Bridging the Gap: From Ad-hoc to Proactive Search in Conversations

Chuan Meng
University of Amsterdam
Amsterdam, The Netherlands
c.meng@uva.nl

Francesco Tonolini
Amazon
London, United Kingdom
tonolini@amazon.com

Fengran Mo
Université de Montréal
Montréal, Canada
fengran.mo@umontreal.ca

Nikolaos Aletras
Amazon & University of Sheffield
London, United Kingdom
aletras@amazon.com

Emine Yilmaz
Amazon & UCL
London, United Kingdom
eminey@amazon.com

Gabriella Kazai
Amazon
London, United Kingdom
gkazai@amazon.co.uk

Abstract

Proactive search in conversations (PSC) aims to reduce user effort in formulating explicit queries by proactively retrieving useful relevant information (e.g., facts) given conversational context. Previous work in PSC either directly uses this context as input to off-the-shelf ad-hoc retrievers or further fine-tunes them on PSC data. However, ad-hoc retrievers are pre-trained on short and concise queries, while the PSC input is longer and more noisy. This input mismatch between ad-hoc search and PSC limits retrieval quality. While fine-tuning on PSC data helps, its benefits remain constrained by this input gap. In this work, we propose Conv2Query, a novel conversation-to-query framework that adapts ad-hoc retrievers to PSC by bridging the input gap between ad-hoc search and PSC. Conv2Query maps conversational context into ad-hoc queries, which can either be used as input for off-the-shelf ad-hoc retrievers or for further fine-tuning on PSC data. A key challenge is that users' search intents for each conversational context are implicit, and no ground-truth ad-hoc query targets revealing the implicit search intents are available to optimise the mapping in Conv2Query. We overcome this by generating pseudo ad-hoc query targets from relevant documents for each conversational context, and then fine-tuning a large language model (LLM) to map conversational contexts to pseudo targets. Furthermore, we devise QF-DC, a query filtering mechanism that selects optimal query targets that are relevant to its source document and aligned with the conversational context. Extensive experiments on two PSC datasets show that Conv2Query significantly improves ad-hoc retrievers' performance, both when used directly and after fine-tuning on PSC.

CCS Concepts

• **Information systems → Information retrieval**.

Keywords

Proactive search, Query prediction, Conversational IR

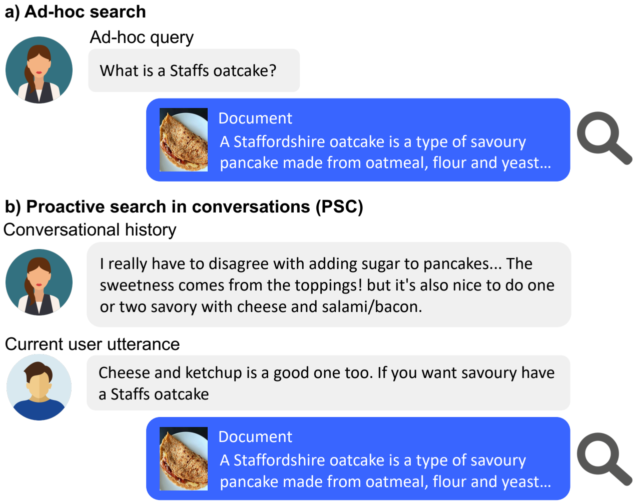
ACM Reference Format:
Chuan Meng, Francesco Tonolini, Fengran Mo, Nikolaos Aletras, Emine Yilmaz, and Gabriella Kazai. 2025. Bridging the Gap: From Ad-hoc to Proactive Search in Conversations. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '25), July 13–18, 2024, Padua, Italy.* ACM, New York, NY, USA, 12 pages.

1 Introduction

Proactive search in conversations (PSC) aims to retrieve relevant documents based on an ongoing conversation without an explicit query from the user [2, 27, 62, 69, 71]. This contrasts with traditional ad-hoc or conversational search which typically follows the popular "query–response" [31, 37, 91, 92] or "query–clarification" paradigms [9, 17, 18, 98], where users issue explicit queries, and then the system retrieves information or asks clarifying questions. PSC has been shown to not only reduce user effort in formulating and refining explicit queries in conversations, but also enrich conversations by proactively introducing relevant facts and ideas [2]. Specifically, PSC supports conversations in two key ways: (i) *Conversation contextualisation* [27, 62, 71]. PSC can proactively retrieve relevant documents to clarify concepts or fact-check claims, before the user explicitly asks for them [27, 62, 71]. E.g., Figure 1b, shows a conversation about "pancake", where a user mentions a specific pancake "Staffs oatcake." The search system returns a document explaining "Staffordshire oatcake," helping the user understand it without needing to search manually (e.g., "What is a Staffs oatcake?" in Figure 1a). (ii) *Interest anticipation* [69, 71]. PSC can proactively retrieve documents aligning with users' next potential interests before they ask for them, also known as *interest anticipation* [69, 71]. As shown in Figure 1b, even without the current user utterance, after a user states, "it's also nice to do one or two savoury," the system proactively retrieves information about "Staffordshire oatcake," a savoury pancake. By retrieving this information in advance, the system eliminates the need for a manual query (e.g., "What pancakes are savoury?") and ensures a natural conversation flow.

Motivation. Existing studies typically feed raw conversational context into off-the-shelf ad-hoc lexical/neural retrievers, or further fine-tune the neural ones on PSC using this raw context [69, 71]. However, previous work faces three key limitations: (i) *Input gap between ad-hoc pre-training and PSC inference.* Directly feeding raw conversational context into neural retrievers pre-trained on ad-hoc search data (e.g., MS MARCO [3]) leads to poor retrieval quality [69].

SIGIR '25, July 13–18, 2024, Padua, Italy



What is a Staffs oatcake?

A Staffordshire oatcake is a type of savoury pancake made from oatmeal, flour and yeast...

I really have to disagree with adding sugar to pancakes... The sweetness comes from the toppings! but it's also nice to do one or two savory with cheese and salami/bacon.

Cheese and ketchup is a good one too. If you want savoury have a Staffs oatcake

A Staffordshire oatcake is a type of savoury pancake made from oatmeal, flour and yeast...

Figure 1: Illustration of proactive search in a multi-party conversation from the ProCIS dataset [71]. Prior work in PSC typically inputs lengthy and noisy conversational context (e.g., concatenated history and the current utterance) into ad-hoc retrievers pre-trained on concise ad-hoc queries.

This is due to the mismatch between the input format used in pre-training and inference: ad-hoc neural retrievers are pre-trained on short and concise queries, whereas in PSC, they receive longer and noisier conversational contexts. Such a distribution shift between training and inference hinders retrieval effectiveness [65, 101, 102]. (ii) *Input gap between ad-hoc pre-training and PSC fine-tuning.* While further fine-tuning ad-hoc neural retrievers on PSC data improves performance [69, 71], the retrieval quality might still be limited by the input mismatches between the source ad-hoc search task (ad-hoc queries) and the target PSC task (conversational context). The discrepancy limits neural retrievers' ability to fully leverage pre-trained ad-hoc knowledge, hindering effective transfer learning [74, 97]. (iii) *Limited performance of lexical retrievers.* Prior work has shown that traditional lexical retrievers (e.g., BM25 [68]) have been found to struggle with verbose conversational contexts [69, 71]. Unlike neural retrievers, they cannot be fine-tuned on PSC data.

A novel framework for PSC. To tackle the above limitations, we propose a *Conversation-to-Query framework* (Conv2Query) for PSC, which aims to effectively adapt ad-hoc neural retrievers to PSC. Conv2Query aims to transform lengthy, noisy conversational context into short, concise ad-hoc queries that closely resemble the format of queries in widely-used ad-hoc search datasets, like MS MARCO [3]); these datasets have been widely-used for training numerous state-of-the-art retrievers [25, 26, 45, 87]. We hypothesise that Conv2Query can effectively improve the performance of ad-hoc neural retrievers on PSC by providing ad-hoc queries during both inference and fine-tuning on PSC data; and Conv2Query can also effectively improve lexical retrievers' performance by delivering concise queries that eliminate noises in conversational contexts.

Learning pseudo ad-hoc query targets. Modelling Conv2Query presents a key challenge. In PSC, users' search intents for each conversational context are implicit, and we need to generate ad-hoc queries that reveal the implicit search intent for each context. However, no ground-truth ad-hoc query training targets revealing implicit search intents are available to optimise the mapping in Conv2Query. Also, our preliminary experiments reveal that directly prompting large language models (LLMs) to generate ad-hoc queries from verbose conversational context yields limited retrieval performance. To overcome the issue, we propose to generate pseudo ad-hoc query training targets for Conv2Query from annotated relevant documents for each conversational context. Because users' implicit search intents are well reflected in the relevant documents, the query targets generated from relevant documents have the potential to capture the implicit search intents. To perform the document-to-query mapping process, we leverage a Doc2Query model [28, 60] because this model has been pre-trained on ad-hoc search data to take a document as input and generate ad-hoc queries that the document might answer. Following prior work [28], we first use Doc2Query to generate a set of queries for each relevant document, and then use query–document relevance filtering to select the optimal query target. The filtering ensures that the selected target is highly relevant to the relevant document for a conversational context.

