially valuable: user choices depend on individual preferences (e.g., budget, travel style, sustainability concerns) and contextual factors (destination popularity, local activities, etc.) [3]. Consequently, **high-quality training data** that reflects real-world travel preferences is essential for building effective and robust Tourism Recommender Systems (TRS).

However, publicly available travel datasets are often limited in **breadth** (e.g., focusing only on popular cities) and **depth** (e.g., missing nuanced preferences like climate consciousness, walkability, or budget constraints). The lack of richly annotated data hinders research on advanced personalization strategies incorporating specialized filters — such as sustainable travel or off-peak tourism.

Large Language Models (LLMs) — such as GPT-4, PaLM, and LLaMA — provide a powerful way to generate human-like text. They have performed remarkably in question-answering, summarization, and creative writing. In the travel context, LLMs can:

(1) Emulate diverse *user personas* (e.g., a budget-conscious student vs. a sustainability-focused family).
(2) Incorporate *structured filters* (e.g., budget limits and environmental impact considerations).
(3) Generate *natural-sounding requests* that integrate factual knowledge (e.g., city highlights, monthly visitor footfall).

However, LLMs are prone to *hallucination* when asked for domain-specific details. To mitigate this, we propose *grounding* LLM responses in curated knowledge bases (KB) to ensure factual correctness regarding destinations, costs, or sustainability metrics.

Given that travel data with rich user and contextual features is notoriously difficult, we aim to share the insights as a *resource paper*. In particular, our goal is to share *(i)* a synthetic dataset that

captures realistic travel constraints; *(ii)* a generation pipeline that enables flexible updates (new personas, destinations, or sustainability constraints); and *(iii)* reproducible evaluation protocols. We hope this will help to address the pressing community need for better coverage, deeper personalization, and a more robust domain grounding in tourism recommendations.

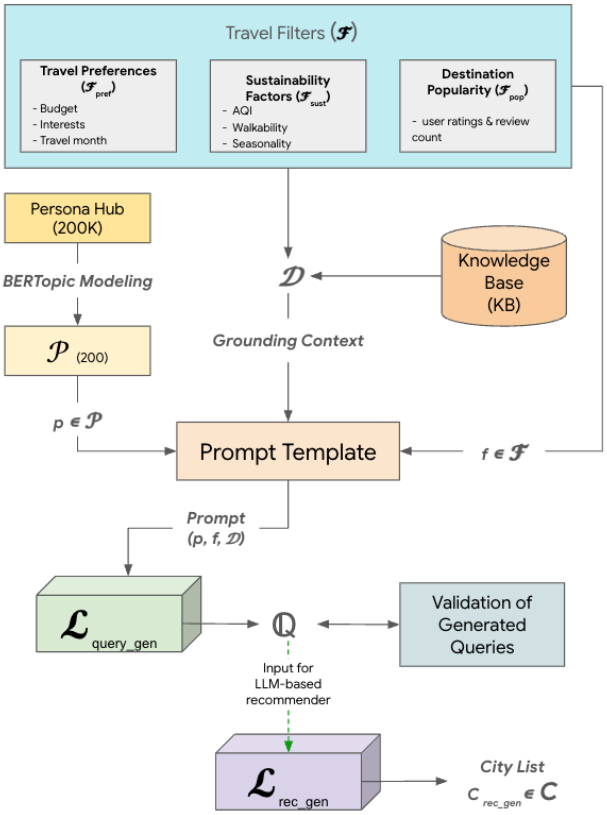

Fig. 1: Our proposed framework for generating synthetic data using LLMs for personalized, sustainable city trips.

Below are some example user queries for a sustainable city trip, ensuring factual accuracy by grounding in a verified KB.

- **Q1.** *"Find a low-budget, walkable city in Europe with unusual museums or a hidden, alternative nightlife scene."*
- **Q2.** *"Quiet European coastal city with good air quality, affordable, not touristy, with interesting nightlife options."*
- **Q3.** *"Best European cities for unique, artistic experiences and independent cinema, avoiding mainstream tourist attractions?"*

Such queries capture complex **personalization** aspects (budget, interests, sustainability) that are typically underrepresented in publicly available travel datasets [36, 38]. *As evident from Figure 1, our resource generation framework, SynthTRIPs, automates the creation of such queries at scale, pairing persona definitions with domain-specific constraints to produce a rich corpus for training and benchmarking TRS models.*

Proposed Resource and Contributions

This paper presents **SynthTRIPs**, a novel resource for building sustainable, persona-aware travel data via LLMs. SynthTRIPs addresses a critical gap in the tourism recommendation community by offering:

- **A reproducible pipeline for large-scale synthetic data generation** rooted in a curated KB that ensures factual grounding. By focusing on European city trips and integrating sustainability metrics such as walkability, off-peak travel, and popularity constraints, SynthTRIPs provides a versatile blueprint adaptable to diverse travel contexts.
- **Open-source datasets and code**: We publicly release all components, including:
  - **Data**: (1) A structured KB containing real-world attributes of European cities, (2) Generated queries that integrate filters (budget, timing, popularity, sustainability)
  - **Code for Query Generation**: (1) Prompt templates leveraging In-Context Learning (ICL) with various persona definitions, (2) Colab notebooks for straightforward replication of the entire pipeline.
  - **Code for Evaluation**: (1) Automated LLM-based evaluation measuring groundedness, factual correctness, persona alignment, and sustainability compliance, (2) Tools for expert (human) evaluation for clarity, diversity, and overall rating.
- **A flexible framework for future expansions**: Researchers can easily adapt SynthTRIPs to other regions or thematic focuses (beyond sustainability) by updating the underlying KB and modifying persona definitions or filter sets.

By offering a complete methodology for generating data and robust evaluation metrics, SynthTRIPs opens new avenues for more **personalized, sustainability-focused** travel recommendations.

Our paper is organized as follows: Section 2 examines existing gaps in the literature and relevant work, while Section 3 provides a detailed formal description and methodology of our query generation framework. Section 4 subsequently presents our primary resources along with their key statistics. Following that, Section 5 outlines the various validation dimensions utilized to evaluate the quality of our generated queries. Finally, Section 6 concludes the paper, discussing its limitations and suggestions for future research.

2 Related Work

Early travel recommenders primarily relied on historical booking logs and user reviews [6, 24, 25]. While these sources offer insights into popular destinations, they rarely include explicit data for *sustainability preferences* or off-peak travel. Subsequent works have integrated *knowledge graphs* to enrich semantic understanding of destination attributes [20], yet they often lack *diverse user persona data*. A popular and effective way of considering the nuances of user preferences is through Conversational Recommender Systems, which provide more interactive recommendations compared to traditional recommenders [18]. Table 1 provides an overview of some of the datasets in this field, across domains like travel [29, 36, 38] and food [40]. However, the dearth of data to evaluate this system is a persistent problem, especially with the time-consuming and resource-intensive nature of manual data collection and annotation.

