Doing Personal LAPS: LLM-Augmented Dialogue Construction for Personalized Multi-Session Conversational Search

Hideaki Joko
Radboud University
Nijmegen, The Netherlands
hideaki.joko@ru.nl

Shubham Chatterjee
University of Edinburgh
Edinburgh, Scotland, UK
shubham.chatterjee@ed.ac.uk

Andrew Ramsay
University of Glasgow
Glasgow, Scotland, UK
andrew.ramsay@glasgow.ac.uk

Arjen P. de Vries
Radboud University
Nijmegen, The Netherlands
a.devries@cs.ru.nl

Jeff Dalton
University of Edinburgh
Edinburgh, Scotland, UK
jeff.dalton@ed.ac.uk

Faegheh Hasibi
Radboud University
Nijmegen, The Netherlands
f.hasibi@cs.ru.nl

ABSTRACT

The future of conversational agents will provide users with personalized information responses. However, a significant challenge in developing models is the lack of large-scale dialogue datasets that span multiple sessions and reflect real-world user preferences. Previous approaches rely on experts in a wizard-of-oz setup that is difficult to scale, particularly for personalized tasks. Our method, LAPS, addresses this by using large language models (LLMs) to *guide* a single human worker in generating personalized dialogues. This method has proven to speed up the creation process and improve quality. LAPS can collect *large-scale, human-written, multi-session, and multi-domain* conversations, including extracting user preferences. When compared to existing datasets, LAPS-produced conversations are as natural and diverse as expert-created ones, which stays in contrast with fully synthetic methods. The collected dataset is suited to train preference extraction and personalized response generation. Our results show that responses generated explicitly using extracted preferences better match user's actual preferences, highlighting the value of using extracted preferences over simple dialogue history. Overall, LAPS introduces a new method to leverage LLMs to create realistic personalized conversational data more efficiently and effectively than previous methods.

CCS CONCEPTS

• **Information systems** → **Users and interactive retrieval**.

KEYWORDS

Personalization, Conversational Search, Dialogue Collection

ACM Reference Format:
Hideaki Joko, Shubham Chatterjee, Andrew Ramsay, Arjen P. de Vries, Jeff Dalton, and Faegheh Hasibi. 2024. Doing Personal LAPS: LLM-Augmented Dialogue Construction for Personalized Multi-Session Conversational Search. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24), July 14–18, 2024, Washington, DC, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3626772.3657815

SIGIR '24, July 14–18, 2024, Washington, DC, USA

ACM ISBN 979-8-4007-0431-4/24/07.
https://doi.org/10.1145/3626772.3657815

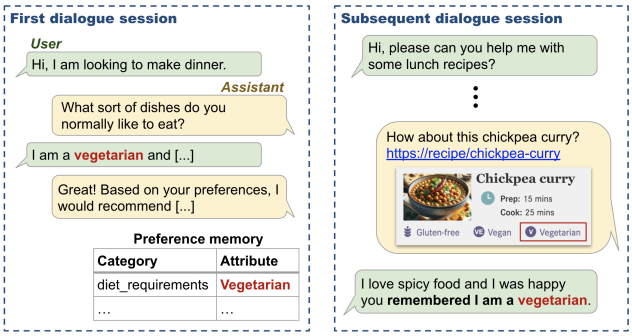

Figure 1: A snippet from a multi-session dialogue. User preferences are extracted and stored in memory to generate personalized recommendations in subsequent sessions.

1 INTRODUCTION

Personalization is paramount for conversational search and recommendation [6, 39]. Conversational agents need to meet users' expectations and provide them with individually tailored responses. In real-world scenarios, where users interact with dialogue agents across multiple sessions, conversational systems need to accurately understand, extract, and store user preferences and articulate personalized recommendations based on the stored user profile; see Figure 1. The Information Retrieval (IR) community has studied various aspects of personalization in conversational systems. For instance, the concept of Personal Knowledge Graph (PKG) [3] is introduced to enable personalization of (conversational) search systems. Similarly, TREC Interactive Knowledge Assistance Track (iKAT) utilizes Personal Text Knowledge Base (PTKB) [1] for persona-based conversational search. Recent years have also witnessed tremendous progress in Large Language Models (LLMs) [7, 43]. Yet LLM-based conversational agents do not effectively handle user preferences, delivering generalized recommendations that fail to capture the nuanced interests of individual users.

A major hindrance in developing a personal conversational system is the unavailability of *large-scale human-written* conversational datasets. These are needed for training new models and understanding user behavior, in expressing their preferences over multiple dialogue sessions [22, 45]. However, constructing such datasets has proven a daunting task [4, 45]. Preference elicitation in conversations is complex, and crowd workers engage poorly with the task. While human experts deliver quality results, recruiting them as

intermediary coaches or as human agents to generate conversations does not scale. The recently-emerged paradigm of collecting synthetic conversational data through LLMs [10, 20, 21, 25, 26, 30, 58, 60] raises concerns about dialogue diversity [13, 14, 44, 47, 59]. Crucially, LLM-generated conversations do not represent actual user preferences and interactions, undermining the credibility for developing *future* personal conversational systems.

The critical question that arises here is ***RQ1:*** *Can we collect large-scale multi-session human-written conversational datasets that contain user preferences?* We address this question by proposing LAPS, an **L**LM-**A**ugmented **P**ersonalized **S**elf-Dialogue method to collect large-scale personal conversations. LAPS employs an LLM to dynamically generate personal *guidance* for crowd workers, playing both user and assistant roles. The guidance is generated based on the previously elicited user preferences and the current state of the dialogue, determined by a dialogue act classifier. After each dialogue session, LAPS extracts preferences from the dialogue and stores them in a *preference memory*. This memory is a key-value store of user preferences, analogous to the PKG [3] and PTKB [1] concepts. Using LAPS, we can collect 1,406 multi-domain multi-session dialogues, paired with 11,215 preferences.

Our next research question concerns the quality of LAPS-generated datasets: ***RQ2:*** *How do the LLM-augmented self-dialogues compare to human- and synthetically-generated conversations?* We compare our conversations with a wide range of widely used conversational datasets and show that LAPS-generated conversations score higher in diversity (based on Dist-n, Ent-n, and SELF-BLEU metrics) and quality (based on UniEval [66]). We further compare LAPS- and LLM-generated dialogues using GPT-3.5 and GPT-4 and show that LLM-generated dialogues are less diverse than those involving humans, even with temperature tuning.

Although LAPS extracts preferences in a semi-structured format and stores them in a preference memory, one could wonder whether such memory is needed, given LLMs' abilities in handling long context from the previous sessions. This leads us to the third research question: ***RQ3:*** *How can preference memory enhance the effective utilization of user preferences in recommendations?* To address this question, we train a preference extraction model on our dataset and use it to build a preference memory from previous sessions. These preferences are then incorporated into the LLM's prompt for generating personalized recommendations. We compare this approach to the baseline prompting method, where dialogue histories of all sessions are appended to the prompt. Our experiments show that by incorporating preference memory, the model can more accurately utilize the users' disclosed preferences for recommendations than the baseline method. The notable advantage of preference memory is that it contributes to more explainable recommendations. Finally, we found that when using the baseline method, the LLM struggles with recalling user preferences; likely due to lengthy prompts.

Contributions of this work are as follows:

- We introduce LAPS method for collecting scalable multi-session personalized dialogues with actual user preferences.
- We analyze and compare various dialogue collection methods, demonstrating that LAPS collects lexically diverse and high-quality dialogues, uncovering the diversity issue of generating fully-synthetic dialogues with LLMs.

