Large Language Models for User Interest Journeys

KONSTANTINA CHRISTAKOPOULOU, ALBERTO LALAMA, CJ ADAMS, IRIS QU, YIFAT AMIR, SAMER CHUCRI, PIERCE VOLLUCCI, FABIO SOLDO, DINA BSEISO, SARAH SCODEL, LUCAS DIXON, ED H. CHI, and MINMIN CHEN, Google Inc., USA

Large language models (LLMs) have shown impressive capabilities in natural language understanding and generation. Their potential for deeper user understanding and improved personalized user experience on recommendation platforms is, however, largely untapped. This paper aims to address this gap. Recommender systems today capture users' interests through encoding their historical activities on the platforms. The generated user representations are hard to examine or interpret. On the other hand, if we were to ask people about interests they pursue in their life, they might talk about their hobbies, like *I just started learning the ukulele*, or their relaxation routines, e.g., *I like to watch Saturday Night Live*, or *I want to plant a vertical garden*. We argue, and demonstrate through extensive experiments, that LLMs as foundation models can reason through user activities, and describe their interests in nuanced and interesting ways, similar to how a human would.

We define *interest journeys* as the persistent and overarching user interests, in other words, the non-transient ones. These are the interests that we believe will benefit most from the nuanced and personalized descriptions. We introduce a framework in which we first perform *personalized* extraction of interest journeys, and then summarize the extracted journeys via LLMs, using techniques like few-shot prompting, prompt-tuning and fine-tuning. Together, our results in prompting LLMs to name extracted user journeys in a large-scale industrial platform demonstrate great potential of these models in providing deeper, more interpretable, and controllable user understanding. We believe LLM powered user understanding can be a stepping stone to entirely new user experiences on recommendation platforms that are *journey-aware*, assistive, and enabling frictionless conversation down the line.

ACM Reference Format:
Konstantina Christakopoulou, Alberto Lalama, Cj Adams, Iris Qu, Yifat Amir, Samer Chucri, Pierce Vollucci, Fabio Soldo, Dina Bseiso, Sarah Scodel, Lucas Dixon, Ed H. Chi, and Minmin Chen. 2023. Large Language Models for User Interest Journeys. 1, 1 (May 2023), 19 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

With the abundance of the internet content, the role of recommendation systems has significantly expanded. In the past, users mainly relied on recommendation systems to make one-off decisions around where to eat, what to buy, or which movie to watch [11, 25, 41]. Nowadays, users expect the recommendation platforms to also support their persistent and overarching interests, including their real-life goals that last days, months or even years [12, 29, 30]. For example, a user who is into stand-up comedy would want recommendations tailored to their tastes, and might expect the system to help them explore other forms of comedy performance. A user who is learning to play an instrument or is improving home decoration, would want recommendations and tips appropriate to their skill level. Or, a user with the goal of

Authors' address: Konstantina Christakopoulou; Alberto Lalama; Cj Adams; Iris Qu; Yifat Amir; Samer Chucri; Pierce Vollucci; Fabio Soldo; Dina Bseiso; Sarah Scodel; Lucas Dixon; Ed H. Chi; Minmin Chen, Google Inc., Mountain View, USA.

Manuscript submitted to ACM

becoming an entrepreneur would want the recommender system to find inspirational content assisting them every step along the way.

If one were to ask a friend for recommendations around any of their journeys, the friend would probably ask them to first describe their interests or needs in detail. Once they get a reply, like *I want to know the history of stand-up comedy and the most famous stand-up comedian at this time*, or *I started playing ukulele a month ago, and I want to improve my strumming skills*, the friend would then be in a much better position to give good recommendations. Conversely, recommender systems make recommendations by predicting the next item a user might want to interact with, given their historical activities [1, 11, 41].

We argue that this type of collaborative filtering based approach [11, 20] does not meet user needs for higher-level semantic preferences. In order for recommender systems to truly assist users through their real-life journeys, they need to be able to understand and reason about interests, needs, and goals users want to pursue [19, 33, 37, 49]. However, the task presents some challenges. First, users often do not explicitly spell out their interests, needs, and real-life goals to the recommenders. As a result, the recommenders need to infer them from the historical activities the users engaged on the platform. Second, users can have multiple journeys intertwined in their activity history at any time. Third and most importantly, journeys are personalized and nuanced. Two users who are both into stand-up comedy can be interested in completely different aspects of it (e.g., *history of stand-up comedy documentaries* vs *Saturday Night Live skits*). This is where Large Language Models (LLMs) come in play. LLMs have demonstrated impressive capabilities for natural language understanding and generation, achieving state-of-the-art performance in a variety of tasks, from coding to essay writing to question answering [10, 13, 16, 50]. What if we power recommender systems with LLMs that can reason through user activities on the platform to uncover the underlying personalized and nuanced user interests, needs and goals, that is: user journeys?

To this end, we propose to build a personalized user journey profile which 1) uses personalized clustering to uncover coherent *user journeys*, i.e., persisting user interests, needs, and goals, from a long sequence of user interaction logs, and 2) leverages the capabilities of LLMs aligned to the user interest journey domain through prompt-tuning [27] and fine-tuning [60] on different data sources, to describe the extracted journeys with interpretable and nuanced names. Together, we make the following contributions:

- A first demonstration of the capabilities of LLMs to uncover and describe in natural language the interests, needs, and goals users pursue, similar to how people would describe them, e.g., *hydroponic gardening*, *playing the ukulele as a beginner*, *cooking italian recipes* (Figure 1). We posit that this will unlock unique user experiences and enable recommenders to assist users throughout their journeys.
- A thorough research study shedding light into the different factors impacting the quality of the generated journey names, e.g., the prompting techniques, the underlying domain data used for prompting, the LLM architecture and size, and the journey extraction technique.
- An at-scale user research study uncovering the taxonomy of real users' journeys and how they pursue them on recommendation platforms.

2 UNDERSTANDING REAL USER JOURNEYS

To build recommender systems that are able to assist users on their journeys, we started by launching a series of user research studies consisting of online surveys and user interviews to understand the types of journeys users pursue. Informed consent from study participants was collected in accordance with company user data and privacy policies,

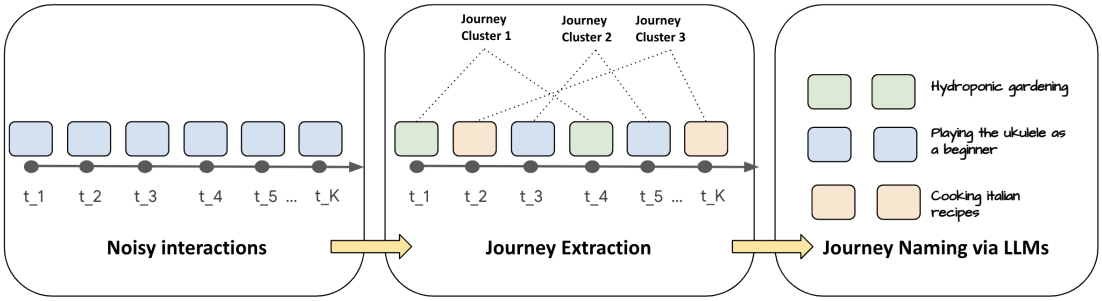

Fig. 1. Our approach uses personalized clustering to uncover coherent user journeys, and names them via prompting LLMs.

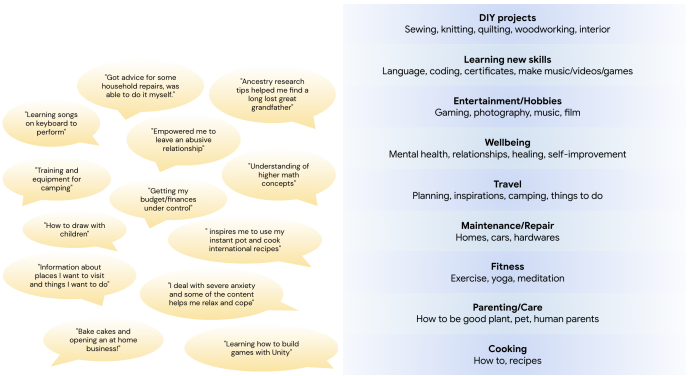

Fig. 2. (left) Sample real journeys described by respondents. (right) Taxonomy of valued journeys uncovered by clustering text user responses via UMAP [32].

which adhere to ethical research standards on human participants policies. Furthermore, all user data was de-identified by replacing names with randomized numbers and securely stored.