Table 1: An overview of the existing datasets and benchmarks as well as SynthTRIPs.

Dataset	Domain	Goal/Task	Description	Size (#queries)	LLMs/Model Used	Data Sources
reddit-travel-QA-finetuning [29]	Travel	Question Answering + Finetuning	Real-time travel discussions (posts and top comments) on top 100 travel-related subreddits	10,000	N/A	Reddit
TravelPlanner [38]	Travel	Travel Plan Generation	Synthetically generated queries and travel itineraries using LLMs, including constraints such as cuisine, accommodation, and transport	1,225	GPT-4	Scraped from the Internet
TravelDest [36]	Travel	City Trip Recommendations	Broad and indirect queries spanning over 700 cities	50	Manually Generated	Wikivoyage
FeB4RAG [34]	General	Federated Search	Information requests collected from the BEIR subcollections across multiple domains	790	GPT-4	BEIR Datasets [33]
Recipe-MPR [40]	Food	Multi-Aspect Recommendation	Manually annotated preference-based queries that contain multi-item descriptions to benchmark conversational recommender systems	500	Manually Generated	FoodKG [14], Recipe1M+ [28]
SynthTRIPs	**Travel**	**City Trip Recommendations**	**Personalized travel queries generated synthetically covering travel filters such as budget, interests integrated with additional sustainability features**	**4,604**	**Llama-3.2-90B, Gemini-1.5-Pro**	**Wikivoyage, Nomadlist, Where And When, Tripadvisor**

Meanwhile, with the recent advancements in LLMs and their ability to generate human-like conversations, there is a growing field of study that explores their potential in simulating users and their behavior [1, 2, 17, 21, 26, 35, 42]. LLM-based synthetic data generation has gained traction in domains such as healthcare [32] and education [19], demonstrating the potential to automatically produce large-scale, high-quality text data.

Nonetheless, existing approaches either (1) do not specifically tailor synthetic data to *travel* use cases, like FeB4RAG [34], or (2) do not incorporate robust *sustainability and personalization* constraints. Furthermore, naive LLM-based query generation may lead to factual errors unless carefully grounded.

Some studies have used LLMs to generate data for TRS, but they are often broad queries that do not necessarily involve a user asking for city trip recommendations. For example, the Reddit Q&A dataset introduced by Meyer et al. [29] consists of 10,000 posts and comments taken from various travel-related subreddits, covers a broad spectrum of travel-related queries, including but not limited to visas, required vaccines, and itinerary suggestions. TravelDest, a benchmark introduced [36], consists of 50 broad and indirect textual queries, which are insufficient to train algorithms. In tourism, Xie et al. [38] uses LLMs to generate synthetic travel queries for origins and destinations in the United States to annotate and benchmark travel itineraries. However, TravelPlanner [38] only focuses on testing the planning capabilities, so it doesn't consider the persona aspect of the query. All the queries sound similar and are in the same template, so we can't test preference extraction capabilities on this dataset.

In contrast, our work directly addresses these **gaps** by:

- **Combining** persona-based preferences (budget, special interests) with explicit *sustainability filters*.
- **Grounding** each query in a KB to reduce hallucination.
- **Introducing** a formal framework and evaluation metrics for query realism and alignment.

3 SynthTRIPs: Query Generation System

We now describe the formal design of SynthTRIPs and illustrate how LLMs are leveraged to generate rich, realistic user queries. Our system is organized around three core modules: *(1) Persona Hub*, *(2) Travel Filters*, and *(3) Contextual Prompting with KB Grounding*, as shown in Figure 1.

3.1 Formal Description

Personas: Prior research has shown that persona-guided LLMs can generate more diverse annotations than standard LLMs, improving the variability of LLM-based data annotation [9, 10]. Hence, to include personalization in our setup, we use personas from the PersonaHub[1] dataset, containing 200,000 unique personas. This dataset is a repository of diverse user profiles (e.g., student, family, digital nomad), where each profile includes attributes such as age, interests, and environmental concerns.

To ensure broader coverage and reduce redundancy, we apply **BERTopic modeling** [12] using `all-MiniLM-L6-v2` embeddings on 196,693 English personas, clustering them into 201 topics (with one outlier group). We then select the highest probability persona from each cluster, resulting in a refined set of 200 diverse and representative personas.

Let $p \in \mathcal{P}$, where $\mathcal{P} = \{p_1, p_2, \ldots, p_n\}$ where $n = 200$, denotes this refined set of personas, and let $f \in \mathcal{F}$ be a set of travel filters (explained below).

[1] https://huggingface.co/datasets/proj-persona/PersonaHub

Knowledge Base (KB). We denote our knowledge base of European cities and their attributes as:

$$KB = (C, A, V),$$

Where:

- C is the set of *city entities* (e.g., "Amsterdam," "Berlin," "Budapest"),
- A is the set of *attribute types* relevant for city travel (e.g., "walkability index," "average cost," "peak season," "AQI"),
- $V : C \times A \rightarrow \mathcal{D}$ is a partial function mapping city-attribute pairs to a value domain $\mathcal{D}$ (e.g., real numbers or categorical labels).

We refer to *KB* as containing all factual knowledge used to ground the LLM generations.

Travel Filters: These filters $\mathcal{F}$ are used to retrieve the relevant domain information from *KB* using structured queries. We define the set of travel filters $\mathcal{F}$ as a combination of **travel filters**, **sustainability factors**, and **destination popularity**:

$$\mathcal{F} = \mathcal{F}_{\text{pref}} \cup \mathcal{F}_{\text{sust}} \cup \mathcal{F}_{\text{pop}} \tag{1}$$

Travel filters ($\mathcal{F}_{\text{pref}}$):

$$\mathcal{F}_{\text{pref}} = \{\mathcal{F}_{\text{budget}}, \mathcal{F}_{\text{month}}, \mathcal{F}_{\text{interests}}\} \tag{2}$$

Where:

- $\mathcal{F}_{\text{budget}}$: (low, medium, high),
- $\mathcal{F}_{\text{month}}$: (January–December),
- $\mathcal{F}_{\text{interests}}$: from five categories [31]: *Arts & Entertainment*, *Outdoors & Recreation*, *Food*, *Nightlife Spots*, *Shops & Services*.

Sustainability Factors ($\mathcal{F}_{\text{sust}}$):

$$\mathcal{F}_{\text{sust}} = \{\mathcal{F}_{\text{seasonality}}, \mathcal{F}_{\text{walkability}}, \mathcal{F}_{\text{AQI}}\} \tag{3}$$

Here, $\mathcal{F}_{\text{seasonality}}$ focuses on less crowded (off-peak) months, $\mathcal{F}_{\text{walkability}}$ imposes a minimum walkability index, and $\mathcal{F}_{\text{AQI}}$ checks city air quality thresholds. Sustainable destinations typically prioritize low seasonality, high walkability, and excellent AQI.