A new query filtering mechanism for PSC. Although query–document relevance filtering ensures the selected pseudo ad-hoc query target is relevant to the source document, we found that this approach often creates a semantic gap between pseudo ad-hoc query targets and their conversational context, making it harder for the LLM to learn an effective mapping. E.g., as shown in Figure 1b, a query like "What is a Staffordshire oatcake?" ranks high in query–document relevance but lack semantic alignment with the conversational context. In contrast, a query like "What are recipes for savoury oatcakes?" is not only relevant to the document but also aligns with the conversation. However, query–document relevance filtering would overlook such contextually aligned queries. To reduce the semantic gap, we propose *QF-DC*, a query filtering mechanism that selects optimal query targets based on both document relevance and contextual alignment.

Experiments. Experimental results show that Conv2Query significantly improves the performance of reusing off-the-shelf ad-hoc lexical/neural retrievers on PSC (see Section 6.1). Conv2Query enables off-the-shelf ad-hoc neural retrievers to achieve retrieval quality on par with or better than the ones fine-tuned on PSC with raw context. Also, Conv2Query significantly improve the performance of ad-hoc neural retrievers after further fine-tuning them on PSC (see Section 6.2). Furthermore, we assess the impact of query filtering (see Section 7.1), indicating that QF-DC leads to faster convergence and higher retrieval quality. Moreover, we analyse the impact of LLM choices on Conv2Query's performance (see Section 7.2); we examine three families of LLMs (Mistral, Llama and Qwen) spanning from 3B to 22B, showing that Conv2Query performs consistently well in various LLM configurations.

Reproducibility. We release our code and data at https://github.com/ChuanMeng/Conv2Query

Contributions. Our main contributions are as follows:

- We propose Conv2Query for PSC, which effectively adapts ad-hoc retrievers to PSC by bridging the input gap between ad-hoc pre-training and PSC fine-tuning/inference.

- We devise a query filtering mechanism (QF-DC) that selects optimal pseudo ad-hoc query targets based on both document relevance and alignment with conversational context.
- Experimental results show that Conv2Query significantly improves the performance of reusing off-the-shelf ad-hoc retrievers and their performance after further fine-tuning on PSC.

2 Related Work

2.1 Proactive search in conversations

2.1.1 Proactive search. Unlike traditional search following a "query–response" paradigm, where users issue explicit queries and the system retrieves information, proactive search (PS) is query-free, which aims to retrieve relevant information without the user explicitly submitting a query [34]. Because there is no explicit user query, PS utilises users' latent information needs inferred from users' contexts to perform search. PS has been explored by using various users' contexts, including but not limited to users' historical queries (past queries issued by the user, or similar queries issued by other users) [22, 72, 73], web pages browsed by users [12, 42], information about tasks being performed by users (e.g., documents/emails they are reading or writing) [21, 36, 66, 67, 81–83], physical attributes (e.g., time or location) [24, 75, 77, 89], query description [1], ongoing stream of TV broadcast news [30], user status text in social media [63], and conversational context [69, 71].

2.1.2 Proactive search in conversations. Proactive search in conversations (PSC) has recently gained increased attention but remains under-explored [27, 69, 71]. Andolina et al. [2] carry out a user study to show the benefits of PSC. In the study, participants have conversations while accessing a proactive search system that monitors conversation, detects entities mentioned in the conversation and proactively retrieves and presents documents/entities relevant to the conversation. They found that PSC supports conversations with facts/ideas, and reduces users' effort needed to formulate and refine explicit queries. A key bottleneck in developing PSC models is the lack of benchmarks. Ganguly et al. [27] introduce the Retrieval from Conversational Dialogues (RCD) benchmark for PSC, which encourages research on developing systems capable of proactively retrieving relevant documents to contextualise hard-to-understand concepts within a conversation. On this dataset, Pal and Ganguly [62] propose to identify text windows that are likely to be hard-to-understand concepts in conversations, and then perform retrieval based on the identified text windows. However, RCD is limited by its small scale and reliance on movie scripts for conversations, making it less realistic. Furthermore, Ros et al. [69] introduce the WebDisc dataset, a larger and more realistic alternative. It uses Reddit threads as conversations, where users include hyperlinks to external webpages. These hyperlinks often serve as citations or provide additional context, supporting ongoing discussions. Recognising this, Ros et al. [69] regard the linked webpages as relevant documents tied to Reddit threads. Recently, Samarinas and Zamani [71] curate the ProCIS dataset using a similar approach. ProCIS stands out with a larger corpus and training dataset. On both datasets, Ros et al. [69], Samarinas and Zamani [71] feed the raw conversational context into either a lexical retriever or a neural retriever pre-trained on ad-hoc search data. They also further fine-tune these neural retrievers on PSC training data before retrieval.

Our work differs from these studies, as we propose to transform conversational context into ad-hoc queries before using ad-hoc lexical/neural retrievers.

2.2 Query prediction

Our work is related to two key research directions in query prediction: Doc2Query and next query prediction.

2.2.1 Doc2Query. Given a source document, Doc2Query is a process of generating queries that the document might answer [61]. Doc2Query has demonstrated benefits in document expansion [5, 28, 60] and synthetic data generation [8, 13, 33]. In the former, Doc2Query generates a set of relevant queries for each document and appends them to the document before indexing; this expansion improves retrieval quality by bridging term mismatches between user queries and relevant documents. In the latter, Doc2Query generates queries likely relevant to a given document, where each query–document pair forms a positive training example used to train traditional neural ranking models [8, 10, 13, 33] or generative retrieval models [95, 101]. Additionally, Previous work [28, 33] found that some generated queries are irrelevant to the source document, potentially hurting the performance of downstream applications using these queries. To address this issue, these studies [28, 33] use a query filtering mechanism to remove irrelevant generated queries based on their relevance to the source document.

Our work differs from previous studies in three key aspects: (i) we propose Conv2Query that generates ad-hoc queries from conversational context instead of documents; (ii) we leverage Doc2Query to create training data of ad-hoc queries, enabling Conv2Query to learn the mapping from conversation context to ad-hoc queries; and (iii) we propose a query filtering mechanism based on *query–conversation relevance*, ensuring that our ad-hoc query pseudo labels align closely with both target document and conversational context.

2.2.2 Query suggestion. Query suggestion (a.k.a query recommendation) is a core task in session search [6, 7, 39]. It aims to predict the next user query given past users' search behaviours. Query suggestion can assist users in formulating their queries [70], which is particularly valuable when an information need requires multiple searches [23]. Specifically, query suggestion has been studied to predict the next query based on various types of information, such as user historical queries in the current or previous search sessions [11, 14, 58, 70, 78], user feedback on the search result (such as browsing and clicks) [85], or pre-search context (e.g., the news article a user browsed before the search) [35].

Unlike query suggestion, which predicts the next query based on session search data (e.g., query logs or clickthrough data), our work is to generate queries directly from noisy and verbose conversational context, without an explicit user query at the moment. Additionally, it is important to note that Yang et al. [90] focus on selecting the next question a user might ask in a conversation from a predefined pool of user question candidates. We differ as we generate queries from the conversational context without relying on an existing set of user question candidates.

2.3 Conversational search

Conversational search aims to retrieve relevant documents for users' context-dependent queries in a multi-turn conversation [53]. These context-dependent queries often contain omissions, coreferences or ambiguities, making it difficult for ad-hoc search methods to capture the underlying information need. Two main research directions address the context-dependent query understanding problem: conversational query rewriting and conversational dense retrieval. Conversational query rewriting aims to transform context-dependent queries into self-contained ones [31, 37, 47, 48, 52, 54, 55, 91–93], while conversational dense retrieval trains a query encoder to encode the current user query and conversational history into a contextualized query embedding that is expected to implicitly represent the information need of the current query in a latent space [29, 43, 46, 49, 50, 56, 57, 94].

The key difference between PSC and conversational search is that conversational search has explicit user query at each turn, whereas PSC operates on conversation context alone, without a current explicit user query. Because there is no explicit user query, existing conversational query rewriting methods cannot be directly applied to the more challenging PSC scenario.

2.4 Proactive response prediction

Proactive conversational response prediction aims to produce a system response that guides the conversation direction [15, 16, 19, 40, 41]. Various types of proactive response prediction have been explored, such as clarifying question prediction [9, 17, 18, 98], user preference elicitation [99], persuasion [51], target-steering [84, 100], item recommendation [79], suggesting follow-up questions [9, 38, 80, 88], and providing additional information [4, 38]. Amongst these, providing additional information is most relevant to PSC, which aims to proactively produce a response offering supplementary and useful information not explicitly requested by users [4, 38]. For example, a recent study [38] prompt LLMs to generate a proactive response that consists of the answer to the user's query and a proactive element, which refers to new information related to the initial query. However, instead of focusing on response generation, PSC focuses on retrieving relevant documents to offer additional information to users in the absence of an explicit user query.

3 Task definition

Given a conversational context C_t at turn t and a corpus of documents $D = \{d_1, d_2, \cdots, d_{|D|}\}$, the goal of PSC is to develop a ranking model that retrieves a ranked list of k documents $D_t = \{d_{t,1}, d_{t,2}, \cdots, d_{t,k}\}$ from D; D_t provides relevant information (e.g., facts or ideas) to support C_t [71]. Note that a user utterance in the conversational context C_t can take any form and is not necessarily a query. Following Ros et al. [69], we study two settings: (i) *Conversation contextualisation* (referred to as the "full" setting in [69]): C_t consists of user utterances from turns 1 to t, including the conversational history $\{u_1, u_2, \ldots, u_{t-1}\}$ (user utterances up to turn $t-1$) and the current user utterance u_t at turn t. The goal is to retrieve relevant documents D_t to clarify hard-to-understand concepts mentioned in u_t or to verify factual claims made by the user in u_t. (ii) *Interest anticipation* (referred to as the "proactive" setting in [69]): C_t consists of conversational history $\{u_1, u_2, \cdots, u_{t-1}\}$ with user utterances up to turn $t-1$. The aim is to retrieve documents D_t aligned with the user's interest at turn t. In other words, the ranking model f must anticipate the information the user is likely to explore at turn t, based on the conversational history up to turn $t-1$. Ros et al. [69] has shown that this setting is more challenging than conversation contextualisation.