Setup. We sent online surveys to approximately 12, 000 users of an online platform, (a representative sample of the US internet user population aged 18-65+), with the goal of answering the questions: (Q1) If and how people use existing online platforms to pursue their journeys (Q2) What types of journeys people pursue online? Based on the analyzed survey data, we set up in-person interviews with a small cohort (N=9) of survey respondents who had multiple journeys to better understand: (Q3) What are the highlights and pain points of pursuing journeys on the internet? and (Q4) How do their journeys evolve? In the surveys, we framed journeys in terms of real-life interests or goals that the online platform has helped users pursue in a meaningful way.

Insight 1: People value content related to entertainment, learning, and community engagement. Figure 2 (left) shows example real journeys as described by our respondents. Figure 2(right) shows the resulting taxonomy of valued interest journeys i.e., journeys they pursue and find satisfactory, by clustering the provided responses via UMAP [32], and identifying roughly nine emerging themes. In a separate survey, we confirmed that entertainment (50 percent) and learning new skills (34 percent) are among the top valued journeys, and other dimensions include physical and mental well-being, caring for others, and community building etc.

Insight 2: People pursue multiple interest journeys concurrently over long periods of time. Our surveys confirm that users do pursue real-life journeys on recommendation platforms. About 66% of survey respondents used the platform recently to pursue a journey they valued. Out of the 66%, about 8 in 10 consumed content relevant to a journey for more than a month, with half saying some journey lasts for more than a year. People reported exploring multiple journeys simultaneously, with 7 in 10 pursuing one to three journeys and 3 in 10 exploring 4 or more in a single session. The large majority reported both exploring new valued interests and continuing existing ones.

Insight 3: People rely more on explicit actions to find content relevant to their interest journeys. Half of our survey respondents picked the search bar as the most-used feature to pursue their journeys, compared to 20% who relied on the recommendation feed, indicating inefficiency in existing recommenders in assisting user interest journeys, and the opportunity for providing more user control in recommendation experiences [21].

Insight 4: People's journeys are nuanced and evolve in a personalized way. Our in-person interviews uncovered a lot of nuance in people's journeys. For example, one participant placed their interest journey in the general category of "gardening". As they dived deeper, they described it in much more nuanced phrases, i.e., "designing hydroponic systems for small spaces." Another aspect uncovered was the right specificity people identify with, e.g., "greenhouse designs for cold climates" was deemed irrelevant for someone pursuing indoor gardening. Finally, participants mentioned how their interest can change based on offline and on-platform interactions. In other words, there are not two identical journeys between users, calling for personalized treatments.

Together these research insights guide our methods and experimental setup, and provide motivation towards a recommendation experience that can offer interpretability and control, enabling users to pursue their interest journeys.

3 METHODS: JOURNEY SERVICE

3.1 Overview

Journeys. We first define *journey* as the umbrella term for user interests, needs, and goals: *a user journey is a sequence of user-item interactions spanning different periods of time, coherent around a certain user interest, need or goal.* Users can have multiple journeys at any time point, and the journeys can be interleaving.

Journey-aware recommendation. We envision a journey-aware recommender system, capable of identifying personalized user interest journeys and making recommendations accordingly to assist these journeys.

Here we focus on the foundation work of extracting and naming user interest journeys, referred to as **journey service**, and leave mapping these extracted and named journeys to recommendations for future works. The proposed journey service consists of two components, as illustrated in Figure 1:

(1) **Journey Extraction:** Maps the sequence of noisy interactions a user had with items on the platform into coherent journey clusters. The journey clusters span different time periods, with singleton journeys (i.e., containing only a single user-item interaction) removed.
(2) **Journey Naming:** Maps the extracted journey clusters to human-readable, nuanced journey names based on content metadata, such as *hydroponic gardening, playing the ukulele as a beginner.* This enables users to comprehend how the system understands their journeys, and gives them control over their recommendations (e.g. they can choose more or fewer recommendations from a certain journey).

We study the necessity of these two components in section 4 and 5. Our user research uncovered properties we need to take into consideration when designing the journey service:

Algorithm 1 Infinite Concept Personalized Clustering (ICPC) on a User

Input: $\mathcal{H}_u = \{i_t, t = 1, \cdots, T\}$ containing list of items the user interacted with; $\epsilon \in [0, 1]$: salient terms similarity threshold; c: Minimum number of items per cluster. Default values: $\epsilon = 0.1, c = 1$.
Initialize: Journey Set $\mathcal{S}_J = \emptyset$
for $t = 1, \cdots, T$ **do**
  $\forall J \in \mathcal{S}_J$, compute ItemJourneySim$(i_t, J)$
  $J^* \leftarrow \arg\max_{J' \in \mathcal{J}}$ ItemJourneySim(i_t, J')
  if *ItemJourneySim*$(J^*, i) \geq \epsilon$ **then**
    $J^* = J^* \cup \{i_t\}$
  else
    Start a new journey $J_{\text{new}} = \{i_t\}$, $\mathcal{S}_J = \mathcal{S}_J \cup \{J_{\text{new}}\}$.
  end
  Update the journey representation based on the added item.
end

Prune journey clusters with less than c items

Fig. 3. Visualization of journeys across users, with each point representing an extracted and named journey. We generate embeddings for each journey name through Universal Sentence Encoder [7] and cluster them with UMAP [32]. Highlighted, a *video game strategy* journey and its nearest neighbors in the embedding space, per cosine similarity.

(I) Granularity of journey clustering. As revealed in the user surveys, journeys are personalized, thus defining the right granularity of journey clusters for all users is challenging. Measuring granularity quantitatively is hard, thus we propose to measure it via precision and recall, relying on proxy data (i.e., human-curated playlists) to align our journey clustering to how a person would define them (Section 4.1).
(II) Nuance & **interestingness of journey names.** We would like to name the extracted journeys in a nuanced and interesting way, just as how the user would have described their own journey. For example, *hydroponic gardening*, or *growing indoor plants* as in Section 2. We measure the quality of the generated journey names w.r.t. (proxy) ground-truth using Bleurt score [43], as well as imputed specificity, interestingness scores [50], and other dimensions (Section 5).
(III) Safety of journey names. As we envision showing users the generated names and allowing them to control and refine their recommendations down the line, it is important that the generated names are safe [50]. We rely on imputed safety score to measure safety of the generated names. We find the safety of the journey name largely depends on the safety of the included items within the journey cluster (Section 5).

3.2 Journey Extraction

The journey extraction component maps a user's overall history into coherent journey clusters. The most straightforward approach is to partition the entire item corpus into clusters, where each cluster represents one journey cluster. The caveat is that a single item can often cover multiple nuanced journeys. For example, an item on "cooking with herbs for improved mental clarity" can be a part of a "Mediterranean cooking journey", "improve your gut health" journey or "grow your herbs garden journey". A global clustering would then mix these more nuanced journeys together. As suggested by our user research, although there are commonalities among user journeys, there are no two identical journeys across users. We thus hypothesize that *a personalized view of journeys is more appropriate than assuming space of journeys is global*, and propose an infinite concept personalized clustering algorithm.