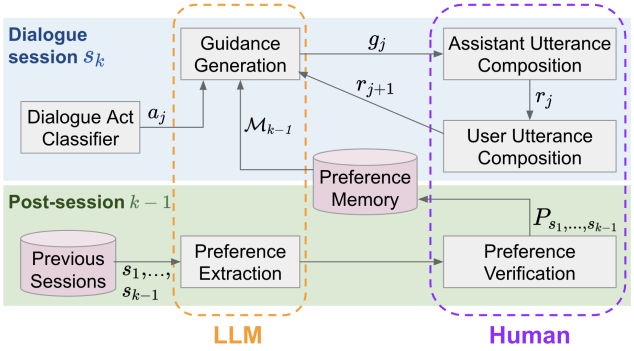

Figure 2: Overview of our dialogue collection method (LAPS).

- We study the benefits of storing and using user preferences in a semi-structured format (preference memory) and show that it helps an LLM in recalling previously disclosed preferences when generating personalized recommendations.
- Enabled by LAPS, we release a unique conversational dataset that is multi-session, human-written, large-scale, and contains users' personal preferences.[1]

2 RELATED WORK

2.1 Dialogue Collection Methods

Human-Human interactions are arguably the optimal strategy for collecting natural dialogues. MultiWOZ [8] and PersonaChat [61] are notable examples, offering task-oriented and chit-chat datasets, respectively. These datasets, however, focus less on real user preferences, relying instead on predefined tasks or personas. Addressing this limitation, Radlinski et al. [45] introduced CCPE-M, a dataset emphasizing actual user preferences. Similarly, Bernard and Balog [4] introduced MG-ShopDial, an e-commerce conversational dataset with genuine preferences. Despite their quality, the small size of these datasets (502 dialogues for CCPE-M and 64 for MG-ShopDial) limits their utility for training large models, and they overlook the multi-session aspect of real-world dialogues.

A quality-quantity trade-off in dialogue collection is highlighted in [4]. Initially attempting crowdsourcing, Bernard and Balog [4] shifted to a volunteer-based collection due to low engagement, resulting in higher quality but fewer dialogues. This underscores the challenge in gathering large, high-quality datasets representing true user preferences.

Self-Dialogue, where a single worker simulates both roles, is an effective approach for large-scale data collection. Introduced by Krause et al. [29], this technique has been shown to produce high-quality data, with Fainberg et al. [16] noting its increased coherence and reduced errors compared to human-human dialogue. Byrne et al. [9] further validated this, emphasizing the superior linguistic diversity and fewer mistakes in self-dialogue data. However, self-dialogue has limitations, especially in managing complex conversations, such as those involving preference elicitation, while

[1]The code and dataset is available at https://github.com/informagi/laps

acting in two distinct roles. Additionally, self-dialogue cannot authentically capture unknown user preferences as the same person plays both user and assistant roles, leading to fewer clarification scenarios than in human-human interactions [16]. Our approach introduces LLM-augmented personalized self-dialogues, leveraging LLMs to ease the cognitive load on workers and addressing the limitation of capturing unknown preferences by involving an LLM, which doesn't have pre-existing knowledge of user preferences.

(Semi)-Synthetic Dialogue Generation offers an alternative to relying on crowd workers [10, 21, 25, 26, 30, 31, 51, 58]. Lee et al. [30] developed PERSONACHATGEN using two LLMs for dialogues between personas, requiring multiple model calls for one dialogue. Chen et al. [10] optimized this by using a single model call with a carefully crafted prompt. Leszczynski et al. [31] took a different approach, generating synthetic dialogues by combining user-generated content and metadata. However, concerns exist about the diversity in LLM-generated texts. Reif et al. [47] noted lexical and syntactic repetition, developing LinguisticLens for syntactic diversity analysis. Chung et al. [13] emphasized the need for human intervention to enhance diversity, and Yu et al. [59] pointed out the uniformity in LLM outputs from simple prompts, suggesting diverse prompts for more varied data. The diversity issue is also evident in fields like social science and business [14, 44].

To address this, semi-synthetic data collection methods like having crowd workers edit LLM-generated texts have been effective [2, 46, 49]. Shah et al. [49] introduced M2M, a framework using templates for initial dialogue generation and crowd workers for rewriting. Similarly, Rastogi et al. [46] used a schema-guided method. Our research improves upon these by using LLMs for providing workers with guidance for response composition, promoting diversity and reducing the influence of generated drafts.

2.2 Personalized Conversational Systems

Problem-driven conversational systems, especially those for search and recommendation, benefit from personalization. Shifting from traditional approaches such as collaborative filtering, recent personalization focuses on more interactive approaches, such as preference elicitation [11, 27, 28, 32, 45, 53, 65]. These elicited preferences can be stored as a knowledge graph [3] through entity linking [23, 24, 55] or in text format [1], and later used for personalization. Storing user preferences in a (semi-)structured format enables conversational agents to better satisfy users' information needs and can be also useful to mitigate bias[19]. Following this line, our method elicits user preferences and stores them in a semi-structured preference memory.

Memory and feedback also offer the promise of making systems better aligned with user needs. This approach, tracing back to ALFRED [48], involves storing past failures to improve future interactions. Recently, Madaan et al. [41] advanced this concept with SELF-REFINE, enabling LLMs to refine their outputs iteratively using their own feedback. For tasks requiring deeper personalization, human feedback is invaluable. Madaan et al. [40] enhanced LLM response quality by adding a memory module that remembers information from the user's past sessions. Aligned with [40], our approach includes a memory module to store user preferences from past interactions, enabling enhanced personalization.

3 DIALOGUE COLLECTION METHOD

We propose LAPS, an LLM-Augmented Personalized Self-Dialogue construction method, capable of collecting large-scale, human-written, multi-session, and multi-domain conversations, paired with extracted user preferences. The method consists of four key elements (cf. Fig. 2): (i) dialogue act classification, (ii) guidance generation, (iii) utterance composition, and (iv) preference extraction.

The dialogue act classifier determines the next action that the assistant should take; e.g., recommend. Based on the dialogue act, the LLM generates guidance considering the dialogue history and the previously extracted preferences. The human agent then composes the assistant response based on the LLM-generated guidance, and then switches to the role of a user, providing a response to the previous utterance. The process continues until a relevant recommendation is made and the dialogue session is completed. Upon completion of a session, the preferences are extracted from the dialogue using an LLM and checked by the human agent. These preferences, once confirmed by the same human agent, are stored in the *preference memory* and used in subsequent sessions for generating personalized guidance. The human agent is then encouraged to initiate a new dialogue session for another scenario in the given domain. This process continues until the human agent exits the job or reaches the end of all pre-defined session scenarios.

3.1 Task Formulation

The objective of this task is twofold: firstly, to create a large-scale collection of multi-session dialogues written by humans, focusing on user preferences; and secondly, to extract and compile the specific user preferences mentioned within these dialogue sessions. Formally, the dialogue collection method F is defined as a mapping from a set of task descriptions $\mathbb{T}$ and human agents $\mathbb{H}$ to a set of dialogue sessions and their extracted preferences $\mathbb{S}$:

$$F : \mathbb{T}, \mathbb{H} \rightarrow \mathbb{S}$$
$$F(t, h) = [(s_1, P_{s_1}), \ldots, (s_n, P_{s_n})],$$

where t is a task description for a topic, h is a human agent with identical preferences for a given topic, and n represents the total number of sessions. The dialogue session s_i and its corresponding preference set P_{s_i} are defined as:

$$s_i = [u_1^i, u_2^i, \ldots, u_m^i],$$
$$P_{s_i} = \{(c, p_j) \mid c \in C, j \in \mathbb{N}\}, \quad (1)$$

where the dialogue session s_i composed of a sequence of utterances u, and the preference set P_{s_i} is a set of category-preference pairs (c, p_j), where the category c belongs to the set of categories C; e.g., {`allergy`, `cuisine`, `diet`}. We note that the preference set P_{s_i} is generated only after the completion of session s_i, by extracting user preferences from the dialogue post-session and validating them with the same worker. The extracted preferences of a human agent are stored in a memory component $\mathcal{M}$, defined as:

$$\mathcal{M}_k = \bigcup_{\substack{1<i\leq k \\ k\leq n}} P_{s_i},$$

where session s_k is the last completed session by the human agent.