Ros et al. [69] also explore the "last" setting, where C_t includes only the current user utterance u_t; they found that retrievers perform similarly whether they use only the current user utterance u_t or combine it with the conversational history. We exclude this setting, as we believe only using the current utterance is insufficient for practical applications that often require context from prior interactions. Additionally, Samarinas and Zamani [71]consider a "reactive" setting, where retrieval occurs only after a conversation reaches its final turn T. We do not adopt this setting, as it not suitable for delivering timely information to support ongoing conversations.

4 Method

Conv2Query consists of five phases: (i) generating ad-hoc queries from documents, (ii) query filtering by relevance to documents and conversations, (iii) learning to generate ad-hoc queries from conversations, (iv) at inference, generating ad-hoc queries for retrieval, and (v) (optionally) fine-tuning ad-hoc retrievers via filtered ad-hoc queries. In (i), we leverage a Doc2Query model to generate n ad-hoc query candidates from a document (see Section 4.1); in (ii), we introduce a novel query filtering mechanism (QF-DC) that evaluates query–document relevance and query–conversation alignment to select the optimal ad-hoc query target that is relevant to both its source documents and conversational context (see Section 4.2). In (iii), we fine-tune Conv2Query model to learn the mapping from a conversational context to its filtered ad-hoc query (see Section 4.3). (iv), at inference, given a conversational context, we generate an ad-hoc query to be used with any ad-hoc retriever (see Section 4.4). (v) is optional: we fine-tune an ad-hoc neural retriever on PSC by using our filtered ad-hoc queries produced in (ii) (see Section 4.5).

4.1 Generating ad-hoc queries from documents

For a conversational turn t annotated with a relevant document d_t^+, we leverage a Doc2Query [5, 60] model to map d_t^+ to a set of n ad-hoc query candidates that d_t^+ might answer. Formally,

$$\{q_{t,1}, q_{t,2}, \ldots, q_{t,n}\} = \mathrm{f}_{\mathrm{Doc2Query}}(d_t^+), \tag{1}$$

where $\{q_{t,1}, q_{t,2}, \ldots, q_{t,n}\}$ represent n generated ad-hoc query candidates. Doc2Query models excel at generating ad-hoc queries from documents due to its pre-training on query–document pairs from MS MARCO [3], a widely-used ad-hoc search dataset; the model architecture can be based on a pre-trained language model (e.g., T5 [28, 60]) or an LLM (e.g., Llama 2 [5]).

4.2 Query filtering by relevance to documents and conversations

We propose a query filtering mechanism that evaluates both query–document relevance and query–conversation relevance, ensuring the selection of queries that are highly relevant to their source document while also align with the corresponding conversational

Instruction: Based on the following conversation history and the current user utterance, please generate a search query that retrieves documents relevant to the current user utterance.
Conversational history: {}
Current user utterance: {}
Generated query:

Figure 2: Prompt of conversation contextualization setting.

context. Specifically, given the generated n ad-hoc query candidates $\{q_{t,1}, q_{t,2}, \ldots, q_{t,n}\}$, we select an optimal query candidate q_t^* by identifying the query candidate with the highest aggregated score across all n candidates. This score is derived by aggregating the query–document relevance and query–conversation relevance scores. Formally,

$$\begin{aligned} i^* &= \arg \max_{i\in\{1,\ldots,n\}} s_{t,i}, \\ q_t^* &= q_{t,i^*}, \\ s_{t,i} &= \mathrm{f}_{\mathrm{aggregate}}(s_{t,i}^{qd}, s_{t,i}^{qc}) \in \mathbb{R}, \end{aligned} \tag{2}$$

where $s_{t,i}$ represents the aggregated score for the i-th query candidate $q_{t,i}$, calculated using the aggregation function $\mathrm{f}_{\mathrm{aggregate}}(\cdot,\cdot)$ (e.g., summation). $s_{t,i}^{qd}$ and $s_{t,i}^{qc}$ denote the query–document relevance score and the query–conversation relevance score for $q_{t,i}$, respectively. We follow Gospodinov et al. [28] to compute the query–document relevance score $s_{t,i}^{qd}$:

$$s_{t,i}^{qd} = \mathrm{f}_{\mathrm{relevance}}(q_{t,i}, d_t^+) \in \mathbb{R}, \tag{3}$$

where $\mathrm{f}_{\mathrm{relevance}}(\cdot,\cdot)$ is a relevance prediction model that maps a query–document pair to a relevance score, with higher scores representing greater relevance. $\mathrm{f}_{\mathrm{relevance}}(\cdot,\cdot)$ can be either a re-ranker (e.g., MonoT5 [59]) or a retriever (e.g., TCT-ColBERT [44]). We calculate the query–conversation relevance score $s_{t,i}^{qc}$ in a similar way:

$$s_{t,i}^{qc} = \mathrm{f}_{\mathrm{relevance}}(q_{t,i}, C_t) \in \mathbb{R}, \tag{4}$$

where the value of $s_{t,i}^{qc}$ is higher if $q_{t,i}$ is more relevant to its corresponding conversational context.

4.3 Learning to generate ad-hoc queries from conversations

We treat the selected ad-hoc query q_t^* as a learning target and pair it with the corresponding conversational context C_t to form a training data point (C_t,q_t^*). It enables us to train our Conv2Query model in mapping from a conversational context C_t to q_t^*, represented as $C_t \rightarrow q_t^*$. The loss function for a conversation with T tuns is defined as follows:

$$\mathcal{L}(\theta_{Conv2Query}) = -\frac{1}{Z} \sum_{t\in\{t|I(t)=1\}}^{T} \log P(q_t^* \mid \mathrm{prompt}(C_t)), \tag{5}$$

where $I(t)$ is an indicator function that equals to 1 if turn t is annotated with a relevant document and 0 otherwise. $Z = \sum_{i=1}^{T} I(t)$. $\mathrm{prompt}(\cdot)$ is a prompt to instruct the Conv2Query model.

Instruction: Based on the following conversation history, please generate a search query that retrieves documents relevant to the next expected utterance.
Conversational history: {}
Generated query:

Figure 3: Prompt for the interest anticipation setting.

4.4 Generating ad-hoc queries for retrieval

At inference time, the trained Conv2Query model takes a conversational context C_t at turn t as input and generates an ad-hoc query q_t'. This query q_t' is then passed to a retrieval system, which returns a ranked list of relevant documents D_t:

$$\begin{aligned} q_t' &= \mathrm{f}_{\mathrm{Conv2Query}}(\mathrm{prompt}(C_t)), \\ D_t &= \mathrm{f}_{\mathrm{retriever}}(q_t'). \end{aligned} \tag{6}$$

4.5 Retriever fine-tuning using pseudo queries

For each conversational context C_t, we pair its selected optimal ad-hoc query q_t^* (Section 4.2) with the corresponding relevant document d_t^+ to create a positive training example (q_t^*, d_t^+). We then sample negative documents for C_t following standard neural retrieval practices and use both positive and negative examples to fine-tune a specific ad-hoc retriever on PSC, such as ANCE [87], SPLADE++ [25] or RepLLaMA [45].

5 Experimental setup

Research questions. Our work is steered by the following research questions:

RQ1 To what extent does Conv2Query bridge input gap between ad-hoc pre-training and PSC inference under conversation contextualisation and interest anticipation settings?

RQ2 How well Conv2Query bridge input gap between ad-hoc pre-training and PSC fine-tuning under the two settings?

RQ3 To what extent does our proposed query filtering mechanism (QF-DC) improve Conv2Query's performance?

RQ4 To what extent the choice of LLMs impact Conv2Query's performance?

Datasets. We use two recent large-scale datasets for proactive search in multi-party conversations: ProCIS [71] and WebDisc [69]:

- **ProCIS** [71] consists of Reddit threads where multiple users engage in discussions; each conversation (thread) includes at least one utterance (comment) that contains Wikipedia hyperlinks; the hyperlink is added by the user when posting the comment. These user-added Wikipedia articles serve as retrieval targets (sparse relevance judgments) because they frequently offer additional context or background information relevant to the ongoing conversation. The dataset has a corpus of 5,315,384 Wikipedia articles; the average article length is 145.88 tokens (Llama tokenizer). The dataset has four subsets: train, dev, future-dev, and test, containing 2,830,107, 4,165, 3,385, and 100 conversations (threads), respectively. The average number of turns per conversation in these subsets is 5.41, 4.91, 4.48, and 4.49. The future-dev set only contains conversations that occur chronologically after those in the training set; so it can be used to evaluate a retrieval model's ability to generalise to newly emerging concepts and topics not seen during training. The test set contains turn-level human-annotated dense relevance judgments, while other sets only has turn-level sparse relevance judgments based on user-included Wikipedia articles. On the test set, each turn with associated relevant documents has an average of 2.30 relevant documents.

- **WebDisc** [69] is built in a similar way to ProCIS, consisting of Reddit threads with turn-level sparse relevance judgments derived from user-added webpage hyperlinks. This dataset has a corpus of 98,231 webpages not limited to Wikipedia. Ros et al. [69] truncate overly long webpages to ensure compatibility with passage ranking models. The dataset is split into train, validation and test sets, containing 128,404, 15,344 and 15,249 turns with user-added webpages, respectively.