Infinite concepts. We start with annotating each item with a set of salient terms (unigrams and bi-grams), each with an associated weight called salience score [17, 28]. Sources for the unigrams and bigrams are the text (e.g., title, description) of the item, search query that lead to clicks on the item, anchor text, and so on. Together these terms provide the semantic meaning of the item. The salience score of each term is always between $[0, 1]$, which is obtained by training a supervised learning model to predict human ratings of relevance between the salient term and the item, with tf/idf demotions built in. One can then obtain the salient term representation of a group of items by aggregating salient terms of individual items, taking into account their salience score, resulting in a new set of salient terms representing the item group. This way, any group of items (or any concept for that matter) can be represented in the salient term space, which can be thought of as an infinite dimension embedding space where one salient term represents one dimension. Furthermore, the similarity between any two concepts can be computed in terms of cosine similarity of the salient terms embeddings, measuring the overlap of salient terms.

Infinite Concept Personalized clustering (ICPC). As depicted in Algorithm 1, our proposed algorithm effectively performs online clustering of a user's interaction history, based on the salient terms of the comprising items. Each user starts with an empty journey cluster set. We parse the user historical interactions one by one, each time assigning an item to the nearest, in terms of salient terms cosine similarity, journey cluster. Meanwhile, the representation of the assigned cluster is updated with the salient terms and salience scores of the newly assigned item. If no cluster is nearby (determined by thresholding the cosine similarity), a new cluster is created with the item assigned to it. The algorithm outputs the resulting journey clusters, which by definition will be thematically coherent, as shown in figure 3. The highlighted journey, extracted and named as "video game strategy" is closest to journeys such as "video game walkthroughs" and "game AI".

3.3 Journey Naming

Once we have extracted journey clusters for a user, we describe each journey in natural language so that it is explainable, scrutable, and controllable [3]. This is where Large Language Models (LLMs) come into play. Here, we describe the underlying models we leverage (Section 3.3.1) and how we adapt them to the domain of user interest journeys (Section 3.3.2) to capture nuanced natural language descriptions.

3.3.1 Models. In this paper, we build on LaMDa [50] and PaLM [10] family of LLMs.

LaMDA [50], aka. Language Models for Dialog Applications, is a family of Transformer-based, decoder-only, neural language models specialized for dialog, which have up to 137B parameters and are pre-trained on 1.56T words of public dialog data and web text. The models have shown state of the art performance across tasks, including the BigBENCH [47], and have been fine-tuned on human rewrites so to provide safe, interesting, sensible, and specific dialog responses.

PaLM [10], aka. Pathways Language Model, is a densely-activated decoder-only transformer language model trained using Pathways [15], a large-scale ML accelerator orchestration system that enables highly efficient training across TPU pods. At the time of release, PaLM 540B achieved breakthrough performance on a suite of multi-step reasoning tasks [10]. Its training corpus consists of 780 billion tokens representing a mixture of webpages, Wikipedia articles, source code, social media conversations, news articles and books [10].

Although we mainly build on LaMDA and PaLM, in our experiments we also considered **Flan-PaLM**, i.e., the instruction-tuned counterpart introduced by Wei et al.[53]. This pre-trained model was fine-tuned on datasets where each example is prefixed with some combination of instructions and/or few-shot exemplars, and was shown in [13] to

Prompt Template	Input	Output
I consumed content with titles: { }. I would describe one of my interests as:	Natural & Cultural Heritage Conservation in the Philippine Seas; Seafaring Vessels of Ancient Filipinos; The Philippines as a Maritime Nation: Opportunities and Challenges	Philippine Maritime History
titles: { } interest_journey:	Two-Angle Bisectors; Equal Angles; Cyclic Quad; Equal Segment; Circumcircles	Olympiad Geometry
keywords: { } interest_journey:	history, microwave, nerd, nostalgia, mp3	Evolution of Personal Technology

Table 1. Prompt formats: (a) natural language titles prompt, used across all prompting strategies unless stated otherwise; (b) structured titles prompt; (c) structured playlist infinite concepts prompt.

outperform PaLM on several benchmarks. We also considered **PaLMChilla**, i.e., a PaLM variant trained on an additional data mixture that follows the Chinchilla compute-optimal training procedure [14] approach.

3.3.2 Prompting LLMs for user interest journeys. Although general-purpose LLMs have reached state of the art performance on a wide variety of tasks on challenging benchmarks [52], for them to have good performance for the domain of user journey understanding, it is necessary to align them using domain-specific data. We explore data-efficient techniques of **few-shot prompting** [16] and **prompt-tuning** [27], and data-rich **end-to-end fine-tuning** [60].

Few-shot prompting Brown et al. [16] demonstrated that LLMs are strong few-shot learners. Fast in-context learning can be achieved through including a handful of demonstration examples as prompts in the input context, *without any gradient updates or fine-tuning*. In this study we focused on standard zero-shot, and few-shot. *Zero-shot prompting* queries these models with an instruction describing the task without any additional examples, i.e., *Summarize my interest journey in a concise and interesting way.* In *few-shot prompting*, the prompt further includes few-shot examples describing the task through text-based demonstrations. These demonstrations are encoded as input-output pairs.

Prompt tuning [27] has been proposed as a data efficient domain adaption technique for LLMs. The key idea is that instead of changing *all* model parameters, only a small subset of parameters corresponding to the prompt embedding will be updated given a small training set of input-output pairs (in the order of tens to hundreds). This can be viewed as a soft prompt, as opposed to the hard prompt encoded in few-shot prompting.

Fine-tuning [60] updates *all* model parameters to adapt to the new domain given a large amount of in-domain data. Typically, fine-tuning can achieve the best in-domain results with in-domain data in the order of thousands, but is also the most costly in compute and maintenance. An advantage of prompt-tuning compared with fine-tuning, is that the same underlying model (with same parameters) can be adapted to different domains by using prompt embeddings tuned on the domain data.

To be consistent across prompting strategies, we pre-processed input-output examples to follow the same format, as shown in Table 1.

3.3.3 Data for aligning LLMs to user journeys. To align LLMs to a new domain, it is important to have high-quality examples from that domain. Different prompting techniques, as discussed above, require different volumes of data. The ideal dataset for our task is derived from asking users to annotate and describe interest journeys they are pursuing on the platform in a nuanced and interesting way. This is, however, difficult and time-consuming to achieve at scale.

Fine-tuning data. To overcome the challenge, we rely on user-curated playlists on the platform as proxy datasets. Each playlist maps groups of content about the same topic to a user-provided name. We only include playlists that have high episodicity (items tend to be consumed in a sequential order) and are classified as learning-focused by a neural network trained on raters data. We refer to this dataset as **learning playlists**. The upside is that there are a lot of learning playlists readily available to fine-tune the model (we used twenty thousands examples); the downside is that the ground-truth playlist names tend to be short and noisy, without necessarily the nuanced nature we are looking for.

Prompt tuning data. Prompt tuning requires much fewer examples compared to fine-tuning. High-quality data, however, is crucial to ensure accurate results. Thus, instead of relying on the noisy learning playlists, we curated three small data sources for prompt tuning our models: (D1) **user interviews**, (D2) **user collections**, (D3) **expert-curated collections**, all in the order of less than a hundred examples. The (D1) source is the closest approximation to the ideal dataset, which is collected during our user interviews. We asked 50 users to describe journeys they pursued on the platform for more than a month, and pick from their interaction history the items that helped them toward these journeys. Source (D2) uses data from a separate platform anchored towards long-term aspirational journeys, where users save items in collections that they annotate with collection names. Lastly, (D3) comes from learning editorial collections, where expert editors have curated collections of items tailored to learning topics, and have named them in a nuanced and interesting way.

Few-shot examples For few-shot prompting, we only need a handful of examples (five to ten) from the domain of user interest journeys. For consistency, we sample examples from one of the above mentioned prompt tuning data.

4 JOURNEY EXTRACTION RESULTS

4.1 Experimental Setup

Metrics. As discussed in Section 3.1, measuring the right granularity of the extracted journey is extremely challenging. We look at qualitative results when extracting journeys on real user histories (evaluation data E1 explained below). Figure 4 shows one comparison between journeys extracted using different methods. To obtain quantitative results comparing different journey extraction algorithms, we measure *precision* (percentage of items correctly assigned the right journey cluster) and *recall* (percentage of items belonging to each journey cluster getting retrieved) on a "golden journey" set detailed in the following (evaluation data E2).