Here we draw an analogy between the preference memory $\mathcal{M}$ and Personal Knowledge Graphs (PKGs) [3], where the personal

Table 1: Overview of the selected baseline datasets and ours. EXP and CW denote an expert and a crowd worker, respectively.

Dataset	Collection Method	#Dial	Tasks	Domains	Scalability	Actual User Preferences	Preference Elicitation	Preference Extraction	Multi-Session
SGD [46]	Semi-synthetic	16,142	Booking, rec., etc.	20 domains, inc. restaurants	✓	×	×	×	×
M2M [49]	Semi-synthetic	3,008	Booking	Restaurants, movies	✓	×	×	×	×
PersonaChatGen [30]	Fully-synthetic	1,649	Personal chit-chat	Open domain	✓	×	×	×	×
Taskmaster-1 [9]	Self-dialogue	7,708†	Booking, ordering	6 domains, inc. restaurants	✓	×	×	×	×
MultiWOZ [8]	Human-human (CW)	8,438	Booking, rec., etc.	7 domains, inc. restaurants	✓	×	×	×	×
CCPE-M [45]	Human-human (EXP)	502	Rec.	Movies	×	✓	✓	△	×
MG-ShopDial [4]	Human-human (EXP)	64	Rec., QA, etc.	E-commerce	×	✓	✓	×	×
LAPS	LAPS	1,406	Rec.	Recipes, movies	✓	✓	✓	✓	✓

† Includes only self-dialogues.
△ Annotations are provided but no entity-relation pair extraction with like/dislike distinction.

information of users is stored according to an ontology. The <subject, predicate, object> triplets in PKGs correspond to human (h), category (c), and preference (p) in preference memory, respectively. We note that unlike a PKG that is built based on a pre-defined ontology, preference memory uses a more relaxed version of categories that are extracted on-the-fly from user utterances. Preference memory can be also viewed as a semi-structured form of PTKB, where free-form sentences about a user's persona are transformed in a key-value format.

3.2 Guidance Generation

Generating guidance for human agents is central to collecting large-scale, high-quality, human-written utterances in LAPS. Large-scale construction of conversational data requires recruiting crowd workers. However, due to the complexity of the task and high cognitive load of generating conversational utterances, crowd workers show poor engagement [4]. The challenge is even more intense for the preference elicitation task, where we need to coach the human agent simulating the system to ask engaging questions to reveal user preferences [45]. A remedy could be utilizing LLMs to generate system utterances. This, however, results in less diverse conversations and is not in line with our aim of generating training data for *future* personalized conversational search and recommendation systems. Even by instructing crowd workers to re-write LLM-generated utterances, we observed (in our pilot studies) that crowd workers become less creative in generating their own utterances and tend to replicate pre-generated utterances.

To alleviate the aforementioned problems, we propose to coach crowd workers throughout the conversation using automatically generated personalized guidance, and let the workers compose their own utterances via dialogue self-play. Using this approach, we reduce the cognitive load of a highly complex task to a minimum, allowing workers to focus on a simple sub-task at a time and generate high-quality and engaging conversations. Formally, to compose the assistance utterances u_j, the human agent receives personalized guidance g_j generated by an LLM. The guidance generation function $\mathcal{G}$ is defined as:

$$\mathcal{G}\left(u_1, \ldots, u_{j-1}, a_j, \mathcal{M}_k\right) = g_j,$$

where $\mathcal{M}_k$ denotes the preference memory extracted from sessions $(s_1, \ldots, s_k)$, utterances $u_1, \ldots, u_{j-1}$ represents conversations history up until turn j, and a_j is the action that needs to be taken for turn j, obtained from the dialogue act classifier (cf. § 3.3).

Instantiation. As an instantiation of this function, we prompt GPT-3.5 `turbo` to generate personalized guidance. The guidance prompt template takes the dialogue history $u_1, \ldots, u_{j-1}$, preference memory $\mathcal{M}_k$, and instructions for the current dialogue act a_j as inputs. The prompts include detailed step-by-step instructions for chain-of-thought prompting [56]. For instance, the prompt for the *preference elicitation* act in a given `DOMAIN` is as follows:

> *You are an advisor, who supports* **`$DOMAIN`** *recommendation assistants to compose responses to users. In this step, the assistant* ***collects information about user's preferences*** *([a_j]).*
>
> Preference memory: [$\mathcal{M}_k$]
> Dialogue history: [$u_1, \ldots, u_{j-1}$]
>
> *Step 1: Identify the last user turn.*
> *Step 2: Explain the intent of the last user utterance.*
> *Step 3:* ***Which preference(s) should the assistant ask next*** *given the dialogue history?*
> *Step 4:* ***Compose very short guidance for the human assistant on how to write a response to the user.***
> *Step 5: Output in JSON format following [...]*
> *Ensure that you distinctly label and delineate Steps 1, 2, 3, 4, and 5. Let's think step by step:*

The guidance prompt for *recommendation* act is similar to the *preference elicitation* act, except that the assistant is instructed to recommend an item based on the user's personal preferences with URLs. The guidance also includes "*When making recommendations, if necessary, effectively utilize the user's preferences, such as [...]*" to encourage the assistant to use the user's preferences disclosed in the previous sessions if necessary.

3.3 Dialogue Act Classification

Dialogue act is an action that can change the (mental) state of conversation and guide the system to generate the next utterance [18]. Dialogue act is used as an input to the guidance generation function and is obtained by the dialogue act classifier $\mathcal{A}$:

$$\mathcal{A}(u_1, \ldots, u_{j-1}, a_{j-1}) = a_j$$

which determines action a_j based on the dialogue history $u_1, \ldots, u_{j-1}$ and the previous dialogue act a_{j-1}.

Instantiation. We instantiate the act classifier function by prompting GPT-3.5 `Turbo`. A series of dialogue acts are defined, outlining the specific actions to be taken sequentially. The primary dialogue

acts in our setup are: (1) *greeting*, (2) *preference elicitation*, (3) *recommendation*, (4) *follow-up questions*, and (5) *goodbye*. A dialogue act is selected upon completion of the previous act, which is determined by the LLM using detailed chain-of-thought instruction prompts; e.g., the *recommendation* act is selected when the LLM produces the response "true" for the instruction prompt "*[...] Has the user shared any of the preferences listed above? Has the assistant collected why the user has the preference? If both are true, return true.*"

When collecting preferences for the first session, the *preference elicitation* act is further divided into three sub-actions for collecting (i) *must-have*, (ii) *should-have*, and (iii) *could-have* preferences. Once the *must-have* and *should-have* preferences are collected, the subsequent sessions only collect *could-have* preferences, using the preference memory for other preferences.

3.4 Utterance Composition

Conversation utterances are composed by the human agent for both user and assistant roles via dialogue self-play. For the assistant role, the composition process is supported by the LLM-generated *guidance*, which enables generating diverse human-written system utterances. For the user role, the human agent mainly needs to elaborate his preferences and state his opinion about the recommendations. The system and user utterance generation functions are defined as:

$$\mathcal{W}_s\left(S_k, g_j, u_1, \ldots, u_{j-1}\right) = u_j,$$
$$\mathcal{W}_u\left(S_k, u_1, \ldots, u_j\right) = u_{j+1},$$

where $\mathcal{W}_s$ and $\mathcal{W}_u$ represent the function for writing system and user responses, respectively. Here, g_j denotes the guidance for generating utterance u_j, and $S_k = \{s_1, \ldots, s_k\}$ represents the session history, comprising all previous dialogue sessions up until, but not including, the current session.

