Knowledge-Augmented Large Language Models for Personalized Contextual Query Suggestion

Jinheon Baek*
KAIST
Seoul, South Korea
jinheon.baek@kaist.ac.kr

Nirupama Chandrasekaran
Microsoft Research
Redmond, WA, USA
niruc@microsoft.com

Silviu Cucerzan
Microsoft Research
Redmond, WA, USA
silviu@microsoft.com

Allen Herring
Microsoft Research
Redmond, WA, USA
allenh@microsoft.com

Sujay Kumar Jauhar
Microsoft Research
Redmond, WA, USA
sjauhar@microsoft.com

ABSTRACT

Large Language Models (LLMs) excel at tackling various natural language tasks. However, due to the significant costs involved in re-training or fine-tuning them, they remain largely static and difficult to personalize. Nevertheless, a variety of applications could benefit from generations that are tailored to users' preferences, goals, and knowledge. Among them is web search, where knowing what a user is trying to accomplish, what they care about, and what they know can lead to improved search experiences. In this work, we propose a novel and general approach that augments an LLM with relevant context from users' interaction histories with a search engine in order to personalize its outputs. Specifically, we construct an entity-centric knowledge store for each user based on their search and browsing activities on the web, which is then leveraged to provide contextually relevant LLM prompt augmentations. This knowledge store is light-weight, since it only produces user-specific aggregate projections of interests and knowledge onto public knowledge graphs, and leverages existing search log infrastructure, thereby mitigating the privacy, compliance, and scalability concerns associated with building deep user profiles for personalization. We validate our approach on the task of contextual query suggestion, which requires understanding not only the user's current search context but also what they historically know and care about. Through a number of experiments based on human evaluation, we show that our approach is significantly better than several other LLM-powered baselines, generating query suggestions that are contextually more relevant, personalized, and useful.

CCS CONCEPTS

• **Computing methodologies** → **Natural language processing**;
• **Information systems** → *Personalization*.

---

*This work was done during an internship at Microsoft Research.
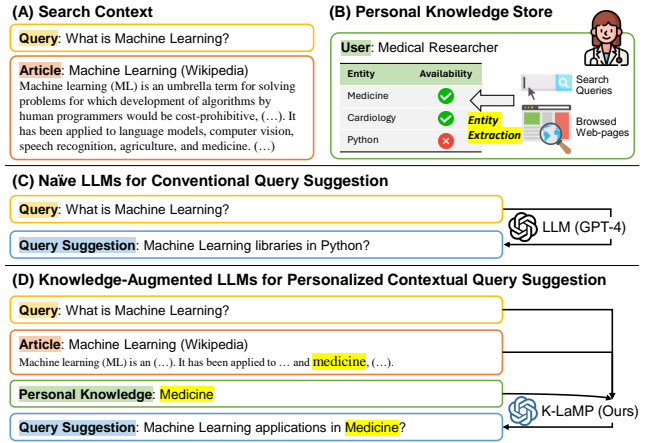

Figure 1: Illustration of our proposed Knowledge-augmented large Language Model for Personalization (K-LaMP) framework for contextual query suggestion. (A) A user's search context includes a current query and an page being viewed. (B) The user has knowledge of medicine, extracted from past search activities. (C) Conventional query suggestion with naïve LLMs generates a query unrelated to the user's knowledge and the context. (D) K-LaMP suggests a query that is personally and contextually relevant.

KEYWORDS

Large Language Models, Personalization, Entity-centric Knowledge, Contextual Query Suggestion

1 INTRODUCTION

Large Language Models (LLMs) [6, 9, 16, 53, 69, 70], such as GPT-4, are multi-billion parameter models trained on massive text corpora, which are capable of internalizing general knowledge across diverse domains [55, 59]. This capability allows them to generate plausible, reasonable, and helpful outputs in response to user inputs that been leveraged with impressive results for a diverse range of natural language tasks, including question answering and dialogue generation, even without any task-specific training [6, 53, 70].

Despite these successes, customizing LLMs to generate personalized responses, which take into account the individual preferences, needs, knowledge and context of users, with the goal of making them more meaningful and relevant to each user, remains challenging. This is due to the fact that re-training or fine-tuning LLMs for individual users is prohibitively expensive. Meanwhile, for some real-world applications, such as query suggestion [22, 65], item recommendations [26, 77], snippet generation [13] or question answering [31], reflecting personal preferences, knowledge and needs of users in the model's outputs is essential.

Several recent studies [20, 29, 32, 36, 46, 48] have tackled the problem of LLM personalization through augmenting the user's input with relevant information, a process known as in-context learning [9, 39, 58, 75]. For example, in order to suggest the next item that the user may interact with, they prepend a sequence of their past item interactions into the LLM's input [14, 46]. Due to the often large volume of users' historical information, more recent work [60, 64, 73, 79] proposes to inject only a fraction of the most relevant history by retrieving it from a complete interaction memory. Such retrieval-based personalization, while demonstrating success on several tasks including product recommendation and writing assistance [14, 44, 60, 73], remains ill-suited for more challenging personalization scenarios where a deeper understanding of users' personal *knowledge* is crucial. Meanwhile, some studies [38, 80] enable personalization through construction of deep user profiles that are then incorporated into LLM prompts. Nevertheless, such profile-based personalization captures user knowledge at the cost of privacy and scalability, requiring online modeling beyond the capabilities of existing logging infrastructure.

Thus, in this paper, we introduce a novel approach to personalizing the output of LLMs. Our method revolves around *an entity-centric light-weight personalization layer* that enables knowledge-augmentation of LLMs with contextual entities retrieved from a *personal knowledge store*. This knowledge store is derived from existing search logs that capture users' interactions with modern search engines. Specifically, this store is built over time by aggregating entities that appeared in queries that the user issued, or web-pages that they browsed, and is further enhanced by different views that capture the entities the user may be familiar or unfamiliar with, and those that may have recently lapsed from their memory.

This entity-centric personalization strategy has several advantages. First, contextually relevant retrievals from this entity-centric knowledge store encourage LLMs to generate outputs that are more deeply grounded in what users know and care about as compared with linearly stored past query logs. At the same time, it largely relies on already existing logging infrastructure, which means that it is more amenable to privacy, flexibility and scalability considerations than profile-based personalization, as it reduces data collection, modeling and update overheads. Also, thanks to its light-weight design, our approach offers easy integration with existing LLMs for other personalization tasks, both in search [13, 31] and beyond [44, 60]. Finally, our knowledge augmentation method is cost efficient since the knowledge injection employs entities as atoms. This results in minimal, succinct additions to the prompt, unlike other LLM contextualization approaches that operate over raw text [39, 58, 75]. We refer to our framework as **K**nowledge-augmented large **La**nguage **M**odels for **P**ersonalization, or K-LaMP.

While our method of personalizing LLM outputs is broadly applicable to problems in search (and beyond), it is especially relevant to tasks that require modeling the *knowledge* of users in addition to their interests. One such challenging task is a new variant of query suggestion [4, 12, 23, 66], that we call *contextual* query suggestion. In this variant, a system must recommend queries to a user conditioned on a web-page they are currently reading, in addition to their historical query information. Thus, in this setting, knowing a user's domain of expertise and proficiency about a particular topic can lead to substantially different suggestions, as shown in Figure 1. It is worth noting that contextual query suggestion is different from existing *context-aware* query suggestion in the literature [5, 15, 34, 37], since the latter neither conditions recommendations on the body of the web-page being viewed by the user, nor explicitly captures the user's knowledge, instead focusing on surface-level relationships between queries and pages, or their titles.

In our study, we validate the effectiveness of K-LaMP for contextual query suggestion, using real-world search logs from the Bing search engine [2], which is also used to construct our entity-centric knowledge store. On a battery of tests conducted via human evaluation, we find that K-LaMP substantially outperforms several LLM-powered (contextual) query suggestion baselines in generating recommendations that are better related and more useful to individuals, while maintaining high search result quality. Further analyses demonstrate that K-LaMP retrieves contextually relevant knowledge in a highly effective manner, and continues to become more performant as longer user interaction histories are processed and stored, neither of which other baselines are capable of doing.

2 BACKGROUND AND RELATED WORK

Large Language Models. Language models [24, 47, 56, 57], which are pre-trained on unannotated text corpora using Transformer architectures [71] based on self-supervised learning objectives, have been shown to acquire knowledge from text corpora [55, 59, 67] and have been successfully used for various natural language tasks, such as question answering and dialogue generation tasks [35, 68]. Recently, Large Language Models (LLMs) [6, 53, 70], which are scaled-up versions of language models, have demonstrated the capability of handling diverse language tasks across various domains. In particular, LLMs have shown increased capacity for knowledge acquisition and retention thanks to their very large number of parameters [49, 78], as well as a remarkable ability to generalize across new domains with no need for additional task-specific fine-tuning and training data [62, 74]. Moreover, they are able to understand the context of given inputs and then generate contextually coherent responses, allowing users and system designers to easily customize LLM responses through prompt engineering [39, 58, 75]. For example, to generate factually correct answers in response to input questions, existing work [7, 40, 63] typically augments the internalized knowledge in LLMs with externally relevant factual knowledge related to questions. However, while they can effectively provide *generic responses* that may apply to a broad range of users, it remains challenging to generate *personalized responses* that capture the unique preferences, needs, and knowledge of individual users.