Evaluation Data We created two evaluation sets to evaluate journey extraction:

- E1 **Unlabeled Histories:** 300 user interaction histories, each with at least 10 valued interactions over 30 days, resulting in a dataset of 18,370 user-item interactions. We define a valued user-item interaction as one where 1) the user engaged with at least X minutes, and 2) a model that predicts user satisfaction surveys has found that the predicted user satisfaction score is in the top percentiles. The 300 user histories have a median length of 31.5, a mean length of 61.23 and a standard deviation of 69.41. The longest one contains 479 items; the shortest, 11.
- E2 **Learning playlists as golden journeys:** We sampled 5,000 learning playlists as explained in Section 3.3.3 as the golden journeys, comprised in total of 130,381 items. For each user we randomly sample and mix two playlists/journeys, and the goal is to cluster the items back into the two playlists.

Baselines. We compared with baseline approaches that uses different strategies to partition the entire item corpus into non-personalized global clusters; each cluster then becomes one journey cluster:

(1) **Clusters based on co-occurence behavior**. A topic cluster for each item is produced by: 1) taking the item co-occurrence matrix, where entry (i, j) counts the number of times item i and j were interacted with by the

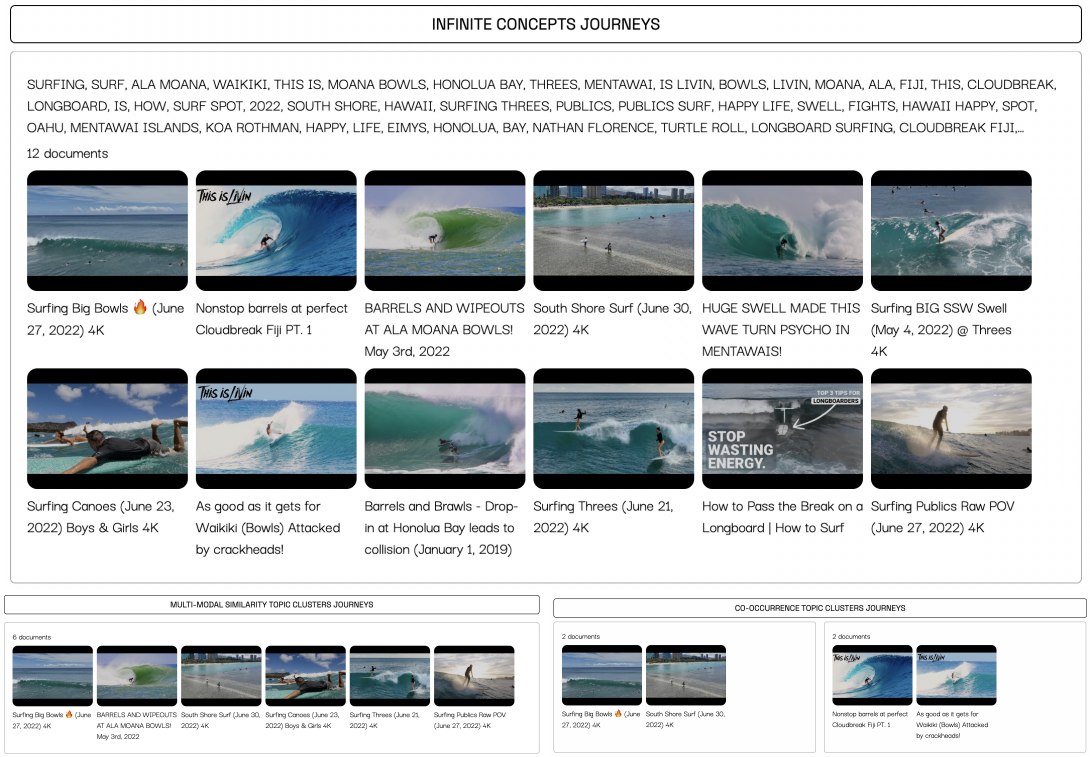

Fig. 4. Visual depiction of an infinite concepts personalized clusters journey extraction for *surfing*. Shown at the top, the journey's salient terms representation. Below, the set of documents with a thumbnail and title for each. In contrast, multi-modal similarity topic clusters journeys only retrieve 6 documents, and co-occurrence topic clusters split these into 2 clusters, each with 2 documents.

same user consecutively; 2) performing matrix factorization to generate one embedding for each item; 3) using k-means to cluster the learned embeddings into K clusters; 4) assigning each item to the nearest cluster centroid.

(2) **Clusters based on multimodal item similarity** These clusters group items together based on their audiovisual similarity, using online agglomerative/hierarchical clustering. The first-level of the hierarchy is capturing the macro-clusters, while the second level contains items all of fixed distance ϵ; each macro cluster would then serve as a journey cluster. Online clustering works by comparing each new item with the micro-clusters in terms of embedding similarity, and assigns it to the closest micro-cluster. If too often an item is found in close proximity to multiple micro-clusters, micro-clusters are merged.

Parameters. For the proposed Infinite Concept Personalized Clustering (we will refer to it here as *ICPC* for brevity), we set the similarity threshold parameter to 0.1 unless explicitly stated otherwise, as we found qualitatively that this results in the most coherent journeys of the right granularity. For the baselines, we used parameters as tuned in a large-scale production recommender system. For the co-occurrence based topic clusters, the total number of clusters K is 10,000; for multi-modal similarity-based clusters, the distance ϵ is set to 70. For all methods, we prune resulting clusters with a single item (we specify when we prune those with less than 5 items).

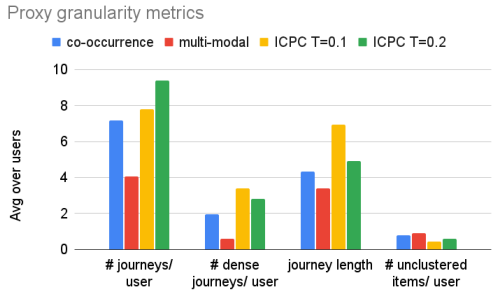

Fig. 5. **Comparison of journey extraction methods across proxy granularity metrics in (E1) setup.**

	Two golden journeys per user		
Method	**Recall**	**# journeys**	**# clusters / journey**
Co-occurrence	0.29	106,936	4.56
Multi-modal	0.16	71,388	3.63
ICPC (**ours**)	**0.82**	**13,488**	**2.42**

Fig. 6. **Comparison of journey extraction methods in E2.**

4.2 Key Results

Here we qualitatively and quantitatively compare the extracted journeys using our proposed ICPC algorithm and baselines across the two experimental setups.

ICPC achieves long coherent journeys, and is more robust to trivial coherence as obtained by singleton journey clusters. As shown in Figure 4 and Figure 5, the proposed method produces longer coherent journeys than baselines. The baselines on the other hand tend to produce trivially coherent journeys by assigning each item to a different cluster. Figure 5 shows that a fraction of 0.79 of items per user are singletons for the co-occurrence based baseline; the number increases to 0.90 for the multi-modal similarity-based ones. This fraction is much lower, but exists for our method too ($\sim$ 0.4).

ICPC corroborates our user research findings that each user pursues a few journeys. As shown in Figure 5, restricting the minimum number of items per journey to a value of 5, i.e., considering only dense journeys, the mean number of journeys per user to significantly decrease. The decrease is very prominent for multi-modal similarity topic clusters, suggesting that clusters are sparser. Conversely, *the cardinality of ICPC clusters and co-occurence topic clusters is consistent with user research* (Section 2), that 7 of out 10 users pursue 1-3 interests simultaneously and 3 of out 10 sustain 4 or more.

ICPC can easily control journey granularity via the similarity threshold. We find that at a lower threshold, more items are grouped into the same journey cluster, resulting in a coarser granularity. Increasing the threshold renders a larger number of journeys that are more specific, unearthing deeper semantic relationships between items. For example, in a public transportation-related example journey, we find that a 0.1 similarity threshold breaks up the *public transportation* journey (with 0.05 threshold) into three journeys: a general *public transportation* journey, a more specific *London public transportation* and a *public transportation video game simulations*. With pre-defined global clusters, it is difficult for the baselines to produce journey clusters at different granularity.