Instantiation. These functions are instantiated by recruiting crowd workers and instructing them to write a self-dialogue for our task: *"Your task is to chat with yourself both as a user (seeking a cooking recipe or movie) and an assistant (offering recipe or movie recommendations). You need to discuss preferences and receive suggestions while playing both roles."* This instruction is followed by brief descriptions of roles, session settings, and chat interface instructions, as well as general information about the payment and rejection policy.

For the user role, workers are instructed to be themselves, provide their preferences, review the recommendations, and offer feedback. In contrast, the assistant role entails more tasks. Assistants must elicit preferences to inform their recommendations and provide URLs for the recommended items. Following user feedback, assistants confirm why the user likes or dislikes the recommendation to obtain a reusable preference for the next session.

3.5 Preference Extraction

Upon completion of a dialogue session, user preferences are extracted from the dialogue. Due to the inherent ambiguity and complexity of human-written conversations, we collect these preferences from the same human agent that generates the conversation. Note that here we only collect user preferences that are mentioned in the course of previous conversation sessions and not general user preferences. Formally, given the dialogue session s_i, the preference extraction function $\mathcal{E}$ is defined as:

$$\mathcal{E}(s_i, C) = P_{s_i}, \tag{2}$$

where C denotes the set of preference categories, and P_{s_i} is the set of category-preference pairs extracted from the dialogue session.

Instantiation. Extraction of preferences is performed using a semi-automated approach, where the initial set of preferences is extracted by an LLM and then validated by the human agent. We prompt GPT4 with chain-of-thought instructions to generate preference attributes for each preference category $c \in C$, given the dialogue session s_i and preference categories C.

Once an initial set of preferences is extracted, the worker is instructed to confirm the correctness of each extracted preference and verify all disclosed preferences during the conversations are extracted. The worker is then directed to start a new conversation session or terminate the task.

3.6 Evaluation

Baseline Datasets. We evaluate our LAPS method by comparing existing conversational datasets with the conversations generated by LAPS. We carefully select the baseline datasets by examining 170 datasets listed by Joko et al. [24], and narrowing down our selection by the following criteria: (i) inclusion of preference elicitation, (ii) utilization of preferences, and (iii) focus on task-oriented dialogues. We prioritize datasets that are both published in peer-reviewed venues and publicly available. The selected datasets are summarized in Table 1. For a fair comparison, when possible, we select a single domain from each baseline dataset that overlaps with the domains of the dataset created by LAPS, namely recipe and movie. For open-domain datasets, e.g., PersonChatGen [30], we select dialogues with food-related personas, categorized under "Food" and "Drink."

Baseline Methods. We tried three other human-based dialogue collection methods: *Human-Human*, *self-dialogue*, and *LLM-human*. These methods do not scale and cannot generate quality conversations. In the *Human-Human* method, we encountered difficulties in pairing up two workers simultaneously due to the high dropout rates, caused by the complex nature of multi-session preference elicitation. For the *self-dialogue* method, we followed [52] and simplified the assistant role by providing users with a predefined response. This, however, led to homogeneous dialogues, even though the workers were instructed to rewrite the response. In the *LLM-human* method, we generated assistant responses using GPT-3.5 `turbo` and asked workers to rewrite the utterances. However, even after experimenting with multiple temperature settings, the dialogue remained homogeneous (a common issue with LLMs reported also in [13, 14, 44, 47, 59]) and workers often accepted the grammatically correct responses without re-writing them.

While we focus on collecting dialogues with actual user preferences, one might wonder whether an LLM could also play the role of a human agent. To address this question, we prompt the LLM to act both as a user and an agent, using the same input and output as human agents in LAPS (cf.§ 3.4). We report on the results obtained from GPT-3.5 `turbo` and GPT-4-1106 `preview` with a temperature parameter of 1.0. These dialogues are 3 sessions long.

Dialogue Diversity. We use three metrics to measure the lexical diversity of the collected conversations: (i) Dist-n [33], (ii) Ent-n [64], and (iii) Self-BLEU [67]. ***Dist-n*** measures the response diversity by computing the ratio of distinct n-grams to all n-grams in the given collection. Following [33], we use Dist-1 and -2 to measure the lexical diversity of conversation utterances. ***Ent-n*** aims to enhance the Dist-n measure by incorporating the frequency differences of n-grams into consideration, leveraging the entropy of the n-gram distribution. We report on Ent-4, following [64]. ***Self-BLEU*** considers one utterance as a hypothesis and the rest of the utterances in the collection as references and calculates the BLEU score for each utterance. Following the NLTK's default setting [5] and [67], we compute the BLEU scores for $n = 1, 2, 3, 4$ for each hypothesis-reference pair and take the mean over all computed BLEU scores.

Normalization. A frequently overlooked pitfall of diversity metrics is their dependency on the total number of words in the dialogues. For instance, Dist-n scores are typically higher for datasets with fewer total number of words, as they are less likely to have repeated words. To address this, we set a word cutoff for all datasets and randomly sample dialogues until the total word count reaches this cutoff. The cutoff is set at 7,012 words, which is the minimum number of total words for user/system utterances across all datasets. We perform 100 random sampling per dataset and report the average scores, as well as two-tailed independent t-test results. Additionally, since M2M [49] dataset is lowercased, we lowercase all other datasets for a fair comparison.

Dialogue Quality. Dialogue quality evaluation is inherently challenging due to its subjective nature. Human evaluation, often considered the gold standard, is known to be highly sensitive to task design and instructions. Even with much care, it still suffers from differing bias and high variance per annotator, especially in crowdsourced environments [15, 17, 34, 50, 63]. Automatic evaluation, while capable of mitigating the aforementioned issues, is also known to have its own limitations, including a bias towards machine-generated responses [38]. Aware of these limitations, we opt for automatic evaluation for two reasons: (1) our aim is to ensure our dialogue quality aligns with other high-quality datasets, rather than attaining state of the art, and (2) we use human workers to compose responses in their own words, which is less likely to be overestimated by metrics biased towards machine-generated responses.

After examining four reference-free automatic evaluation methods [35, 38, 42, 66], we select **UniEval** [66] as our automatic evaluation metric considering availability, cost, and performance. For evaluation aspects, we choose naturalness, understandability, and coherence from Zhong et al. [66]. The aspects that require the conditioning fact as an input are not used, as the factuality of user preferences falls outside the scope of our study. We use the official implementation of UniEval[2] and its default settings. To ensure that the evaluation is computationally feasible using our available computational resources, we randomly select 100 dialogues (consisting of 1.8K responses on average) from each dataset. For significant testing, we use two-tailed independent t-tests ($p < 0.05$).

[2]https://github.com/maszhongming/UniEval

Table 2: Statistics of LAPS dataset.

Domain	Split	#Dialogue Sets			#Pref	#Utt	#Dial
		Single-Session	Two-Session	Three-Session			
Recipe	Train	163	24	160	5,538	9,342	691
	Val	24	5	21	772	1,333	97
	Test	48	10	41	1,600	2,610	191
	Total	**235**	**39**	**222**	**7,910**	**13,285**	**979**
Movie	Train	46	14	72	2,225	3,974	290
	Val	5	1	13	351	642	46
	Test	11	4	24	729	1,220	91
	Total	**62**	**19**	**109**	**3,305**	**5,836**	**427**

3.7 Experimental Setup

Domains. Our domains are *recipe* and *movie*. The recipe domain involves planning for the next dinner (session 1), breakfast (session 2), and lunch (session 3). The movie domain involves planning to watch a movie with family, friends, or alone (session 1), exploring another movie by the same director or actress/actor as the previous recommendation (session 2), and watching with different people or a different occasion (session 3). For each domain, we curated a list of categories that are relevant to the task and categorized them into categories of *must-have*, *should-have*, and *could-have*. These categories are detailed in our online repository.