Large Language Models for Personalization. In order to yield outputs that are customized to individual users, recent studies [10,

45] propose to personalize the generations of LLMs, with applications spanning various tasks and domains. These include product or content recommendations [14, 27, 30, 73], dialogue generations [64, 79], writing assistants [44, 60], and even robotic systems [76]. Specifically, early work [20, 29, 32, 36, 46, 48] proposes to incorporate the historical sequence of the user's interactions (e.g., recent purchase logs of items) into LLMs prompts, thereby allowing LLMs to generate outputs that are personalized (e.g., next item recommendation). While this simple, linear injection mechanism can effectively provide LLMs with relevant contextual information for personalization, it is limited by often very large interaction histories which exceed the capacity of LLM prompt windows. Also, not all of this history is relevant to every query. Based on this observation, recent work [60, 64, 73, 79] instead retrieves relevant content from an external memory [54] that stores the user's historical information. A few studies [44, 79] go a step further, processing the information in the interaction memory – for example, with summarization or key-word extraction – to gain higher-level insights from the user's history when contextualizing LLMs for personalization.

Unlike prior work in LLM output personalization that focuses largely on modeling the *interests* of users, our work additionally targets their *knowledge* over topics and domains of interest. Accordingly, rather than only leveraging the linearly stored interaction histories of users, in this work, we build a personal knowledge store consisting of entities mined from search queries and page visitations. This mechanism, which provides a lens through which user knowledge can be captured, has two additional advantages: it enables light-weight personalization by retrieval from the knowledge store without requiring explicit profiling of users [38, 80]; the knowledge represented as entities is succinct, thereby leading to efficiency gains through reductions in input context length when compared with existing LLM contextualization work [58].

Search Query Suggestion. The goal of query suggestion is to recommend new queries of potential interest to users, based on current and previous queries in and across search sessions. This task is both highly practical and useful, having been shipped in web-scale search engines such as Google and Bing, as well as been widely applied to other tasks and domains, such as task-oriented search [25, 28] and recruitment platforms [82]. Early work on query suggestion has used frequency-based statistical (probabilistic) methods, which include Markov or LDA models [11, 33, 37, 50, 72]. More recently, neural network methods based on recurrent or attention-based architectures [12, 23, 34, 66, 81] have been leveraged to better model past query sequences and generalize to unseen and long-tail queries. Meanwhile, other studies have proposed to improve training strategies by performing either multi-task learning with a document ranker [4, 5, 15] or reinforcement learning [8]. Finally, other recent work [51, 52] uses pre-trained language models [24, 43] to achieve superior performances with larger model capacity.

In comparison to prior work on query suggestion, we tackle the novel but practical task of *contextual* query suggestion, where recommendations are additionally conditioned on the web-page a user is currently viewing. This task is notably different from existing query suggestion work that leverages previously clicked pages [5, 15, 34, 37] only through surface-level association (such as relationships between past queries and page titles), because it

requires contextualizing the full text of the page. This novel task is of particular interest to us because it exposes the need for personalized models that recommend queries based not only on what users are interested in, but also on what and how much they *know*.

3 K-LAMP: KNOWLEDGE-AUGMENTED LLMS FOR PERSONALIZED QUERY SUGGESTION

In this section, we introduce our approach to generating personalized outputs using a novel knowledge-augmentation method for LLMs and our entity-centric personal knowledge store. We also detail its application to the novel task of contextual query suggestion.

3.1 Problem Statement

We begin with preliminaries, formally introducing LLMs and the problem of contextual query suggestion.

Large Language Models. Let us define an LLM as a model, parameterized by θ, that takes both an input sequence of tokens $\boldsymbol{x} = [x_1, x_2, ..., x_n]$ and a supplemental sequence of context tokens $\boldsymbol{c} = [c_1, c_2, ..., c_k]$ as a prompt, and then generates an output sequence of tokens $\boldsymbol{y} = [y_1, y_2, ..., y_m]$. Then, formally, the inference of an LLM can be represented as: $\boldsymbol{y} = \text{LLM}_\theta(\boldsymbol{x}, \boldsymbol{c})$. Here, θ is the set of fixed model parameters typically pre-trained auto-regressively on massive text corpora; $\boldsymbol{x}$ is a task-dependant user-issued prompt or set of instructions; and $\boldsymbol{c}$ is some additional context provided by an auxiliary system that helps augment, ground, or otherwise improve the quality of the input, so that the LLM is able generate outputs $\boldsymbol{y}$ more effectively. This paper particularly focuses on the nature of $\boldsymbol{c}$ for the task of contextual query suggestion defined below, and with the use of entity-centric knowledge for more personalized outputs.

Contextual Query Suggestion. Before formalizing Contextual Query Suggestion, we first define the task of conventional Query Suggestion. Let q_j be the most recent query issued by a user and $\boldsymbol{q}_h = [q_1, q_2, ..., q_{j-1}]$ be a sequence of their historical queries. Then, a query suggestion model QS aims to predict new queries q_{j+1} that an individual user with current query q_j and query history $\boldsymbol{q}_h$ might be likely to find useful. This process can be summarized as follows: $q_{j+1} = \text{QS}_\theta(q_j, \boldsymbol{q}_h)$.

Contextual query suggestion expands on this definition to incorporate a broader set of context $\boldsymbol{c} = [c_1, c_2, ..., c_k]$ linearized as sequences of text. Specifically, let us first assume that $\boldsymbol{x}$ is an input query: $\boldsymbol{x} = q_j$. Then, $\boldsymbol{q}_h \in \boldsymbol{c}$, meaning that the query history is one of the contextual signals capable of being leveraged for query suggestion. In this task, the text of a web-page $\boldsymbol{w}$ currently being consumed by the user is also included in $\boldsymbol{c}$, as follows: $\boldsymbol{w} \in \boldsymbol{c}$. Formally, this task can be summarized as follows: $q_{j+1} = \text{QS}_\theta(\boldsymbol{x}, \boldsymbol{c})$[1].

In this work, we attempt to solve the problem of contextual query suggestion by leveraging a knowledge-augmented model to yield more personalized outputs. Formally, for $q_{j+1} = \text{QS}_\theta(\boldsymbol{x}, \boldsymbol{c})$, we set *QS* to be an LLM (e.g., GPT-4 [53]) and include aggregated entity-centric knowledge from users' historical interactions in the context $\boldsymbol{c}$, in order to generate *better* recommendations q_{j+1}, as measured by a set of personalization-focused metrics (see Section 4.3).

---

[1]Without the hard requirement of an input web document included in $\boldsymbol{c}$, this definition may be relaxed to capture prior work on *context-aware* query suggestion [5, 15, 34, 37].

(A) Previous Search Logs

Query: Bayesian algorithm (Jul 16, 2023)
The Bayesian algorithm is a technique in machine learning. It provides a probabilistic framework for inferring unknown parameters based on data.

Query: Supervised learning examples (Oct 10, 2023)
In supervised machine learning, examples include regression for predicting housing prices and classification for filtering spam emails.

(B) Current Search Context

Query: Email spam filtering (Oct 11, 2023)
Email spam filtering has evolved tremendously with the integration of machine learning techniques. Specifically, unsupervised machine learning allows systems to identify potential spam patterns without explicitly being trained on labeled spam or non-spam data. One of the prominent methods used in this domain is the Bayesian algorithm, which calculates the probability of an email being spam based on the frequency of certain words or phrases.

(C) Memory Stream

Instance	Time Stamp
Bayesian algorithm	Jul 16, 2023
…	…
Supervised learning examples	Oct 10, 2023

Entity Extraction

(D) Entity-based Knowledge Store

Entity	Count	Last Time
Machine Learning	11	Oct 10, 2023
Classification	6	Oct 10, 2023
Bayesian algorithm	3	Jul 16, 2023
Unsupervised ML	0	N/A

(E) Query Suggestion

Naïve Query Suggestion:
Whitelist vs. blacklist in spam filtering.