For both datasets, raw links are already removed from Reddit utterances. In a conversation (i.e., thread), only certain user utterances (i.e., comments) are associated with hyperlinks or human-labelled documents. Thus, the assumption is that a PSC retriever should perform retrieval at those turns.

Baselines. We use retrievers under two settings: off-the-shelf ad-hoc retrievers or these further fine-tuned on PSC.

For off-the-shelf ad-hoc retrievers, we evaluate them by feeding them three types of input without any fine-tuning on PSC data.[1] First, following Ros et al. [69], we feed conversational context into off-the-shelf ad-hoc retrievers. Specifically, we use one lexical retriever BM25 [68], and three neural retrievers that are pre-trained on the widely-used ad-hoc search dataset MS MARCO [3]. For the neural ones, we consider one learned sparse retriever, SPLADE++ [25] (`splade-cocondenser-ensembledistil`), and two dense retrievers: ANCE [87] (`ance-msmarco-passage`), and RepLLaMA [45] (`repllama-v1-7b-lora-passage`), an LLM-based state-of-the-art retriever. Second, we consider two methods specifically designed for PSC, which first process the conversational context before using off-the-shelf ad-hoc retrievers: (i) Text Window [62] first extracts text segments with size k for the conversational context, and selects those likely to achieve high retrieval quality. Following Pal and Ganguly [62], we use the query performance prediction (QPP) method NQC [76] to estimate retrieval quality and set $k = 5$. However, we replace the original LM-Dirichlet [96] retriever with the more recent and effective RepLLaMA. (ii) LMGR [71] first uses an LLM to generate n text descriptions for conversational context; then it retrieves k documents per description, and use an LLM to select the document that best matches each description. We follow Samarinas and Zamani [71] in using OpenChat-3.5 (enhanced Mistral-7B) as the LLM, with $n = 20$ and $k = 5$, but replace the original ANCE retriever with the more recent RepLLaMA for consistency. Third, we assess a Conv2Query variant that relies solely on prompting (only using Equation 4.4). It directly prompts an LLM to generate ad-hoc queries from conversational context. We apply 1-shot and 2-shot prompting, denoted as Conv2Query-1-S and Conv2Query-2-S. [2] The queries generated by Conv2Query-1-S/-2-S are then fed into RepLLaMA for retrieval.

Regarding the ad-hoc retrievers further fine-tuned on PSC, we use the three neural retrievers, ANCE [87], SPLADE++ [25] and RepLLaMA [45], each fine-tuned on PSC data. All are fed with conversational context during fine-tuning on PSC. These baselines can be viewed as adapted conversational dense retrieval methods on PSC, as they follow a similar technical paradigm [46, 57, 64].

Evaluation metrics. We follow Ros et al. [69] to use Precision@1 (P@1) and MRR@10 as our evaluation metrics for dev/val sets of both datasets. This choice is made because these sets only contain sparse relevance judgments. Moreover, as PSC is designed for conversational scenarios, it prioritises retrieving the most relevant document at the top of the ranked list. Therefore, precision-oriented metrics like Precision and MRR are particularly suitable.

For the ProCIS test set, which contains dense relevance judgments, we further use npDCG, a metric proposed by Samarinas and Zamani [71] specifically tailored to PSC. Unlike nDCG [32], npDCG has three features: (i) it aggregates DCG/iDCG values across all turns per conversation into a single score; (ii) it avoids rewarding a retrieval model for returning the same relevant document across multiple turns; and (iii) it evaluates the timing prediction of a PSC system: a retriever can gain only when retrieving at turns with annotated judgments; the retriever incurs no gain/cost if it skips retrieval at a turn or retrieves at a turn without any annotated judgments. Because each turn with associated judgements has 2.30 relevant documents on average, we use a cut-off of 5 for npDCG. Note that our work focuses only on what to retrieve, and leaves retrieval timing prediction for future work. So we assume perfect timing prediction and perform retrieval only at turns with annotated documents.

Implementation details. We perform BM25 retrieval via `Pyserini`; for ProCIS, we set $k_1 = 0.9$ and $b = 0.4$; for WebDisc, we follow Ros et al. [69] to set $k_1 = 8, b = 0.99$ for the conversation contextualisation setting, and $k_1 = 7, b = 0.99$ for the interest anticipation setting. For all neural retrievers fed with conversational context, we observe that the default truncation length (e.g., 32 tokens) of the query encoders used on MS MARCO substantially reduces retrieval quality, as lots of important information in the lengthy conversational context is truncated. Thus, we set the truncation length for query encoders to 512, which exceeds the average conversational context length. To further fine-tune a neural retriever on PSC, we randomly sample negative documents from a mix of top 200 hard negatives examples retrieved by BM25 using the conversational history alone and the history combined with the user's current utterance, increasing the diversity of the negatives. For a fair comparison, all neural retrievers use the same negative samples during fine-tuning on PSC data.

Regarding our method, for the Doc2Query model used in Equation 1, we use doc2query-T5[3]; we generate 100 queries per document ($n = 100$ in Equation 1); to ensure that the generated ad-hoc query candidates are both diverse and relevant to their source document, we follow [60, 101] to adopt a top-k sampling strategy

[1] Note that we do not use conversational query rewriting methods [31, 37, 48, 52, 91, 92] as baselines, because there is no current explicit user query in PSC for rewriting, making these methods inapplicable to PSC.

[2] We randomly sample one or two training examples respectively, each consisting of a conversational context and its ad-hoc query selected by QF-DC from Doc2Query candidates. We found that adding more than two examples significantly increases inference costs without notable gains.

[3] Doc2Query-T5: https://huggingface.co/BeIR/query-gen-msmarco-t5-large-v1. We also experimented with Doc2Query-Llama2 released by Basnet et al. [5], but our preliminary results showed no notable improvement over Doc2Query-T5.

($k = 10$) during the query generation. We use RankLLaMA [45][4] as the relevance model (Equations 3 and 4) for query filtering. We use summation as the aggregation function in Equation 2. For the Conv2Query model (Equations 5 and 6), we initialise it with `Mistral-7B-Instruct-v0.3`. We use the prompts illustrated in Figures 2 and 3 for the contextualisation and interest anticipation settings, respectively. We fine-tune the Conv2Query model on the training set using QLoRA [20] for one epoch. All experiments are conducted on 4 NVIDIA A100 GPUs (40GB). For neural retrievers using queries generated by Conv2Query, we set the truncation length of the query encoders to 32 tokens. Note that when fine-tuning neural retrievers on PSC, we ensure they use the same set of negative examples, regardless of using conversational context or pseudo ad-hoc queries as input.

6 Results

6.1 From ad-hoc pre-training to PSC inference

To answer **RQ1**, we present results of off-the-shelf ad-hoc retrievers using conversational context, text windows/descriptions and ad-hoc queries generated by Conv2Query and its prompting-only variant (Conv2Q-1-S/-2-S), on ProCIS and WebDisc, under the conversation contextualisation and the interest anticipation settings in Table 1.

We have four main observations. First, Conv2Query-generated queries leads to a significant improvement in retrieval quality for the lexical retriever BM25 compared to using conversational context, across all metrics, evaluation sets and settings. E.g., Conv2Query improves BM25's MRR@10 values by 0.241, 0.198, and 0.236 on the ProCIS dev, future-dev, and test sets under conversation contextualisation, and by 0.131, 0.144, and 0.087 under interest anticipation. We attribute this improvement to Conv2Query's ability to remove noise from raw conversational contexts and generate queries containing keywords that reflect users' implicit information needs.

Second, all ad-hoc neural retrievers using Conv2Query-generated queries significantly outperform their counterparts using conversational context across all metrics, evaluation sets and settings. For instance, Conv2Query improves RepLLaMA's MRR@10 values by 0.355, 0.257 and 0.330 on the ProCIS dev, future-dev, and test sets under conversation contextualisation, and by 0.122, 0.117 and 0.094 under interest anticipation. The improvement demonstrates that Conv2Query effectively adapts off-the-shelf ad-hoc neural retrievers to PSC by resolving input mismatches, without requiring retriever fine-tuning on PSC.

Third, Conv2Query outperforms Text Window and LMGR by a large margin across all settings. We attribute this to two key advantages of Conv2Query. (i) Conv2Query generates ad-hoc queries that closely resemble those in ad-hoc search datasets used for ad-hoc retriever pre-training, whereas Text Window and LMGR return text segments or descriptions, causing an input format mismatch. And (ii) Conv2Query captures users' implicit search intents by learning to generate queries that effectively retrieve the annotated relevant documents for a given conversational context. However, the two baselines lack an effective strategy to ensure their text windows or descriptions accurately match users' implicit information needs.

Fourth, Conv2Query significantly outperforms its prompting-only variant (Conv2Q-1-S/-2-S) in retrieval performance. We think this is because fine-tuning provides Conv2Query with extensive training signals to learn to generate queries that accurately align with the annotated relevant documents for each conversational context, effectively capturing users' implicit information needs.

6.2 From ad-hoc pre-training to PSC fine-tuning

To answer **RQ2**, we examine the performance of further fine-tuning ad-hoc three neural retrievers (ANCE, SPLADE++ and RepLLaMA) on PSC training data using raw conversational context and our generated pseudo ad-hoc queries, on ProCIS and WebDisc, under the conversation contextualisation and interest anticipation settings, in Table 2 Note that for further fine-tuning using pseudo ad-hoc queries, we fine-tune each retriever using optimal pseudo ad-hoc queries selected by QF-DC from Doc2Query-generated candidates (see Section 4.5).