Participants and Quality Control. We recruited Prolific[3] workers from English-speaking countries, having ≥ 98% approval rate and ≥ 1000 previous submissions. For the movie domain, we only invited workers that accurately performed that task for the recipe domain. Throughout the experiments, we actively communicated with workers, answering over 250 questions and incorporating their feedback to clarify instructions. £14 was paid for completing three sessions, which took ~65 minutes. Workers were also allowed to terminate the task at an earlier session and receive partial payment. This allowed us to collect high-quality multi-session dialogues, as workers completing all sessions tend to be more engaged in the task. After crowdsourcing, we manually reviewed the dialogues to ensure data quality, making corrections or deletions as necessary. Common errors (aside from malicious behavior of not providing meaningful responses) include failing to include URLs in recommendations and misunderstanding their current role in the conversation.

Chat Interface. We collect conversations using TaskMAD [52] and further develop it to support new features for our task. The human agent interacts with two chat interfaces for system and user roles, and each interface consists of a text box for composing responses and an instruction box to clarify role responsibilities.

4 PERSONAL RECOMMENDATION METHOD

The end goal for large-scale preference elicitation conversational datasets is to enable personal conversational search and recommendation. Based on LAPS, we collect a large-scale dataset for the movie and recipe domains and use it to train a model for extracting personal preferences from conversations. The preferences are then used to generate personalized recommendations.

[3]https://www.prolific.com/

Table 3: Lexical diversity scores. Significance against all baselines is marked by $^+$.

Dataset	Dist-1/2	Ent-4	Self-BLEU$^\blacktriangledown$
SGD	0.179 / 0.538	8.311	0.964
M2M	0.057 / 0.290	7.922	0.955
PersonaChatGen	0.165 / 0.523	8.261	0.970
Taskmaster-1	0.207 / 0.644	8.384	0.949
MultiWOZ	0.158 / 0.505	8.345	0.966
CCPE-M	0.175 / 0.571	8.414	0.961
MG-ShopDial	**0.234** / 0.653	8.199	**0.935**
LAPS-Recipe	0.207 / 0.650	8.563$^+$	0.955
LAPS-Movie	0.222$^+$ / **0.666**$^+$	**8.593**$^+$	0.954

$^\blacktriangledown$ Lower is better.

Table 4: Lexical diversity scores of synthetic dialogue generation. Significance against all baselines is marked by $^+$.

Domain	Method	Dist-1/2	Ent-4	Self-BLEU$^\blacktriangledown$
Recipe	Synthetic GPT-3.5	0.127 / 0.430	8.308	0.981
	Synthetic GPT-4	0.183 / 0.597	**8.601**	0.976
	LAPS	**0.207**$^+$ / **0.65**$^+$	8.563	**0.955**$^+$
Movie	Synthetic GPT-3.5	0.138 / 0.444	8.331	0.977
	Synthetic GPT-4	0.178 / 0.559	8.481	0.979
	LAPS	**0.222**$^+$ / **0.666**$^+$	**8.593**$^+$	**0.954**$^+$

$^\blacktriangledown$ Lower is better.

4.1 Preference Extraction

In this task, we aim to automatically extract user preferences from a dialogue session (cf. Eq. 2). We cast this task as a seq-to-seq QA [12] and fine-tune an LLM with instruction prompts eliciting user preference regarding category and conversation session. Formally, our method decomposes Eq. 2 as $\mathcal{E}(s, C) = \bigcup_{c \in C} \mathcal{E}'(s, c)$, where $\mathcal{E}'$ is the preference extraction function for individual category c.

We use FlanT5 [12] as the base model and use instruct prompts to read the given session and answer questions about the user's preferences, such as *"What cuisine does the user like?"* For the recipe domain, we use FlanT5 Small (80M parameters), Base (250M), and Large (780M) models and fine-tune them on our recipe dataset. For the movie domain, we explore a domain adaptation approach, where the initially fine-tuned model on the recipe domain (FlanT5-Large) is further fine-tuned on the movie domain.

4.2 Personalized Recommendation

With the personalized recommendation task, we aim to generate a recommendation response based on user's personal preferences. Personal preferences are stated in the conversation history h and all previously completed sessions S_k. By utilizing the preference extraction method (cf. § 4.1), we construct the preference memory $\mathcal{M}_k$ based on session history S_k and use it for recommendation. Formally, recommendation utterance u^{pred} is generated by the recommendation generation function R, defined as $R(h, \mathcal{M}_k) = u^{pred}$. Recommendation responses are generated by an LLM (LLaMA-2-7B [54]) using zero-shot prompting. The prompt contains instructions to generate a personalized recommendation appended with the preference memory and history of the current conversation.

4.3 Evaluation

Recommendation Baseline. For our personalized recommendation method, we consider a baseline method, where the preference memory $\mathcal{M}_k$ is replaced with raw utterances from session history S_K. This baseline allows us to measure the effectiveness of using semi-structured fine-grained preferences over raw conversational utterances.

Recommendation Human Evaluation. Two human experts are instructed to evaluate the *rationale* and *relevance* aspects of recommendations: "Which assistant's rationale is more in line with the user's preferences?", and " Which assistant's recommendation item meets user preferences?" Following Li et al. [34], we conduct pairwise comparisons between responses from our method and a baseline. For each aspect, the annotators choose win, lose, or tie options, and disagreements are resolved through discussion. For evaluation, we randomly select 50 and 10 samples from the recipe and movie domains, respectively.

Recommendation Automatic Evaluation. Automatic machine translation measures such as ROUGE and BLEU assume significant overlap between ground truth and valid responses. This strong assumption do not hold for dialogue systems and in particular for personal recommendations, where valid responses represent high level of diversity. The lack of correlation between human evaluation is reported in previous study [36] and has been observed in our study as well. Addressing this challenge, we introduce an automatic reference-based evaluation metric, **Preference Utilization (PU)**, which aims to measure the utilization of user preferences in the recommendation response. Let $\mathcal{P} = \{p_1, p_2, \ldots\}$ denote the set of preferences (without their corresponding categories as defined in Eq. 1). The subsets $\mathcal{P}^{pred} \subseteq \mathcal{P}$ and $\mathcal{P}^{ref} \subseteq \mathcal{P}$ denote preferences $p_i \in \mathcal{P}$ that appear in predicted and reference recommendation responses, respectively. Preference utilization precision P_{PU} and recall R_{PU} are computed as:

$$P_{PU} = \frac{|\mathcal{P}^{pred} \cap \mathcal{P}^{ref}|}{|\mathcal{P}^{pred}|}, \quad R_{PU} = \frac{|\mathcal{P}^{pred} \cap \mathcal{P}^{ref}|}{|\mathcal{P}^{ref}|}.$$

In our experiments, we perform exact string matching to generate $\mathcal{P}^{pred}$ and $\mathcal{P}^{ref}$. When refference utterance u^{ref} includes URLs, we extract the content from the corresponding web page and use it in string matching. For our ground truth, we only use the assistant responses about recommendations which are accepted by the user.

Preference Extraction Evaluation. We evaluate preference extraction using exact match and BERTScore [62]. Exact match assesses the case-insensitive string match of preferences between predictions and ground truth, while BERTScore accounts for different expressions of identical preferences. Here, preferences within each category are flattened into a single string with commas as delimiters and used for computation of BERTScore.

4.4 Experimental Setup

For preference extraction, we fine-tuned Flan-T5 models on our training set using the HuggingFace Transformers library [57]. For both domains, the batch size is set to 8, the learning rate to 5e-5, and

Table 5: UniEval scores. Significance against all baselines is marked by $^{+}$.