Familiar Entity: Machine Learning
Query Suggestion w/ Familiar Entity:
ML algorithms for spam filtering

Unfamiliar Entity: Unsupervised ML
Query Suggestion w/ Unfamiliar Entity:
Unsupervised spam filtering algorithms

Lapsed Entity: Bayesian algorithm
Query Suggestion w/ Lapsed Entity:
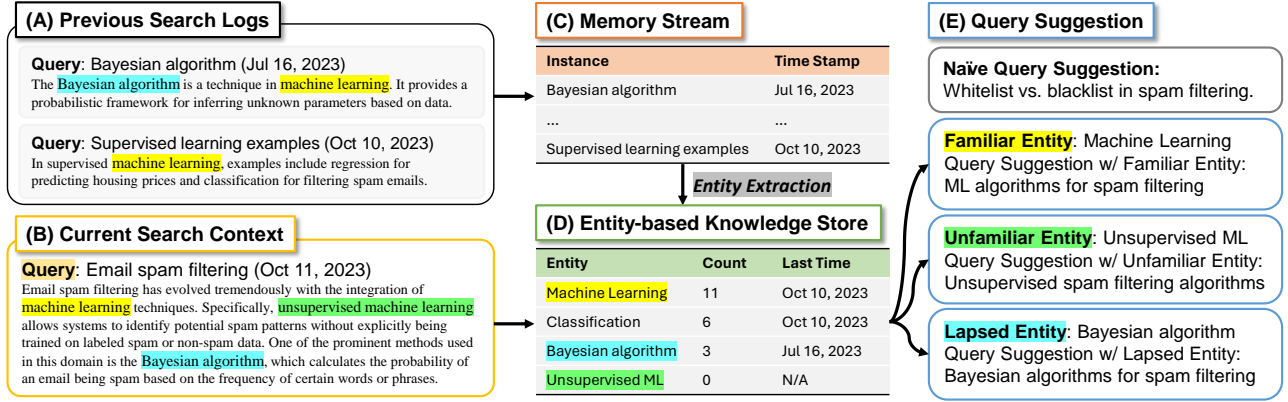Bayesian algorithms for spam filtering

Figure 2: Overview of K-LaMP. The inputs to the system are (A) previous search logs and (B) the current search context of a user, which consist of a query and an associated web-page. (C) Each search record is stored in a memory stream along with a time stamp. (D) The entity-based knowledge store is constructed by aggregating entities extracted from the memory stream. (E) K-LaMP augmented with entities retrieved from the entity-based personal knowledge store, generates query suggestions that are compatible with the user's knowledge and interests.

3.2 Knowledge Store for Output Personalization

We now discuss our knowledge-augmented LLM framework for personalization and describe two instantiations of this framework.

Recall that the supplemental context of an LLM c is provided by auxiliary sources or systems that help enrich the input prompt to the model. According to our goal of personalizing the LLM outputs, these auxiliary sources should ideally consist of data that captures the personal preferences, interests, and knowledge of users. Thus, if $\mathcal{K}$ is a knowledge store that encapsulates these user-specific data, and $k \in \mathcal{K}$ is a contextually relevant subset linearized as text, then, for contextual query suggestion, the context c can be defined as follows: $c = [q_h \cdot w \cdot k]$, where $[\cdot]$ is the concatenation operation.

Given this general formulation, the important questions to answer are: (1) How is the personal knowledge store $\mathcal{K}$ constructed? and (2) How are contextually relevant items k retrieved from the knowledge-store $\mathcal{K}$. We answer both of these questions below, focusing on construction in Section 3.2.1 and retrieval in Section 3.2.2.

3.2.1 Constructing Personal Knowledge Stores. While a variety of sources may be used to construct a compendium of what a user cares about and knows, in this paper, we leverage their interaction histories with a web search engine. We argue that this is a natural choice given that users express goals, desires, interests, and depth of knowledge both explicitly and implicitly through issued queries, clicked web-pages, consumed content and behavioral patterns over time. It is also an especially relevant source of knowledge for contextual query suggestion, where generating helpful search suggestions from search histories is the eventual goal of the task. Given this source, we build two distinct instantiations of the knowledge store $\mathcal{K}$: a simple variant that linearly captures historical user queries and browsing patterns ($\mathcal{K}_s$), and an entity-centric variant that aggregates users' personal interests and knowledge ($\mathcal{K}_e$).

Personal Knowledge Store from Search History ($\mathcal{K}_s$). The intuition behind this instantiation of the knowledge store $\mathcal{K}_s$, is that users issue queries and click on web-pages that they are interested in or care about. And, when accumulated over time, these also start to construct a picture of what users know and how deeply do they know them. For example, if a user issues multiple queries over time that include "Machine Learning", "ML", "Optimization", "SGD",

"Deep Learning" and clicks a number of web-pages resulting from these queries, we can assume that this user is at least familiar with the general concept of "Machine Learning".

In order to operationalize this intuition, we construct a time-stamped memory stream consisting of the queries issued by users and the web-pages associated with the results they clicked on (see Figure 2). Note that this is an extremely light-weight instantiation of the knowledge store, being only a partial view of user actions and interactions already logged by modern web-scale search engines. As a result, there are no privacy or scalability concerns beyond those already inherent in the search engine's logging system.

Personal Entity-centric Knowledge Store ($\mathcal{K}_e$). While building a memory stream over users' search histories is simple, there are a few limitations that stem from its design. Firstly, because search queries and web-page visitations are stored and retrieved linearly, it is difficult to perform aggregations on the fly. Yet, such aggregation can be greatly beneficial for personalization. For example, knowing that a user clicked on web-pages associated with "Machine Learning" multiple times, while only clicking on a single web-page stemming from the query "Computational Biology" would tend to indicate a greater affinity for and knowledge of the machine learning subject. Other issues include the fact that individual web-pages visited by a user may contain mixtures of several different topics and domains, distracting LLMs in generating outputs consistent with the context, and the fact that including large amounts of text from lengthy web-pages renders LLM usage slow and expensive.

In order to address these concerns, we construct an entity-centric instance of the knowledge store $\mathcal{K}_e$ (see Figure 2). Entities are useful atoms for capturing the interests and knowledge of users because they consist of the nouns (proper or otherwise) that describe the people, places, organizations, topics and domains that the users care and know about. Additionally, because they tend to be relatively short and easy to aggregate, and because entity recognition and linking [18, 41] are well-studied problems, the process of operationalizing the creation of this store is greatly simplified.

We employ an off-the-shelf state-of-the-art entity linking system [19] to tag and canonicalize the entities that appear in the search queries and associated web-pages visited by users. While

individual occurrences of entities in the knowledge store are time-stamped, additional aggregation is done by counting the number of occurrences of entities in full user interaction histories.

While this entity-centric knowledge store instantiation $\mathcal{K}_e$ may not be as minimalist as $\mathcal{K}_s$, it is still relatively light-weight when compared with systems that personalize through the construction of deep profiles. The solution is scalable since the only external dependency is an existing entity linker, which can typically process thousands of tokens per second. Additionally, privacy concerns are also small because entity linking projects and aggregates mentions in text onto sub-graphs of public knowledge bases (e.g. Wikipedia). Aggregation of entity occurrences lends itself naturally to common privacy mitigation practices such as k-anonymization [61]. Additionally, records are amenable to easy removal upon request by simply eliminating associated entities from the store.

3.2.2 Contextual Retrieval from Personal Knowledge Stores. We now turn to the question of retrieving contextually relevant items k from a knowledge-store $\mathcal{K}$, conditioned on an input query q_j and a document w that the user is currently interacting with. A carefully considered retrieval step is essential in augmenting the capability of LLMs to produce personalized outputs, since it grounds generation in the historical interests and knowledge of users. In the following, we show how retrieval is performed for both instantiations of the knowledge store $\mathcal{K}_s$ and $\mathcal{K}_e$ described in Section 3.2.1.

In the case of $\mathcal{K}_s$ over users' search and browsing history, retrieval is done by finding and returning the most similar queries and previously visited web-pages to the current input x. In practise, the queries are then elided from this result since they yield little benefit over the much longer text present in web-pages. To operationalize the retrieval step, we first represent all records in the knowledge-store $\mathcal{K}_s$ using embeddings, then compute embedding-level similarities with the representation of current query q_j using Contriever [42]. The most similar records k are finally returned.

Meanwhile, for the entity-centric knowledge store $\mathcal{K}_e$, retrieval is conditioned on the entities present in the current query q_j and the web-page w, which are further matched against $\mathcal{K}_e$. Given that entities are atomic units with associated counts and time-stamps, the matching and retrieval process can be operationalized in flexible ways (See Figure 2). We particularly explore three strategies for matching entities: familiar (entities frequently encountered by the user), unfamiliar (entities the user has encountered infrequently or not at all), and lapsed (entities that the user used to encounter previously but hasn't done so more recently). Specifically, for familiar entities, we sort the entities appearing in the search context $[x \cdot w]$ by frequency of occurrence in the knowledge store $\mathcal{K}_e$, then sample 5 entities proportionately to their frequency. For unfamiliar entities, a similar process is used for sampling, except that entities are sorted inversely with respect to their occurrence in $\mathcal{K}_e$. Finally, for lapsed entities, we start by filtering entities in $[x \cdot w]$ by time-stamp to retain only those that occur in $\mathcal{K}_e$, but haven't been engaged with in the preceding two weeks. Then we sample from this filtered set of entities by frequency, much like we do with familiar entities.