ICPC achieves the highest recall. Considering the (E2) setup of two golden journeys/playlists per user, we see from Figure 6 that our ICPC algorithm achieves much higher recall compared with baselines. On average, 2.42 journey

clusters are extracted per user using ICPC; close to the ground truth of 2 journeys per user. From the total # of journeys, we see that our method retrieves 1.3 more journey clusters than golden, versus co-occurrence gets around 10.7 more journey clusters.

Together, our findings suggest that our proposed ICPC approach can effectively cluster noisy interaction histories into journeys with the desired properties, significantly outperforming the non-personalized baselines.

5 JOURNEY NAMING RESULTS

Next, we evaluate the quality of journey naming in describing user interest journeys. We provide extensive analysis to understand the effect of different prompting techniques, underlying data, and models.

5.1 Experiment setup

We evaluate journey naming on three experimental setups.

- N1 **Expert-curated collections**: Data of 100 item collections, named with interesting engaging names from editorial teams. The set comes from the same distribution as (D3) used for prompt-tuning (Section 3.3.3), but it is from a different unseen split.
- N2 **Learning Playlists**: Larger-scale 10,000 playlists, of high episodicity focused in learning-related goals. The set comes from the same distribution as the learning playlists considered for fine-tuning (Section 3.3.3); again a different unseen split. As explained before, the names are shorter and noisier, but still convey meaning about the items saved in the playlists.
- N3 **Unlabeled Histories**: Larger scale 10,000 unlabeled extracted journeys, from histories of 2,000 users spanning a month (same distribution as E1 in Section 4.1). The journeys have been extracted using our ICPC algorithm. These journeys do not have expert or user-associated names.

For each journey, we concatenate titles of all the items belonging to the journey as input context to the LLMs as shown in Table 1, and generate the journey name using different LLMs with different prompting techniques. Unless otherwise specified, we mainly build on top of the LaMDA-137B foundation model [50]. We set the prompt embedding size to 5 tokens. We perform 10,000 prompt tuning steps, and 20,000 fine tuning steps. The learning rate and dropout rate is set to 3e-5 and 0.1 respectively across prompt and fine-tuning. During inference, we set beam size to 1, temperature to 0, max decoding steps to 256 and feature lengths 256 tokens for the input and 64 tokens for the targets.

To quantitatively compare the different naming user journeys approaches, we rely mainly on BLEURT [43] and SacreBLEU [35] scores.While SacreBLEU measures the overlap between the generated and the ground truth name, BLEURT, as an embedding-based metric, measures their semantic similarity. For the (N3) setup where we do not have the ground truth names, we propose to rely on a BLEURT score between the concatenated names of input items comprising the user journey and the generated name as the metric.[1]

5.2 Key Result: Prompting LLMs can reason well through user interest journeys.

As shown in Figure 7, our journey naming service can provide accurate, nuanced names describing the user interest journeys, like *the art of photography*, or *go kart racing*. These qualitative results illustrated are from a prompt-tuned

[1]BLEU has been developed and can be interpreted for the domain of machine translation; thus the interpretation of the values for our domain does not have a known reference point. That's why we use it for comparison rather than absolute scores, and we rely on qualitative evaluation as well.

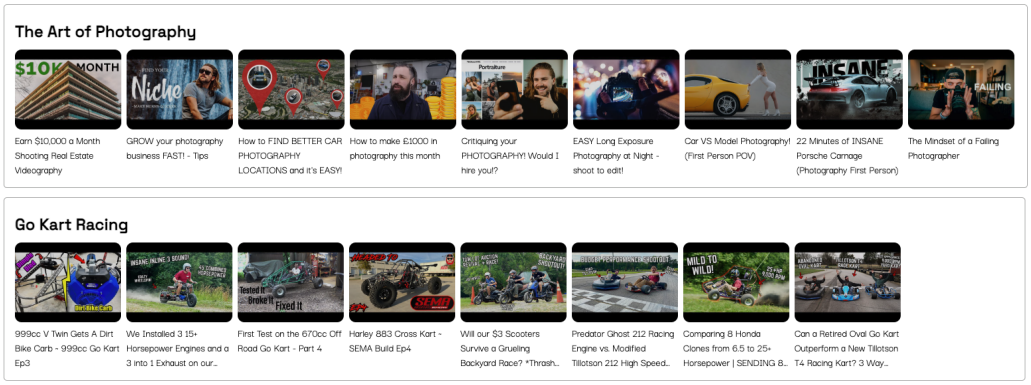

Fig. 7. Examples of real extracted and named user journeys through our journey service.

LaMDA-137B model using expert-curated collections as explained in Section 3.3.3 (prompt tuning data), used in inference mode on top of real user journeys on an industrial recommendation platform, extracted via our ICPC method.

5.3 RQ1: Which prompting technique performs best for journey naming?

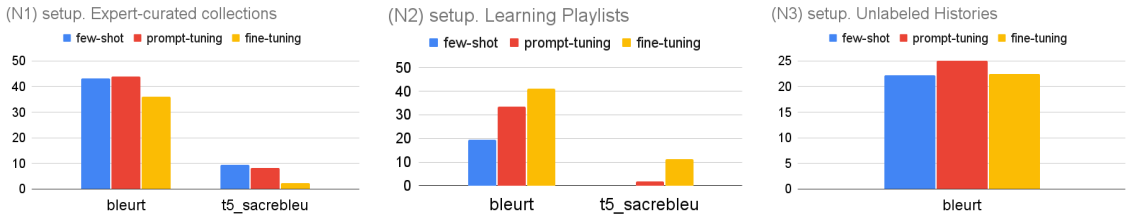

Fig. 8. **Few shot vs. prompt tuning vs. fine-tuning**. (left) Expert collections. (middle) Learning playlists. (right) Unlabeled Histories.

Prompt tuning on small high quality data outperforms few-shot prompt engineering. Figure 8(middle) and (right) compares the different prompting techniques for LaMDA 137B in the larger experimental setups (N2) and (N3), respectively. We can see that for both setups, prompt-tuning performs significantly better compared to few-shot, as measured both by SacreBLEU and BLEURT scores. In Figure 8(left) we also find similar results in the smaller but higher quality evaluation (N1) setup of the 100 expert-curated named collections: prompt-tuning outperforms few-shot prompt engineering both in terms of SacreBLEU and in BLEURT score. We also found that few-shot with 5 examples (bleurt 43.28) significantly outperforms zero-shot prompting (bleurt score 31.68).

Fine-tuning outperforms prompt-tuning in-domain, but prompt-tuning has better generalization capability. Fine-tuning is the default strategy for aligning LLMs to new domains. We tested whether fine-tuning LaMDA 137B on 20,000 user-generated playlists (with input the items and output the corresponding user-given names) could outperform learning the prompt embedding based on the (D3) smaller dataset of expertly curated named collections. The answer is: it depends. Based on results in Figure 8, we find that fine-tuning outperforms prompt tuning for the (N2) evaluation setup where examples come from the same distribution as fine-tuning examples, aka Learning Playlists data. However, testing the generalization power of both prompting techniques in (N3) setup of unlabeled histories, prompt tuning

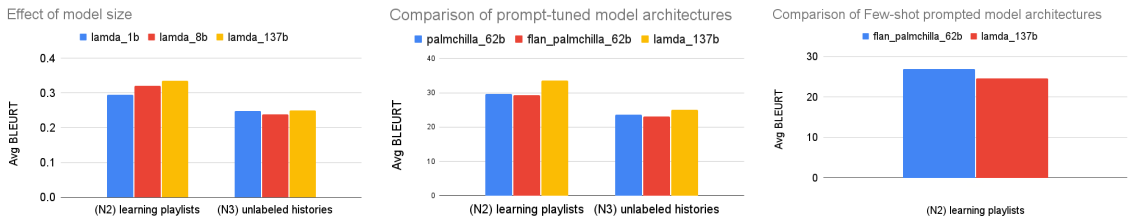

Fig. 9. (left) **Effect of model size.** (middle, right) **Effect of model architectures.**

outperforms fine-tuning. Also, from setup (N1) we find that prompt-tuning with small in-domain examples outperforms fine-tuning with out-of-domain large scale data.