We have two main observations. First, compared to the results in Tables 1 in Section 6.1, we found that ad-hoc neural retrievers using Conv2Query-generated queries without fine-tuning on PSC achieve comparable or superior retrieval performance to retrievers fine-tuned on PSC using conversational context. This finding reiterates the effectiveness of Conv2Query in adapting ad-hoc retrievers to PSC, even without retriever fine-tuning.

Second, we found that all retrievers fine-tuned on our pseudo ad-hoc query targets and inferred with Conv2Query-generated queries significantly outperform those fine-tuned and inferred using raw conversational context across all metrics, datasets, and settings. E.g., RepLLaMA fine-tuned and inferred with ad-hoc queries surpasses its conversational context counterpart in MRR@10 by 0.045, 0.030, and 0.031 on the ProCIS dev, future-dev, and test sets under conversation contextualisation, and by 0.034, 0.057, and 0.055 under interest anticipation. We attribute this improvement to the reduced domain distance between ad-hoc pre-training (source) and PSC fine-tuning (target) by consistently using ad-hoc query formats. As a result, retrievers fine-tuned on our pseudo ad-hoc queries can fully leverage pre-trained ad-hoc knowledge gained during ad-hoc pre-training, leading to more effective transfer learning.

7 Analysis

7.1 Impact of query filtering

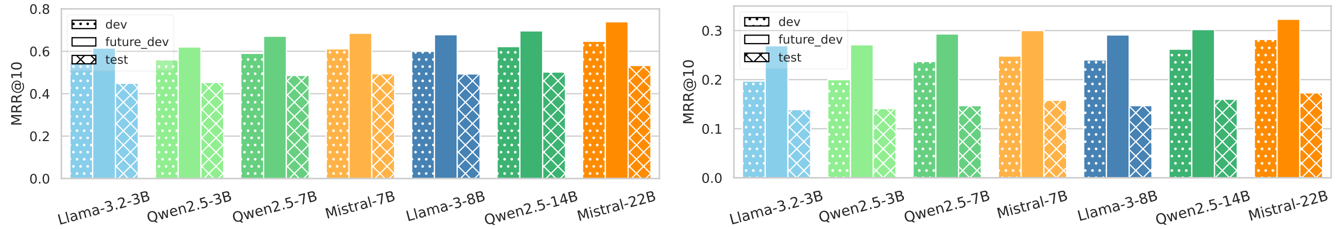
To answer **RQ3**, we study the impact of query mechanism by examining Conv2Query 's learning curves under three settings: (i) *QF-DC* (our final approach) uses both query–document relevance and query–conversation alignment; (ii) *QF-D* uses query filtering based only on query–document relevance [28, 33] by removing Equation 3 and excluding $s^{qc}_{t,i}$ in Equation 2; and (iii) *Random* randomly selects an ad-hoc query from candidates generated by a Doc2Query model (See implementation details in Section 5). Figure 5 presents the retrieval quality (MRR@10) of RepLLaMA (pre-trained on MS MARCO) using Conv2Query-generated queries w.r.t. different training steps, with the three query filtering settings; the results are reported on the dev, future-dev, and test sets of ProCIS, under both the conversation contextualisation and interest anticipation settings.

We have two main observations. First, QF-D leads to higher retrieval quality than Random. This suggests that before directly using Doc2Query-generated candidates to fine-tune Conv2Query, it is

[4] https://huggingface.co/castorini/rankllama-v1-7b-lora-passage

Table 1: Results of reusing off-the-shelf ad-hoc retrievers under the conversation contextualisation and interest anticipation settings. Conversational context in the former includes both conversational history and the current user utterance, while in the latter, it consists only of conversational history. "PT" and "Inf" indicate retriever inputs during ad-hoc pre-training and PSC inference, respectively; "Q" denotes ad-hoc queries; "Conv." denotes conversational context; "Text win" and "LMGR" denote two baselines that convert conversational context into text segments and descriptions, respectively; Conv2Q-1-S/-2-S denote the prompting-only variant of our method; and " Conv2Q" denotes our method Conv2Query. The best value in each column is bold-faced. * denotes a significant improvement when a retriever uses Conv2Query-generated queries at inference, compared to the same retriever with other inputs (paired *t*-test, *p*-value < 0.05).

				ProCIS						WebDisc			
				dev		future-dev		test		val		test	
	Retriever	PT	Inf	P@1	MRR@10	P@1	MRR@10	npDCG@5	MRR@10	P@1	MRR@10	P@1	MRR@10
Conversational Contextual.	BM25	-	Conv	0.082	0.123	0.265	0.295	0.043	0.052	0.205	0.281	0.199	0.277
	ANCE	Q	Conv	0.067	0.093	0.187	0.215	0.031	0.044	0.112	0.157	0.111	0.155
	SPLADE++	Q	Conv	0.144	0.219	0.343	0.398	0.115	0.136	0.170	0.250	0.160	0.249
	RepLLaMA	Q	Conv	0.186	0.256	0.377	0.428	0.132	0.164	0.204	0.280	0.199	0.274
	RepLLaMA	Q	Text win	0.187	0.252	0.401	0.452	0.139	0.174	0.225	0.297	0.218	0.291
	RepLLaMA	Q	LMGR	0.203	0.267	0.387	0.440	0.146	0.184	0.222	0.291	0.213	0.295
	RepLLaMA	Q	Conv2Q-1-S	0.311	0.385	0.459	0.522	0.261	0.368	0.234	0.302	0.232	0.314
	RepLLaMA	Q	Conv2Q-2-S	0.315	0.393	0.462	0.527	0.266	0.369	0.246	0.311	0.247	0.316
	BM25	-	Conv2Q	0.323*	0.409*	0.399*	0.493*	0.209*	0.288*	0.283*	0.358*	0.274*	0.349*
	ANCE	Q	Conv2Q	0.434*	0.501*	0.516*	0.576*	0.289*	0.386*	0.284*	0.352*	0.278*	0.347*
	SPLADE++	Q	Conv2Q	0.522*	0.588*	0.612*	0.665*	0.351*	0.477*	0.312*	0.381*	0.302*	0.375*
	RepLLaMA	Q	Conv2Q	**0.556***	**0.611***	**0.638***	**0.685***	**0.361***	**0.494***	**0.341***	**0.417***	**0.333***	**0.410***
Interest Anticipation	BM25	-	Conv	0.028	0.043	0.051	0.070	0.017	0.015	0.088	0.131	0.085	0.127
	ANCE	Q	Conv	0.038	0.051	0.056	0.070	0.019	0.021	0.056	0.082	0.054	0.080
	SPLADE++	Q	Conv	0.079	0.095	0.096	0.147	0.049	0.059	0.071	0.114	0.086	0.119
	RepLLaMA	Q	Conv	0.090	0.126	0.137	0.183	0.053	0.064	0.098	0.143	0.096	0.141
	RepLLaMA	Q	Text win	0.092	0.129	0.141	0.183	0.057	0.065	0.101	0.143	0.098	0.143
	RepLLaMA	Q	LMGR	0.098	0.133	0.145	0.191	0.059	0.070	0.105	0.151	0.102	0.149
	RepLLaMA	Q.	Conv2Q-1-S	0.111	0.155	0.168	0.212	0.084	0.113	0.127	0.172	0.127	0.172
	RepLLaMA	Q.	Conv2Q-2-S	0.113	0.159	0.171	0.216	0.087	0.116	0.130	0.175	0.130	0.175
	BM25	-	Conv2Q	0.134*	0.174*	0.170*	0.214*	0.073*	0.102*	0.145*	0.209*	0.145*	0.205*
	ANCE	Q	Conv2Q	0.172*	0.206*	0.223*	0.252*	0.095*	0.121*	0.141*	0.198*	0.142*	0.190*
	SPLADE++	Q	Conv2Q	0.208*	0.238*	0.261*	0.286*	0.101*	0.131*	0.151*	0.208*	0.152*	0.204*
	RepLLaMA	Q	Conv2Q	**0.218***	**0.248***	**0.272***	**0.300***	**0.123***	**0.158***	**0.166***	**0.238***	**0.166***	**0.235***



(a) Conversational contextualisation (CC) **(b) Interest anticipation (IA)**

Figure 4: Retrieval quality (MRR@10) of RepLLaMA using ad-hoc queries generated by Conv2Query with three families of LLMs from 3B to 22B, on the ProCIS dev, future-dev, test sets.

essential to ensure high-quality ad-hoc query learning targets that can effectively retrieve their corresponding source documents. This finding aligns with previous research on query filtering based on query–document relevance [28, 33]. Second, QF-DC results in faster convergence and higher retrieval quality than QF-D. We think this is because our introduced query–conversation alignment in QF-DC ensures that the selected ad-hoc query learning targets are pertinent to their corresponding conversational context (Conv2Query's learning inputs); this reduced semantic gap between inputs and targets enables Conv2Query to more effectively learn to generate

Table 2: Results of fine-tuning retriever under the conversation contextualisation and interest anticipation settings. Conversational context in the former includes both conversational history and the current user utterance, while in the latter, it consists only of conversational history. "PT", "FT" and "Inf" indicate retriever inputs during ad-hoc pre-training, PSC fine-tuning and PSC inference, respectively; "Q" denotes ad-hoc queries; "Conv." denotes conversational context; "Conv2Q" denotes ad-hoc queries produced by Conv2Query. The best value in each column is bold-faced. * denotes a significant improvement when a retriever uses ad-hoc queries during PSC fine-tuning and PSC inference, compared to the same retriever using conversational context (paired t-test, p-value < 0.05).