Dataset	NAT	UND	COH	Avg.
SGD	0.794	0.781	0.758	0.778
M2M	0.634	0.616	0.701	0.650
Taskmaster-1	0.792	0.779	0.782	0.784
MultiWOZ	0.870	0.860	0.848	0.859
CCPE-M	0.716	0.708	0.689	0.704
MG-ShopDial	0.743	0.730	0.687	0.720
LAPS-Recipe	0.867	0.860	0.891^{+}	0.872^{+}
LAPS-Movie	**0.874**	**0.868**	**0.897**$^{+}$	**0.880**$^{+}$
PersonaChatGen†	0.894	0.887	0.738	0.839

† Scores provided as a reference, but do not represent fair comparison.

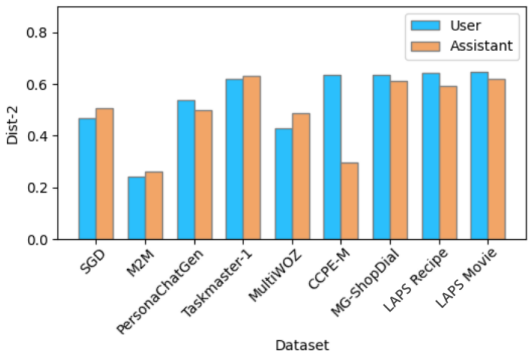

Figure 3: Lexical Diversity of user and assistant utterances.

the AdamW optimizer is used. Training is conducted for 10 epochs, with the best checkpoint selected based on the validation set. We ran all our experiments on a single GPU (NVIDIA A100 40GB). For a personalized recommendation, we use Llama-2-7B [54] due to its ability to handle a sufficiently long context, while being capable of running on a single GPU. We run the model (from the HuggingFace model hub) 10 times for each dialogue and report the average score.

5 RESULTS

This section presents the results of the proposed methods for dialogue collection (§5.1) and personal recommendation (§5.2)

5.1 Dialogue Collection Evaluation

Using LAPS, we can collect a large-scale multi-session personalized dataset, as shown in Table 2. The number of preferences is the total number of (s, p, o) triples, where s is the user, p is the preference category, and o is the preference attribute. Using human verification, we identified 4.5% error rate in preferences extracted by GPT-4, highlighting the need for human involvement for accurate preference extraction.

Lexical Diversity. Table 3 shows the results of lexical diversity evaluation, indicating that LAPS-Movie and -Recipe achieve the highest diversity scores with respect to Dist-2 and Ent-4. Considering all metrics, LAPS-Movie is on par with MG-ShopDial, a dataset of human-human dialogues involving trained volunteers as an assistant role. These results demonstrate that LAPS collects lexically diverse dialogues as effectively as human-human dialogue collection methods.

Table 6: Preference extraction results. Domain adapt. represents the model is first fine-tuned on the recipe domain and then further fine-tuned on the movie domain.

Domain	Model Size	Domain Adpt.	Exact Match P	Exact Match R	Exact Match F	BERTScore P	BERTScore R	BERTScore F
Recipe	Small	–	0.454	0.408	0.412	0.590	0.589	0.589
	Base	–	0.464	0.476	0.454	0.622	0.623	0.622
	Large	–	**0.532**	**0.493**	**0.494**	**0.651**	**0.650**	**0.650**
Movie	Large	×	0.453	0.415	0.425	0.598	0.592	0.595
		✓	**0.470**	**0.424**	**0.432**	**0.618**	**0.614**	**0.616**

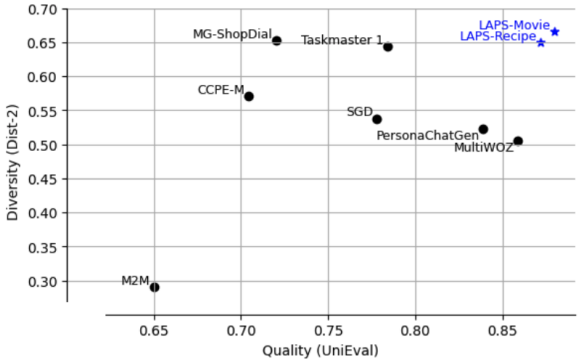

Figure 4: LAPS collects diverse and high-quality dialogues compared to other dialogue collection methods.

Comparing the lexical diversity of user and system utterances in Figure 3, we observe that our method achieves high lexical diversity for both user and assistant utterances. This suggests that LLM's guidance can help workers compose diverse responses. Notably, we observe that assistant utterances in CCPE-M exhibit less diversity than those of the user. This could be attributed to the small number of participants acting as assistants in CCPE-M and their often short and direct responses. This highlights that achieving diversity is non-trivial, even with trained experts. LAPS's success in achieving diversity further demonstrates the effectiveness of our method.

Table 4 compares LAPS- and LLM-generated conversations. The results show that synthetic dialogues are less diverse than LAPS, even with GPT-4 temperature tuning. This suggests potential diversity pitfalls in synthetic personalized dialogue generation using LLMs. One way to mitigate this issue is using a synthetic persona, as in PersonaChatGen [30]. However, as Table 3 shows, it still falls short of human-involved LAPS, suggesting the diverse nature of human preferences.

Dialogue Quality. Table 5 shows the results of the dialogue quality evaluation. For coherence, LAPS outperforms the other datasets. For naturalness and understandability, PersonaChatGen, which is an LLM-based fully-synthetic dataset, outperforms the other datasets. This is consistent with the findings from Liu et al. [38], which shows that LLM-based synthetic dialogues tend to have higher scores for language-model-based automatic evaluation metrics. Excluding fully-synthetic datasets, LAPS's performance is among the best, demonstrating that our method can collect high-quality dialogues as effectively as human-human dialogue collection methods. On average, LAPS outperforms the other datasets, highlighting the effectiveness of our method.

Table 7: Recommendation results. Significance against Standard is marked by $^+$.

Domain	Prompting Method	#Prompt Tokens	Preference Utilization		
			P_{PU}	R_{PU}	F_{PU}
Recipe	Standard	880	**0.554**	0.311	0.398
	Memory	**308**	0.470	**0.411**$^+$	**0.438**$^+$
Movie	Standard	957	**0.508**	0.364	**0.424**
	Memory	**311**	0.443	**0.397**$^+$	0.419

Discussion. Based on these results we can positively answer our first and second research questions: ***RQ1:*** *Using LAPS, we can collect large-scale multi-session human-written conversations that contain actual user preferences.* and ***RQ2:*** *LAPS-collected dialogues show high diversity and quality, on par with expert-involved human-human dialogues, as highlighted in Figure 4.*

5.2 Personal Recommendation Evaluation

Preference Extraction. Table 6 shows the preference extraction performance of different FlanT5 pre-trained model sizes for the recipe topic. The results show that the performance for the Recipe domain improves as the model size increases. Using the recipe domain as the source domain, we further fine-tune the FlanT5-Large fine-tuned model on the target movie domain. The results show that domain adaptation consistently outperforms direct fine-tuning. This demonstrates the adaptability of our dataset to other domains through domain adaptation.

Recommendation Human Evaluation. Table 8 presents the human evaluation results for recommendation quality. In the recipe domain, using preference memory outperforms the baseline, for both recommendation quality and rationale. For the movie domain, the Memory method excels in rationale but not in recommendation quality. Given that improvements are consistently found in the rationale aspect, using preference memory is effective in improving the rationale for recommendations, a critical factor for transparent and explainable recommendations.