4 EXPERIMENTAL SETUP
In this section, we outline the datasets and models used in our evaluation setup, as well as implementation details.

4.1 Data
We use real large-scale search logs from the Bing search engine [2]. Specifically, we sample three months of search logs, from May 01, 2023 to July 31, 2023. We then filter and sample this dataset to make it suitable for evaluating our task, which includes the following steps. First, because the task we are tackling is *contextual* query suggestion – i.e., recommendations are predicated on a current web-page the user is viewing – we filter out sessions that do not contain any clicked search results. We further filter the data to discard click events that lead to pages in domains other than Wikipedia or a curated set of 500 high-traffic news publishing sites. We do this because the entity linker [19] we use maps onto Wikipedia, and we want to maximize the chances of encountering linked entities. It is worth noting that K-LaMP itself is agnostic to the choice of the linker or its underlying knowledge graph, and our approach could readily be applied to a different domain, for example, using a linker over a product graph for shopping. Finally, we filter the remaining data to discard users who had fewer than 100 page visitations for three months, who we assume are infrequent search users. In addition, we perform and apply enterprise-level privacy checks and filters, such as using search queries requested from at least 50 individuals, to ensure that the data remains suitably anonymized.

The resulting data is still extremely large; therefore, we further randomly sample a subset of 1,000 users in order to get the benchmark set that forms the basis for all the evaluations we perform in this paper (Section 4.3). This final dataset, on average, contains 493 queries, 109 sessions, 177 clicked articles, and 3,053 encountered entities per user. For testing, we split the dataset and reserve the most recent 10 sessions of every user as prediction targets for contextual query suggestions and use all the earlier sessions to build search-and-browsing based ($\mathcal{K}_s$) and entity-centric ($\mathcal{K}_e$) personal knowledge stores for users, as described in Section 3.2.1.

4.2 Baselines and Our Model
We compare our approach to knowledge-augmented LLMs for output personalization against several relevant baselines that make query suggestions based on the search context of users. We note that, for the fairest comparison, all baselines and our model use LLMs (specifically GPT-4) to make query suggestions. Notably, we do not include prior query suggestion approaches based on older modeling techniques (e.g., RNNs or BART [51, 66]) in our main experiments due to their limited capacity for understanding longer context and complex data inputs, particularly without dedicated training data (the framework we propose needs none). Moreover, these query suggestion techniques are not designed to handle longer contexts such as the web-page a user is currently viewing, thereby making a direct comparison trivially relevant at best. Nevertheless, in the interest of completeness we have included some anecdotal comparisons in Appendix B.

The models evaluated in this work are listed as follows: (1) **Query Suggestion** – which uses a current query q_j and historical queries from q_h in the same session to suggest the next query q_{j+1}; (2) **Contextual Query Suggestion** – which is similar to Query Suggestion, but additionally conditions the recommendation of the next query q_{j+1} on a web-page w, clicked as a result of current query q_j; (3) **Contextual Query Suggestion w/ $\mathcal{K}_s$** – which includes

Table 1: Main results on our contextual query suggestion task. The best results are highlighted in bold.

Types	Models	Validness (↑)	Relatedness (↑)	Usefulness (↑)	Ranking (↓)
Baselines	Query Suggestion	1.769	0.962	0.948	2.736
	Contextual Query Suggestion	**1.966**	1.267	1.245	2.415
	Contextual Query Suggestion w/ $\mathcal{K}_s$	1.822	1.192	1.166	2.654
Ours	**K-LaMP (Ours)**	**1.966**	**1.482**	**1.455**	**2.160**

retrievals from the knowledge store $\mathcal{K}_s$ over users' historical search and browsing activities, as additional context to personalize the outputs of the LLM; (4) **K-LaMP** – which is our full model that augments LLMs with entity-centric knowledge from the knowledge store $\mathcal{K}_e$ in order to perform contextual query suggestion.

4.3 Evaluation Setup

To evaluate the effectiveness of different query suggestion models on generating personalized outputs, a suitable evaluation metric should ideally not only capture whether the suggested queries are contextually relevant, but also whether they align well with the user's interests and knowledge. Given that contextual query suggestion is a novel problem we propose in this paper, no existing evaluation metrics are available for the task. In particular, metrics for evaluating conventional query suggestion are not applicable here because they do not account for the full *context* present in our task – namely the input document being consumed by the user. Therefore, we turn to human evaluation in order to measure and compare the different models on our experimental benchmark.

It is worth noting that human evaluation of any form of personalization is difficult, since the person performing the evaluation is rarely the person from whom the data originated. Short of flighting and A/B testing our system in a real-world setting – an engineering endeavor well beyond the scope of the scientific exploration in this paper – any evaluation on our task and dataset must be bounded by this constraint. Nevertheless, we attempt to provide annotators with as much information as possible in order to understand both the user's current search context and their personal interests and knowledge (see Table 6 for an example). In particular, to summarize the current search context for annotators, we show them the current and previous search queries of a user in a given session, the web-page the user clicked on after issuing the current search query, and a list of 20 trending entities which capture statistical surges in search volume across users. Additionally, in order to present an encapsulation of the personal interests and knowledge of users, we show annotators a list of the 30 most frequent entities from the user's personal entity-centric knowledge store, as well as a GPT-4 generated summary from these entities that states what topics or domains the user may know or care about much.

Presented with this data and recommended queries from the different baselines and our model (where the system names are obscured to annotators), a human judge is asked to evaluate the following three metrics on a 3-point Likert scale[2]: (1) **Validity** – whether an output query can be input into a search engine and be expected to yield relevant results; (2) **Relatedness** – whether the output query closely relates to the user's personal interests

Table 2: Results of inter-annotator agreements on all query suggestion results evaluated by humans annotators.

Agreements	Metrics	Scores (↑)
Exact match	Validness	0.963
	Relatedness	0.850
	Usefulness	0.819
Cohen's kappa coefficient	Validness	0.606
	Relatedness	0.652
	Usefulness	0.622
Spearman's correlation coefficient	Ranking	0.654

and knowledge; and (3) **Usefulness** – whether the user is likely to click on the output query, given their historical interests and knowledge as well as their current search context. Finally, we also ask the annotators for a fourth measure: (4) **Ranking** – where the outputs of the different systems are ranked according to the order in which they are likely to be clicked, based on their affinity to the user's interests, knowledge, and search context. Collectively, these four evaluation metrics capture not only how good the different query suggestions are, – both individually and in relation to one another – but also how well they align with the *personal* aspects of our evaluation task; namely, what users care about and know.

To perform evaluations with human judges, we recruit 12 annotators in India through a third-party vendor company [3]. They were provided with a guideline document, which includes instructions for the task, metrics and some annotated examples, and they were paid $11.98 per hour for the time they spent working on the task. Over several rounds of judgement and refinement, we obtain manual evaluation results for 1, 309 sets of contextual query suggestion results from all four models listed in Section 4.2 (effectively a total of 5, 236 annotations for individual query suggestions).

Additionally, in order to validate the quality of annotations and to measure inter-annotator agreement, approximately 27% of the data is annotated by two human judges. Specifically for Validity, Relatedness, and Usefulness, we measure an exact match score, which checks often annotators provide the same score on the 3-point likert scale, and Cohen's kappa coefficient [17] which additionally discounts for chance agreement. For Ranking, we report Spearman's correlation coefficient [1], which measures the correlation between two sets of ranked systems, averaged across pairs of users and data instances. As shown in Table 2, we observe that inter-annotator agreement is moderate to high, indicating that judges are in fact able to make reasonably informed decisions about personalized contextual query suggestion from the data we provide them with.

Finally, regarding implementation details, we use the GPT-4 [53] release from June 13, 2023, as the basis for query suggestion across all baselines and model variants, for a fair comparison. We set the

---
[2]The 3-point Likert scale is composed of agree (2), neutral (1), and disagree (0).

Table 3: Results of different retrieval strategies on Retrieval Relevance to the user's current search context.

Retrieval	Types	Retrieval Relevance (↑)
History-based Retrieval ($\mathcal{K}_s$)	Past Documents	0.299
Entity-centric Retrieval ($\mathcal{K}_e$)	Familiar Entities	**0.936**
	Unfamiliar Entities	**0.810**
	Lapsed Entities	**0.849**

hyperparameters of GPT-4 as temperature = 0.7 and top$_p$ = 0.95. The entity linker used to construct instantiations of the knowledge store is NEMO [18, 19][3]. Prompts used to elicit responses from GPT-4 for query suggestion are in Appendix A.