	Retriever	PT	FT	Inf	ProCIS						WebDisc			
					dev		future-dev		test		val		test	
					P@1	MRR@10	P@1	MRR@10	npDCG@5	MRR@10	P@1	MRR@10	P@1	MRR@10
Conv. Context.	ANCE	Q	Conv	Conv	0.446	0.529	0.548	0.613	0.281	0.420	0.246	0.318	0.247	0.313
	SPLADE++	Q	Conv	Conv	0.491	0.574	0.593	0.658	0.326	0.465	0.311	0.373	0.292	0.376
	RepLLaMA	Q	Conv	Conv	0.520	0.603	0.623	0.687	0.355	0.494	0.340	0.402	0.321	0.405
	ANCE	Q	Conv2Q	Conv2Q	0.484*	0.563*	0.579*	0.650*	0.321*	0.457*	0.319*	0.385*	0.308*	0.378*
	SPLADE++	Q	Conv2Q	Conv2Q	0.552*	0.620*	0.641*	0.697*	0.386*	0.507*	0.352*	0.424*	0.334*	0.411*
	RepLLaMA	Q	Conv2Q	Conv2Q	**0.588***	**0.648***	**0.662***	**0.717***	**0.397***	**0.525***	**0.383***	**0.458***	**0.364***	**0.445***
Inter. Antic.	ANCE	Q.	Conv.	Conv.	0.127	0.156	0.144	0.180	0.061	0.076	0.068	0.093	0.061	0.090
	SPLADE++	Q	Conv	Conv	0.154	0.211	0.189	0.241	0.080	0.101	0.105	0.174	0.100	0.171
	RepLLaMA	Q	Conv	Conv	0.188	0.244	0.225	0.277	0.106	0.131	0.160	0.229	0.162	0.229
	ANCE	Q	Conv2Q	Conv2Q	0.201*	0.234*	0.252*	0.280*	0.123*	0.151*	0.165*	0.229*	0.160*	0.215*
	SPLADE++	Q	Conv2Q	Conv2Q	0.236*	0.265*	0.291*	0.315*	0.131*	0.160*	0.177*	0.238*	0.181*	0.232*
	RepLLaMA	Q	Conv2Q	Conv2Q	**0.244***	**0.278***	**0.304***	**0.334***	**0.147***	**0.186***	**0.197***	**0.270***	**0.199***	**0.270***

queries that accurately retrieve their source documents. Our finding align with previous research [86] showing that narrowing the semantic gap between learning inputs and targets facilitates the learning process and leads to better performance.

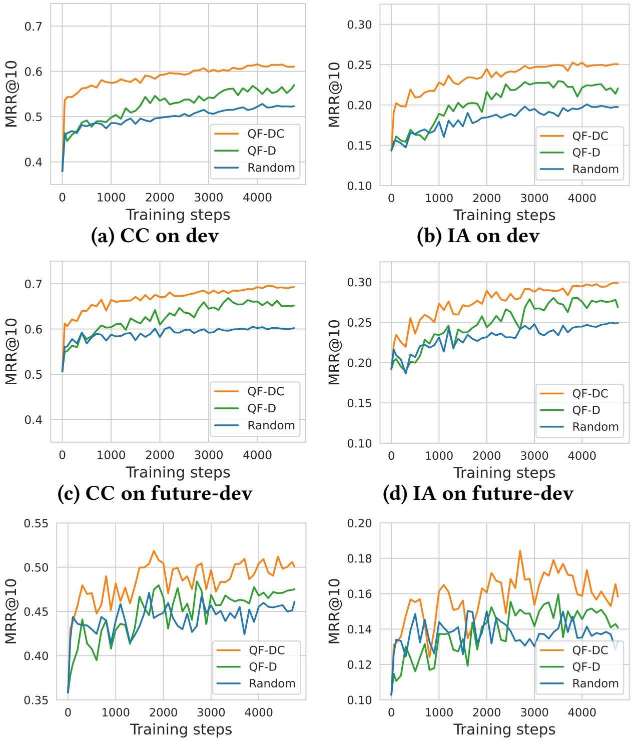
7.2 Impact of the choice of LLMs

To answer **RQ4**, we examine how the choice of LLM impacts Conv2Query's performance. We follow the same fine-tuning setup (see implementation details in Section 5) to evaluate three widely-used LLMs families, Mistral, Llama and Qwen, spanning from 3B to 22B. Specifically, for Mistral, we use Mistral-7B-Instruct-v0.3 and Mistral-Small-Instruct-2409 (denoted it as "Mistral-22B-Instruct"); for llama, we use Llama-3.2-3B-Instruct and Llama-3.1-8B-Instruct; for Qwen, we use Qwen2.5-3B/7B/14B-Instruct. The results, presented in Figure 4, reveal two key insights. First, Conv2Query performs consistently well across different LLM configurations, highlighting its robustness and generalisability across models of varying sizes. Second, scaling LLM size leads to a steady increase in Conv2Query's performance. Mistral-22B, the largest model in our evaluation, results in state-of-the-art retrieval quality.

8 Conclusions & Future Work

We have proposed Conv2Query, a novel framework for PSC, which effectively adapts ad-hoc retrievers to PSC by bridging the input gap between ad-hoc pre-training and PSC fine-tuning/inference. Conv2Query learns to map conversational contexts to pseudo ad-hoc query targets that capture users' implicit information needs. To do so, we have leveraged Doc2Query to generate a set of pseudo queries from relevant documents for each conversational context. Furthermore, we have devised QF-DC, a novel query filtering mechanism that selects the optimal query target for each conversational context based on document relevance and conversation alignment. Extensive experimental results have shown that Conv2Query significantly improve the performance of ad-hoc retrievers, whether used directly or after fine-tuning. QF-DC accelerates convergence while improving retrieval performance, and Conv2Query remains robust and generalisable across various LLM configurations.

Our work has the following limitations. First, we focus on what to retrieve in PSC without exploring the prediction of retrieval timing. Future work includes integrating what and when into a unified model for PSC. Second, Conv2Query incurs extra query latency due to the overhead of query generation. We plan to distill the knowledge of LLM-based Conv2Query into smaller models. Third, we evaluate our method on two PSC datasets based on multi-party Reddit threads, which might not fully represent the diverse scenarios of PSC. As no better PSC datasets are available at the time of writing, it is valuable to curate a more realistic PSC dataset in the future.



(a) CC on dev

(b) IA on dev

(c) CC on future-dev

(d) IA on future-dev

(e) CC on test

(f) IA on test

Figure 5: Conv2Query 's learning curves on ProCIS with QF-DC, query–document relevance filtering (QF-D) and no filtering (Random), on the ProCIS dev, future-dev, test sets, under conversational contextualisation (CC) and interest anticipation (IA) settings. Each plot shows the retrieval quality (MRR@10) of RepLLaMA using ad-hoc queries generated by Conv2Query at different training steps.

References

[1] T Ahmed and S Bulathwela. 2022. Towards Proactive Information Retrieval in Noisy Text with Wikipedia Concepts. In *CEUR Workshop Proceedings*, Vol. 3318. 1–12.

[2] Salvatore Andolina, Valeria Orso, Hendrik Schneider, Khalil Klouche, Tuukka Ruotsalo, Luciano Gamberini, and Giulio Jacucci. 2018. Investigating Proactive Search Support in Conversations. In *DIS*. 1295–1307.

[3] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Xiaodong Liu Jianfeng Gao, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2016. MS MARCO: A Human Generated Machine Reading Comprehension Dataset. In *NIPS*.

[4] Vevake Balaraman and Bernardo Magnini. 2020. Proactive Systems and Influenceable Users: Simulating Proactivity in Task-oriented Dialogues. In *SEMDIAL*.

[5] Soyuj Basnet, Jerry Gou, Antonio Mallia, and Torsten Suel. 2024. DeeperImpact: Optimizing Sparse Learned Index Structures. *arXiv preprint arXiv:2405.17093* (2024).

[6] Paolo Boldi, Francesco Bonchi, Carlos Castillo, Debora Donato, Aristides Gionis, and Sebastiano Vigna. 2008. The Query-flow Graph: Model and Applications. In *CIKM*. 609–618.

[7] Francesco Bonchi, Raffaele Perego, Fabrizio Silvestri, Hossein Vahabi, and Rossano Venturini. 2012. Efficient Query Recommendations in the Long Tail via Center-Piece Subgraphs. In *SIGIR*. 345–354.

[8] Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. 2022. InPars: Unsupervised Dataset Generation for Information Retrieval. In *SIGIR*. 2387–2392.

[9] Yash Butala, Siddhant Garg, Pratyay Banerjee, and Amita Misra. 2024. ProMISe: A Proactive Multi-turn Dialogue Dataset for Information-seeking Intent Resolution. In *EACL*. 1774–1789.

[10] Ramraj Chandradevan, Kaustubh Dhole, and Eugene Agichtein. 2024. DUQGen: Effective Unsupervised Domain Adaptation of Neural Rankers by Diversifying Synthetic Query Generation. In *NAACL*, Kevin Duh, Helena Gomez, and Steven Bethard (Eds.).

[11] Wanyu Chen, Fei Cai, Honghui Chen, and Maarten de Rijke. 2018. Attention-based Hierarchical Neural Query Suggestion. In *SIGIR*. 1093–1096.