Recommendation Automatic Evaluation. To evaluate the effectiveness of our preference-based prompting, we compare it with the baseline standard prompting method. The downstream recommendation task results are depicted in Table 7. The Preference Utilization results show a similar trend to the human evaluation results, demonstrating the overall win of Memory over the Standard method. In the recipe domain, Memory outperforms Standard in F_{PU} score (0.438 vs. 0.398), whereas in the movie domain, their scores show no statistical significance (0.419 vs. 0.424). Error analysis indicates that the preference memory in the movie domain is sometimes insufficient for making recommendations, suggesting a need for improving the preference extraction method. Nevertheless, preference-based prompting for movies reduces the prompt length threefold while maintaining a comparable F_{PU} score. This threefold reduction has significant implications for real-world applications where inference cost is a critical factor.

Preference Utilization by Session. Figure 5 shows the Preference Utilization scores by session in recipe recommendations. The x-axis represents the session where preferences are disclosed, as detailed in Section 4.3. The analysis focuses on the recommendations in the third session to examine Preference Utilization across different sessions. The graph shows methods' struggle with utilizing preferences from earlier sessions (1st and 2nd) than those from the ongoing (3rd) session. A similar phenomenon is reported in [37] in a retrieval augmentation setting, referred to as *LLMs' recall issue in long prompt inputs*. The graph also demonstrates that preference-based prompting more effectively utilizes earlier session preferences than standard prompting, suggesting that preference memory can mitigate long prompt recall issues. We observe that this pattern (recall issues in earlier sessions and the effectiveness of preference-based prompting in addressing them) is generally consistent across the movie domain and different sessions, though not always statistically significant. Overall, our analysis shows the effectiveness of preference memory in long prompts.

Table 8: Human evaluation results of recommendations.

Domain	Prompting Method	Rationale			Relevance		
		Win	Lose	Tie	Win	Lose	Tie
Recipe	Standard	17	29	4	18	22	10
	Memory	**29**	**17**	4	**22**	**18**	10
Movie	Standard	3	6	1	**4**	**3**	3
	Memory	**6**	**3**	1	3	4	3

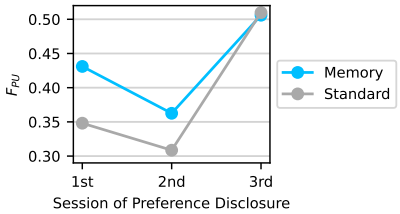

Figure 5: Breakdown of Preference Utilization (F_{PU}, recipe domain).

Discussion. Based on these results we can positively answer our last research questions: ***RQ3*** *Preference memory enhances effective utilization of user preferences in recommendations, improves the rationale of recommendations, and mitigates long prompt recall issues.*

6 CONCLUSION

In this research, we proposed a method to collect large-scale multi-session personalized conversations reflecting actual user preferences. Our method, LAPS, employs LLMs to generate personalized guidance for human workers, reducing the cognitive load for a highly complex task. Extensive experiments demonstrate, while being a scalable and high-quality data collection method, LAPS can collect utterances as diverse as the expert-involved methods. We further showed that utilizing extracted user preferences results in more effective personal recommendations compared to using raw user utterances of previous sessions. In our experiment, fully-synthetic LLM-based methods does not yield diverse conversations. Using actual user preferences from LAPS as personas is a promising avenue to explore for future.

Acknowledgments. This work is supported by the Radboud-Glasgow Collaboration Fund and in part by the Engineering and Physical Sciences Research Council (EPSRC) Grant EP/V025708/1.

REFERENCES

[1] Mohammad Aliannejadi, Zahra Abbasiantaeb, Shubham Chatterjee, Jeffery Dalton, and Leif Azzopardi. 2024. TREC iKAT 2023: The Interactive Knowledge Assistance Track Overview. *arXiv preprint arXiv:2401.01330* (2024).

[2] Sanghwan Bae, Donghyun Kwak, Sungdong Kim, Donghoon Ham, Soyoung Kang, Sang-Woo Lee, and Woomyoung Park. 2022. Building a Role Specified Open-Domain Dialogue System Leveraging Large-Scale Language Models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

[3] Krisztian Balog and Tom Kenter. 2019. Personal Knowledge Graphs: A Research Agenda. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*.

[4] Nolwenn Bernard and Krisztian Balog. 2023. MG-ShopDial: A Multi-Goal Conversational Dataset for e-Commerce. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, July 23–27, 2023, Taipei, Taiwan*.

[5] Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.".

[6] Penelope Brown and Stephen C Levinson. 1987. *Politeness: Some universals in language usage*. Cambridge university press.

[7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*.

[8] Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

[9] Bill Byrne, Karthik Krishnamoorthi, Chinnadhurai Sankar, Arvind Neelakantan, Ben Goodrich, Daniel Duckworth, Semih Yavuz, Amit Dubey, Kyu-Young Kim, and Andy Cedilnik. 2019. Taskmaster-1: Toward a Realistic and Diverse Dialog Dataset. 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)*.

[10] Maximillian Chen, Alexandros Papangelis, Chenyang Tao, Seokhwan Kim, Andy Rosenbaum, Yang Liu, Zhou Yu, and Dilek Hakkani-Tur. 2023. PLACES: Prompting Language Models for Social Conversation Synthesis. In *Findings of the Association for Computational Linguistics: EACL 2023*.

[11] Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. 2016. Towards Conversational Recommender Systems. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[12] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416* (2022).

[13] John Chung, Ece Kamar, and Saleema Amershi. 2023. Increasing Diversity While Maintaining Accuracy: Text Data Generation with Large Language Models and Human Interventions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*.

[14] Fabrizio Dell'Acqua, Edward McFowland III, Ethan Mollick, Hila Lifshitz-Assaf, Katherine C. Kellogg, Saran Rajendran, Lisa Krayer, François Candelon, and Karim R. Lakhani. 2023. Navigating the Jagged Technological Frontier: Field Experimental Evidence of the Effects of AI on Knowledge Worker Productivity and Quality. *Harvard Business School Working Paper* (September 2023).

[15] Jan Deriu, Alvaro Rodrigo, Arantxa Otegi, Guillermo Echegoyen, Sophie Rosset, Eneko Agirre, and Mark Cieliebak. 2021. Survey on evaluation methods for dialogue systems. *Artificial Intelligence Review* 54 (2021).

[16] Joachim Fainberg, Ben Krause, Mihai Dobre, Marco Damonte, Emmanuel Kahembwe, Daniel Duma, Bonnie Webber, and Federico Fancellu. 2018. Talking to myself: self-dialogues as data for conversational agents. *arXiv preprint arXiv:1809.06641* (2018).

[17] Sarah E. Finch, James D. Finch, and Jinho D. Choi. 2023. Don't Forget Your ABC's: Evaluating the State-of-the-Art in Chat-Oriented Dialogue Systems. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*.

[18] Jianfeng Gao, Michel Galley, and Lihong Li. 2019. Neural Approaches to Conversational AI. *Foundations and Trends® in Information Retrieval* 13 (2019).

[19] Emma J. Gerritse, Faegheh Hasibi, and Arjen P. de Vries. 2020. Bias in Conversational Search: The Double-Edged Sword of the Personalized Knowledge Graph. In *Proceedings of the 2020 ACM SIGIR on International Conference on Theory of Information Retrieval*.

[20] Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. ChatGPT outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences* (2023).

[21] 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).

[22] Hideo Joho, Lawrence Cavedon, Jaime Arguello, Milad Shokouhi, and Filip Radlinski. 2017. First International Workshop on Conversational Approaches to Information Retrieval (CAIR'17). In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[23] Hideaki Joko and Faegheh Hasibi. 2022. Personal Entity, Concept, and Named Entity Linking in Conversations. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management*.