5 EXPERIMENTAL RESULTS

We now present the set of experimental results from our evaluation, and report findings from various auxiliary studies and analyses.

Our main results are shown in Table 1. This confirms that our K-LaMP framework consistently and significantly outperforms all other baselines across Relatedness, Usefulness, and Ranking metrics. While it ties Contextual Query Suggestion on the Validity metric, this finding is overall a positive albeit somewhat expected one – since, intuitively, inclusion of personal context does not necessarily lead to queries that are more *valid* for search engine retrieval. Meanwhile, there are a few other interesting insights that can be gleaned from this table. Contextual Query Suggestion with $\mathcal{K}_s$ does not outperform Contextual Query Suggestion. We hypothesize that this is because the information retrieved from the memory store ($\mathcal{K}_s$) has poor relevance to the current search context, leading to spurious augmentations that distract rather than help the LLM.

To investigate this hypothesis further, we conduct a supplementary evaluation, which asks human annotators to rate the information retrieved from knowledge stores for a particular search context (see Section 3.2.2 for details). Specifically, we report Retrieval Relevance from both instantiations of our knowledge stores in Table 3. This metric is the average score from a Yes/No question: whether the retrieved context is relevant to the current search context (1) or not (0). As shown in Table 3, the quality of retrievals from the entity-centric knowledge store is superior to those from the linear search history-based store. This is because we have far greater control with entities being the atomic units of the knowledge representation space, and are able to *exactly* match entities in the context against entities in the store, rather than rely on a fuzzy similarity-driven retrieval process with an embedding-based dense retriever [42].

5.1 Additional Studies and Analyses

Ablation over Entity Matching Strategies. Recall that K-LaMP relies on a combination of matching and retrieval of several different types of entities from its entity-centric knowledge store, namely: familiar, unfamiliar and lapsed entities (see Section 3.2.2). In order to individually measure the contribution of each strategy, we generate knowledge-augmented query suggestions on 313 search contexts using only one type of entity and ask human annotators to evaluate the results on Validity, Relatedness, and Usefulness. The comparative results are presented in Figure 3. Firstly, they reaffirm the fact
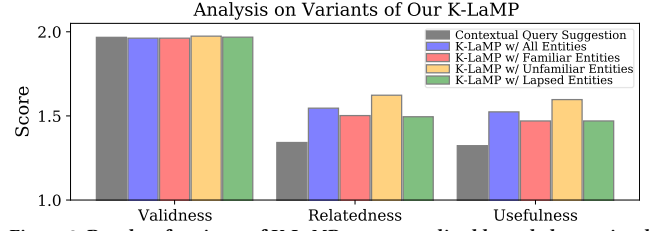
Figure 3: Results of variants of K-LaMP on personalized knowledge retrieval strategy, and results without retrieval.
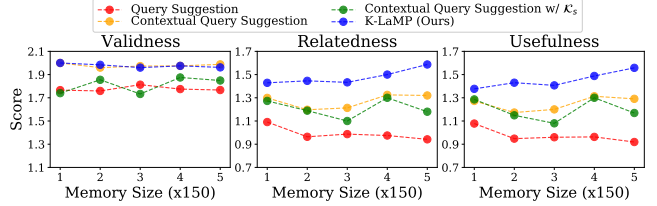
Figure 4: Results of baselines and K-LaMP with different numbers of previous queries (i.e., different memory sizes).

that Validity is practically invariant to the choice of knowledge ingestion, since personal information does not affect whether a query is *valid* or not. The Relatedness and Usefulness metrics, however, are clearly impacted by the choice of entity matching strategy in consistent ways. In particular, using only "unfamiliar" entities yields the highest scores across both metrics, even outperforming the full K-LaMP model. This seems to suggest that queries stemming from new (to the user) entities, which implicitly encourage exploration, are preferred over queries that revisit familiar ground.

While these results are true on this data, we note that real-world users may approach web search with different goals in mind: such as research, exploration, or revision, and it is difficult for judges to assess these goals from limited albeit rich data. As a result, we argue that having a comprehensive approach to capturing different views of the users' knowledge so that it may be deepened, expanded, or revived as the use-case may demand, is a robust strategy.

Analysis over Interaction History Length. A fundamental assumption in our setup for LLM output personalization is that we can learn about users as they interact with search engines. A natural follow-up question to this assumption is to ask how the performance of systems that rely on personal knowledge change as a function of the length of the interaction history. To answer this question, we conduct an analysis with varying history lengths and report results in Figure 4. From this, we observe again that Validity is not affected by the length of the interaction history, while Relatedness and Usefulness are. In particular, K-LaMP is the only model demonstrating consistent improvement with longer interaction histories, showcasing its ability to grow richer representations of personal interests and knowledge over time. A reasonable explanation for this increment is the aggregation that happens in K-LaMP's entity-centric knowledge store, which contrasts with the linearly stored histories of the other approaches.

Analysis using different LLMs. Finally, we conduct an auxiliary analysis to see how the quality of query recommendations from different systems change if an LLM other than GPT-4 is used. Specifically, we use the June 13, 2023 version of GPT-3.5-Turbo as the LLM

Table 4: Results with different LLMs: GPT-3.5 and GPT-4.

Methods	LLMs	Validness	Relatedness	Usefulness
Query Suggestion	GPT-3.5	1.767	1.077	1.069
	GPT-4	1.747	1.080	1.060
Contextual Query Suggestion	GPT-3.5	1.967	1.177	1.202
	GPT-4	1.987	1.367	1.313
K-LaMP (Ours)	GPT-3.5	2.000	1.279	1.303
	GPT-4	1.983	**1.653**	**1.600**

Table 5: Results with automatic evaluation metrics.

Types	Validness	Relatedness	Usefulness
Correlation w/ Human Evaluation	**0.445**	**0.397**	-0.016
Query Suggestion	1.784	1.189	0.882
Contextual Query Suggestion	1.891	1.340	0.831
Contextual Query Suggestion w/ $\mathcal{K}_s$	1.828	1.271	0.847
K-LaMP (Ours)	1.910	1.472	0.845

on 128 sets of query suggestions from two baselines (Query Suggestion and Contextual Query Suggestion) and our K-LaMP framework, and compare the results with GPT-4; these are shown in Table 4. Firstly, Query Suggestion is agnostic to the choice of LLMs, while Contextual Query Suggestion and K-LaMP are not. This is likely due to the fact that the latter two approaches must incorporate information from full web-pages as context and therefore benefit from the representational, reasoning and generative capabilities of the larger model. More relevant to the contributions in this paper, we find that, even with GPT-3.5-Turbo, K-LaMP shows comparable performance on the Usefulness metric with the second best model – Contextual Query Suggestion – despite the latter using GPT-4. This demonstrates the significant edge that an entity-centric representation of a user's personal interests and knowledge provides, for knowledge-augmented personalization of LLMs outputs, with high efficiency of deploying K-LaMP in real-world applications.

5.2 Automatic Evaluation Setup

While human evaluation is useful for measuring systems and gaining insights, especially on a new task like the one we introduce, the process is slow and expensive, and therefore not scalable to bigger datasets, or future extensions. To address these issues, we explore an initial set of automatic evaluation metrics mirroring the ones described in Section 4.3 that may be used in the absence of human judgement. Recall that even human evaluation for tasks that deal with personalization is non-trivial; therefore, automatically evaluating the outputs of a contextual query system while conditioning on complex personal preference and knowledge data is very difficult.

Nevertheless, we propose and experiment with the following automatic formulations: (1) **Validity** – we compute the similarity between the query suggestion output of a system and the top search result (title and snippet) returned from issuing that query to the web search engine (to see if the query yields reasonable search results); (2) **Relatedness** – we measure the similarity between the query suggestion and the set of contextual personal entities retrieved from the user's entity-centric knowledge store (to ensure that the query is grounded in the personal context of the user); (3) **Usefulness** – we calculate the similarity between the query suggestion and the real subsequent queries that the user ended up issuing (in order to compare recommendations against the user's true actions). In each of these three metrics[4], similarity is computed by calculating the dot product of representations obtained from Contriever [42].

We validate these automatic evaluation metrics by ranking the systems on the test set, then computing Spearman's correlation against the ranking obtained by human judgement scores. As shown

in Table 5, we find a moderate correlation on Validity and Relatedness, indicating that our proposed automatic metrics for these measures may be used as proxies in the absence of human labeling. However, there is no correlation between automatic and human Usefulness metrics. This is expected since (contextual) query recommendation is not expected to align perfectly with actual user behavior, which is the basis of our formulation for automatically computing Usefulness; users *should* be surprised and delighted by suggestions they would not have otherwise thought about.