5.4 RQ2: Which underlying model is better, and under which circumstances?

Effect of model size. Figure 9(left) compares how prompt tuning different sized LaMDA models affects the quality of generated journey names, considering the large-scale setups (N2) and (N3). One surprising observation is that while the performance on (N2) learning playlists suggests prompt tuning larger sized model leads to better naming quality, there is not a monotonic improvement on the (N3) unlabeled histories case. Overall, we explain these results by hypothesizing that when increasing the model size, on the one hand, the model has more capacity to infer good quality labels; on the other hand, the conditioning of model on the learned prompt embedding might become less prominent. Another hypothesis is the difference in the data among the two setups: (N3) compares output to input, while (N2) compares output to human labels. This trade-off among model size and prompt tuning needs further research investigation.

Effect of different model architectures. In Figure 9(middle), we compare prompt-tuned models: PaLMChilla 62B, and its corresponding instruction tuned variant Flan-PaLMChilla 62B, with LaMDA 137B. We see that for both (N2) and (N3) setups, LaMDA 137B achieves the best BLEURT score. We also see that the performance of instruction-tuned PaLMchilla does not differ much from its non-instruction-tuned counterpart. Interestingly, when comparing few-shot prompted models FLAN PaLMChilla 62B with LaMDA 137B (Figure 9(right)), the order is reversed: FLAN PaLMChilla outperforms LaMDA in (N2). We think that the instruction tuning of the former makes it better in following the few-shot prompting. Either way, both few-shot prompted models achieve lower BLEURT compared to the worst prompt-tuned model in the (N2) setup shown in Figure 9 (middle).

RQ3: How construction of the prompt affects the quality of generated interest names?

Which metadata are useful to be part of prompt? Figure 10 shows for (N1) setup, how different ways of representing the items: (1) using item titles/names, or (2) using keywords representing the playlist, or (3) with both item names and playlist keywords (as discussed in Table 1 for structured prompts) interact with different prompting techniques. Overall, we find that the titles of the items is the most informative feature, with some additive value provided when including playlist keywords. Therefore, unless specified otherwise, we infer journey name given only titles metadata.

Effect of prompt tuning data on success of learned prompt embeddings. As expected, the quality of the data used to learn the prompt embedding plays a key role in how successful prompt tuning LLMs can be. In Figure 11, we compare how learning the prompt on top of three different small quality datasets generalizes in the task of inferring journey names for the larger (N2) and (N3) experimental tasks. The prompt tuning data are the (D1) user interviews, (D2) user

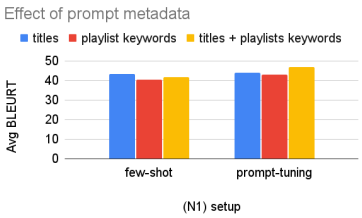

Fig. 10. Effect of prompt metadata.

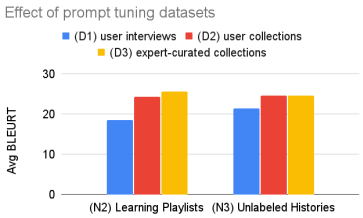

Fig. 11. Effect of prompt-tuning datasets.

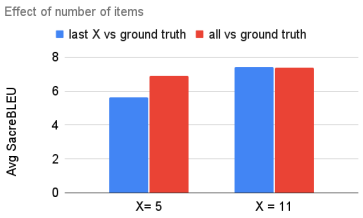

Fig. 12. Effect of number of journey items.

collections, and (D3) expert-curated collections, as described in Section 3.3.2. In both (N2) and (N3) setups, we see a big difference in performance based on the data we use to learn the prompt, with prompt embeddings learned from expertly curated names of collections providing the highest generalization capability. We argue that this is the case because the expertly named collections provides higher quality data to learn the prompt compared to user provided names, which tend to be more noisy, especially in the context of an interview where recipients might provide data of mixed quality. These results also point to the potential of mixed methods, i.e., mixing expert-provided labels, with a subset of higher quality user-provided labels, for enabling LLMs to accurately describe user interests.

Effect of journey length. We also conduct experiments to understand the number of items in the journey that should be included in order to generate a good name. Figure 12 shows that if we generate names only given the last five items in the journey, the SacreBLEU score between ground truth and generated name is significantly lower compared to if we were to use all the items in the journey[2]. However, when comparing journey names as inferred by the last eleven items only (now in histories of more than eleven items), there is little drop in performance compared to using all items in journeys. This can be a guiding point to reduce the context length in generating journey names.

5.5 RQ4: How safe and interesting generated user journey names are?

As pioneered in [50], besides the accuracy of responses generated by LLMs, other evaluation dimensions should be considered as well. These include how safe/responsible the responses are; how interesting they are; whether they are sensible; and whether they are specific enough, with many of these dimensions competing one against the another (e.g., the safer the response, the less interesting it might be). We utilize the already trained LaMDA scorer [50] (without the prompting done to align to the journey interest domain), as the imputation model for the aforementioned dimensions. . Figure 13 plots the imputed scores for the generated, by the prompt-tuned LaMDA-137B model, names, in the three setups (N1), (N2) and (N3). Our key takeaway is that the degree to which the generated journey names are safe/ interesting/ and so on, is largely affected by the underlying input given in inference to produce the names. This is evident by the fact that the more in-the-wild unlabeled histories have names which have lower safety score compared to names produced by the same model for the user-curated playlists, or the expert-collections.

5.6 RQ5: Do we need journey extraction, or can we rely on LLMs on both journey extraction & naming?

A natural question that might arise to the readers is that given LLMs evidently can be used to generate nuanced interesting names of the interest journeys users are pursuing, why not also task them with the job of both extracting and naming simultaneously? To answer this question, we use the evaluation setup (N3). The approaches we compare

[2]This comparison is done limiting to journeys that have more than five items, in the (N2) setup.

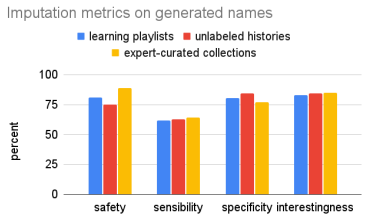

Fig. 13. Imputed aspects for names.

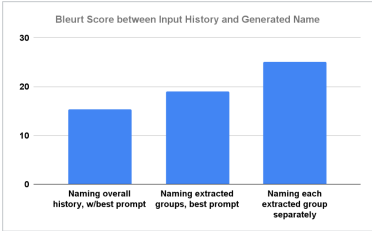

Fig. 14. Value of extracting journeys, separate than naming.

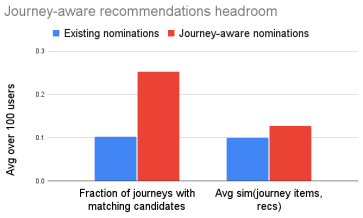

Fig. 15. Headroom of journey-aware recommendations.

are: (1) Providing the whole user history to the LLM and asking the model to name the interest journeys the user is pursuing. (2) Extracting journeys using our ICPC, and concatenating the journey groups (compared to the mixed history in (1)) at once in a single inference to the LLM and asking it to name the journey groups. (3) Naming each extracted group separately, with a different LLM call per journey. We compute the BLEURT score between the whole input history and the concatenated journey names as the final evaluation score and compare the three approaches. It's evident from Figure 14 that the LLM can generate significantly better journey names when it's only given the journey-specific history, thus further validating the need for our journey extraction component in the journey service.