[24] Hideaki Joko, Faegheh Hasibi, Krisztian Balog, and Arjen P. de Vries. 2021. Conversational Entity Linking: Problem Definition and Datasets. In *Proceedings of the 44rd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[25] Hyunwoo Kim, Jack Hessel, Liwei Jiang, Peter West, Ximing Lu, Youngjae Yu, Pei Zhou, Ronan Le Bras, Malihe Alikhani, Gunhee Kim, Maarten Sap, and Yejin Choi. 2023. SODA: Million-scale Dialogue Distillation with Social Commonsense Contextualization. *arXiv preprint arXiv:2212.10465* (2023).

[26] Minju Kim, Chaehyeong Kim, Yong Ho Song, Seung-won Hwang, and Jinyoung Yeo. 2022. BotsTalk: Machine-sourced Framework for Automatic Curation of Large-scale Multi-skill Dialogue Datasets. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*.

[27] Ivica Kostric, Krisztian Balog, and Filip Radlinski. 2021. Soliciting User Preferences in Conversational Recommender Systems via Usage-related Questions. In *Proceedings of the Fifteenth ACM Conference on Recommender Systems*.

[28] Ivica Kostric, Krisztian Balog, and Filip Radlinski. 2023. Generating Usage-related Questions for Preference Elicitation in Conversational Recommender Systems. *ACM Trans. Recomm. Syst.* (2023).

[29] Ben Krause, Marco Damonte, Mihai Dobre, Daniel Duma, Joachim Fainberg, Federico Fancellu, Emmanuel Kahembwe, Jianpeng Cheng, and Bonnie Webber. 2017. Edina: Building an open domain socialbot with self-dialogues. *arXiv preprint arXiv:1709.09816* (2017).

[30] Young-Jun Lee, Chae-Gyun Lim, Yunsu Choi, Ji-Hui Lm, and Ho-Jin Choi. 2022. PERSONACHATGEN: Generating Personalized Dialogues using GPT-3. In *Proceedings of the 1st Workshop on Customized Chat Grounding Persona and Knowledge*.

[31] Megan Leszczynski, Ravi Ganti, Shu Zhang, 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).

[32] Belinda Z. Li, Alex Tamkin, Noah Goodman, and Jacob Andreas. 2023. Eliciting Human Preferences with Language Models. *arXiv preprint arXiv:2310.11589* (2023).

[33] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and William B Dolan. 2016. A Diversity-Promoting Objective Function for Neural Conversation Models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

[34] Margaret Li, Jason Weston, and Stephen Roller. 2019. Acute-eval: Improved dialogue evaluation with optimized questions and multi-turn comparisons. *arXiv preprint arXiv:1909.03087* (2019).

[35] Yen-Ting Lin and Yun-Nung Chen. 2023. LLM-Eval: Unified Multi-Dimensional Automatic Evaluation for Open-Domain Conversations with Large Language Models. In *Proceedings of the 5th Workshop on NLP for Conversational AI*.

[36] Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How NOT To Evaluate Your Dialogue System: An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Jian Su, Kevin Duh, and Xavier Carreras (Eds.).

[37] Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the Middle: How Language Models Use Long Contexts. *arXiv preprint arXiv:2307.03172* (2023).

[38] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-Eval: NLG Evaluation using Gpt-4 with Better Human Alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

[39] Liangchen Luo, Wenhao Huang, Qi Zeng, Zaiqing Nie, and Xu Sun. 2019. Learning Personalized End-to-end Goal-oriented Dialog. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[40] Aman Madaan, Niket Tandon, Peter Clark, and Yiming Yang. 2022. Memory-assisted prompt editing to improve GPT-3 after deployment. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*.

[41] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-Refine: Iterative Refinement with Self-Feedback. *arXiv preprint arXiv:2303.17651* (2023).

[42] Shikib Mehri and Maxine Eskenazi. 2020. USR: An Unsupervised and Reference Free Evaluation Metric for Dialog Generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

[43] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems* (2022).

[44] Peter S. Park, Philipp Schoenegger, and Chongyang Zhu. 2023. Diminished Diversity-of-Thought in a Standard Large Language Model. *arXiv preprint arXiv:2302.07267* (2023).

[45] Filip Radlinski, Krisztian Balog, Bill Byrne, and Karthik Krishnamoorthi. 2019. Coached Conversational Preference Elicitation: A Case Study in Understanding Movie Preferences. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*.

[46] Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards Scalable Multi-Domain Conversational Agents: The Schema-Guided Dialogue Dataset. *Proceedings of the AAAI Conference on Artificial Intelligence* 34 (2020).

[47] Emily Reif, Minsuk Kahng, and Savvas Petridis. 2023. Visualizing Linguistic Diversity of Text Datasets Synthesized by Large Language Models. *arXiv preprint arXiv:2305.11364* (2023).

[48] Christopher Riesbeck. 1981. Failure-Driven Reminding for Incremental Learning. In *IJCAI*.

[49] Pararth Shah, Dilek Hakkani-Tür, Bing Liu, and Gokhan Tür. 2018. Bootstrapping a Neural Conversational Agent with Dialogue Self-Play, Crowdsourcing and On-Line Reinforcement Learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

[50] Eric Smith, Orion Hsu, Rebecca Qian, Stephen Roller, Y-Lan Boureau, and Jason Weston. 2022. Human Evaluation of Conversations is an Open Problem: comparing the sensitivity of various methods for evaluating dialogue agents. In *Proceedings of the 4th Workshop on NLP for Conversational AI*.

[51] Heydar Soudani, Evangelos Kanoulas, and Faegheh Hasibi. 2023. Data Augmentation for Conversational AI. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*.

[52] Alessandro Speggiorin, Jeffrey Dalton, and Anton Leuski. 2022. TaskMAD: A Platform for Multimodal Task-Centric Knowledge-Grounded Conversational Experimentation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[53] Yueming Sun and Yi Zhang. 2018. Conversational Recommender System. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*.

[54] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023).

[55] Johannes M. van Hulst, Faegheh Hasibi, Koen Dercksen, Krisztian Balog, and Arjen P. de Vries. 2020. REL: An Entity Linker Standing on the Shoulders of Giants. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[56] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems*.

[57] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*.

[58] Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley. 2023. Baize: An open-source chat model with parameter-efficient tuning on self-chat data. *arXiv preprint arXiv:2304.01196* (2023).

[59] Yue Yu, Yuchen Zhuang, Jieyu Zhang, Yu Meng, Alexander Ratner, Ranjay Krishna, Jiaming Shen, and Chao Zhang. 2023. Large Language Model as Attributed Training Data Generator: A Tale of Diversity and Bias. *arXiv preprint arXiv:2306.15895* (2023).

[60] Li Yunxiang, Li Zihan, Zhang Kai, Dan Ruilong, and Zhang You. 2023. Chatdoctor: A medical chat model fine-tuned on llama model using medical domain knowledge. *arXiv preprint arXiv:2303.14070* (2023).

[61] Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing Dialogue Agents: I have a dog, do you have pets too?. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

[62] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating Text Generation with BERT. In *International Conference on Learning Representations*.

[63] Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B. Hashimoto. 2023. Benchmarking Large Language Models for News Summarization. *arXiv preprint arXiv:2301.13848* (2023).

[64] Yizhe Zhang, Michel Galley, Jianfeng Gao, Zhe Gan, Xiujun Li, Chris Brockett, and Bill Dolan. 2018. Generating Informative and Diverse Conversational Responses via Adversarial Information Maximization. In *Advances in Neural Information Processing Systems*.

[65] Canzhe Zhao, Tong Yu, Zhihui Xie, and Shuai Li. 2022. Knowledge-aware Conversational Preference Elicitation with Bandit Feedback. In *Proceedings of the ACM Web Conference 2022*.

[66] Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022. Towards a Unified Multi-Dimensional Evaluator for Text Generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*.

[67] 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*.