There are several ways to improve the process of automatically evaluating contextual query suggestion. For example, we could use another LLM to perform a rubric-based evaluation of Validity, Relatedness and Usefulness, relying on its capacity to account for complex personal and preferential data. Or we could train parametrized versions of the automatic metrics we have proposed on manually labeled data with the goal of increasing correlation with human judgement. We leave these and other explorations to future work.

6 CONCLUSION

In this work, we proposed a knowledge-augmentation framework for LLM output personalization called K-LaMP, that leverages historical user interactions with a search engine. The core of the personal knowledge we used for LLM augmentation relies on a novel lightweight entity-centric personal knowledge store, constructed from the queries that users issue and the web-pages that they viewed as they search and browse the web. To stress-test our personalization framework, we focused on the novel task of contextual search query suggestion, which crucially requires modeling both the contextual interests and the knowledge of users. Through human evaluation on an extensive test set, we showed that our entity-centric knowledge-augmented LLM produces personalized query recommendations that are better related to users' intent, more useful, and consistently ranked above those produced by several other LLM-powered query suggestion models. Our findings show that entities are effective atomic units for the representation of personal knowledge, offering a robust middle-ground of performance, flexibility, privacy and scalability, when compared with other personalization approaches that rely either on deep profile building or simple linearization of a user's historical interactions. We believe that K-LaMP has the potential to impact both future research and product innovation. The use of personalized knowledge-augmentation for other search tasks such as snippet generation or question answering, the incorporation of other sources of data such as shopping or media-consumption histories, and the application to domains outside of search such as personal AI assistants, are all exciting avenues of future work. At the same time, enhanced evaluation remains an important future goal, with improved automatic metrics and real-world deployment as potential directions for exploration.

---

[4]We don't specify an automatic measure of Ranking, since this can be done trivially by scoring then sorting systems by one or more of the other automatic metrics.

REFERENCES

[1] 2008. *Spearman Rank Correlation Coefficient*. 502–505.
[2] 2023. *Bing Search*. https://www.bing.com/
[3] 2023. *KarmaHub*. https://www.mykarmahub.com/
[4] Wasi Uddin Ahmad, Kai-Wei Chang, and Hongning Wang. 2018. Multi-Task Learning for Document Ranking and Query Suggestion. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.
[5] Wasi Uddin Ahmad, Kai-Wei Chang, and Hongning Wang. 2019. Context Attentive Document Ranking and Query Suggestion. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*. ACM, 385–394.
[6] Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernández Ábrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan A. Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vladimir Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, and et al. 2023. PaLM 2 Technical Report. *arXiv preprint arXiv:2305.10403* (2023).
[7] Jinheon Baek, Alham Fikri Aji, and Amir Saffari. 2023. Knowledge-Augmented Language Model Prompting for Zero-Shot Knowledge Graph Question Answering. In *Proceedings of the 1st Workshop on Natural Language Reasoning and Structured Explanations (NLRSE)*. Association for Computational Linguistics.
[8] Praveen Kumar Bodigutla. 2021. High Quality Related Search Query Suggestions using Deep Reinforcement Learning. *arXiv preprint arXiv:2108.04452* (2021).
[9] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher 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 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.
[10] Jin Chen, Zheng Liu, Xu Huang, Chenwang Wu, Qi Liu, Gangwei Jiang, Yuanhao Pu, Yuxuan Lei, Xiaolong Chen, Xingmei Wang, Defu Lian, and Enhong Chen. 2023. When Large Language Models Meet Personalization: Perspectives of Challenges and Opportunities. *arXiv preprint arXiv:2307.16376* (2023).
[11] Wanyu Chen, Fei Cai, Honghui Chen, and Maarten de Rijke. 2017. Personalized Query Suggestion Diversification. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*. ACM, 817–820.
[12] Wanyu Chen, Fei Cai, Honghui Chen, and Maarten de Rijke. 2018. Attention-based Hierarchical Neural Query Suggestion. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*. ACM, 1093–1096.
[13] Wei-Fan Chen, Shahbaz Syed, Benno Stein, Matthias Hagen, and Martin Potthast. 2020. Abstractive Snippet Generation. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*. ACM / IW3C2, 1309–1319.
[14] Zheng Chen. 2023. PALR: Personalization Aware LLMs for Recommendation. *arXiv preprint arXiv:2305.07622* (2023).
[15] Qiannan Cheng, Zhaochun Ren, Yujie Lin, Pengjie Ren, Zhumin Chen, Xiangyuan Liu, and Maarten de Rijke. 2021. Long Short-Term Session Search: Joint Personalized Reranking and Next Query Prediction. In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*. ACM / IW3C2, 239–248.
[16] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. PaLM: Scaling Language Modeling with Pathways. *arXiv preprint arXiv:2204.02311* (2022).
[17] Jacob Cohen. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement* 20, 1 (1960), 37–46.
[18] Silviu Cucerzan. 2007. Large-Scale Named Entity Disambiguation Based on Wikipedia Data. In *EMNLP-CoNLL 2007, Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, June 28-30, 2007, Prague, Czech Republic*. ACL, 708–716.
[19] Silviu Cucerzan. 2012. The MSR System for Entity Linking. In *Text Analysis Conference 2012*. NIST.
[20] Sunhao Dai, Ninglu Shao, Haiyuan Zhao, Weijie Yu, Zihua Si, Chen Xu, Zhongxiang Sun, Xiao Zhang, and Jun Xu. 2023. Uncovering ChatGPT's Capabilities in Recommender Systems. In *Proceedings of the 17th ACM Conference on Recommender Systems, RecSys 2023, Singapore, Singapore, September 18-22, 2023*. ACM, 1126–1132.
[21] Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. Autoregressive Entity Retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.
[22] Mostafa Dehghani, Sascha Rothe, Enrique Alfonseca, and Pascal Fleury. 2017. Learning to Attend, Copy, and Generate for Session-Based Query Suggestion. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, November 06 - 10, 2017*. ACM, 1747–1756.
[23] Mostafa Dehghani, Sascha Rothe, Enrique Alfonseca, and Pascal Fleury. 2017. Learning to Attend, Copy, and Generate for Session-Based Query Suggestion. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, November 06 - 10, 2017*. ACM, 1747–1756.
[24] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*. Association for Computational Linguistics.
[25] Heng Ding, Shuo Zhang, Darío Garigliotti, and Krisztian Balog. 2018. Generating High-Quality Query Suggestion Candidates for Task-Based Search. In *Advances in Information Retrieval - 40th European Conference on IR Research, ECIR 2018, Grenoble, France, March 26-29, 2018, Proceedings (Lecture Notes in Computer Science, Vol. 10772)*. Springer, 625–631.
[26] Wenqi Fan, Zihuai Zhao, Jiatong Li, Yunqing Liu, Xiaowei Mei, Yiqi Wang, Jiliang Tang, and Qing Li. 2023. Recommender Systems in the Era of Large Language Models (LLMs). *arXiv preprint arXiv:2307.02046* (2023).
[27] Luke Friedman, Sameer Ahuja, David Allen, Zhenning Tan, Hakim Sidahmed, Changbo Long, Jun Xie, Gabriel Schubiner, Ajay Patel, Harsh Lara, Brian Chu, Zexi Chen, and Manoj Tiwari. 2023. Leveraging Large Language Models in Conversational Recommender Systems. *arXiv preprint arXiv:2305.07961* (2023).
[28] Darío Garigliotti and Krisztian Balog. 2017. Generating Query Suggestions to Support Task-Based Search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*. ACM, 1153–1156.
[29] 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 *RecSys '22: Sixteenth ACM Conference on Recommender Systems, Seattle, WA, USA, September 18 - 23, 2022*. ACM, 299–315.
[30] Zhankui He, Zhouhang Xie, Rahul Jha, Harald Steck, Dawen Liang, Yesu Feng, Bodhisattwa Prasad Majumder, Nathan Kallus, and Julian J. McAuley. 2023. Large Language Models as Zero-Shot Conversational Recommenders. *arXiv preprint arXiv:2308.10053* (2023).
[31] Chao-Chun Hsu, Eric Lind, Luca Soldaini, and Alessandro Moschitti. 2021. Answer Generation for Retrieval-based Question Answering Systems. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021 (Findings of ACL, Vol. ACL/IJCNLP 2021)*. Association for Computational Linguistics, 4276–4282.
[32] Jianchao Ji, Zelong Li, Shuyuan Xu, Wenyue Hua, Yingqiang Ge, Juntao Tan, and Yongfeng Zhang. 2023. GenRec: Large Language Model for Generative Recommendation. *arXiv preprint arXiv:2307.00457* (2023).
[33] Di Jiang, Kenneth Wai-Ting Leung, Jan Vosecky, and Wilfred Ng. 2014. Personalized Query Suggestion With Diversity Awareness. In *IEEE 30th International Conference on Data Engineering, Chicago, ICDE 2014, IL, USA, March 31 - April 4, 2014*. IEEE Computer Society, 400–411.
[34] Jyun-Yu Jiang and Wei Wang. 2018. RIN: Reformulation Inference Network for Context-Aware Query Suggestion. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*. ACM, 197–206.
[35] Minki Kang, Jinheon Baek, and Sung Ju Hwang. 2022. KALA: Knowledge-Augmented Language Model Adaptation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*. 5144–5167.
[36] Wang-Cheng Kang, Jianmo Ni, Nikhil Mehta, Maheswaran Sathiamoorthy, Lichan Hong, Ed H. Chi, and Derek Zhiyuan Cheng. 2023. Do LLMs Understand User Preferences? Evaluating LLMs On User Rating Prediction. *arXiv preprint arXiv:2305.06474* (2023).
[37] Eugene Kharitonov, Craig Macdonald, Pavel Serdyukov, and Iadh Ounis. 2013. Intent models for contextualising and diversifying query suggestions. In *22nd ACM*

International Conference on Information and Knowledge Management, CIKM'13, San Francisco, CA, USA, October 27 - November 1, 2013. ACM, 2303–2308.