[12] Zhicong Cheng, Bin Gao, and Tie-Yan Liu. 2010. Actively Predicting Diverse Search Intent from User Browsing Behaviors. In *WWW*. 221–230.

[13] Zhuyun Dai, Vincent Y Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith Hall, and Ming-Wei Chang. 2023. Promptagator: Few-shot Dense Retrieval From 8 Examples. In *ICLR*.

[14] Mostafa Dehghani, Sascha Rothe, Enrique Alfonseca, and Pascal Fleury. 2017. Learning to Attend, Copy, and Generate for Session-Based Query Suggestion. In *CIKM*. 1747–1756.

[15] Yang Deng, Wenqiang Lei, Minlie Huang, and Tat-Seng Chua. 2023. Rethinking Conversational Agents in the Era of LLMs: Proactivity, Non-collaborativity, and Beyond. In *SIGIR*. 298–301.

[16] Yang Deng, Wenqiang Lei, Wai Lam, and Tat-Seng Chua. 2023. A Survey on Proactive Dialogue Systems: Problems, Methods, and Prospects. In *IJCAI*. 6583–6591.

[17] Yang Deng, Wenqiang Lei, Wenxuan Zhang, Wai Lam, and Tat-Seng Chua. 2022. PACIFIC: Towards Proactive Conversational Question Answering over Tabular and Textual Data in Finance. In *EMNLP*. 6970–6984.

[18] Yang Deng, Lizi Liao, Liang Chen, Hongru Wang, Wenqiang Lei, and Tat-Seng Chua. 2023. Prompting and Evaluating Large Language Models for Proactive Dialogues: Clarification, Target-guided, and Non-collaboration. In *EMNLP*. 10602–10621.

[19] Yang Deng, Lizi Liao, Zhonghua Zheng, Grace Hui Yang, and Tat-Seng Chua. 2024. Towards Human-centered Proactive Conversational Agents. In *SIGIR*. 807–818.

[20] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient Finetuning of Quantized LLMs. *arXiv preprint arXiv:2305.14314* (2023).

[21] Susan Dumais, Edward Cutrell, Raman Sarin, and Eric Horvitz. 2004. Implicit Queries (IQ) for Contextualized Search. In *SIGIR*. 594–594.

[22] Desmond Elliott and Joemon M Jose. 2009. A Proactive Personalised Retrieval System. In *CIKM*. 1935–1938.

[23] Henry Feild and James Allan. 2013. Task-Aware Query Recommendation. In *SIGIR*. 83–92.

[24] Stephen Fitchett and Andy Cockburn. 2012. AccessRank: Predicting What Users Will Do Next. In *SIGCHI*. 2239–2242.

[25] Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2022. From Distillation to Hard Negative Sampling: Making Sparse Neural IR Models More Effective. In *SIGIR*. 2353–2359.

[26] Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. SPLADE: Sparse Lexical and Expansion Model for First Stage Ranking. In *SIGIR*. 2288–2292.

[27] Debasis Ganguly, Dipasree Pal, Manisha Verma, and Procheta Sen. 2020. Overview of RCD-2020, the FIRE-2020 track on retrieval from conversational dialogues. In *FIRE*. 33–36.

[28] Mitko Gospodinov, Sean MacAvaney, and Craig Macdonald. 2023. Doc2Query–: When Less is More. In *ECIR*. Springer, 414–422.

[29] Nam Le Hai, Thomas Gerald, Thibault Formal, Jian-Yun Nie, Benjamin Piwowarski, and Laure Soulier. 2023. CoSPLADE: Contextualizing SPLADE for Conversational Information Retrieval. In *ECIR*.

[30] Monika Henzinger, Bay-Wei Chang, Brian Milch, and Sergey Brin. 2003. Query-Free News Search. In *WWW*. 1–10.

[31] Yunah Jang, Kang-il Lee, Hyunkyung Bae, Seungpil Won, Hwanhee Lee, and Kyomin Jung. 2024. IterCQR: Iterative Conversational Query Reformulation with Retrieval Guidance. In *EMNLP*. 8121–813.

[32] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated Gain-Based Evaluation of IR Techniques. *TOIS* 20, 4 (2002), 422–446.

[33] Vitor Jeronymo, Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, Roberto Lotufo, Jakub Zavrel, and Rodrigo Nogueira. 2023. InPars-v2: Large Language Models as Efficient Dataset Generators for Information Retrieval. *arXiv preprint arXiv:2301.01820* (2023).

[34] Gareth JF Jones, Procheta Sen, Debasis Ganguly, and Emine Yilmaz. 2022. Workshop on Proactive and Agent-Supported Information Retrieval (PASIR). In *CIKM*. 5167–5168.

[35] Weize Kong, Rui Li, Jie Luo, Aston Zhang, Yi Chang, and James Allan. 2015. Predicting Search Intent Based on Pre-Search Context. In *SIGIR*. 503–512.

[36] Markus Koskela, Petri Luukkonen, Tuukka Ruotsalo, Mats Sjöberg, and Patrik Floréen. 2018. Proactive Information Retrieval by Capturing Search Intent from Primary Task Context. *TiiS* 8, 3 (2018), 1–25.

[37] Yilong Lai, Jialong Wu, Congzhi Zhang, Haowen Sun, and Deyu Zhou. 2025. AdaCQR: Enhancing Query Reformulation for Conversational Search via Sparse and Dense Retrieval Alignment. *COLING* (2025).

[38] Jing Yang Lee, Seokhwan Kim, Kartik Mehta, Jiun-Yu Kao, Yu-Hsiang Lin, and Arpit Gupta. 2024. Redefining Proactivity for Information Seeking Dialogue. *arXiv preprint arXiv:2410.15297* (2024).

[39] Ruirui Li, Ben Kao, Bin Bi, Reynold Cheng, and Eric Lo. 2012. DQR: A Probabilistic Approach to Diversified Query Recommendation. In *CIKM*. 16–25.

[40] Lizi Liao, Grace Hui Yang, and Chirag Shah. 2023. Proactive Conversational Agents. In *WSDM*. 1244–1247.

[41] Lizi Liao, Grace Hui Yang, and Chirag Shah. 2023. Proactive Conversational Agents in the Post-ChatGPT World. In *SIGIR*. 3452–3455.

[42] Daniel J Liebling, Paul N Bennett, and Ryen W White. 2012. Anticipatory Search: Using Context to Initiate Search. In *SIGIR*. 1035–1036.

[43] Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. 2021. Contextualized Query Embeddings for Conversational Search. In *EMNLP*. 1004–1015.

[44] Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. 2021. In-Batch Negatives for Knowledge Distillation with Tightly-Coupled Teachers for Dense Retrieval. In *RepL4NLP-2021*. 163–173.

[45] Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. 2024. Fine-Tuning LLaMA for Multi-Stage Text Retrieval. In *SIGIR*. 2421–2425.

[46] Kelong Mao, Chenlong Deng, Haonan Chen, Fengran Mo, Zheng Liu, Tetsuya Sakai, and Zhicheng Dou. 2024. ChatRetriever: Adapting Large Language Models for Generalized and Robust Conversational Dense Retrieval. In *EMNLP*. 1227–1240.

[47] Kelong Mao, Zhicheng Dou, Bang Liu, Hongjin Qian, Fengran Mo, Xiangli Wu, Xiaohua Cheng, and Zhao Cao. 2023. Search-oriented conversational query editing. In *Findings of ACL*. 4160–4172.

[48] Kelong Mao, Zhicheng Dou, Fengran Mo, Jiewen Hou, Haonan Chen, and Hongjin Qian. 2023. Large Language Models Know Your Contextual Search Intent: A Prompting Framework for Conversational Search. In *EMNLP*. 1211–1225.

[49] Kelong Mao, Zhicheng Dou, and Hongjin Qian. 2022. Curriculum Contrastive Context Denoising for Few-shot Conversational Dense Retrieval. In *SIGIR*. 176–186.

[50] Kelong Mao, Hongjin Qian, Fengran Mo, Zhicheng Dou, Bang Liu, Xiaohua Cheng, and Zhao Cao. 2023. Learning denoised and interpretable session representation for conversational search. In *WWW*. 3193–3202.

[51] Kshitij Mishra, Azlaan Mustafa Samad, Palak Totala, and Asif Ekbal. 2022. PEPDS: A Polite and Empathetic Persuasive Dialogue System for Charity Donation. In *COLING*. 424–440.

[52] Fengran Mo, Abbas Ghaddar, Kelong Mao, Mehdi Rezagholizadeh, Boxing Chen, Qun Liu, and Jian-Yun Nie. 2024. CHIQ: Contextual History Enhancement for Improving Query Rewriting in Conversational Search. In *EMNLP*. 2253–2268.

[53] Fengran Mo, Kelong Mao, Ziliang Zhao, Hongjin Qian, Haonan Chen, Yiruo Cheng, Xiaoxi Li, Yutao Zhu, Zhicheng Dou, and Jian-Yun Nie. 2024. A Survey of Conversational Search. *arXiv preprint arXiv:2410.15576* (2024).

[54] Fengran Mo, Kelong Mao, Yutao Zhu, Yihong Wu, Kaiyu Huang, and Jian-Yun Nie. 2023. ConvGQR: Generative Query Reformulation for Conversational Search. In *ACL*. 4998–5012.

[55] Fengran Mo, Jian-Yun Nie, Kaiyu Huang, Kelong Mao, Yutao Zhu, Peng Li, and Yang Liu. 2023. Learning to relate to previous turns in conversational search. In *SIGKDD*. 1722–1732.

[56] Fengran Mo, Chen Qu, Kelong Mao, Yihong Wu, Zhan Su, Kaiyu Huang, and Jian-Yun Nie. 2024. Aligning query representation with rewritten query and relevance judgments in conversational search. In *CIKM*. 1700–1710.

[57] Fengran Mo, Chen Qu, Kelong Mao, Tianyu Zhu, Zhan Su, Kaiyu Huang, and Jian-Yun Nie. 2024. History-Aware Conversational Dense Retrieval. *Findings of ACL* (2024).

[58] Cristina Ioana Muntean, Franco Maria Nardini, Fabrizio Silvestri, and Marcin Sydow. 2013. Learning to Shorten Query Sessions. In *WWW*. 131–132.

[59] Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. Document Ranking with a Pretrained Sequence-to-Sequence Model. In *EMNLP*. 708–718.

[60] Rodrigo Nogueira and Jimmy Lin. 2019. From doc2query to docTTTTTquery. (2019).

[61] Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document Expansion by Query Prediction. *arXiv preprint arXiv:1904.08375* (2019).

[62] Dipasree Pal and Debasis Ganguly. 2021. Effective Query Formulation in Conversation Contextualization: A Query Specificity-based Approach. In *ICTIR*. 177–183.

[63] Dae Hoon Park, Yi Fang, Mengwen Liu, and ChengXiang Zhai. 2016. Mobile App Retrieval for Social Media Users via Inference of Implicit Intent in Social Media Text. In *CIKM*. 959–968.

[64] Chen Qu, Liu Yang, Cen Chen, Minghui Qiu, W Bruce Croft, and Mohit Iyyer. 2020. Open-retrieval Conversational Question Answering. In *SIGIR*. 539–548.

[65] David Rau and Jaap Kamps. 2022. The Role of Complex NLP in Transformers for Text Ranking. In *ICTIR*. 153–160.

[66] Bradley James Rhodes and Pattie Maes. 2000. Just-In-Time Information Retrieval. *IBM Systems journal* 39, 3.4 (2000), 685–704.

[67] Bradley J Rhodes and Thad Starner. 1996. Remembrance Agent: A Continuously Running Automated Information Retrieval System. In *PAAM*, Vol. 96. 487–495.

[68] Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at TREC-3. *Nist Special Publication Sp* 109 (1995), 109.

[69] Kevin Ros, Matthew Jin, Jacob Levine, and ChengXiang Zhai. 2023. Retrieving Webpages Using Online Discussions. In *ICTIR*. 159–168.

[70] Corbin Rosset, Chenyan Xiong, Xia Song, Daniel Campos, Nick Craswell, Saurabh Tiwary, and Paul Bennett. 2020. Leading Conversational Search by Suggesting Useful Questions. In *The web conference*. 1160–1170.

[71] Chris Samarinas and Hamed Zamani. 2024. ProCIS: A Benchmark for Proactive Retrieval in Conversations. In *SIGIR*. 830–840.

[72] Procheta Sen, Debasis Ganguly, and Gareth Jones. 2018. Procrastination is the Thief of Time: Evaluating the Effectiveness of Proactive Search Systems. In *SIGIR*. 1157–1160.

[73] Procheta Sen, Debasis Ganguly, and Gareth JF Jones. 2021. I Know What You Need: Investigating Document Retrieval Effectiveness with Partial Session Contexts. *TOIS* 40, 3 (2021), 1–30.

[74] Haizhou Shi, Zihao Xu, Hengyi Wang, Weiyi Qin, Wenyuan Wang, Yibin Wang, Zifeng Wang, Sayna Ebrahimi, and Hao Wang. 2024. Continual Learning of Large Language Models: A Comprehensive Survey. *arXiv preprint arXiv:2404.16789* (2024).

[75] Milad Shokouhi and Qi Guo. 2015. From Queries to Cards: Re-ranking Proactive Card Recommendations Based on Reactive Search History. In *SIGIR*. 695–704.

[76] Anna Shtok, Oren Kurland, David Carmel, Fiana Raiber, and Gad Markovits. 2012. Predicting Query Performance by Query-Drift Estimation. *TOIS* 30, 2 (2012), 1–35.

[77] Yang Song and Qi Guo. 2016. Query-Less: Predicting Task Repetition for NextGen Proactive Search and Recommendation Engines. In *WWW*. 543–553.

[78] 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 *CIKM*. 553–562.

[79] Yueming Sun and Yi Zhang. 2018. Conversational Recommender System. In *SIGIR*. 235–244.

[80] Anuja Tayal and Aman Tyagi. 2024. Dynamic Contexts for Generating Suggestion Questions in RAG Based Conversational Systems. In *WWW*. 1338–1341.

[81] Tuan A Tran, Sven Schwarz, Claudia Niederée, Heiko Maus, and Nattiya Kanhabua. 2016. The Forgotten Needle in My Collections: Task-Aware Ranking of Documents in Semantic Information Space. In *CHIIR*. 13–22.

[82] Tung Vuong, Giulio Jacucci, and Tuukka Ruotsalo. 2017. Proactive Information Retrieval via Screen Surveillance. In *SIGIR*. 1313–1316.

[83] Tung Vuong, Giulio Jacucci, and Tuukka Ruotsalo. 2017. Watching inside the Screen: Digital Activity Monitoring for Task Recognition and Proactive Information Retrieval. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 3 (2017), 1–23.

[84] Jian Wang, Yi Cheng, Dongding Lin, Chak Leong, and Wenjie Li. 2023. Target-oriented Proactive Dialogue Systems with Personalization: Problem Formulation and Dataset Curation. In *EMNLP*. 1132–1143.

[85] Bin Wu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. 2018. Query Suggestion with Feedback Memory Network. In *Proceedings of the 2018 World Wide Web Conference*. 1563–1571.

[86] Yun Xing, Jian Kang, Aoran Xiao, Jiahao Nie, Ling Shao, and Shijian Lu. 2024. Rewrite Caption Semantics: Bridging Semantic Gaps for Language-Supervised Semantic Segmentation. *NeurIPS* 36 (2024).

[87] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. In *ICLR*.

[88] Rui Yan and Dongyan Zhao. 2018. Smarter Response with Proactive Suggestion: A New Generative Neural Conversation Paradigm.. In *IJCAI*. 4525–4531.

[89] Liu Yang, Qi Guo, Yang Song, Sha Meng, Milad Shokouhi, Kieran McDonald, and W Bruce Croft. 2016. Modeling User Interests for Zero-Query Ranking. In *ECIR*. Springer, 171–184.

[90] Liu Yang, Hamed Zamani, Yongfeng Zhang, Jiafeng Guo, and W Bruce Croft. 2017. Neural Matching Models for Question Retrieval and Next Question Prediction in Conversation. *arXiv preprint arXiv:1707.05409* (2017).

[91] Fanghua Ye, Meng Fang, Shenghui Li, and Emine Yilmaz. 2023. Enhancing Conversational Search: Large Language Model-Aided Informative Query Rewriting. In *EMNLP*. 5985–6006.

[92] Chanwoong Yoon, Gangwoo Kim, Byeongguk Jeon, Sungdong Kim, Yohan Jo, and Jaewoo Kang. 2024. Ask Optimal Questions: Aligning Large Language Models with Retriever's Preference in Conversational Search. *arXiv preprint arXiv:2402.11827* (2024).

[93] Shi Yu, Jiahua Liu, Jingqin Yang, Chenyan Xiong, Paul Bennett, Jianfeng Gao, and Zhiyuan Liu. 2020. Few-shot Generative Conversational Query Rewriting. In *SIGIR*. 1933–1936.

[94] Shi Yu, Zhenghao Liu, Chenyan Xiong, Tao Feng, and Zhiyuan Liu. 2021. Few-shot Conversational Dense Retrieval. In *SIGIR*. 829–838.

[95] Hansi Zeng, Chen Luo, Bowen Jin, Sheikh Muhammad Sarwar, Tianxin Wei, and Hamed Zamani. 2024. Scalable and Effective Generative Information Retrieval. In *WWW*. 1441–1452.

[96] Chengxiang Zhai and John Lafferty. 2001. A Study of Smoothing Methods for Language Models Applied to Ad Hoc Information Retrieval. In *SIGIR*. 334–342.

[97] Wen Zhang, Lingfei Deng, Lei Zhang, and Dongrui Wu. 2022. A Survey on Negative Transfer. *IEEE/CAA Journal of Automatica Sinica* 10, 2 (2022), 305–329.

[98] Xuan Zhang, Yang Deng, Zifeng Ren, See-Kiong Ng, and Tat-Seng Chua. 2024. Ask-before-Plan: Proactive Language Agents for Real-World Planning. *arXiv preprint arXiv:2406.12639* (2024).

[99] Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and W Bruce Croft. 2018. Towards Conversational Search and Recommendation. In *CIKM*. 177–186.

[100] Yutao Zhu, Jian-Yun Nie, Kun Zhou, Pan Du, Hao Jiang, and Zhicheng Dou. 2021. Proactive Retrieval-based Chatbots based on Relevant Knowledge and Goals. In *SIGIR*. 2000–2004.

[101] Shengyao Zhuang, Houxing Ren, Linjun Shou, Jian Pei, Ming Gong, Guido Zuccon, and Daxin Jiang. 2022. Bridging the Gap Between Indexing and Retrieval for Differentiable Search Index with Query Generation. *arXiv preprint arXiv:2206.10128* (2022).

[102] Shengyao Zhuang and Guido Zuccon. 2021. Dealing with Typos for BERT-based Passage Retrieval and Ranking. In *EMNLP*. 2836–2842.