Destination Popularity ($\mathcal{F}_{\text{pop}}$): We estimate popularity from the normalized number of Tripadvisor reviews [4]. For a city c, let $R(c)$ be the number of reviews. We define:

$$\text{popularity}(c) = \frac{R(c) - \min\limits_{c' \in C} R(c')}{\max\limits_{c' \in C} R(c') - \min\limits_{c' \in C} R(c')} \tag{4}$$

We then split the normalized scores into three tiers (*low*, *medium*, *high*). We separate popularity from sustainability constraints to ensure a balanced dataset covering all popularity ranges.

Complexity Levels of Filters. Inspired by TravelPlanner [38], we define four levels of complexity:

$$\begin{aligned}
\textbf{Easy:} \quad & f_{\text{easy}} = \{f\}, \quad f \in_R \mathcal{F}_{\text{pref}} \\
\textbf{Medium:} \quad & f_{\text{med}} = \{f_1, f_2\}, \quad f_1, f_2 \in_R \mathcal{F}_{\text{pref}}, f_1 \neq f_2 \\
\textbf{Hard:} \quad & f_{\text{hard}} = \mathcal{F}_{\text{pref}} \\
\textbf{Sustainable:} \quad & f_{\text{sust}} = \{f_1, f_2\} \cup \{s\}, \quad f_1, f_2 \in_R \mathcal{F}_{\text{pref}}, s \in_R \mathcal{F}_{\text{sust}}
\end{aligned}$$

This ensures a mix of simple and more complex queries. Formally, let

$$\mathcal{F}_{\text{complexity}} = \{f_{\text{easy}}, f_{\text{med}}, f_{\text{hard}}, f_{\text{sust}}\}.$$

Overall Key Functions. For each persona $p \in \mathcal{P}$, we generate queries across all four complexity levels and three city popularity tiers (*low*, *medium*, *high*). This yields:

$$|\mathcal{P}| \times |\mathcal{F}_{\text{complexity}}| \times |\mathcal{F}_{\text{pop}}| = 200 \times 4 \times 3 = 2{,}400$$

possible key functions, denoted $\ell(p, f)$.

Retrieval of Grounding Context. When a key function $\ell(p, f)$ is chosen, we run a structured query on *KB*:

$$\mathscr{D} \subset \mathcal{D} \;=\; \text{Retrieve}(KB,\; f),$$

Thus obtaining only the relevant facts (e.g., cities $c_{\mathscr{D}}$ and their corresponding attributes satisfying the filter constraints). If $\text{Retrieve}(KB, f) = \emptyset$, the filter is deemed invalid. This results in 2302 valid key functions for our use case.

Prompt Construction. The context $\mathscr{D}$, along with the persona p and the filter set f, is assembled into a structured prompt:

$$\text{Prompt}\ (p, f, \mathscr{D}).$$

We feed this prompt to a *Query Generator LLM*, denoted $\mathcal{L}_{\text{query\_gen}}$. The resulting final user query is:

$$\mathbb{Q} = \mathcal{L}_{\text{query\_gen}}\left(\text{Prompt}(p, f, \mathscr{D})\right).$$

City Recommendation (Optional). If a subsequent step uses an LLM or any recommender model $\mathcal{L}_{\text{rec\_gen}}$ for city recommendations (given the newly generated user query), we denote that:

$$c_{rec\_gen} \;=\; \mathcal{L}_{\text{rec\_gen}}\ (\mathbb{Q}),$$

Where c_{rec_gen} is the list of recommended cities provided by $\mathcal{L}_{\text{rec\_gen}}$. However, this step is optional for creating our *synthetic user query* dataset since we primarily focus on generating queries in **SynthTRIPs**. Still, the framework naturally extends to a multi-turn pipeline.

3.2 Goal

The goal of SynthTRIPs is to produce a dataset $\mathbb{Q} = \{q_1, q_2, \dots\}$ and accompanying cities in the context $c_{\mathscr{D}}$, ensuring the queries are realistic, personalized, and grounded in factual knowledge.

Example

Consider a configuration where:

- **Persona (p)**: *"A wanderlust-filled trader who appreciates and sells the artisan's creations in different corners of the world"*
- **Filters (f)**:
  - Popularity: Low
  - Interests: Nightlife Spot

Given this configuration, the pipeline generates a variety of queries:

- **Vanilla Query ($q_{\mathbf{v}}$)**: *"Recommend off-the-beaten-path European cities with low popularity for a nightlife-focused trip with a mix of bars, clubs, and live music venues."*
- **Persona-Specific Query 1 ($q_{\mathbf{p0}}$)**: *"Unique nightlife and cultural experiences in off-the-beaten-path European cities for a budget-conscious traveler interested in local artisans."*
- **Persona-Specific Query 2 ($q_{\mathbf{p1}}$)**: *"Which European cities offer a rich cultural heritage, historic centers, and local artisan markets to explore?"*

This example highlights how combining personas and filters shapes the generated queries, providing general (vanilla) and tailored (persona-specific) outputs. We explain the different outputs (vanilla and personalized) in Section 3.4.

3.3 In-Context Learning and Prompt Templates

SynthTRIPs employs few-shot or single-shot **ICL** to guide the LLM to produce queries that are *stylistically aligned* with the persona and *faithful* to the filters. Our prompt template [7] typically includes:

(1) **Task Description**: Instruct the LLM to generate user queries given a specific persona, set of travel filters, and retrieved KB data.
(2) **In-Context Examples**: One or more example queries illustrating the desired style and structure.
(3) **Persona, Filters, & Context**: The explicit persona traits, the chosen filter constraints, and relevant city attributes from the KB.
(4) **Output Requirements**: The final request to produce a single-sentence or multi-sentence user query, possibly with formatting constraints (e.g., no direct city names).

3.4 Response Generation

For each of the 2,302 valid key functions, we experiment with three conditions using *llama-3.2-90b* and *gemini-1.5-pro* as the $\mathcal{L}_{\text{query\_gen}}$ with a *temperature* setting of 0.5 and *top_p* value of 0.95:

- $q_{\mathbf{v}}$**: Vanilla (Non-Personalized)**: Persona information is omitted.
- $q_{\mathbf{p0}}$**: Personalized Zero-Shot**: Persona information is included, but without an example.
- $q_{\mathbf{p1}}$**: Personalized Single-Shot**: Persona information is included along with an example.

Hallucination Mitigation via Grounding

LLMs can produce *hallucinated* statements when lacking access to factual references [15]. To address this in SynthTRIPs, we explicitly **ground** the LLM output by:

- **Strict retrieval from** *KB*: Only validated attributes from *KB* that meet the filter constraints are provided in the prompt.
- **Post-generation checks**: We implement an output parser to ensure properly formatted outputs. When the LLM generates overly verbose responses, we first apply regex-based extraction. If unsuccessful, we use an LLM-based parser using *gemini-1.5-pro* followed by manual verification.