[38] Mojtaba Komeili, Kurt Shuster, and Jason Weston. 2022. Internet-Augmented Dialogue Generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

[39] Andrew K. Lampinen, Ishita Dasgupta, Stephanie C. Y. Chan, Kory W. Mathewson, Mh Tessler, Antonia Creswell, James L. McClelland, Jane Wang, and Felix Hill. 2022. Can language models learn from explanations in context?. In *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*. Association for Computational Linguistics, 537–563.

[40] Angeliki Lazaridou, Elena Gribovskaya, Wojciech Stokowiec, and Nikolai Grigorev. 2022. Internet-augmented language models through few-shot prompting for open-domain question answering. *arXiv preprint arXiv:2203.05115* (2022).

[41] Martin Josifoski Sebastian Riedel Luke Zettlemoyer Ledell Wu, Fabio Petroni. 2020. Zero-shot Entity Linking with Dense Entity Retrieval. In *EMNLP*.

[42] Yibin Lei, Liang Ding, Yu Cao, Changtong Zan, Andrew Yates, and Dacheng Tao. 2023. Unsupervised Dense Retrieval with Relevance-Aware Contrastive Pre-Training. In *Findings of the Association for Computational Linguistics: ACL 2023*.

[43] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*. Association for Computational Linguistics, 7871–7880.

[44] Cheng Li, Mingyang Zhang, Qiaozhu Mei, Yaqing Wang, Spurthi Amba Hombaiah, Yi Liang, and Michael Bendersky. 2023. Teach LLMs to Personalize - An Approach inspired by Writing Education. *arXiv preprint arXiv:2308.07968* (2023).

[45] Lei Li, Yongfeng Zhang, Dugang Liu, and Li Chen. 2023. Large Language Models for Generative Recommendation: A Survey and Visionary Discussions. *arXiv preprint arXiv:2309.01157* (2023).

[46] Junling Liu, Chao Liu, Renjie Lv, Kang Zhou, and Yan Zhang. 2023. Is ChatGPT a Good Recommender? A Preliminary Study. *arXiv preprint arXiv:2304.10149* (2023).

[47] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692* (2019).

[48] Hanjia Lyu, Song Jiang, Hanqing Zeng, Yinglong Xia, and Jiebo Luo. 2023. LLM-Rec: Personalized Recommendation via Prompting Large Language Models. *arXiv preprint arXiv:2307.15780* (2023).

[49] Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Hannaneh Hajishirzi, and Daniel Khashabi. 2022. When Not to Trust Language Models: Investigating Effectiveness and Limitations of Parametric and Non-Parametric Memories. *arXiv preprint arXiv:2212.10511* (2022).

[50] Qiaozhu Mei, Dengyong Zhou, and Kenneth Ward Church. 2008. Query suggestion using hitting time. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM 2008, Napa Valley, California, USA, October 26-30, 2008*. ACM, 469–478.

[51] Agnès Mustar, Sylvain Lamprier, and Benjamin Piwowarski. 2020. Using BERT and BART for Query Suggestion. In *Proceedings of the First Joint Conference of the Information Retrieval Communities in Europe (CIRCLE 2020), Samatan, Gers, France, July 6-9, 2020 (CEUR Workshop Proceedings, Vol. 2621)*. CEUR-WS.org.

[52] Agnès Mustar, Sylvain Lamprier, and Benjamin Piwowarski. 2022. On the Study of Transformers for Query Suggestion. *ACM Trans. Inf. Syst.* 40, 1 (2022), 18:1–18:27.

[53] OpenAI. 2023. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774* (2023).

[54] Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative Agents: Interactive Simulacra of Human Behavior. *ariv preprint arXiv:2304.03442* (2023).

[55] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick S. H. Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander H. Miller. 2019. Language Models as Knowledge Bases?. 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 2019, Hong Kong, China, November 3-7, 2019*. Association for Computational Linguistics, 2463–2473.

[56] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training. (2018).

[57] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.* (2020).

[58] Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-Context Retrieval-Augmented Language Models. *arXiv preprint arXiv:2302.00083* (2023).

[59] Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How Much Knowledge Can You Pack Into the Parameters of a Language Model?. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*. Association for Computational Linguistics, 5418–5426.

[60] Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2023. LaMP: When Large Language Models Meet Personalization. *arXiv preprint arXiv:2304.11406* (2023).

[61] Pierangela Samarati and Latanya Sweeney. 1998. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. https://api.semanticscholar.org/CorpusID:2181340

[62] Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Févry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multitask Prompted Training Enables Zero-Shot Task Generalization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*.

[63] Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023. REPLUG: Retrieval-Augmented Black-Box Language Models. *arXiv preprint arXiv:2301.12652* (2023).

[64] Kurt Shuster, Jing Xu, Mojtaba Komeili, Da Ju, Eric Michael Smith, Stephen Roller, Megan Ung, Moya Chen, Kushal Arora, Joshua Lane, Morteza Behrooz, William Ngan, Spencer Poff, Naman Goyal, Arthur Szlam, Y-Lan Boureau, Melanie Kambadur, and Jason Weston. 2022. BlenderBot 3: a deployed conversational agent that continually learns to responsibly engage. *arXiv preprint arXiv:2208.03188* (2022).

[65] Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. 2015. A Hierarchical Recurrent Encoder-Decoder for Generative Context-Aware Query Suggestion. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19 - 23, 2015*. ACM, 553–562.

[66] Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. 2015. A Hierarchical Recurrent Encoder-Decoder for Generative Context-Aware Query Suggestion. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19 - 23, 2015*. ACM, 553–562.

[67] Mujeen Sung, Jinhyuk Lee, Sean S. Yi, Minji Jeon, Sungdong Kim, and Jaewoo Kang. 2021. Can Language Models be Biomedical Knowledge Bases?. In *EMNLP*. Association for Computational Linguistics.

[68] Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Kathleen S. Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, Viktoriya Kuzmina, Joe Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Agüera y Arcas, Claire Cui, Marian Croak, Ed H. Chi, and Quoc Le. 2022. LaMDA: Language Models for Dialog Applications. *arXiv preprint arXiv:2201.08239* (2022).

[69] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. *arXiv preprint arXiv:2302.13971* (2023).

[70] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv preprint arXiv:2307.09288* (2023).

[71] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *NeurIPS*.

[72] Thanh Vu, Alistair Willis, Udo Kruschwitz, and Dawei Song. 2017. Personalised Query Suggestion for Intranet Search with Temporal User Profiling. In *Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval, CHIIR 2017, Oslo, Norway, March 7-11, 2017*. ACM, 265–268.

[73] Yancheng Wang, Ziyan Jiang, Zheng Chen, Fan Yang, Yingxue Zhou, Eunah Cho, Xing Fan, Xiaojiang Huang, Yanbin Lu, and Yingzhen Yang. 2023. RecMind: Large Language Model Powered Agent For Recommendation. *arXiv preprint arXiv:2308.14296* (2023).

[74] 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. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*.

[75] Jerry W. Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, and Tengyu Ma. 2023. Larger language models do in-context learning differently. *arXiv preprint arXiv:2303.03846* (2023).

[76] Jimmy Wu, Rika Antonova, Adam Kan, Marion Lepert, Andy Zeng, Shuran Song, Jeannette Bohg, Szymon Rusinkiewicz, and Thomas A. Funkhouser. 2023. TidyBot: Personalized Robot Assistance with Large Language Models. *arXiv preprint arXiv:2305.05658* (2023).

[77] Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, Hui Xiong, and Enhong Chen. 2023. A Survey on Large Language Models for Recommendation. *arXiv preprint arXiv:2305.19860* (2023).