5.7 RQ6: Can the extracted and named journeys enable an improved recommendation experience?

Lastly, we provide some early analysis on the degree to which a user's journeys can be better assisted when the recommender is aware of the underlying journeys. We compare the journeys extracted from real user histories over the period of a month (same setup as N3) with the recommendations given to the user the day after the month on the industrial platform. For each journey of a user, we compute the similarity of the average salient terms embedding of the journey and that of all the recommended items[3]. When the similarity is larger than a threshold (set to 0.2), we track the journey as being served. We can then compare this with a new recommendation process using our journey service and search: 1) we extract journeys from the one-month user history and name the extracted journeys using prompt-tuned LaMDA 137B on (D3); 2) we generate recommendations for each journey through searching the platform using the generated journey name. We then compute the similarities between these new journey-aware recommendations and journeys for comparison. We can see in Figure 15 that on average over a hundred sampled users, around 1 out of 10 journeys is served (0.102) under the current recommendation system. This number is significantly improved when the recommender is journey-aware (0.253). We also report the average similarity score between journeys and candidates (without the thresholding). We can see that overall, the provided recommedation candidates have higher similarity in the case of journey aware nominations (0.127 vs 0.099). Of course, to fully answer this question, ideally we need novel recommendation approaches mapping the journeys to recommendations. We also need A/B user facing experiments to measure the value added to the user. Both will be part of our future work. Nevertheless, these early results already point to headroom of improving recommendations via making them aware of user interest journeys.

6 RELATED WORK

Intent, Tasks, User Needs We compare journeys with related work on understanding and modeling user needs, intent, and tasks. Intents has long been considered in the search domain [38, 58]. Recent works on intent-aware recommendation

[3]For computation purposes, we reduce comparison of each journey with each recommended item, to comparing with the entire recommendation set

[5, 33] shows modeling intent improves user satisfaction. While intent captures the high-level context of the user's visit, it is much coarser than journeys. For example, the intent of "I want to do something inspiring" covers "becoming an entrepreneur" and "learning the ukulele" journeys. Another related work is modeling task behavior, which goes beyond the current single query, in search engines to improve search ranking (e.g., [31, 54]). However, search tasks are typically based on in-session interactions [31] or across only a few sessions [26], while users are found to pursue interest journeys for a month, or more. In addition, interest journeys also cover persisting interest such as entertainment, which is less task-oriented. User needs in Web search were categorized in [6, 39, 56] as transactional, navigational, or informational. A similar categorization on *why* users utilize recommendation systems has not been defined. Our findings suggest a new dimension of user needs: the dimension of using recommenders alongside real-life interests and aspirations, i.e., learning new skills, entertainment, and sense of belonging. This confirms our insight that the recommendation process should be transformed to satisfy this additional need. Broadly speaking, the goals of journey-aware recommendation align with those of personal assistants [19, 40, 46, 49] and context-aware recommenders [4]. Our journey extraction technology can be utilized as context within a proactive personal assistant to tailor recommendations.

User Interests Modeling Another close line of work is user modeling in recommender systems, and particularly modeling changing user interests via sequential models (e.g., RNNs) on user actions [8, 55]. Although powerful, sequence models such as RNNs tend to focus only on the last interactions, forgetting long-term user interests. To mitigate the recency bias, a recent thread has focused on incorporating users' long-term history data into recommendation models [34, 36, 59] by powering models with attention-based mechanisms [22, 48]. The produced user representation however is hard to examine or interpret. In this work, we take a different approach: extracting longer-term valued journeys, separately from the recommendation task. This allows for modularity of involved components and for interpretability [37], to enable the system explain recommendations via the lens of journeys [51]. Our journey extraction service allows to easily inspect and control the granularity of extracted journeys, while naming each journey via LLMs makes our approach interpretable. Furthermore, by introducing the journey entity, we can eventually enable users to interact with their extracted journeys; e.g., control whether they want more or less recommendations related to that journey. Also, our approach of extracting journeys via clustering, while paying attention to content understanding is related to user topic modeling [2, 42, 57] and clustering content by user needs [23, 24]. Also, works on uncovering coherent trails [9, 24, 44, 45] inspired us greatly.

Interpretable/Transparent User Models Our work is related to many past works advocating for the need of *explainable recommendation* [51], as well as *user control* [21]. Our contribution is the first-ever study showing the power of prompted LLMs so that they can reason through user interests; as well as shedding light to the angle of persistent real-life user journeys that users expect recommenders to help them with besides the shorter-term tasks. Perhaps the most close work to ours is [37] which laid out the vision for scrutable Natural Language (NL)-based recommendation. Our work however has several differences: Rather than a complete summary of user interests, we aim at short sentences describing different user interest journeys. Also, while [37] focuses on scrutable profiles, with the aim of recommendation, we focus on investigating the degree to which our personalized clustering approach coupled with prompting LLMs can provide deeper user understanding. Perhaps the most important difference is that, to our knowledge, our work is the first-ever experimental study of aligning LLMs to reason through user journeys. Furthermore, our work differs significantly from [3] where the authors showed how a restricted representation of users as weighted pairs of tag interaction can be verbalized as NL statements using templates. Our models rely on free-form natural language, rather than templates to output interest names.

Large Language Models (LLMs) Last but not least, our work is the first to demonstrate the application of LLMs in the user interest journey domain, for deeper user understanding in recommendation systems. While [37] provided some key directions along the usage of LLMs in the recommendation space as mentioned above, to our knowledge this is the first study to address issues like which data to use to fine-tune and prompt-tune LLMs to align them with this domain, how different prompting techniques perform, among others. We argue that there is large untapped potential in LLMs for the domain of recommender systems. We know that recent large pre-trained language models have been shown to have impressive generative fluency [10, 50] as well as effective few-shot learning capabilities [16] in the domain of natural language generation and understanding, and have passed several benchmarks of various tasks. However, so far, the key application to the recommendation system domain has been the P5 paper [18] which effectively maps recommendation as a language processing task, framing a variety of recommendation tasks into an end-to-end transformers framework. Our work is orthogonal; it leverages the power of LLMs as a way to reason through user interests, so to provide the interpretability a human would offer if they were to be asked for a recommendation.

7 CONCLUSIONS AND FUTURE WORK

Our work aims to bridge the gap in LLMs for understanding user interest journeys and improving user experience on recommendation platforms. We provide the first-ever demonstration of LLMs to reason through user interests and describe them similarly to how people would. We uncover important aspects to enable such capabilities, such as data, prompting techniques, and evaluation methodology. We show that prompt-tuning LLMs based on small but quality data of expertly curated and named collections has the most generalization capability. We also find first extracting the interest journeys through a personalized clustering procedure is critical. We validate our findings on a large-scale industrial recommender platform, and together suggest the promise of utilizing our proposed journey service as a way to gain deeper user understanding and building novel user experiences that offer more user control.

REFERENCES

[1] [n. d.].

[2] Seyed Ali Bahrainian, Fattane Zarrinkalam, Ida Mele, and Fabio Crestani. 2019. Predicting the topic of your next query for just-in-time ir. In *European Conference on Information Retrieval*. Springer, 261–275.

[3] Krisztian Balog, Filip Radlinski, and Shushan Arakelyan. 2019. Transparent, scrutable and explainable user models for personalized recommendation. In *Proceedings of the 42nd international acm sigir conference on research and development in information retrieval*. 265–274.

[4] Linas Baltrunas, Bernd Ludwig, and Francesco Ricci. 2011. Matrix factorization techniques for context aware recommendation. In *Proceedings of the fifth ACM conference on Recommender systems*. 301–304.

[5] Biswarup Bhattacharya, Iftikhar Burhanuddin, Abhilasha Sancheti, and Kushal Satya. 2017. Intent-aware contextual recommendation system. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, 1–8.

[6] Andrei Broder. 2002. A taxonomy of web search. In *ACM Sigir forum*, Vol. 36. ACM New York, NY, USA, 3–10.

[7] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175* (2018).

[8] Minmin Chen, Alex Beutel, Paul Covington, Sagar Jain, Francois Belletti, and Ed H Chi. 2019. Top-k off-policy correction for a REINFORCE recommender system. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 456–464.

[9] Ed H Chi, Peter Pirolli, Kim Chen, and James Pitkow. 2001. Using information scent to model user information needs and actions and the Web. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 490–497.

[10] Aakanksha et al. Chowdhery. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311* (2022).

[11] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*. 191–198.

[12] Michael D Ekstrand and Martijn C Willemsen. 2016. Behaviorism is not enough: better recommendations through listening to users. In *Proceedings of the 10th ACM conference on recommender systems*. 221–224.

[13] Hyung Won Chung et al. 2022. Scaling Instruction-Finetuned Language Models. arXiv:2210.11416 [cs.LG]

[14] Jordan Hoffmann et al. 2022. Training Compute-Optimal Large Language Models. arXiv:2203.15556 [cs.CL]

[15] Paul Barham et al. 2022. Pathways: Asynchronous Distributed Dataflow for ML. arXiv:2203.12533 [cs.DC]

[16] Tom B. Brown et al. 2020. Language Models are Few-Shot Learners. arXiv:2005.14165 [cs.CL]

[17] Michael Gamon, Patrick Pantel, Xinying Song, Tae Yano, and Johnson Tan Apacible. 2016. Identifying salient items in documents. US Patent 9,251,473.

[18] Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2022. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In *Proceedings of the 16th ACM Conference on Recommender Systems*. 299–315.

[19] Ramanathan Guha, Vineet Gupta, Vivek Raghunathan, and Ramakrishnan Srikant. 2015. User modeling for a personal assistant. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*. 275–284.

[20] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. arXiv:1708.05031 [cs.IR]

[21] Dietmar Jannach, Sidra Naveed, and Michael Jugovac. 2017. User control in recommender systems: Overview and interaction challenges. In *E-Commerce and Web Technologies: 17th International Conference, EC-Web 2016, Porto, Portugal, September 5-8, 2016*. Springer, 21–33.

[22] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*. IEEE, 197–206.

[23] Jing Kong, Alex Scott, and Georg M Goerg. 2016. Improving topic clustering on search queries with word co-occurrence and bipartite graph co-clustering. (2016).

[24] Weize Kong, Mike Bendersky, Marc Najork, Brandon Vargo, and Mike Colagrosso. 2020. Learning to Cluster Documents into Workspaces Using Large Scale Activity Logs. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '20)*. 2416–2424.

[25] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[26] Alexander Kotov, Paul N Bennett, Ryen W White, Susan T Dumais, and Jaime Teevan. 2011. Modeling and analysis of cross-session search tasks. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. 5–14.

[27] Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. arXiv:2104.08691 [cs.CL]

[28] Zhen Li, Huazhong Ning, Liangliang Cao, Tong Zhan, Yihong Gong, and Thomas S. Huang. 2011. Learning to Search Efficiently in High Dimensions. In *Neural Information Processing Systems*.

[29] Yu Liang. 2019. Recommender system for developing new preferences and goals. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 611–615.

[30] Yu Liang, Aditya Ponnada, Paul Lamere, and Nediyana Daskalova. 2023. Enabling Goal-Focused Exploration of Podcasts in Interactive Recommender Systems. In *Proceedings of the 28th International Conference on Intelligent User Interfaces*. 142–155.

[31] Claudio Lucchese, Salvatore Orlando, Raffaele Perego, Fabrizio Silvestri, and Gabriele Tolomei. 2011. Identifying task-based sessions in search engine query logs. In *Proceedings of the fourth ACM international conference on Web search and data mining*. 277–286.

[32] Leland McInnes, John Healy, and James Melville. 2020. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. arXiv:1802.03426 [stat.ML]

[33] Rishabh Mehrotra, Mounia Lalmas, Doug Kenney, Thomas Lim-Meng, and Golli Hashemian. 2019. Jointly leveraging intent and interaction signals to predict user satisfaction with slate recommendations. In *The World Wide Web Conference*. 1256–1267.

[34] Nikil Pancha, Andrew Zhai, Jure Leskovec, and Charles Rosenberg. 2022. PinnerFormer: Sequence Modeling for User Representation at Pinterest. *arXiv preprint arXiv:2205.04507* (2022).

[35] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. Association for Computational Linguistics, USA.

[36] Qi Pi, Weijie Bian, Guorui Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Practice on long sequential user behavior modeling for click-through rate prediction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2671–2679.

[37] Filip Radlinski, Krisztian Balog, Fernando Diaz, Lucas Dixon, and Ben Wedin. 2022. On Natural Language User Profiles for Transparent and Scrutable Recommendation. *arXiv preprint arXiv:2205.09403* (2022).

[38] Filip Radlinski, Martin Szummer, and Nick Craswell. 2010. Inferring query intent from reformulations and clicks. In *Proceedings of the 19th international conference on World wide web*. 1171–1172.

[39] Daniel E Rose and Danny Levinson. 2004. Understanding user goals in web search. In *Proceedings of the 13th international conference on World Wide Web*. 13–19.

[40] Tara Safavi, Adam Fourney, Robert Sim, Marcin Juraszek, Shane Williams, Ned Friend, Danai Koutra, and Paul N Bennett. 2020. Toward activity discovery in the personal web. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 492–500.

[41] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*. 285–295.

[42] Ugo Scaiella, Paolo Ferragina, Andrea Marino, and Massimiliano Ciaramita. 2012. Topical clustering of search results. In *Proceedings of the fifth ACM international conference on Web search and data mining*. New York, NY, USA, 223–232.

[43] Thibault Sellam, Dipanjan Das, and Ankur P. Parikh. 2020. BLEURT: Learning Robust Metrics for Text Generation. arXiv:2004.04696 [cs.CL]

[44] Dafna Shahaf and Carlos Guestrin. 2010. Connecting the dots between news articles. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. 623–632.

[45] Dafna Shahaf, Carlos Guestrin, and Eric Horvitz. 2012. Trains of thought: Generating information maps. In *Proceedings of the 21st international conference on World Wide Web*. 899–908.

[46] Yang Song and Qi Guo. 2016. Query-less: Predicting task repetition for nextgen proactive search and recommendation engines. In *Proceedings of the 25th International Conference on World Wide Web*. 543–553.

[47] Aarohi et al. Srivastava. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615* (2022).

[48] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 1441–1450.

[49] Yu Sun, Nicholas Jing Yuan, Yingzi Wang, Xing Xie, Kieran McDonald, and Rui Zhang. 2016. Contextual intent tracking for personal assistants. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 273–282.

[50] Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239* (2022).

[51] Nava Tintarev. 2007. Explanations of recommendations. In *Proceedings of the 2007 ACM conference on Recommender systems*. 203–206.

[52] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2020. SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems. arXiv:1905.00537 [cs.CL]

[53] Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned Language Models Are Zero-Shot Learners. arXiv:2109.01652 [cs.CL]

[54] Ryen W White, Wei Chu, Ahmed Hassan, Xiaodong He, Yang Song, and Hongning Wang. 2013. Enhancing personalized search by mining and modeling task behavior. In *Proceedings of the 22nd international conference on World Wide Web*. 1411–1420.

[55] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J Smola, and How Jing. 2017. Recurrent recommender networks. In *Proceedings of the tenth ACM international conference on web search and data mining*. 495–503.

[56] Xiaoxin Yin and Sarthak Shah. 2010. Building taxonomy of web search intents for name entity queries. In *Proceedings of the 19th international conference on World wide web*. 1001–1010.

[57] Manzil Zaheer, Amr Ahmed, and Alexander J Smola. 2017. Latent LSTM allocation: Joint clustering and non-linear dynamic modeling of sequence data. In *International Conference on Machine Learning*. PMLR, 3967–3976.

[58] Yuchen Zhang, Weizhu Chen, Dong Wang, and Qiang Yang. 2011. User-click modeling for understanding and predicting search-behavior. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1388–1396.

[59] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Deep interest evolution network for click-through rate prediction. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 5941–5948.

[60] Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2020. Fine-Tuning Language Models from Human Preferences. arXiv:1909.08593 [cs.CL]