SynthTRIPs significantly reduces inaccuracies and fosters factual consistency through prompt guidance and post-generation verifications.

4 Resources

This section outlines our key resources: the Knowledge Base (Section 4.1) used for query generation and the final dataset (Section 4.2, which includes the generated queries along with their associated information.

4.1 Knowledge Base

Our *KB* construction is inspired by the data sources and methodology outlined in Banerjee et al. [5]. We utilize Wikivoyage's *Listings*[2] data for detailed tabular information on POIs, including places to visit, accommodations, and dining options in 141 unique European cities. The POIs in this dataset are categorized into different activity categories such as *'see'*, *'do'*, *'eat'*, *'drink'*, *'sleep'*, and *'go'*.

To map the wikivoyage listings POIs to the five travel preferences in Section 3.1, we use `bart-large-mnli`, a BART [22] variant trained on the MNLI dataset [37], for zero-shot classification. This pre-trained Natural Language Inference (NLI) model enables sequence classification without additional training [39] and assigns a probability score indicating how well each POI aligns with the defined travel filters. The top three most probable items in each activity sub-category per city are selected to maintain consistency, reduce anomalies, and remain within the LLM's context window.

Publicly available data from Tripadvisor[3] and Where and When (W2W)[4] is utilized to incorporate both popularity and seasonality into the sustainability features of our *KB*. Tripadvisor reviews and opinions represent city popularity, while W2W's monthly visitor footfall data indicates seasonality [5]. Additional cost labels, AQI levels, and walkability data are sourced from Nomadlist[5].

Basic Statistics. The final dataset comprises 200 cities from 43 European countries, formed by combining the cities from the aforementioned four datasets. Analyzing the distribution of popularity in our *KB* shows that 59.16% of the cities are categorized as high popularity, with 29.66% and 11.18% categorized as medium and low popularity respectively. To ensure that this skew in the data does not reflect in our generated queries, we use a popularity-based stratification while generating the configurations such that there is an equal representation of low, medium, and popular cities.

Furthermore, looking into the seasonality reveals that the peak seasons across cities are the summer months in Europe, from May – September. December also sees a slight increase in seasonality, potentially due to the Christmas holiday season, which precedes the decrease in seasonality over the winter. January – March is the low-season months across cities.

Following the classification of the interests into the 5 categories as described in Section 3.1, we observe that the distribution of travel interests in our *KB* is predominantly related to *Arts & Entertainment* and *Food*, comprising of almost 70% of the dataset, followed by *Outdoors & Recreation*, *Nightlife Spot* and *Shops & Services*. These interest categories cover a wide range of points of interest, from museums, galleries, and theaters to restaurants, bars, and natural sights, ensuring diversity of travel interests during query generation.

4.2 Generated Queries

Our framework results in a total of 4,604 diverse queries generated from 2,302 key functions in 3 experimental settings (q_{v}, q_{p0}, and q_{p1}) for each model namely *gemini-1.5-pro* and *llama-3.2-90b* . The

[2]https://github.com/baturin/wikivoyage-listings
[3]https://tripadvisor.com
[4]https://www.whereandwhen.net
[5]https://nomadlist.com

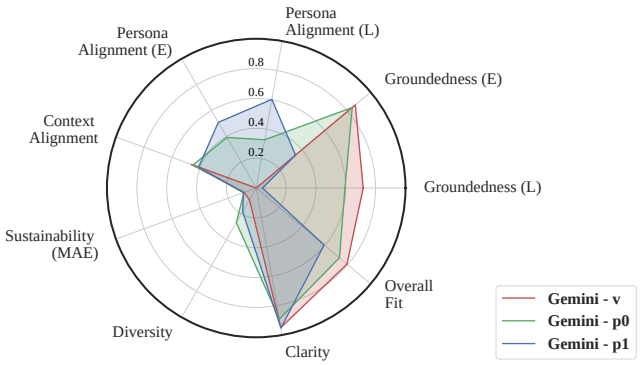

Fig. 2: Radar Chart showing the different dimensions of validation and performance of queries generated by *Gemini*. *Llama* shows similar performance across the dimensions and hence is excluded from the paper. L (E) denotes LLM (Expert) validations.

dataset consists of the mapping between key functions, retrieved-context, and generated queries. The configuration setting includes the selected persona, city popularity, complexity, and travel filters. The retrieved context includes the information used to ground the query ($f \in \mathcal{F}$) and the set of cities from our knowledge base that align with the travel filters ($\mathscr{D} \subset \mathcal{D}$). Additionally, the dataset contains three types of queries, corresponding to each experimental setting.

Basic Statistics. Each query in our generated queries dataset is mapped to an un-ranked list of ground truth cities. To ensure balanced exposure for each city, we stratify our key functions based on the city category described in Section 4.1. This results in a well-proportioned distribution of generated queries: 739 low-popularity cities (32%), 784 medium-popularity cities (34%), and 779 high-popularity cities (33%). In this dataset, we have, on average, 11 cities from our *KB* mapped to a generated query.

5 Validation of the Generated Queries

In this section, we present the validation of the generated queries, focusing primarily on assessing their quality. Additionally, we demonstrate their practical utility by leveraging a recommender LLM, denoted as $\mathcal{L}_{\text{rec\_gen}}$, to generate travel city recommendations.

We validate the generated queries across the following dimensions: groundedness with travel filters (Section 5.2), alignment with context (Section 5.4), alignment with personas (Section 5.3), adherence to sustainability filters (Section 5.5), diversity of generated queries (Section 5.6), and overall clarity and quality assessment (Section 5.7). Additionally, Section 5.1 provides an overview of our experimental setup. Figure 2 shows a radar chart showing the different dimensions of validation and performance of queries generated by *Gemini*.

5.1 Experimental Setup

We use a combination of offline experiments and expert (human) evaluations to assess the quality of the generated queries. The subsequent discussion provides a detailed overview of our experimental setup across these validation criteria.

5.1.1 Offline Experiments. We assess the quality of the generated queries through the following offline experiments: (1) JudgeLLM, or J-LLM (*gpt-4o*) to evaluate alignment with travel filters and persona, (2) semantic similarity for contextual alignment, (3) Mean Average Error (MAE) to measure sustainability alignment, and (4) Self-BLEU scores to assess query diversity.

5.1.2 Expert Evaluation. To ensure that the responses generated by the J-LLM align with human judgments, we conduct an expert evaluation using 60 example queries from each model. Two domain experts independently assess these queries using a custom evaluation tool developed specifically for this study. The tool, built with Python's Streamlit library[6], is fully open-source, responsive, and publicly available on Hugging Face Spaces[7] and can be accessed here. Figure 3 shows a screenshot from a part of this tool.

The queries generated by all three methods — q_{v}, q_{p0}, and q_{p1} — are reviewed by experts using the tool, without knowledge of which model or settings were used for the generation. This approach helps minimize bias. Each expert is also provided with the relevant travel filters and persona for context. One alternative we considered was to ask users to generate realistic queries themselves based on the filters and travel constraints rather than evaluate system-generated ones. However, this would have required recruiting linguists and domain experts capable of producing high-quality, representative queries — resources that are both scarce and difficult to source. Hence, we proceeded with our expert-based evaluation approach. Future work could explore this approach to create more naturalistic evaluation benchmarks.

Our interface features detailed instructions on the evaluation process, initially collapsed for a streamlined experience but expandable for full details. Experts are able to submit responses incrementally, save progress, and resume later, with all responses securely stored in a **Firebase real-time database** [8], ensuring seamless retrieval of the evaluation state. The evaluation focused on multiple dimensions, including alignment with personas, travel filters, clarity, and overall fit, as these aspects are often nuanced and challenging for LLMs to capture.

The following sections provide a detailed analysis of the evaluation outcomes for each query dimension.

5.2 Groundedness with Travel Filters

To measure how well the generated queries represent the input constraints or the travel filters, we measure the groundedness using LLMs as an evaluator [41]. We use *gpt-4o* as our JudgeLLM (J-LLM) to evaluate the queries generated by *llama-3.2-90b* and *gemini-1.5-pro* by providing the generated query and the travel filters and asking it to find the number of filters present in the query, along with an explanation. We then use these matches to compute the Mean Recall (MR) for each query setting as follows:

[6]https://streamlit.io
[7]https://huggingface.co/spaces

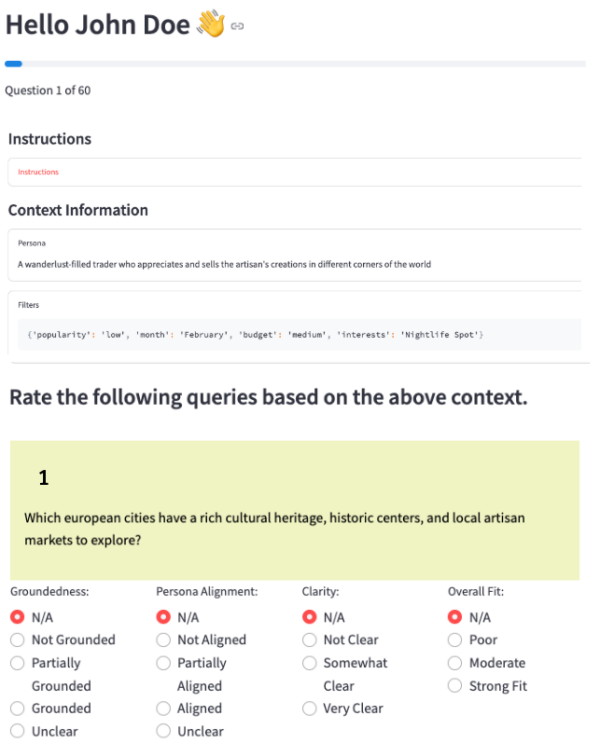

Fig. 3: Screenshot showing a part of the evaluation tool developed for the expert study. The full version can be found here.

$$MR = \sum_{i=1}^{N} \frac{len(matches)_i}{len(filters)_i} \tag{5}$$

The results of the MR scores can be found in Table 2. We can see that the vanilla setting (q_{v}) consistently appears to outperform the other two settings across models, with the personalized zero-shot (q_{p0}) setting showing the second-best performance. However, the personalized single-shot (q_{p1}) setting shows extremely low recall values across models. We theorize that this is due to the models overfitting on the persona and the ICL example provided in the prompts, which also led to lower diversity scores for this setting.

Expert evaluations align with the observed trends of J-LLM in Table 2, with evaluators categorizing queries into four levels: 0 for *Unclear*, 1 for *Not grounded*, 2 for *Partially grounded* and 3 for *Grounded*, based on their alignment with travel filters. The percentage of queries rated as *Grounded* (level 3) is reported in Table 2. To further analyze the reliability of expert judgments, we measured inter-evaluator agreement, where we obtained a Mean Absolute Error (MAE) of 0.083, indicating a high level of consistency between the two evaluators.

Table 2: Validation results assessing the alignment of generated queries with travel filters ($\mathcal{F}$) and personas (p), as evaluated by both J-LLM and domain experts. The values in bold (*italics*) denote the maximum (minimum) values for each model across all the settings — q_{v}, q_{p0}, and q_{p1}. ↑(↓) means higher (lower) is better.

Model	Query Setting	Alignment (%) ↑			
		$\mathcal{F}$		p	
		J-LLM	Expert	J-LLM	Expert
Llama	q_{v}	**0.759**	**0.869**	N/A	N/A
	q_{p0}	0.644	0.806	*34.348*	*33.333*
	q_{p1}	*0.043*	*0.346*	**61.538**	**59.167**
Gemini	q_{v}	**0.717**	**0.867**	N/A	N/A
	q_{p0}	0.595	0.840	*32.841*	*39.167*
	q_{p1}	*0.042*	*0.344*	**60.382**	**50.833**

5.3 Persona Alignment

We assess the alignment of generated queries with personas using expert evaluation and J-LLM (*gpt-4o*), following a two-fold approach similar to that described in Section 5.2. Both LLM-based and expert evaluators are provided with the persona description, the query, and four rating options: *Not Aligned*, *Partially Aligned*, *Aligned*, and *Unclear*. They are asked to determine how likely the persona would formulate the given query, focusing on tone and phrasing rather than inferred filters.

To mitigate potential biases associated with specific personas, the evaluation prompts explicitly instruct the LLMs and expert evaluators to assess only the tone and linguistic style of the query rather than making assumptions about the persona's domain expertise or preferences. This ensures that alignment ratings reflect the stylistic coherence between the query and persona rather than pre-existing biases.

Table 2 presents the results of the persona alignment for each model and query setting, comparing J-LLM and expert evaluations. We measure the persona alignment as the percentage of queries rated as *Aligned*. Although q_{p1} shows weaker groundedness with the travel filters, it achieves a higher persona alignment score than q_{p0}, according to J-LLM and expert evaluations. This suggests a potential trade-off between persona alignment and query groundedness.

To quantify inter-evaluator agreement, we calculate the MAE score between expert ratings, obtaining a moderate agreement level with an MAE of 0.3. The relatively high MAE suggests that persona interpretation is inherently subjective and can vary across individuals, highlighting the challenges of achieving consistent persona alignment in query generation.

5.4 Contexual Alignment

Besides evaluating alignment with travel filters and personas, we assess query groundedness within the context to minimize LLM hallucinations and ensure factual accuracy from our *KB*. To quantify this, we perform a semantic search for the query q using our *KB* (now represented as a vector database) to retrieve the query

Table 3: Evaluation of context groundedness, sustainability features, and diversity in the generated queries. The values in bold (*italics*) denote the maximum (minimum) values for each model across all the settings – $q_{\mathbf{v}}$, $q_{\mathbf{p0}}$, and $q_{\mathbf{p1}}$. ↑(↓) means higher (lower) is better.

Model	Query Setting	Contextual Alignment ↑	Sustainability		Diversity ↑				
			Similarity ↑	MAE ↓	Overall	Query Level			
						$f_{\mathbf{easy}}$	$f_{\mathbf{med}}$	$f_{\mathbf{hard}}$	$f_{\mathbf{sust}}$
N/A	**Baseline**	0.073	N/A	N/A	-	-	-	-	-
Llama	q_{v}	**0.482**	**0.659**	*0.077*	*0.101*	*0.138*	0.136	*0.169*	*0.179*
	q_{p0}	0.455	0.624	0.078	**0.221**	**0.352**	**0.274**	0.299	0.319
	q_{p1}	*0.430*	*0.605*	**0.091**	0.177	0.307	*0.241*	**0.315**	**0.327**
Gemini	q_{v}	**0.463**	**0.648**	*0.086*	*0.091*	*0.140*	*0.119*	*0.113*	*0.167*
	q_{p0}	0.451	0.589	0.087	**0.267**	**0.415**	**0.321**	0.327	0.360
	q_{p1}	*0.411*	*0.526*	**0.091**	0.182	0.346	0.251	**0.342**	**0.378**

context and cities, respectively. Our assumption follows that if the queries are factually grounded in the *KB*, then a subsequent semantic retrieval using these queries from the same *KB* should result in the same information. This retrieved context is compared with the original context used during query generation by computing the cosine similarity between the respective contextual embeddings generated using the *all-mpnet-base-v2*[8] model.

From Table 3, we observe that the models and query settings show similar performance across metrics, with the vanilla setting appearing to barely outperformthe other two settings. However, there is a sizeable increase in performance when compared to the baseline, which is computed by comparing a paragraph of Lorem Ipsum text with the original context, indicating that the addition of the *KB* in the query generation leads to queries that are factually grounded.

5.5 Sustainability Alignment

The ***sustainable*** query complexity level f_{sust} contains two non-sustainable travel filters and one sustainable filter, along with the destination popularity information. While we want to introduce sustainable travel into our dataset in a seamless manner, it is also important to verify that the sustainable queries do not show substantial differences semantically when compared to the other queries. In order to measure this, we compute the cosine similarity scores using the *all-mpnet-base-v2* model between the sustainable queries and non-sustainable queries, as shown in Table 3.

The results show moderately high scores between the sustainable and non-sustainable queries across model and query settings, indicating semantic similarity between the two sets. However, this does not indicate whether the sustainable filter is well represented within each sustainable query or if it is compromised in favor of the non-sustainable travel filters.

To evaluate how well the generated queries consider the sustainability constraints, we compute the cosine similarities between each travel filter and the query belonging to the sustainable complexity (q_i). We further compute the MAE between the similarities of the sustainability and the average similarity of the non-sustainability filters (represented by *j*) for each query setting as the following:

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left| sim_{\mathrm{sust},q_i} - \frac{1}{M}\sum_{j=1}^{M} sim_{j,q_i} \right| \tag{6}$$

The results in Table 3 show low MAE scores across query settings and models, suggesting that the LLMs do not overlook sustainability in favor of the other travel filters, ensuring a balanced representation of sustainability during query generation.

5.6 Diversity of the Generated Queries

Given that the filter and context control the sentence semantics, we measure the diversity of the generated queries at the token level. We use Self-BLEU [43], a metric derived from the BLEU score [30], which calculates sentence similarity between a hypothesis and a set of reference texts using n-gram matching. For each generated query q_{i}, we measure Self-BLEU against the other queries as the reference set using 4-grams. We define the diversity score as

$$Diversity = 1 - \frac{(\Sigma_{i=1}^{|G|} Self\text{-}BLEU(q_i))}{|G|}, \quad q \in G, G \subseteq \mathbb{Q} \tag{7}$$

A higher diversity score signals better diversity.

Table 3 shows that personalization improves query diversity, with q_{p0} and q_{p1} yield higher scores than q_{v}. We observe that queries generated with the same persona tend to sound similar across different key functions. This resulted in a lower overall diversity score since each persona is assigned to 12 key functions. When analyzing the scores at the query level, it becomes apparent — especially for sustainable queries, adding more contextual dimensions leads to higher diversity. However, potential overfitting in q_{p1} suggests that queries become less diverse when closely mirroring examples in the prompt. This underscores the need for careful prompt design to balance personalization and diversity in generated queries.

[8] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

5.7 Clarity and Overall Quality Assessment

While the J-LLM evaluations can assess most aspects of query quality, they often fail to capture subtle nuances in language and semantics. For instance, consider the following persona: *"A franchise owner of a major retail brand planning to establish a branch in the town,"* with a preference for low popularity and low budget. The generated query for this persona, q_{p1}, is: *"Low-cost European city with untapped market potential for retail."*

An LLM-based evaluation assigns this query a perfect persona alignment and groundedness score. However, expert evaluators highlight an important nuance: the phrase *"untapped market potential for retail"* is not directly aligned with the stated preferences. Consequently, they assign a moderate overall fit score rather than a perfect one. This discrepancy underscores the importance of expert evaluations, particularly in assessing clarity and overall fit. Hence, we measure two additional aspects—*clarity* and *overall fit*—through expert review. *Clarity* refers to how grammatically correct, understandable, and interpretable a query is, while *overall fit* captures a holistic assessment considering relevance, clarity, and persona alignment.

For *clarity*, the average score, normalized over the scale of evaluation choices, is high across models and settings, with q_{p1} showing the highest scores at 0.947 and 0.953 for *llama-3.2-90b* and *gemini-1.5-pro* , respectively. Furthermore, the mean absolute error (MAE) for clarity ratings across two independent evaluators is extremely low (0.011), indicating a high degree of agreement.

Similarly, evaluators provide consistent ratings across all queries and models for overall fit, averaging 0.7 with minimal deviation. However, we see the opposite trends in *overall fit* as compared to *clarity*, with q_{v} having the highest scores in both models, likely due to the overfitting of the q_{p1} setting to the persona, indicating a correlation between the *overall fit* and other validation criteria, such as groundedness in filters and context. The intra-evaluator MAE for overall fit is 0.16, further confirming the reliability of their judgments. These results reinforce the necessity of expert evaluation to capture subtleties overlooked by automated methods.

5.8 Preliminary Analysis using RecGenLLM

As a preliminary analysis, we evaluated the queries generated by Gemini as $\mathcal{L}_{\text{query\_gen}}$ using the same Gemini model (*gemini-1.5-pro*), now serving as our recommender LLM (RecGenLLM, $\mathcal{L}_{\text{rec\_gen}}$), grounded with web search [11]. While query generation and recommendation were performed using *gemini-1.5-pro* in our setup, other LLMs could also function as $\mathcal{L}_{\text{rec\_gen}}$, as their generative capabilities are inherently task- and prompt-dependent [27]. We instructed RecGenLLM to recommend cities from the predefined 200 cities in our *KB*, in zero- and few-shot settings. Grounding it with web search allowed it to provide citations from web searches if available.

On average, RecGenLLM recommends 10 cities per query, of which eight cities (80%) are found in our knowledge base (KB). To assess the impact of popularity bias, we computed the percentage of recommended cities classified as "high-popularity" based on the popularity scores in our *KB* and averaged this across all queries. On average, 79% of the cities recommended by $\mathcal{L}_{\text{rec\_gen}}$ are highly popular, regardless of the query constraints. However, on a closer look, we observe that the $\mathcal{L}_{\text{rec\_gen}}$ tends to recommend popular cities 77% of the time, even when the queries specifically request for cities with low popularity. For example, in July, the c_{rec_gen} for the query *"Trip to a less-visited European city with historical sites and museums. High budget."* include cities like *Budapest (Hungary) and Prague (Czechia)*, which are not only popular destinations but also have peak season in July. In contrast, the $c_{\mathscr{D}}$ for this query consists of less popular cities like *Syktyvkar (Russia), Malatya and Kars (Türkiye), Ioannina (Greece)*.

This behavior is likely due to the limited capability of the $\mathcal{L}_{\text{rec\_gen}}$ to understand nuanced queries and predilection to recommend more common travel destinations. Addressing this issue in future work will require developing retrieval strategies for RecGenLLM that actively mitigate popularity bias, ensuring more balanced and diverse recommendations [13, 23].

6 Conclusion

In conclusion, this paper introduces SynthTRIPs, a novel framework leveraging LLMs and a curated knowledge base for generating synthetic travel queries, addressing the need for more personalized and sustainable travel recommendation datasets. Through persona-based preferences, sustainability filters, and grounding techniques, SynthTRIPs effectively creates realistic and diverse queries, as validated by human experts and automatic LLM-based assessments. The framework's reproducible pipeline, open-source resources, and flexible design offer valuable tools for researchers in the tourism recommendation community.

However, SynthTRIPs has limitations too, including the potential for subjective persona interpretation and the observed trade-off between personalization and groundedness, where highly personalized queries may exhibit lower factual accuracy or alignment with their constraints. Furthermore, the personas in SynthTRIPs are not always domain-aligned. Since PersonaHub consists of synthetically generated data, it often lacks diversity and realism. While clustering them by topic marginally improves diversity, the issue of unrealism in the personas persists and remains evident in our dataset.

Furthermore, our preliminary analysis reveals a potential popularity bias in a zero/few-shot setting through a RecGenLLM to generate city recommendations. In this case, the model often favors high-popularity destinations, even when users explicitly specify a preference for less popular options. Since SynthTRIPs ensures an equal representation of popular and less popular travel destinations, future work could use this to focus on mitigating popularity bias in zero/few-shot recommendations and developing more sophisticated retrieval strategies to ensure balanced and diverse results. Future work could extend SynthTRIPs by incorporating more user filters (e.g., traveling with families, solo travels, accessibility, etc.), expanding beyond European cities to enhance global applicability, and developing strategies to maintain the knowledge base as real-world travel data evolves.

Acknowledgments

We thank the Google AI/ML Developer Programs team for supporting us with Google Cloud Credits.

References

[1] Zahra Abbasiantaeb, Yifei Yuan, Evangelos Kanoulas, and Mohammad Aliannejadi. 2024. Let the llms talk: Simulating human-to-human conversational qa via zero-shot llm-to-llm interactions. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*. 8–17.

[2] Trevor Ashby, Adithya Kulkarni, Jingyuan Qi, Minqian Liu, Eunah Cho, Vaibhav Kumar, and Lifu Huang. 2024. Towards Effective Long Conversation Generation with Dynamic Topic Tracking and Recommendation. In *Proceedings of the 17th International Natural Language Generation Conference*. 540–556.

[3] Gokulakrishnan Balakrishnan and Wolfgang Wörndl. 2021. Multistakeholder Recommender Systems in Tourism. *Proc. Workshop on Recommenders in Tourism (RecTour 2021)* (2021).

[4] Ashmi Banerjee, Tunar Mahmudov, Emil Adler, Fitri Nur Aisyah, and Wolfgang Wörndl. 2025. Modeling sustainable city trips: integrating CO 2 e emissions, popularity, and seasonality into tourism recommender systems. *Information Technology & Tourism* (2025), 1–38.

[5] Ashmi Banerjee, Adithi Satish, and Wolfgang Wörndl. 2024. Enhancing Tourism Recommender Systems for Sustainable City Trips Using Retrieval-Augmented Generation. *arXiv preprint arXiv:2409.18003* (2024).

[6] Kinjal Chaudhari and Ankit Thakkar. 2020. A comprehensive survey on travel recommender systems. *Archives of computational methods in engineering* 27 (2020), 1545–1571.

[7] Yashar Deldjoo. 2023. Fairness of chatgpt and the role of explainable-guided prompts. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 13–22.

[8] Google Firebase. 2025. *Firebase Realtime Database Documentation*. https://firebase.google.com/docs/database

[9] Leon Fröhling, Gianluca Demartini, and Dennis Assenmacher. 2024. Personas with Attitudes: Controlling LLMs for Diverse Data Annotation. *arXiv preprint arXiv:2410.11745* (2024).

[10] Tao Ge, Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, and Dong Yu. 2024. Scaling synthetic data creation with 1,000,000,000 personas. *arXiv preprint arXiv:2406.20094* (2024).

[11] Google. [n. d.]. *Ground Gemini to your data*. https://cloud.google.com/vertex-ai/generative-ai/docs/multimodal/ground-with-your-data#private-ground-gemini

[12] Maarten Grootendorst. 2022. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. *arXiv preprint arXiv:2203.05794* (2022).

[13] Yufei Guo, Muzhe Guo, Juntao Su, Zhou Yang, Mengqiu Zhu, Hongfei Li, Mengyang Qiu, and Shuo Shuo Liu. 2024. Bias in large language models: Origin, evaluation, and mitigation. *arXiv preprint arXiv:2411.10915* (2024).

[14] Steven Haussmann, Oshani Seneviratne, Yu Chen, Yarden Ne'eman, James Codella, Ching-Hua Chen, Deborah L McGuinness, and Mohammed J Zaki. 2019. FoodKG: a semantics-driven knowledge graph for food recommendation. In *The Semantic Web–ISWC 2019: 18th International Semantic Web Conference, Auckland, New Zealand, October 26–30, 2019, Proceedings, Part II 18*. Springer, 146–162.

[15] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2024. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems* (2024).

[16] Folasade Olubusola Isinkaye, Yetunde O Folajimi, and Bolande Adefowoke Ojokoh. 2015. Recommendation systems: Principles, methods and evaluation. *Egyptian informatics journal* 16, 3 (2015), 261–273.

[17] Pegah Jandaghi, XiangHai Sheng, Xinyi Bai, Jay Pujara, and Hakim Sidahmed. 2023. Faithful persona-based conversational dataset generation with large language models. *arXiv preprint arXiv:2312.10007* (2023).

[18] Dietmar Jannach and Li Chen. 2022. Conversational recommendation: A grand AI challenge. *AI Magazine* 43, 2 (2022), 151–163.

[19] Mohammad Khalil, Farhad Vadiee, Ronas Shakya, and Qinyi Liu. 2025. Creating Artificial Students that Never Existed: Leveraging Large Language Models and CTGANs for Synthetic Data Generation. *arXiv preprint arXiv:2501.01793* (2025).

[20] Jian Lan, Runfeng Shi, Ye Cao, and Jiancheng Lv. 2022. Knowledge Graph-based Conversational Recommender System in Travel. In *2022 International Joint Conference on Neural Networks (IJCNN)*. 1–8. doi:10.1109/IJCNN55064.2022.9892176 ISSN: 2161-4407.

[21] Megan Leszczynski, Shu Zhang, Ravi Ganti, Krisztian Balog, Filip Radlinski, Fernando Pereira, and Arun Tejasvi Chaganty. 2023. Talk the Walk: Synthetic Data Generation for Conversational Music Recommendation. *arXiv preprint arXiv:2301.11489* (2023).

[22] Mike Lewis. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461* (2019).

[23] Jan Malte Lichtenberg, Alexander Buchholz, and Pola Schwöbel. 2024. Large language models as recommender systems: A study of popularity bias. *arXiv preprint arXiv:2406.01285* (2024).

[24] Kwan Hui Lim, Jeffrey Chan, Christopher Leckie, and Shanika Karunasekera. 2018. Personalized trip recommendation for tourists based on user interests, points of interest visit durations and visit recency. *Knowledge and Information Systems* 54 (2018), 375–406.

[25] Eric Hsueh-Chan Lu, Ching-Yu Chen, and Vincent S Tseng. 2012. Personalized trip recommendation with multiple constraints by mining user check-in behaviors. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*. 209–218.

[26] Yu Lu, Junwei Bao, Zichen Ma, Xiaoguang Han, Youzheng Wu, Shuguang Cui, and Xiaodong He. 2023. AUGUST: an automatic generation understudy for synthesizing conversational recommendation datasets. *arXiv preprint arXiv:2306.09631* (2023).

[27] Daniel Machlab and Rick Battle. 2024. LLM In-Context Recall is Prompt Dependent. *arXiv preprint arXiv:2404.08865* (2024).

[28] Javier Marın, Aritro Biswas, Ferda Ofli, Nicholas Hynes, Amaia Salvador, Yusuf Aytar, Ingmar Weber, and Antonio Torralba. 2021. Recipe1m+: A dataset for learning cross-modal embeddings for cooking recipes and food images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43, 1 (2021), 187–203.

[29] Sonia Meyer, Shreya Singh, Bertha Tam, Christopher Ton, and Angel Ren. 2024. A Comparison of LLM Finetuning Methods & Evaluation Metrics with Travel Chatbot Use Case. *arXiv preprint arXiv:2408.03562* (2024).

[30] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 311–318.

[31] Pablo Sanchez and Linus W Dietz. 2022. Travelers vs. Locals: The Effect of Cluster Analysis in Point-of-Interest Recommendation. In *Proceedings of the 30th ACM Conference on User Modeling, Adaptation and Personalization*. 132–142.

[32] Ruixiang Tang, Xiaotian Han, Xiaoqian Jiang, and Xia Hu. 2023. Does synthetic data generation of llms help clinical text mining? *arXiv preprint arXiv:2303.04360* (2023).

[33] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663* (2021).

[34] Shuai Wang, Ekaterina Khramtsova, Shengyao Zhuang, and Guido Zuccon. 2024. Feb4rag: Evaluating federated search in the context of retrieval augmented generation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 763–773.

[35] Xi Wang, Hossein A Rahmani, Jiqun Liu, and Emine Yilmaz. 2023. Improving Conversational Recommendation Systems via Bias Analysis and Language-Model-Enhanced Data Augmentation. *arXiv preprint arXiv:2310.16738* (2023).

[36] Qianfeng Wen, Yifan Liu, Joshua Zhang, George Saad, Anton Korikov, Yury Sambale, and Scott Sanner. 2024. Elaborative Subtopic Query Reformulation for Broad and Indirect Queries in Travel Destination Recommendation. http://arxiv.org/abs/2410.01598 arXiv:2410.01598 [cs].

[37] Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (New Orleans, Louisiana). Association for Computational Linguistics, 1112–1122. http://aclweb.org/anthology/N18-1101

[38] Jian Xie, Kai Zhang, Jiangjie Chen, Tinghui Zhu, Renze Lou, Yuandong Tian, Yanghua Xiao, and Yu Su. 2024. Travelplanner: A benchmark for real-world planning with language agents. *arXiv preprint arXiv:2402.01622* (2024).

[39] Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. 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)*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, Hong Kong, China, 3914–3923. doi:10.18653/v1/D19-1404

[40] Haochen Zhang, Anton Korikov, Parsa Farinneya, Mohammad Mahdi Abdollah Pour, Manasa Bharadwaj, Ali Pesaranghader, Xi Yu Huang, Yi Xin Lok, Zhaoqi Wang, Nathan Jones, and Scott Sanner. 2023. Recipe-MPR: A Test Collection for Evaluating Multi-aspect Preference-based Natural Language Retrieval. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23)*. Association for Computing Machinery, New York, NY, USA, 2744–2753. doi:10.1145/3539618.3591880

[41] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems* 36 (2023), 46595–46623.

[42] Lixi Zhu, Xiaowen Huang, and Jitao Sang. 2024. A LLM-based Controllable, Scalable, Human-Involved User Simulator Framework for Conversational Recommender Systems. *arXiv preprint arXiv:2405.08035* (2024).

[43] Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A Benchmarking Platform for Text Generation Models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval* (Ann Arbor, MI, USA) *(SIGIR '18)*. Association for Computing Machinery, New York, NY, USA, 1097–1100. doi:10.1145/3209978.3210080