[78] Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2023. Generate rather than Retrieve: Large Language Models are Strong Context Generators. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*.

[79] Kai Zhang, Fubang Zhao, Yangyang Kang, and Xiaozhong Liu. 2023. Memory-Augmented LLM Personalization with Short- and Long-Term Memory Coordination. *arXiv preprint arXiv:2309.11696* (2023).

[80] 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 (Volume 1: Long Papers)*.

[81] Jianling Zhong, Weiwei Guo, Huiji Gao, and Bo Long. 2020. Personalized Query Suggestions. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*. ACM, 1645–1648.

[82] Zile Zhou, Xiao Zhou, Mingzhe Li, Yang Song, Tao Zhang, and Rui Yan. 2022. Personalized Query Suggestion with Searching Dynamic Flow for Online Recruitment. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*. ACM, 2773–2783.

A PROMPTS

In this section, we provide the prompt for the K-LaMP framework in Table 8, which we use for eliciting responses from LLMs for personalized contextual query suggestion. In addition, we provide the prompts for other approaches, namely query suggestion, contextual query suggestion, and contextual query suggestion w/ $\mathcal{K}_s$ in Table 9, Table 10, and Table 11, respectively.

B QUERY SUGGESTION EXAMPLES

We provide examples based on a real-world comparison of different query suggestion approaches, including commercial query suggestion models deployed in public web-search engines such as Microsoft Bing and Google. For this comparison, we use the manufactured example in Table 6, which encapsulates a user query about "Tim Cook" that leads to a web page about "Tim Cook's leadership", as well as a number of personal entities that a user has historically been interested in (such as Apple products). Example query suggestions from different approaches are shown in Table 7. These demonstrate how K-LaMP better contextualizes suggestions on the current page a user is consuming. For example, the query "Difference between Tim Cook's and Steve Jobs' approach to Apple's product development" is a contrastive information-seeking request that leverages the user context, versus "Tim Cook leadership style" (from the non-contextual query suggestion method) which isn't particularly helpful given that this is already the topic of the current page. K-LaMP's suggestions also better capture the implicit knowledge and interests of the user, based on historical browsing behavior. For example, the suggestion "Tim Cook's impact on Apple's product line" aligns more closely with the user's ostensibly technophile interests when compared with the celebrity gossip-focused suggestion "Tim Cook boyfriend" from the commercial query suggestion methods in web-search engines.

Table 6: A manufactured example showing the type of data that was provided to human judges.

Types	Texts
Query	Tim Cook
Session	Apple | Tim Cook
Article	Tim Cook Leadership: A new profile examines how Apple CEO Tim Cook, with "cautious, collaborative and tactical" leadership, honed the Cupertino tech giant into the world's largest company. (…)
Trending entities	'GPT-4', 'OpenAI', 'Google Bard', 'Microsoft Copilot', 'Elon Musk', …
Personal summary	The user is interested in Apple products and technology ('Macbook', 'macOS', and 'Apple TV'). They have a keen interest in ML, with topics like 'Supervised Learning' and 'Optimization'. Additionally, they enjoy animation, showing interest in 'Studio Ghibli', 'Walt Disney', and 'Pixar'. Their preferences also extend to home entertainment ('DVD' and 'HDTV'). Lastly, they follow baseball ('MLB' and 'New York Yankees').
Personal entities	'Macbook', 'macOS', 'Machine Learning', 'Optimization', 'Supervised Learning', 'Apple TV', 'Animation', 'Studio Ghibli', 'DVD', 'Walt Disney', 'Pixar Animation Studios', 'Apple Inc.', 'Baseball', 'HDTV', 'Major League Baseball', 'New York Yankees', …

Table 7: Query suggestion results of different approaches including non-contextual and commercial methods, based on the data example in Table 6.

Models	Query Suggestion Results
Microsoft Bing Search	"Tim Cook Boyfriend", "Tim Cook Latest News"
Google Search	"Tim Cook Quotes", "Tim Cook Boyfriend"
Query Suggestion w/o Context	"Tim Cook leadership style", "Tim Cook net worth"
K-LaMP (Ours)	"Tim Cook's impact on Apple's product line", "Difference between Tim Cook's and Steve Jobs' approach to Apple's product development"

Table 8: The prompt used in the full instantiation of K-LaMP.

Types	Texts
System Message	You are an AI assistant whose primary goal is to suggest a next search query, in order to help a user search and find information better on the search engine. Two different queries and entities are separated by the token '|'. For example, 'Microsoft' and 'Google' would appear as 'Microsoft'	'Google'.
User Message	You are going to suggest a search query that the user would search next based on the current query, the current session, the current article, and the personal entities. The explanations of the query, session, article, and personal entities are as follows: - The query is a specific set of phrases that the user enters into the search engine to find the information or resources related to a particular topic, question, or interest. - The session refers to a sequence of queries requested by the user on the search engine, within a certain period of time or with regard to the completion of a task. - The article refers to a specific webpage that the user clicks and reads from several search results displayed by the search engine in response to the requested query. - The personal entity refers to a topic, keyword, person, event, or any subject that is specifically relevant or appealing to the individual user based on their personal interests. Read the following query, session, article, and personal entities of the user as the context information, which might be helpful and relevant to suggest the next query. Query: {Query} Session: {Session} Article Title: {Article['Title']} Article Text: {Article['Text']} Personal Entities: {Entities} Based on the above query, session, article, and personal entities, please generate one next query suggestion with the rationale, in the format of Query Suggestion: Rationale:

Table 9: The prompt used in the instantiation of the query suggestion baseline.

Types	Texts
System Message	You are an AI assistant whose primary goal is to suggest a next search query, in order to help a user search and find information better on the search engine. Two different queries and entities are separated by the token '|'. For example, 'Microsoft' and 'Google' would appear as 'Microsoft'	'Google'.
User Message	You are going to suggest a search query that the user would search next based on the current query, and the current session. The explanations of the query, and session are as follows: - The query is a specific set of phrases that the user enters into the search engine to find the information or resources related to a particular topic, question, or interest. - The session refers to a sequence of queries requested by the user on the search engine, within a certain period of time or with regard to the completion of a task. Read the following query, and session of the user as the context information, which might be helpful and relevant to suggest the next query. Query: {Query} Session: {Session} Based on the above query, and session, please generate one next query suggestion with the rationale, in the format of Query Suggestion: Rationale:

Table 10: The prompt used in the instantiation of the contextual query suggestion baseline.

Types	Texts
System Message	You are an AI assistant whose primary goal is to suggest a next search query, in order to help a user search and find information better on the search engine. Two different queries and entities are separated by the token '|'. For example, 'Microsoft' and 'Google' would appear as 'Microsoft'	'Google'.
User Message	You are going to suggest a search query that the user would search next based on the current query, the current session, and the current article. The explanations of the query, session, and article are as follows: - The query is a specific set of phrases that the user enters into the search engine to find the information or resources related to a particular topic, question, or interest. - The session refers to a sequence of queries requested by the user on the search engine, within a certain period of time or with regard to the completion of a task. - The article refers to a specific webpage that the user clicks and reads from several search results displayed by the search engine in response to the requested query. Read the following query, session, and article of the user as the context information, which might be helpful and relevant to suggest the next query. Query: {Query} Session: {Session} Article Title: {Article['Title']} Article Text: {Article['Text']} Based on the above query, session, and article, please generate one next query suggestion with the rationale, in the format of Query Suggestion: Rationale:

Table 11: The prompt used in the instantiation of the contextual query suggestion w/ $\mathcal{K}_s$ baseline.

Types	Texts
System Message	You are an AI assistant whose primary goal is to suggest a next search query, in order to help a user search and find information better on the search engine. Two different queries and entities are separated by the token '|'. For example, 'Microsoft' and 'Google' would appear as 'Microsoft'	'Google'.
User Message	You are going to suggest a search query that the user would search next based on the current query, the current session, the current article, and the related article. The explanations of the query, session, article, and related article are as follows: - The query is a specific set of phrases that the user enters into the search engine to find the information or resources related to a particular topic, question, or interest. - The session refers to a sequence of queries requested by the user on the search engine, within a certain period of time or with regard to the completion of a task. - The article refers to a specific webpage that the user clicks and reads from several search results displayed by the search engine in response to the requested query. - The related article refers to a specific webpage that the user had previously read with interest, which may be relevant to the current query, session, and article. Read the following query, session, article, and related article of the user as the context information, which might be helpful and relevant to suggest the next query. Query: {Query} Session: {Session} Article Title: {Article['Title']} Article Text: {Article['Text']} Related Article Title: {RelatedArticle['Title']} Related Article Text: {RelatedArticle['Text']} Based on the above query, session, article, and related article, please generate one next query suggestion with the rationale, in the format of Query Suggestion: Rationale: