Spoken Conversational Context Improves Query Auto-completion in Web Search

TUNG VUONG, University of Helsinki, Finland
SALVATORE ANDOLINA, University of Palermo, Italy and University of Helsinki, Finland
GIULIO JACUCCI, University of Helsinki, Finland
TUUKKA RUOTSALO, University of Helsinki, Finland and University of Copenhagen, Denmark

Web searches often originate from conversations in which people engage before they perform a search. Therefore, conversations can be a valuable source of context with which to support the search process. We investigate whether spoken input from conversations can be used as a context to improve query auto-completion. We model the temporal dynamics of the spoken conversational context preceding queries and use these models to re-rank the query auto-completion suggestions. Data were collected from a controlled experiment and comprised conversations among 12 participant pairs conversing about movies or traveling. Search query logs during the conversations were recorded and temporally associated with the conversations. We compared the effects of spoken conversational input in four conditions: a control condition without contextualization; an experimental condition with the model using search query logs; an experimental condition with the model using spoken conversational input; and an experimental condition with the model using both search query logs and spoken conversational input. We show the advantage of combining the spoken conversational context with the Web-search context for improved retrieval performance. Our results suggest that spoken conversations provide a rich context for supporting information searches beyond current user-modeling approaches.

CCS Concepts: • **Information systems → Information retrieval query processing**; • **Human-centered computing → Sound-based input / output**;

Additional Key Words and Phrases: QAC, query auto-completion, speech input, voice, background speech

ACM Reference format:
Tung Vuong, Salvatore Andolina, Giulio Jacucci, and Tuukka Ruotsalo. 2021. Spoken Conversational Context Improves Query Auto-completion in Web Search. *ACM Trans. Inf. Syst.* 39, 3, Article 31 (May 2021), 32 pages.
https://doi.org/10.1145/3447875

This research was funded by the project COADAPT (Human and Work Station Adaptation Support to aging citizens, Grant Agreement No. 826266) and the project PON AIM (ID No. AIM1875400-1, CUP No. B74I18000210006), and was partially supported by the Academy of Finland (Flagship programme: Finnish Center for Artificial Intelligence FCAI and decision numbers: 322653, 328875, 336085).

Authors' addresses: T. Vuong and G. Jacucci, University of Helsinki, Pietari Kalmin katu 5 00560 Helsinki, Finland; emails: vuong@cs.helsinki.fi, giulio.jacucci@helsinki.fi; S. Andolina, University of Palermo, Via Archirafi 34 90123 Palermo, Italy and University of Helsinki, Pietari Kalmin katu 5 00560 Helsinki, Finland; email: salvatore.andolina@unipa.it; T. Ruotsalo, University of Helsinki, Pietari Kalmin katu 5 00560 Helsinki, Finland, University of Copenhagen, Universitetsparken 1 2100 Copenhagen, Denmark; email: tuukka.ruotsalo@helsinki.fi.

1046-8188/2021/05-ART31 $15.00
https://doi.org/10.1145/3447875

1 INTRODUCTION

Web searches are becoming an integral element of everyday conversations, whether in meetings, collaborative activity planning, or shopping. The emergence of practical approaches to interpreting spoken and written natural language and to supporting collaboration among people searching for information together has enabled systems that can assist people in their everyday information-seeking activities [33]. The ability to conduct searches quickly during conversational exchanges between people can enrich a conversation with new facts, facilitating common ground and the reinforcement of mutual beliefs. However, current search systems are not tailored to support the conversational process as such but require full human control in formulating and inputting queries when new information is needed [30]. **Query auto-completion (QAC)** can address this challenge by predicting the intended query as the user types, thus allowing users to concentrate on the conversation [7, 52]. QAC approaches are typically based on search logs [17]. Given a query prefix, traditional QAC models first select a list of queries that match the leading characters from the logs as potential candidates for the intended query. This list is then ranked based on popularity or personalization models. However, because the prefix is often short and ambiguous, immediate pre-search context can be sparse, and the number of candidates may be large, such an approach makes contextual queries hard to predict.

A context-aware approach has been proven to be more effective for boosting QAC performance. Researchers have studied several context sources in the literature, including prior queries made by the user [7], the location and application context [41, 79], profile context information such as age and gender [74], and interactions with graphical user interfaces that visualize the search space [67, 68]. However, using spoken context for a QAC to support search during conversations has received less attention. Research shows that conversations are valuable sources of information needs and, in turn, engage people in search activities [12, 73]. Searching at a particular point in a conversation is not a random occasion but is dependent on a conversational topic [15]. Specific query terms are grounded in the cognitive processes related to information processing and are associated with the information needs of the discourse [42, 43]. Search queries potentially can be any spoken word that is recorded [2], and this source of information be can useful to mine the context of search [73]. Consider, for example (Figure 1), two people having a conversation on movies who need to recall related information and perform searches. Such a conversational context can be leveraged to optimize query suggestions. Yet, QAC systems fail to utilize conversational information and are limited to more conventional human-computer interaction contexts, such as clicks, typed queries, or page visits occurring prior to searching [46]. Such information is highly useful but fails to capture the more comprehensive context of the user. Therefore, the main goal of the present research is to study the benefit of considering spoken context for query prediction. We compare different types of speech-to-text conversion to capture spoken context: automatic transcription and ideal transcription.

To investigate whether spoken input from conversations between people can be used as a context to improve QAC, we model the temporal dynamics of the spoken conversational context preceding the queries and use these models to re-rank real-world QAC suggestions. Subsequently, we simulate the ideal transcription by using human annotators to manually translate the speech. Accordingly, we analyze the contextual QAC performance with regard to different speech recognizers (state-of-the-art and ideal). To these ends, we aim to answer the following research questions:

- Does the use of spoken conversation as a context improve the ranking of query suggestions?
- Does the spoken context help to reduce user effort in typing queries?
- How does the accuracy of speech recognition affect the ranking of query suggestions?

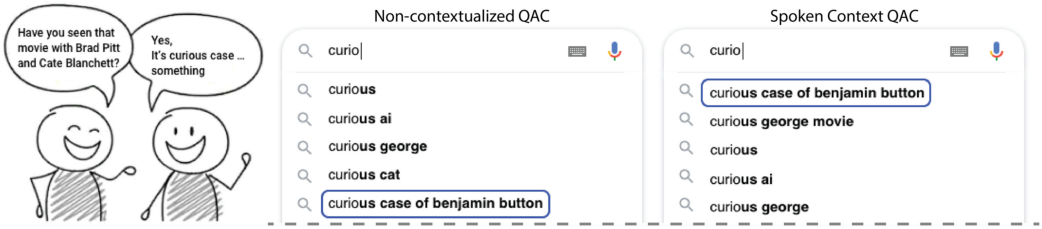


Fig. 1. Example of Query Auto-completion (QAC) for searches conducted during spoken conversations between people. Given a few characters typed in a search box, conventional QAC retrieves a ranked list of suggested queries, in which the user's intended query may not be ranked high enough to be visible for the user. Instead, a QAC method leveraging the spoken context preceding the search allows for predicting user intentions and ranking the intended query at the top of the suggestion list with less user effort.

To answer these research questions, we conducted a controlled task-based information-seeking experiment in which 12 pairs of participants had conversations about movies or travel lists and supported that conversation by performing Web searches. The conversations were both automatically and manually transcribed into textual transcripts, and the queries that the participants inputted into the search interfaces during their conversations were collected. The Google QAC suggestions service was used as a source of initial query suggestion ranking. In the experiments, we built topic models for re-ranking query suggestions. To understand whether spoken information is useful for improving query predictions, the Google QAC service was used as a control condition, and we manipulated the context source leveraged to construct the model for experimental conditions. The source determined the information used for training the model, consisting of (1) transcripts of spoken conversations, (2) search history, and (3) combined information from both spoken conversations and search history.

The results show that re-ranking with the conversational context significantly outperforms QAC without contextualization. When combining spoken conversations with search history, these context sources complement each other and further improve the performance of QAC. Our findings suggest that spoken conversations provide a rich context for supporting information searches beyond current user-modeling approaches. Unleashing sources of contextual information from the users' activity shows a high potential for personalization and ubiquitous user modeling.

The rest of the article is organized as follows. Section 2 provides some background with which to position our work in the context of current literature. Section 3 describes the data-collection experiment used to build our unique dataset consisting of transcripts and query logs. Section 4 introduces our context-modeling method for QAC. Section 5 describes the evaluation of context QAC relative to the control condition without contextualization. In Section 6, we present the results of the evaluation. We conclude by providing our discussion and conclusions in Sections 7 and 8. The structure of the article is reflected in Figure 2, which illustrates the overall procedure followed in this work.

2 BACKGROUND

2.1 Speech Input in Web Search

In recent years, advances in automatic speech recognition have led to a returning interest in speech-based systems [59]. Early works discussed the interleaving of automatic speech recognition with information-retrieval systems for query modeling [56]. Many follow-up studies focused on improving various spoken dialogue systems but mainly addressed the challenges of speech recognition itself such as vocabulary matching and detection, and handling background noise [44, 70].

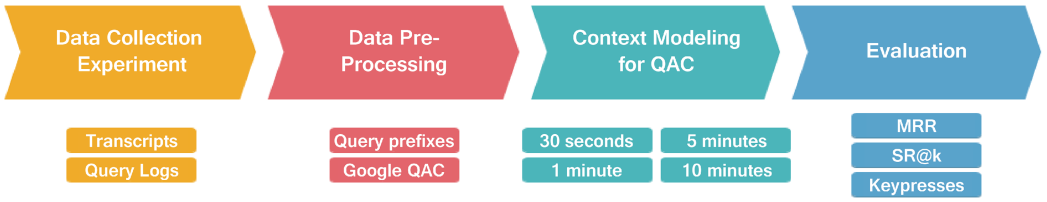


Fig. 2. The overall procedure consisted of four phases: the data-collection experiment for collecting transcripts and query logs; data processing, including extracting prefixes and generating QAC suggestions; context modeling for re-ranking the query suggestions originating from Google's QAC service; and evaluating our QAC approach using three commonly used metrics (MRR, SR@k, and average number of key presses).

Generally, the underlying technology in these systems allows users to input queries into retrieval systems with a spoken input [29].

Another more recent line of research promotes the use of spoken conversational input to enable natural language communication with intelligent personal assistants [44]. Commercial examples of such agents include *Apple's Siri*, *Microsoft's Cortana*, *Google Assistant*, and *Amazon Alexa*. They are typically available on people's personal devices or via smart speakers. In addition to voice system responses, intelligent assistants provide users with a diverse set of services, ranging from Web-search results [28] and direct answers to questions [65] to proactive recommendations [75]. The main difference between traditional conversational search and intelligent assistants is the conversational nature of the interaction. In the conversation mode with an intelligent assistant, the technology can refer to users' previous interactions and requests to understand a conversation's context [22]. However, intelligent assistants only receive voice commands from an individual, whereas in the present study we use rich context from conversations between people and build a model to predict their intended queries based on the topic of the entire conversation.

Spoken conversational input also is increasingly being studied as a part of more traditional information retrieval methods. This is the case, for example, of conversational recommendation systems [6, 19, 61], which elicit spoken input from users to learn their preferences, thus addressing the cold-start problem. Another area is conversational search [64, 78], in which search queries are formulated through natural spoken language. These systems are characterized by a dialogue mode of interaction in which the conversational context is used as explicit input [54].

A less investigated area is the use of conversational context as implicit input in Web-search tasks. McMillan et al. [53] suggested that spoken conversational context, in the form of a continuous speech stream, could be used to identify users' next actions such as searches. Oard [60]'s work encouraged searchers to speak at length about what they are looking for. However, such an approach could not mitigate the challenge of properly matching verbose multi-term queries. Andolina et al. [2, 3] used spoken conversational context to perform proactive Web searches that could be useful during informal conversations. Similarly to prior work, we also use spoken conversational context as implicit input to improve the Web-search process, but instead of using the context to perform searches, we use it to improve the ranking of relevant candidate queries in QAC. That is, we do not rely on explicit queries prompted by the user, but use the conversational context to predict the exact query from a set of query candidates.

Studies in conversational search are also conducted in settings where people engage in conversations with each other and not just with intelligent agents. A recently emergent area of multimodal conversational systems fuses various kinds of user inputs from interactive communications between people, including speech, hand gestures, and explicit interactions such as clicks, to understand their search intent [40, 68, 80]. Empirical evidence has revealed that user effort in finding

relevant information during collaborative search conversations can be reduced by adapting speech recognition to additional contexts [34]. However, the majority of previous work has focused on reducing the word-error rate in speech recognition but has not utilized the spoken context to optimize search performance. Instead of addressing speech-recognition errors, Shiga et al. [73] modeled users' information needs based on spoken conversations. Utterances were manually annotated, and those followed by search activity were classified as information requests and used to train a predictive model [63]. Although the model accurately detected when users were searching, it was not successful in predicting the actual query or supporting users in writing the query.

Another body of research involves the problem of spoken query reformulation, showing that users often respond to automatic speech-recognition errors by repeating the query [36, 37, 66, 69]. Consequently, researchers have built classifiers to categorize and predict the reformulation patterns in real time [37]. Immediate corrective actions such as re-ranking transcriptions can be carried out to avoid the same error occurring repeatedly [31]. Researchers have also found that users tend to switch from voice to typing to refine queries [76]. Unlike these approaches, our study does not focus on query reformulation, but we investigate the use of spoken conversational inputs as a search context to improve the ranking of query suggestions in QAC and, in turn, assist with formulating the query more rapidly.

2.2 Context-aware QAC

QAC has been widely adopted by Internet browsers, development environments, websites, desktop searches, operating systems, databases, email clients, and search engines [17]. One of the first examples is Google Suggest,[1] a service launched in 2004, which provides users with query completions in real time and shows those completions below the search box as a drop-down menu while the users type [11]. The main objective of QAC is to support the rapid formulation and refinement of a Web-search query [52]. Query candidates are matched against the prefix on the fly using a variety of information-retrieval and natural-language-processing techniques [27, 58].

Early research in this area has focused on the use of predictive models. For example, Grabski and Scheffer [27] proposed an index-based retrieval algorithm and a cluster-based approach and presented users with a complete query given an initial prefix. A similar approach involved learning a linearly interpolated n-gram model to support users in completing a sentence in natural language [14]. While these previous studies inspired our work, they focused on predicting the query as a complete sentence for document retrieval. By contrast, the problem we discuss here is how to re-rank a set of Web-search query suggestions given by a real-world search engine.

To this end, Fan et al. [24] proposed a generative model that learns topics from relevant documents based on Latent Dirichlet Allocation, which they used to generate terms that are topically coherent to a query. However, the model only focused on term-by-term suggestions, as oppose to predicting the complete query. Chaudhuri and Kaushik [20] captured input-typing errors by calculating the edit distance and proposed an error-tolerance QAC model that suggests the correct completion even when a user mistypes a query. Marchionini and White [52] investigated how suggesting query words as the user enters a query affects the query formulation. An analysis of query quality showed that offering query completion improved the quality of initial queries, making it potentially useful when initiating a search, when searchers may be in most need of support [52]. Another example is the work done by Bhatia et al. [13], in which frequently occurring phrases and n-grams from text collections were used to generate and rank auto-completion candidates for partial queries in the absence of search logs.

[1] https://googleblog.blogspot.com/2004/12/ive-got-suggestion.html.

Several QAC approaches have been proposed to extend these underlying approaches in various ways. The most common approaches are based on search logs and consist of three main phases [48]:

(1) Inspect the search log to retrieve a set of candidate queries that match the input prefix provided by the user;
(2) Rank those candidates by their frequency, either in conjunction with Step 1 or by applying a separate ranking operation; and
(3) Optionally, re-rank an initial subset of the sorted candidate list, based on a second more complex ranking criterion such as, for example, predicted popularity, search context, or personalization concerns.

In the first and second phases, a list of candidate queries matching each prefix is generated in advance and stored in efficient data structures such as prefix trees for fast look-ups [10, 11, 20, 35]. Although this approach is very effective and can handle large-scale datasets, the suggestions are often considered to be the same for all users. Hence, for a given prefix, all users are presented with the same set of suggestions. By contrast, in the present work, we focus on the third phase by investigating how the spoken conversational context can be used to improve traditional QAC methods.

For a given prefix and its context, context-aware methods focus on re-ranking the candidate queries according to the dependency between the candidate and the given context [8, 38]. Previous research in this area has explored various search-context sources. Bar-Yossef and Kraus [7] considered the user's recent queries as context and took into account the similarity of QAC candidates within this context for ranking. Cai and de Rijke [16] proposed a model for selectively personalizing query auto-completion by encoding the ranking signal as a trade-off between query popularity and the search context. At run time, these models are used to predict a user's intended query from prior queries when a short input prefix is provided. However, it is unclear whether they can deal with contexts that have never occurred in search engine logs. In comparison, our model is adapted to more of a real-world context. It can capture an intent of the user query that has already been presented in many casual conversations between people, but it does not require any prior interaction with the search engine itself.

Of many other possible sources of context, location information and personal profile also have been used widely, as they are readily available to search engines. For example, Kamvar and Baluja [41] used several contextual signals, such as location, time, and day of week, to improve QAC on mobile devices. Shokouhi [74] compared the effectiveness of various user-specific and demographic contextual features such as users' age, gender, location, and longer search history. They showed that certain demographic features such as location are more effective than others and that adding more context based on users' demographics and search history leads to further effectiveness improvements [74]. While the idea is very intuitive and a majority of scenarios have used contextual information to disambiguate user intent [18] and provide query recommendations [25], to our knowledge, no published work has utilized the spoken conversational context for query-completion applications.

Our work contributes to this research area by investigating the use of a rich source of relevant presearch context that has received little attention in prior work: the spoken conversational context. As major Web-search engine providers have made major advances in this area, as demonstrated by prior work [7, 41, 74], we chose to utilize the QAC method used by Google, which we accessed through publicly available APIs. Google QAC is strong and as competitive as any other QAC-modeling approach. Therefore, we opted for Google QAC, which outputs the plausible and practically accepted query suggestions. In the experiments, Google QAC without context information was utilized as a control condition. To study the effect of spoken context on query prediction,

we incorporated different contextual signals such as spoken conversation, search history, and both spoken conversation and search history into the prediction model as the experimental conditions.

3 DATA-COLLECTION EXPERIMENT

A controlled laboratory experiment was designed to build the dataset needed for our research. While informal conversations may occur in any place, executing an experiment in a natural environment would have been difficult, led to noisy and potentially error-prone speech input, and influenced the type and quality of the spoken conversational context preceding Web searches with a number of uncontrollable factors (e.g., ambient noise and incidental conversations). Therefore, we chose to limit the potential confounding factors by designing a data-collection experiment conducted in a controlled laboratory setting. The research followed the ethical guidelines and was approved by the Ethical Committee of the University of Helsinki.[2]

3.1 Apparatus

For the present experiment, each participant used a MacBook Pro 15-inch laptop connected to a Samson Meteor microphone. The experimental session was video-recorded using a Panasonic camcorder. Additionally, the laptop screen was recorded using the Screencast-O-Matic software, which also recorded the participants' faces with the webcam embedded in the laptop.

3.2 Participants

We recruited the participants by sending out invitations to the University of Helsinki's mailing lists. The eligibility criteria for taking part in the experiment were having a higher education background and high English proficiency (IELTS = 6.5 and above). It was assumed that people satisfying these criteria would be more likely to have good communication skills and sufficient fluency in the English language.

A total of 24 participants (12 pairs) took part in the present study. The participants included 12 men and 12 women, with an average age of 27 years (SD = 3.87). The participants were randomly assigned into groups, each consisting of two participants. Of the participants, 12 were undergraduate students, five were doctoral students, three were research assistants, three were post-doc researchers, and one was a nurse. Eleven participants reported having previous experience with conversational agents, and all of them reported rare usage of them. Each participant received two movie tickets worth around 20 euros as compensation for participating in the experiment.

3.3 Setting

The experiment took place in a laboratory. We set up the room to resemble a comfortable and informal environment where the participants could feel at ease. They sat at a table across from each other. Each participant had a laptop connected to a tabletop microphone in front of him/her (Figure 3). The microphones were placed to pick up all voice and capture the whole conversational context. Therefore, speaker detection was excluded as a factor of the experiments.

3.4 Task

The participants were asked to converse with the other participant in the group on two topics: a list of movies that they planned to watch or a list of places that they wanted to visit. The tasks were assigned to the groups in a counterbalanced order. The task assigned was not meant to generate a specific outcome; rather, it was intended to provide only a general shape for the conversations.

[2] https://www.helsinki.fi/en/research/ethical-review-board-in-the-humanities-and-social-and-behavioural-sciences.

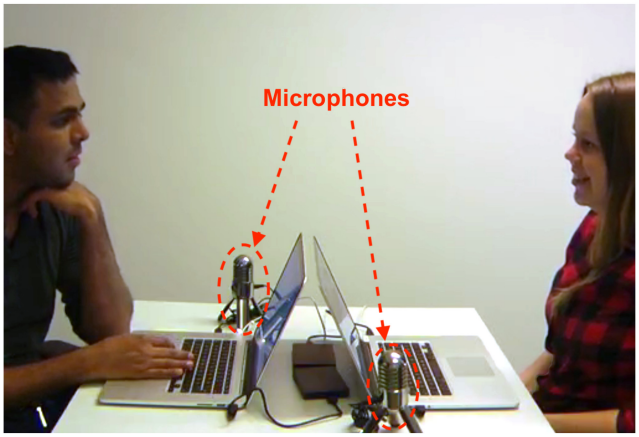


Fig. 3. Experimental setup. The participants sat at a table, and a laptop was placed in front of each participant. The laptops displayed the search interface. Microphones were placed on the table to record the participants' conversation.

More specifically, we asked the participants to share their experiences regarding movies or travels that had impressed them and to get inspiration from the other participant's utterances.

3.5 Procedure

First, the experimenter welcomed the participants and introduced them to the experiment's main goals and overall procedure. Afterward, the participants signed informed consent forms. During the experiment, the experimenter simply described the task and then left the room to allow the participants to talk freely. The experimenter followed the experiment through a video connection and was reachable in case the participants needed assistance. Each conversation session lasted 20 min, after which the experimenter returned to the laboratory to end the session. The participants were not forced to perform Web searches during the conversation, but they were allowed to search freely for additional information according to their needs. The only service the participants were allowed to use was our search engine, to ensure that all search inputs would be captured. To control the search engine, disable personalization, and parameterize the search result output, the search interface used in the study was an instance of Google Custom Search.

3.6 Transcript and Web-search Logging

We used two transcription methods: automatic and ideal. The automatic transcription was conducted using an automatic speech-recognition service. The ideal transcription was manually conducted by a professional transcription service. Figure 4 illustrates a snippet of a conversation in which speakers' utterances were transcribed by the two transcription methods. Web-search logs were also collected and temporally associated with the conversations.

3.6.1 Automatic Transcription. An automatic speech-to-text system continuously recorded conversations through two microphones, one for each participant. Speech recognition was performed using Google's implementation of the HTML5 Web Speech API.[3] The speech API takes an audio recording as a voice input and outputs a transcript in natural language. The speech recognizer continuously recorded each conversation and directly transcribed the audio input when

[3]https://www.google.com/intl/en/chrome/demos/speech.html.

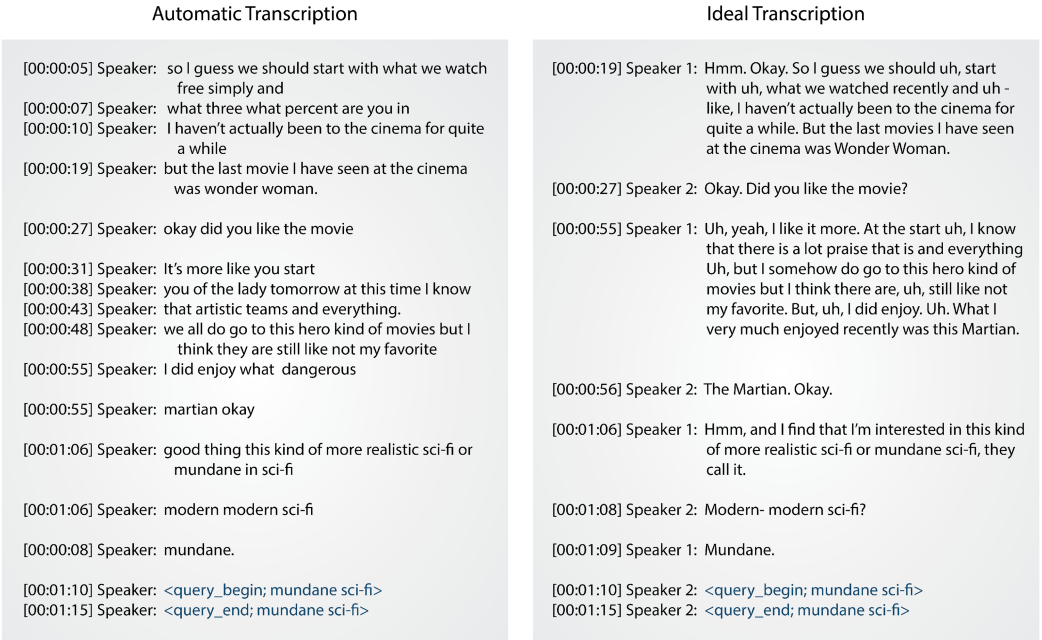


Fig. 4. Examples of automatic and ideal transcriptions.

a voice activity was detected. After the voice activity stopped, the systems recognized an utterance and returned a sentence transcript. As soon as the recognized transcript was available, it was saved as a text unit containing the sentence transcript and an associated timestamp, as illustrated in Figure 4. This procedure ensured that the speech recognizer had access only to the conversations that occurred prior to the search and was unable to use the post-search conversations when creating the transcripts.

3.6.2 Ideal Transcription. Besides automatically processed transcripts, the output of the data-collection experiment also contained high-quality video recordings. We obtained ideal transcriptions through manual annotation of the video recordings. A professional transcription company was hired to transcribe the video recordings. Speakers' turns were identified, and each turn was associated with an end timestamp, as shown in Figure 4. The end timestamps were obtained whenever the speaker changed. Furthermore, we manually checked and verified the correctness of the individual timestamps. Two coders manually transcribed the recordings and agreed on 100% of the transcribed texts except for the use of plurals and prepositions, which were difficult to identify. However, these did not affect the results as the text was stemmed and stop words were removed before the models were trained.

3.6.3 Web-search History. The effectiveness of using users' search history to contextualize QAC was also investigated. The search history consists of queries submitted and Web pages browsed in the same session prior to searching. To extract the text from HTML responses, we used the content and comment extractors[4] of the Dragnet [62].

[4]https://github.com/dragnet-org/dragnet.

3.7 Data Preprocessing

The resulting experimental data comprises all queries and browsing activities prior to searching, as well as the transcripts of each session. The data was first segmented into search activities with a time threshold. Each search activity is composed of a current query, recent activity (previous queries and browsed Web pages), and speakers' utterances within the time threshold preceding the search.

To understand the amount of context information that is sufficient to improve QAC, we used the full data and composed four subsets of data by varying context sizes, using time thresholds of 30 s, 1 min, 5 min, and 10 min.

3.7.1 Utterances. The utterances produced through automatic speech recognition were considered text units. In ideal transcription, we considered an utterance in each turn of a speaker as a single text unit. We discarded filler words such as "Yeah," "OK," or "Hmm," because context-aware methods need at least some useful spoken text as context.

3.7.2 Web-search History. Web-search history was captured from the search sessions. The history consisted of the browsing history of Web pages that the participants encountered during the experimental session. We decomposed the Web pages by paragraphs and considered them single text units to train the model. Text units were timestamped using their associated Web entries in the log. Query contents were also considered separate text units.

3.7.3 Query Prefixes and Original QACs. All possible prefixes of the query that the user actually wrote and submitted to the search system were generated and sent to the Google Query Suggestion Service[5] to retrieve a set of 20 query candidates. For example, considering the query "alice in wonderland," we first send the single-character prefix "*a*" to the Google service, which will return "amazon," "airbnb," "aliexpress," and so on.[6] We then apply the same procedure to the other prefixes ("*al*," "*ali*," etc.) of the query.

The cleaned datasets, including truncated streams of text units and the query prefixes, were further used to build the models for contextual QAC. For each session, we treated the submitted query as the ground truth q_T, that is, the intended query we wanted to re-rank.

Given the determined context size, we preprocessed text units produced from previous steps. Text units were preprocessed through stopword removal, stemming, and lemmatization. The preprocessed text units were later used to train the re-ranking models.

3.8 Data-Collection Results

Table 1 presents the results of the data collection and preprocessing. For clarity, in the following, we refer to lemmatized and stemmed terms as words. The resulting dataset consists of 12 transcribed sessions. There were 21,624 words in total, an average of 1,802 (SD = 462) recognized words per session with automatic transcription. Manual transcribing resulted in 33,238 words, an average of 2,770 (SD = 631) words per session. The **word error rate (WER)** [57] was computed as the probability of incorrect word recognition of the automatic transcription. Average WER per participant, computed by comparing the words in each speaker turn (with filler word removal and expanding word contractions) between the automatic transcript and the ideal transcript, was 44.67%. Percentage of keywords recognized correctly from the speech was computed by dividing the number of keywords recognized correctly (in the automatic transcript) by the total number of actual keywords (in the manual transcript) in one session. We used AllenNLP [26] to extract

[5] http://clients1.google.com/complete/search?&q=⟨prefix⟩&client=chrome.
[6] The prefix x was submitted to the Google query completion API on September 1, 2019, in private browsing mode.

Table 1. Results of the Data-Collection Experiment

Number of transcribed sessions	12
Number of spoken words (Automatic Transcription)	21,624 (M=1,802, SD=462)
Number of spoken words (Ideal Transcription)	33,238 (M=2,770, SD=631)
Word error rate	44.67%
Percentage of keywords recognized correctly	61.96%
Number of submitted queries	214 (M=18, SD=14)
Number of browsed Web pages	149 (M=13, SD=11)
Number of prefixes	2,930
Average number of characters per query	14 (SD = 5)

(a) Descriptive Information of Experimental Dataset

	Prior queries		*Recently browsed Web pages*		
Context Size (min)	*Number of queries*	*Words per query*	*Number of Web pages*	*Paragraphs per Web page*	*Words per paragraph*
0.5	0.71 (0.84)	2.41 (1.47)	0.27 (0.74)	93.74 (44.69)	11.91 (6.08)
1	1.32 (1.13)	2.53 (1.47)	0.71 (1.88)	114.11 (69.89)	11.37 (7.59)
5	5.93 (3.76)	2.82 (1.11)	2.48 (4.39)	260.25 (211.69)	9.76 (8.34)
10	10.21 (7.01)	2.88 (0.97)	3.98 (5.54)	378.02 (308.14)	9.64 (7.95)

(b) Web-search History

	Automatic Transcription		*Ideal Transcription*	
Context Size (min)	*Number of utterances*	*Words per utterance*	*Number of speakers' turns*	*Words per speaker turn*
0.5	6.61 (1.67)	5.02 (1.58)	6.53 (3.31)	8.29 (5.42)
1	12.86 (2.84)	4.63 (1.11)	12.78 (5.59)	6.65 (2.96)
5	57.41 (18.21)	4.43 (0.59)	56.79 (22.81)	5.02 (1.31)
10	98.03 (43.53)	4.03 (0.66)	97.41 (49.53)	4.39 (1.39)

(c) Transcripts

keywords from the transcripts. Overall, an average of 61.96% keywords per participant in the spoken conversations were recognized correctly by the automatic transcription system.

Participants browsed 149 Web pages in conversations, an average of 13 Web pages per session (SD = 11). Of 214 queries in total, an average of 18 (SD = 14) were determined in the query logs, with an average of 14 characters per query (SD = 5). These were used to generate 2,930 different query prefixes.

Tables 1(b) and 1(c) present the results for the formed datasets using context sizes of 30 s and 1, 5, and 10 min. Datasets using larger sizes involve more context information in search history and spoken conversation. Participants used few words to construct their queries, with an average of 2.41, 2.53, 2.82, and 2.88 for 30-s, 1-min, 5-min, and 10-min context sizes, respectively. Participants searched and browsed the Web over the conversations; however, the number of clicked on and visited Web pages prior to search was small, even within a longer context, such as 3.98 in a 10-min context. Average user utterances using automatic transcription were 6.61, 12.86, 57.41, and 98.03 for 30-s, 1-min, 5-min, and 10-min context sizes, respectively. The ideal transcription included an average of 6.53, 12.78, 56.79, and 97.41 spoken turns for 30-s, 1-min, 5-min, and 10-min context sizes, respectively. Average number of words per utterance was four to five, whereas spoken turns contained more words, up to an average of eight words per turn within a shorter context of 30 s.

4 CONTEXT MODELING FOR QAC

Two context sources were leveraged—spoken conversational input and search history (browsed Web pages and prior queries)—for re-ranking QACs. The sources determined the information used to build the context models. The sources used to construct the three models are described below.

- **Search Context Model.** The search context model was constructed based on a user's Web-search activity followed by a subsequent search or the current query. The textual content of browsed Web pages and queries of prior searches were utilized to train the model. We assumed that if a user searched and opened a Web document, the content might influence the user's subsequent search and contain useful information for modeling. Text units of browsed Web pages and prior queries processed in the early step were used to train the model.
- **Spoken Context Model.** The spoken context model was constructed based on spoken conversation between users that occurred prior to the current search query. The information comprised text units produced from automatic or ideal transcription.
- **Combined Context Model (Spoken + Search Context).** The combined context model was created using a combination of spoken conversational inputs and a user's search history. Outputs from the two separate models were combined.

4.1 Dirichlet-Hawkes Processes

We used **Dirichlet–Hawkes processes (DHP)** [23] for topic modeling of search and spoken context. DHP is a time-dependent topic model that combines Dirichlet [4] and Hawkes processes [32] to uncover meaningful topics and their temporal dynamics in the temporal stream of a conversation. The Hawkes process model has been used in prior QAC research [46] to model word sequence data. The occurrence of a particular word or phrase in the past can influence specific queries to be issued in the future.

Our pilot tests show that DHP is particularly suitable to model the evolving nature of informal conversational topics when compared with conventional topic models that do not consider the temporal dynamics.

DHP was utilized to discover topic clusters from a stream of text units based on both the contents and temporal dynamics of their occurrence. The model is estimated through an online inference algorithm that jointly learns the cluster pattern and the parameters of the Hawkes processes for each cluster [23]. The use of DHP relies on the assumption that speakers' utterances, Web queries, and Web documents with similar topics emerging closely in time are related to each other. The co-occurrence of spoken words in a topic cluster (e.g., "curious" and "case") will influence the later QAC prediction (e.g., "curious case of benjamin button"). In the DHP model, each text unit (an utterance, a prior query, and a browsed Web page) was considered an input unit. After model estimation, the resulting topic clusters are used to re-rank query suggestions.

4.2 Modeling Technique

The main notation used is described in Table 2. We denote a query that the user submitted to the search system at time T as q_T. Each query q_T is decomposed into a set of prefixes $R = \{r_1, r_2, \ldots, r_j\}$ of length j, where j is equal to the number of characters required to enter the entire query if no query completion interface was available. For each r_j, we generate a list of query candidates $C = \{c_1, c_2, \ldots c_i\}$ predicting the query q_T.

To incorporate contextual signals in the prediction model, a set of text units preceding the query q_T is extracted, denoted as $D = \{d_1, \ldots, d_t\}$ consisting of all words $W = \{w_1, \ldots w_n\}$, truncated by length len before time T as $d_{T-len:T}$ with $len = \{30\text{-s}, 1\text{-}, 5\text{-}, 10\text{-min}\}$.

Table 2. Main Notation Used in the Article

Notation	Description
T	logged timestamp associated to a user activity
q_T	a query submitted by the user at time T in a conversation session
R	set of all prefixes of a q_T
r_j	a prefix of length j; $(j = 1, 2, \ldots, \lvert R\rvert)$
C	a list of QAC candidates for prefix r_j returned by non-contextualized QAC
c_i	the ith query candidate in a QAC list
len	a context size with a threshold of 30 seconds or 1, 5, or 10 min
D	set of text units truncated by len before time T using $d_{T-len:T}$
d_t	the tth text unit in D; $(t = 1, 2, \ldots, \lvert D\rvert)$
W	vocabulary of D
w_n	the nth word in W; $(n = 1, 2, \ldots, \lvert W\rvert)$
f_n^t	word count of w_n if w_n exists in d_t, or 0 otherwise
K	number of topics produced by DHP
z_k	the kth topic in K topics; $(k = 1, 2, \ldots, \lvert K\rvert)$

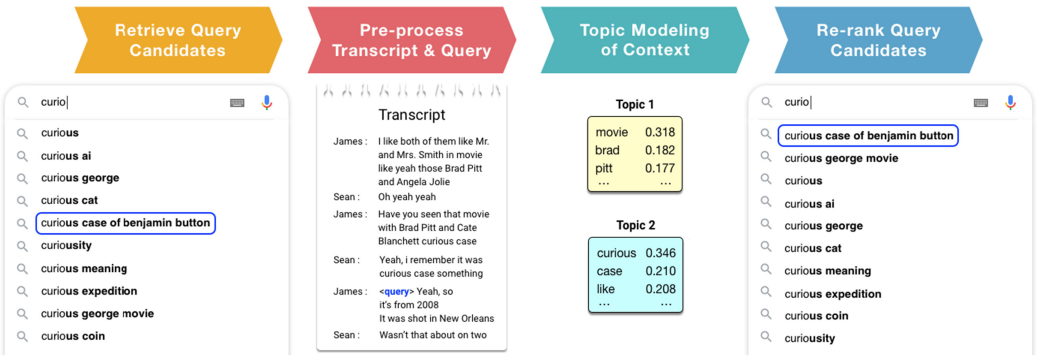


Fig. 5. Context modeling for QAC includes three main steps: (1) Use the query prefix to retrieve query candidates; (2) Use text units preceding the query to build a topic model of the conversation; (3) Use the topic distribution to re-rank the candidates.

Given each prefix r_j and contextual signals in D originating from the user, our approach is based on the three following steps, illustrated in Figure 5:

(1) Use r_j to retrieve query candidates C using a non-contextualized QAC approach.
(2) Use D to build a topic model of the context preceding the search.
(3) Use the topic distribution of context by aggregating topic distributions of all past text units D to re-rank the list of query candidates.

Step 1: Retrieving query candidates. We retrieve a list of candidate query suggestions C predicting the intended query q_T from the Google Query Suggestion Service. The list is limited to the top 20 query candidates returned by the service.

Step 2: Topic modeling of context. Given D, the context at each time step is defined as a vector over W words that represent the conversation at that specific time. Each text unit is treated as a document that is represented as a bag of words in which non-zero elements are the words

present in the current text unit. The context is stored in the matrix $X \in \mathcal{R}^{|W|\times|D|}$:

$$X = \begin{array}{c} \\ \mathrm{w}_1 \\ \mathrm{w}_2 \\ \\ \mathrm{w}_n \end{array} \begin{array}{c} \begin{array}{cccc} \mathrm{d}_1 & \mathrm{d}_2 & \dots & \mathrm{d}_t \end{array} \\ \begin{bmatrix} f_1^1 & f_1^2 & \cdots & f_1^t \\ f_2^1 & f_2^2 & \cdots & f_2^t \\ \vdots & \vdots & \ddots & \vdots \\ f_n^1 & f_n^2 & \cdots & f_n^t \end{bmatrix} \end{array},$$

where the element f_n^t is the word count of w_n if w_n exists in the text unit d_t, or 0 otherwise; each d_t is associated with a timestamp; $|W|$ and $|D|$ are the set sizes; and $n = \{1, 2, \dots, |W|\}$ and $t = \{1, 2, \dots, |D|\}$.

The DHP approach projects X into a low-dimensional latent space such that co-occurring words in text units should have similar representation. The model automatically yields a fixed K number of topics, for which we denote each topic as z_k.

Step 3: Using the topic distribution of context to re-rank query candidates. The resulting topic model assigns each text unit d_t a probability distribution over topics, in which $p(z_k|d_t)$ is the probability of a topic z_k given a text unit d_t. By aggregating such topic distributions across text units D, we can infer a prior topic distribution as follows:

$$p(z_k|D) = \frac{1}{|D|}\sum_{t=1}^{|D|} p(z_k|d_t).$$

Intuitively, a topic that was discussed frequently would obtain a higher probability.

Given $p(z_k|D)$, each query suggestion c_i can be assigned a rank based on the probability of c_i assuming K topics (denoted as $p(c_i)$) by computing the product of $p(z_k|D)$ and the Dirichlet-Multinomial [23] log likelihood of c_i belonging to each topic $p(c_i|z_k)$ as follows:

$$\begin{aligned} p(c_i) &= p(z_k|D) \cdot p(c_i|z_k) \\ &= p(z_k|D) \cdot \left(\frac{\Gamma(f^{z_k \setminus c_i} + |W|) \prod_{n=1}^{|W|} \Gamma(f_n^{z_k \setminus c_i} + f_n^{c_i} + \theta_0)}{\Gamma(f^{z_k \setminus c_i} + f^{c_i} + |W|) \prod_{n=1}^{|W|} \Gamma(f_n^{z_k \setminus c_i} + \theta_0)} \right), \end{aligned}$$

where $f^{z_k \setminus c_i}$ is the word count of the topic z_k excluding the candidate c_i; $f_n^{z_k \setminus c_i}$ refers to the count of the nth word; f^{c_i} is the word count in c_i; and θ_0 is obtained from the DHP. Query suggestions are ranked by sorting $p(c_i)$ in ascending order. That is, the query suggestions that are more semantically related to the topic discussed would be ranked higher.

4.3 Combining Spoken and Search Context Models

To form a single rank distribution from the two separate models, we used the Dempster-Shafer theory [71] that allows for combining various sources of evidence. Two rank distributions, B and C produced by search and spoken context models, respectively, can be combined to form a single rank distribution A. The combination rule for fusing two pieces of evidence m_1 and m_2 respective to $p(c_i)$ of rank distributions B and C, is defined as follows:

$$m_{12}(A) = m_1 \oplus m_2(A) = \frac{\sum_{B\cap C}\{m_1(B).m_2(C)\}}{1 - L},$$

when $m_{12}(A) \neq \emptyset$ and $m_{12}(\emptyset) = 0$.

$$L = \sum_{B\cap C=\emptyset} \{m_1(B).m_2(C)\},$$

Table 3. Configurations of Each Compared Condition

	Control (C)	*Search (S) Context*	*Spoken Context*	*Combined Context*
Search history	-	✓	-	✓
Spoken conversation	-	-	✓	✓
Context model	-	✓	✓	✓
Google QAC	✓	✓	✓	✓

Context model with the inputs: search history and spoken conversation are additive.

where L is the degree of conflict in two sources of evidence and the denominator $(1 - L)$ is a normalization factor, which ignores all the conflicting evidences and is calculated by adding up the products of $p(c_i)$ of two distributions where the intersection is $\emptyset$.

5 EVALUATION

The collected data and the QAC model in varying conditions were evaluated in an offline experiment. Here, we explain the configuration for each condition and evaluation metrics used to measure the QAC performance in these conditions.

5.1 Conditions

To study the utility of spoken context in QAC, we tested the QAC model in four conditions: the control condition, the search context condition, the spoken context condition, and the combined context condition. Table 3 shows model configurations in these conditions, which are described in more detail below.

- *Control.* In the control condition, QAC initially produced by the Google API service was used, but the QAC did not account for any context information from the conversation. We turned off the personalization feature in Google API to avoid any confounding factors that might affect the initial ranking of QAC. For instance, different users might have different tastes in movies and travel present in their long-term search history prior to the experiment, and Google API would have this information and personalized QACs, which would have become a factor in the experiment.
- *Search Context.* In this condition, we included the search context model, which leveraged only a user's search context information, to re-rank Google QACs.
- *Spoken Context.* In this condition, we included the spoken context model, which leveraged only spoken context information, to re-rank Google QACs.
- *Combined Context.* In this condition, both spoken and search context information was leveraged to re-rank Google QACs.

5.2 Evaluation Setup

The models were evaluated with queries and trained with data occurring prior to the queries against to which they were tested. This is a natural split to testing and training data. More formally, the models for a query q_T are trained with data $D = \{d_1, \ldots, d_t\}$ occurring before time T. This ensures that the evaluation follows a real-world scenario, in which only data occurring before the query was issued are used as evidence for the models.

5.3 Evaluation Metrics

With the ground truth query q_T, we can evaluate the QAC performance by three metrics, including **mean reciprocal rank (MRR)** and **success rate at top-k (SR@k)** and average number of keystrokes needed to take the correct suggestion matching the intended query to position 1. MRR

Table 4. An Example of Computing Inverse Hitrank and SR@k for a Given Query

j	1		2		3		4
Prefix	d	⇒	do	⇒	doc	⇒	doct
c1	dropbox		donald trump		docs		doctor strange ✓
c2	drive		docker		doctor strange ✓		doctor who
c3	dhl		doctor strange ✓		doc martin		doctor sleep
c4	duckduckgo		doodle		doc martens		doctor mike
$hitrank^{-1}$	0.00		0.33		0.50		1.00
SR@1	0.00		0.00		0.00		1.00
SR@2	0.00		0.00		1.00		1.00
SR@3	0.00		1.00		1.00		1.00

For an intended query "doctor strange" submitted by the user, all prefixes are first extracted. For each prefix, the top-ranked candidates matched in the auto-completion are collected from the Google query complete API. In the example, the first row represents the prefix being formulated, while c1, c2, c3, and c4 denote the top 4 auto-completion suggestion candidates returned for the prefix. The correct suggestion matching the intended query in each list is specified by a checkmark (✓).

is a commonly used metric for QAC as the multiplicative inverse of the rank of the actual query q_T in the ranking list [7]. The MRR for each q_T is computed as follows:

$$MRR_{q_T} = \frac{1}{|R|} \sum_{j=1}^{|R|} \frac{1}{hitrank(q_T, r_j)},$$

where R is all possible prefixes of q_T, $hitrank(q_T, r_j)$ is a rank of the intended query q_T in the suggestion list given each prefix r_j of length j. The MRR for QAC in each condition is computed by averaging the MRRs of all queries per each conversation session.

The second metric is the success rate at top-k (SR@k) denoting the average percentage of the intended queries that can be found in the top-k query suggestions. Both of the metrics are widely used for the task with a ground truth of only one instance such as query completion [39]. Table 4 illustrates how SR@1, SR@2, and SR@3 are computed.

The third metric is the average number of keystrokes needed to enter queries that can be saved per query suggested with our approach. We count the number of characters needed for the QAC to obtain the correct suggestion matching the intended query in position 1. For example, let us examine the intended query "curious case of benjamin button." It takes 5 characters, "curio," to bring the correct query candidate to position 1 using our approach, whereas with Google QAC, it takes 10 characters, "curious cas," to do the same.[7]

5.4 Statistical Testing Procedure

We applied a paired-samples t-test to determine whether there were statistically significant differences in QAC performance in different conditions. To test the significance levels, we used MRR, SR@k, and average number of characters (or keystrokes) as dependent variables and the conditions as independent variables. RStudio software v1.1.4 was used for the calculation of statistical significance.

We also applied Bonferroni correction [72] to adjust for multiple pairwise comparisons. In addition, Cohen's d values for the t-test were computed to measure the effect sizes between the approaches.

[7]The prefix was submitted to the Google query completion API on September 1, 2019, in private browsing mode.

Table 5. The QAC Performance in Terms of MRR, SR@1, SR@2, and SR@3 in Different Conditions with All Possible Prefixes Under Context Sizes of 30 s and 1, 5, and 10 min

Measure	*Context Size (min)*	*Control (C)*	*Search Context (Se)*	*p-value (vs. C)*	*Spoken Context (Sp)*	*p-value (vs. C)*	*p- value (vs. Se)*	*Spoken + Context*	*p-value (vs. C)*	*p-value (vs. Se)*	*p-value (vs. Sp)*
MRR	0.5	0.47	0.49	1	0.53	0.2	0.2	**0.55**	**0.04**	**0.03**	1
	1		0.49	1	**0.55**	**0.04**	**0.03**	**0.57**	**0.01**	**0.01**	1
	5		0.50	1	**0.56**	**0.02**	**0.02**	**0.57**	**0.01**	**0.01**	1
	10		0.50	1	**0.56**	**0.02**	**0.02**	**0.57**	**0.01**	**0.01**	1
SR@1	0.5	0.40	0.42	1	0.49	0.2	0.3	**0.52**	**0.03**	**0.02**	0.9
	1		0.42	1	**0.51**	**0.04**	**0.04**	**0.53**	**0.01**	**0.01**	1
	5		0.43	1	**0.52**	**0.02**	**0.02**	**0.53**	**0.01**	**0.01**	1
	10		0.43	1	**0.52**	**0.02**	**0.02**	**0.53**	**0.01**	**0.01**	1
SR@2	0.5	0.48	0.50	1	0.54	0.3	0.3	**0.56**	**0.05**	0.1	0.9
	1		0.50	1	0.55	0.1	0.1	**0.58**	**0.01**	**0.01**	1
	5		0.51	1	**0.56**	**0.05**	0.1	**0.58**	**0.01**	**0.01**	1
	10		0.51	1	**0.56**	**0.05**	0.1	**0.58**	**0.01**	**0.01**	1
SR@3	0.5	0.51	0.53	1	0.56	0.4	0.5	0.57	0.09	0.1	1
	1		0.53	1	0.57	0.08	0.1	**0.60**	**0.01**	**0.01**	1
	5		0.54	1	0.58	0.06	0.1	**0.60**	**0.01**	0.06	1
	10		0.54	1	0.58	0.06	0.1	**0.60**	**0.01**	0.06	1

(a) Automatic Transcription

Measure	*Context Size (min)*	*Control (C)*	*Search Context (Se)*	*p-value (vs. C)*	*Spoken Context (Sp)*	*p-value (vs. C)*	*p- value (vs. Se)*	*Spoken + Context*	*p-value (vs. C)*	*p-value (vs. Se)*	*p-value (vs. Sp)*
MRR	0.5	0.47	0.49	1	**0.58**	**0.01**	**0.01**	**0.59**	**0.01**	**0.01**	1
	1		0.49	1	**0.59**	**0.01**	**0.01**	**0.60**	**0.01**	**0.01**	1
	5		0.50	1	**0.59**	**0.01**	**0.01**	**0.61**	**0.01**	**0.01**	1
	10		0.50	1	**0.59**	**0.01**	**0.01**	**0.60**	**0.01**	**0.01**	1
SR@1	0.5	0.40	0.42	1	**0.55**	**0.01**	**0.01**	**0.56**	**0.01**	**0.01**	1
	1		0.42	1	**0.57**	**0.01**	**0.01**	**0.58**	**0.01**	**0.01**	1
	5		0.43	1	**0.57**	**0.01**	**0.01**	**0.58**	**0.01**	**0.01**	1
	10		0.43	1	**0.57**	**0.01**	**0.01**	**0.58**	**0.01**	**0.01**	1
SR@2	0.5	0.48	0.50	1	**0.58**	**0.01**	**0.01**	**0.59**	**0.01**	**0.01**	1
	1		0.50	1	**0.60**	**0.01**	**0.01**	**0.61**	**0.01**	**0.01**	1
	5		0.51	1	**0.60**	**0.01**	**0.01**	**0.61**	**0.01**	**0.01**	1
	10		0.51	1	**0.60**	**0.01**	**0.01**	**0.61**	**0.01**	**0.01**	1
SR@3	0.5	0.51	0.53	1	**0.59**	**0.02**	0.09	**0.60**	**0.01**	0.09	1
	1		0.53	1	**0.61**	**0.01**	**0.04**	**0.62**	**0.01**	**0.01**	1
	5		0.54	1	**0.61**	**0.01**	**0.04**	**0.62**	**0.01**	**0.04**	1
	10		0.54	1	**0.61**	**0.01**	**0.04**	**0.62**	**0.01**	**0.04**	1

(b) Ideal Transcription

6 RESULTS

The results, comparing the QAC performance among the conditions, are reported in the following with respect to the research questions defined earlier: ranking performance and effect of typed-query input.

6.1 Ranking Performance

6.1.1 Automatic Transcription. Table 5 shows the experimental results for MRR and SR@k. In general, QAC in the spoken context condition performed better in terms of MRR compared with the model in the control and search context conditions. The MRR was 0.55 for the 1-min context and 0.56 for both 5- and 10-min contexts. Significant differences were found in the performance among the conditions (paired-samples t-tests, $p < 0.04$, $d > 0.87$). Furthermore, the QAC model in

the spoken context condition performed well when measured using SR@1 with 1-, 5-, and 10-min context sizes; SR@1 was 0.51 for the 1-min context and 0.52 for the 5- and 10-min contexts (paired-samples t-test, $p < 0.4$, $d > 0.87$). However, no significant differences were found between the control condition and the spoken context condition at SR@1, SR@2, or SR@3 for the 30-s context.

The results show that the MRRs in the combined context condition were higher for all context sizes; MRR was 0.55 for the 30-s context and 0.57 for the 1-, 5-, and 10-min contexts. Differences between the combined context condition and the control condition were significant ($p < 0.04$, $d > 0.87$). However, no significant differences in performance were found between the search context condition and the control condition. This indicates that fusing the two data sources positively contributed to the context modeling. SR@k in the combined context condition also improved as a result of data fusion. MRR and SR@1 in the combined context condition also largely improved over the search context condition for all context sizes (p-values < 0.03, d-values > 1.02). Such results suggest that, when engaged in spoken conversations, the participants conducted searches originating from their discussions, but the searches were also influenced by the contents of Web documents. In our experiments, jointly utilizing the two sources in modeling could successfully predict a user's search intent and improve QAC over a model that relies on a single source. Nevertheless, no significant differences were found in MRR and SR@k between the spoken context condition and the combined context condition.

6.1.2 Ideal Transcription. QAC in the spoken context condition achieved better performance than the model in the search context condition in terms of MRRs, SR@1, and SR@2. The paired-samples t-test revealed a significant effect of using spoken context to improve QAC for all context sizes ($p < 0.01$, $d > 0.89$), whereas the difference in SR@3 between the search context condition and the spoken context condition in the 30-s context was not significant. The results indicate that QAC in the spoken context condition using ideal transcription predicted the user queries most accurately within the 1-, 5-, and 10-min contexts. More context seems, therefore, preferable in predicting query completion.

The QAC model performed best in the combined context condition among the conditions. The paired t-test revealed that all differences between the combined context condition and the control condition for MRRs and SR@k were significant in all context sizes (p-values < 0.01, d-values > 1.02). The QAC model performance was actually better within longer contexts (1 min, 5 min, and 10 min). This suggests that setting the time threshold for the spoken context as 1 to 5 min is recommended, because the model performance will be consistent regardless of how far back in a user's history we go.

Quality of top-1 suggestions with combined contexts is generally better than the model using search context alone. The results show that MRR, SR@1, and SR@2 in the combined context condition were the largest compared to those values in the search context condition and control condition. Paired-samples t-test confirmed the differences between the combined context condition and the search context condition were significant (p-values < 0.01, d-values > 0.81). The results demonstrate not only the usefulness of conversational inputs for improving QAC but also the extreme impact of the combined contexts. Although there were no significant effects in SR@3 using a different 30-s context, the results indicate that larger data actually produced higher improvements over the smaller one. There was no significant difference found between the spoken context condition and the combined context condition when using ideal transcription.

6.2 Effect of Typed-Query Input: Overall Performance

To understand how much user effort invested in typing queries can be reduced if the spoken context approach is used, we inspected the average prefix length and keystrokes saved per predicted query for SR@k, where k ranged from 1 to 5.

6.2.1 Automatic Transcription. Figure 6 shows the QAC's performance in the control condition and experimental conditions in terms of the average number of keystrokes or characters needed to obtain the right suggestion matching the intended query in the first position of the ranked list. The results show that QAC in the spoken context condition using automatic transcription reduced user typing effort over the control condition. At least, a single key press can be saved using our spoken context QAC. On average, 4.52, 4.09, and 4.02 keystrokes were required to enter the set of queries and 14.23%, 22.39%, and 22.91% decreased in user effort can be obtained using our QAC with context sizes of 30 s, 1 min, 5 min, and 10 min, respectively.

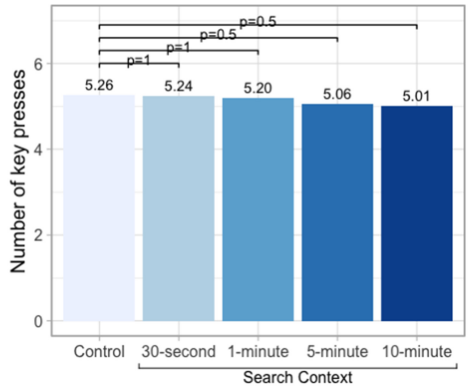
In the search context condition, the QAC model can also slightly reduce the number of keystrokes, but there were no significant differences found between the search context condition and the control condition. This is probably because the model did not have access to personal preferences from complete user histories (e.g., long-term Web activities before the experiment). Because users' topical interests were highly dynamic within a short period of time in conversation, it was hard to obtain a comprehensive context based on only Web browsing activities. However, the search context played an important role in understanding users' instant intent when combined with spoken context, which made it more powerful in the proposed model. This can be seen in the results showing that an average number of keystrokes needed to enter a query was reduced when using the combined context approach: two key press can be saved using the combined context information. The number of keystrokes was reduced from 5.23 (in the control condition) to 3.88 for the 1-min context, to 3.90 for 5-min context, and 3.89 for 10-min context. Percent decreases were improved: 26.37%, 26%, and 26.18% for 1-, 5-, and 10-min contexts, respectively. Paired-samples t-test revealed that differences between the combined context condition and the control condition were also significant.

6.2.2 Ideal Transcription. QAC in the spoken context condition using ideal speech recognition system consistently lowered user typing effort over both the control condition and the search context condition, reducing the number of keystrokes when taking into account the contextual signals from the conversation preceding the query. Paired-samples t-test revealed a significant difference among the models ($p < 0.01$, $d > 1.6$).

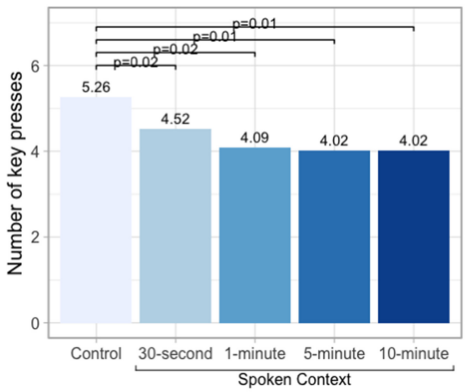
Table 6 presents the reduced typing effort in percent decrease in the number of keystrokes required to make the right suggestion in the first position in the ranked list. QAC in the spoken context condition requires an average of four keystrokes with a 24.28% decrease in keystrokes already within a short context of 30 s. QAC in the combined context condition reduces more keystrokes with an average of 3.9 and a 26% decrease.

By considering the longer context, QAC in the spoken context condition can further decrease the number of keystrokes needed to enter a query with an average of 3.83, 3.79, and 3.78 keystrokes and an average of 27.32%, 28.08%, and 28.27% decrease in typing effort with context sizes of 1 min, 5 min, and 10 min, respectively. Here, the QAC approach in the combined context condition yielded even better results, which may be attributed to modeling the relationship between users' browsing behaviors and spoken conversation within the same session. The average keystrokes needed to enter a query was 3.75 for the 1-min context, 3.7 for the 5-min context, and 3.71 for the 10-min context. The results indicate a further improvement in percent decrease: up to a 28.84% reduction in keystrokes for the 1-min context, 29.79% for the 5-min context, and 29.60% for the 10-min context.

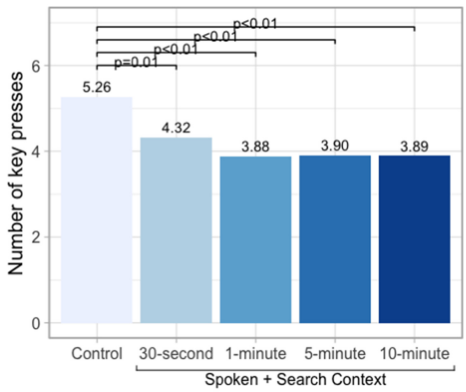
QAC in the search context condition performed worse compared to the model in spoken and combined context conditions, since it pays no attention to users' spoken conversation. No significant differences were found between the QAC's performance in the search context condition and control condition.



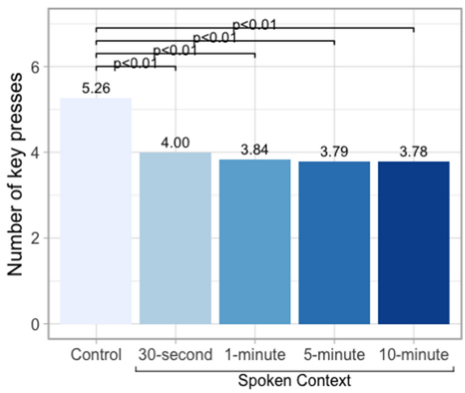
(a) Search Context



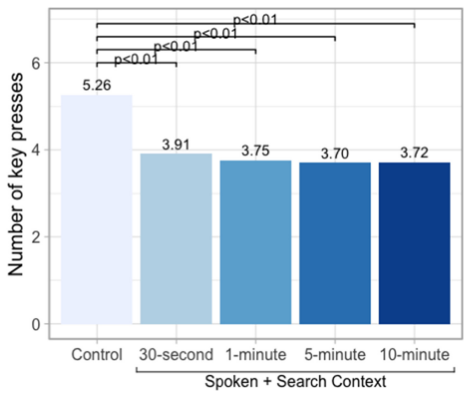
(b) Spoken Context (Automatic Transcription)



(c) Spoken + Search Context (Automatic Transcription)



(d) Spoken Context (Ideal Transcription)



(e) Spoken + Search Context (Ideal Transcription)

Fig. 6. Number of keystrokes or characters needed to obtain the suggestion matching the intended query at the first position in the ranked list of suggestions.

Table 6. Percent Decrease (Higher Is Better) in the Number of Keystrokes Required to Enter the Set of Queries with the Context QAC Relative to the Keystrokes Needed to Enter the Same Set of Queries with QAC in the Control Condition (cf. Figure 6)

	Search Context			*Spoken Context*				*Spoken + Search Context*				
Context Size (min)	*# of key presses*	*Percent decrease*	*p-value (vs. C)*	*# of key presses*	*Percent decrease*	*p-value (vs. C)*	*p-value (vs. Se)*	*# of key presses*	*Percent decrease*	*p-value (vs. C)*	*p-value (vs. Se)*	*p-value (vs. Sp)*
0.5	5.23	0.95%	1	**4.52**	**14.23%**	**0.03**	**0.05**	**4.31**	**18.22%**	**0.01**	**0.01**	0.3
1	5.20	1.33%	1	**4.09**	**22.39%**	**0.03**	0.07	**3.88**	**26.37%**	**0.01**	**0.01**	1
5	5.05	4.18%	1	**4.01**	**23.91%**	**0.02**	0.09	**3.90**	**26.00%**	**0.01**	**0.02**	1
10	5.01	4.93%	0.9	**4.01**	**23.91%**	**0.02**	0.1	**3.89**	**26.18%**	**0.01**	**0.04**	1

(a) Automatic Transcription

	Search Context			*Spoken Context*				*Spoken + Search Context*				
Context Size (min)	*# of key presses*	*Percent decrease*	*p-value (vs. C)*	*# of key presses*	*Percent decrease*	*p-value (vs. C)*	*p-value (vs. Se)*	*# of key presses*	*Percent decrease*	*p-value (vs. C)*	*p-value (vs. Se)*	*p-value (vs. Sp)*
0.5	5.23	0.95%	1	**4.00**	**24.28%**	**0.01**	**0.01**	**3.90**	**26.00%**	**0.01**	**0.01**	0.9
1	5.20	1.33%	1	**3.83**	**27.32%**	**0.01**	**0.01**	**3.75**	**28.84%**	**0.01**	**0.01**	0.4
5	5.05	4.18%	1	**3.79**	**28.08%**	**0.01**	**0.01**	**3.70**	**29.79%**	**0.01**	**0.01**	0.5
10	5.01	4.93%	0.9	**3.78**	**28.27%**	**0.01**	**0.01**	**3.71**	**29.60%**	**0.01**	**0.01**	1

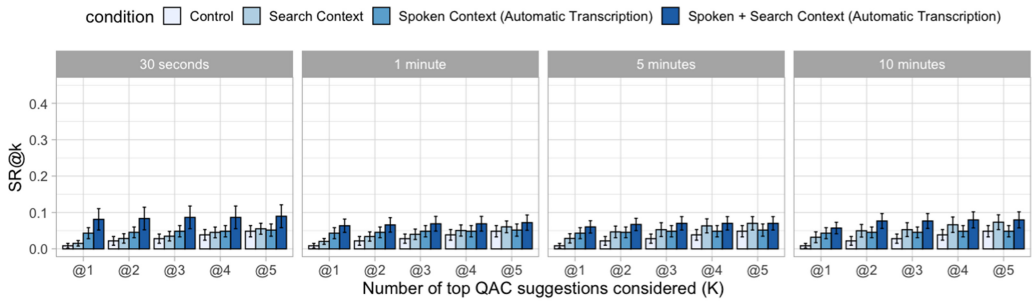
(b) Ideal Transcription

The results of query typing effort are based on the assumption that we can only show one query suggestion on the user interface. By using this constraint, we may not be exploiting the full information gain inherent to the user's context. The tradeoff between the improvement in query entry and the number of suggestions added to the user interface should be evaluated in future work. However, by comparing the number of keystrokes needed to enter queries to rank the correct query suggestion in the first position, we found advantages of the spoken context in QAC. This indicates that the model in the spoken context condition can recommend user-intended queries higher with less keystrokes. The model in the combined context condition with different context sizes also demonstrates its robustness and consistency in improvement.

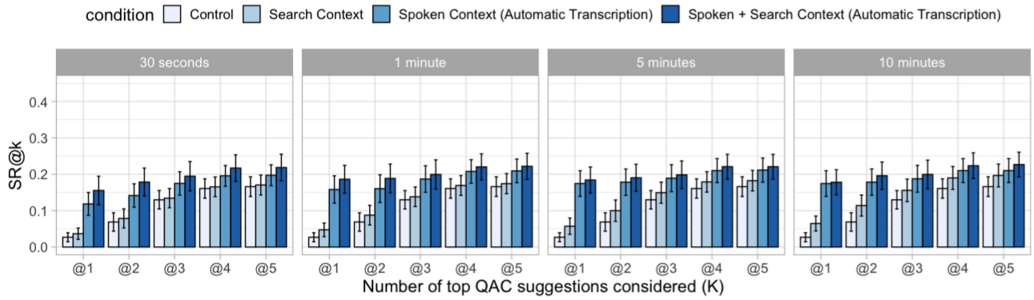
6.3 Effect of Typed-query Input: Character-level Performance

Another way to investigate the performance is to measure how many characters need to be typed to achieve a certain performance. Figures 7(a), 7(b), and 7(c) show the QAC's SR@k performance in the spoken context condition using automatic transcription when 1, 2, or 3 characters are typed with a growing length of top-k suggestions; while Figures 8(a), 8(b), and 8(c) show the QAC's SR@k performance in the spoken context condition using ideal transcription.

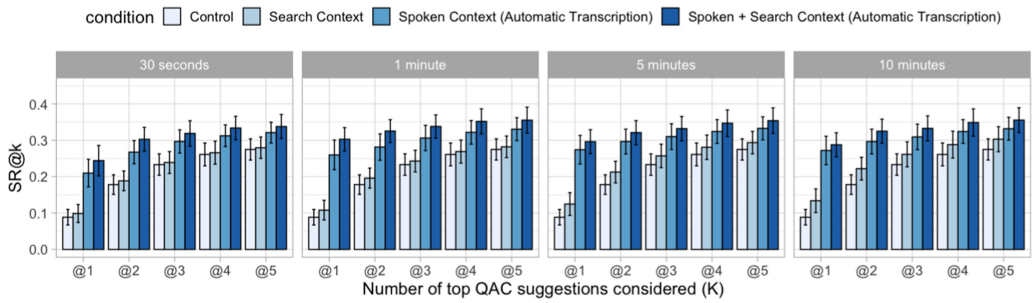
Unsurprisingly, the longer the input prefix, the better the query prediction, as more characters of the intended user query are available to the QAC model. Nevertheless, results show that only one typed character is enough for relatively good performance when the context models are used. Paired-samples t-test revealed significant differences in the performance between the spoken context condition using ideal transcription and the control condition (p-values < 0.01, $d > 1.2$). Differences in SR@k between the combined context condition using ideal transcription and the control condition were also significant (p-values < 0.01, $d > 1.4$). This showed that QAC in the experimental conditions was good at predicting the correct completions. A possible explanation is that some of the submitted queries were highly popular (e.g., netflix, momondo, and finnair), while Google may employ a popularity-based QAC approach that is very successful at predicting such queries [8]. Although such an approach results in a good QAC ranking, it may fail to produce hits at the top positions (SR@5) when the user input is very short (one character). By considering spoken conversational context, QAC ranking performance improves, indicating that such



(a) 1-character prefix



(b) 2-character prefix

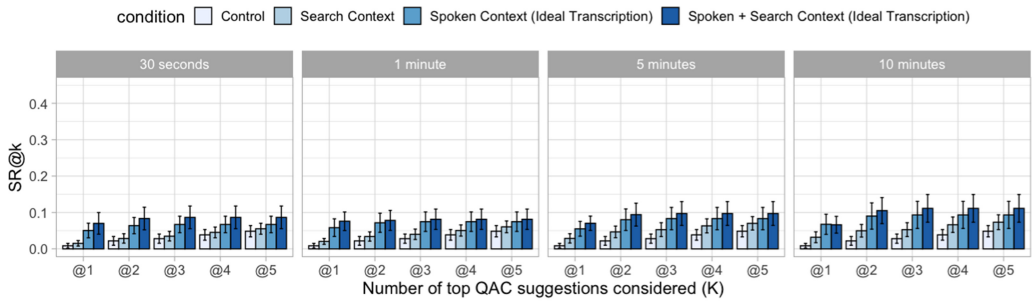


(c) 3-character prefix

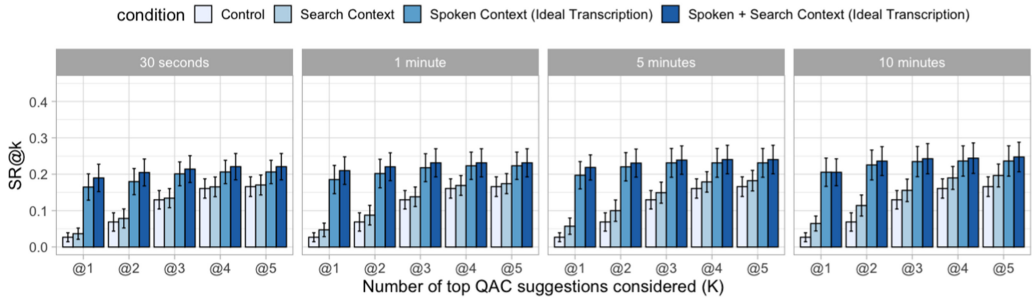
Fig. 7. Results for automatic transcription. Performance in terms of success rate at top-K (SR@K), $K = 1, \ldots, 5$, for QACs in control condition and experimental conditions when considering input prefixes of various sizes: (a) one character; (b) two characters; (c) three characters. The charts illustrate SR@K for context sizes of 30 s and 1, 5, and 10 min.

information was useful in improving query prediction. Nevertheless, QAC in the spoken context and the combined conditions when using automatic transcription did not outperform QAC in the control condition, indicating that speech recognition accuracy plays an important role in providing the required contextual information and, in turn, predicting queries.

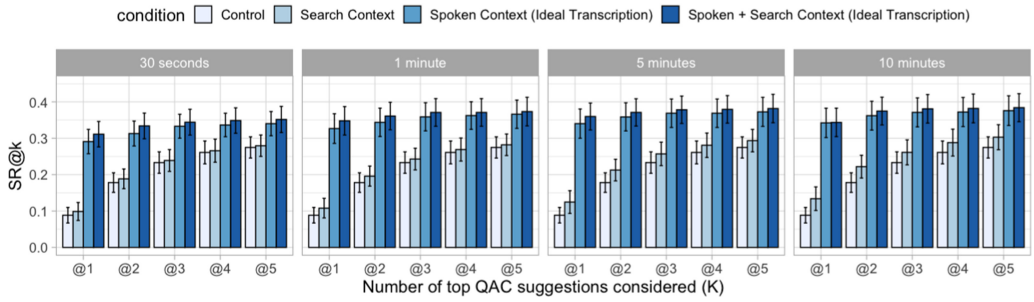
With two and three characters, QAC in the spoken context condition and combined context condition, using either ideal or automatic speech recognition system, significantly outperformed QAC in the control condition in terms of SR@k (p-values < 0.01, $d > 1.4$). Additionally,



(a) 1-character prefix



(b) 2-character prefix



(c) 3-character prefix

Fig. 8. Results for ideal transcription. Performance in terms of success rate at top-K (SR@K), $K = 1, \ldots, 5$, for QAC in control condition and experimental conditions when considering input prefixes of various sizes: (a) one character; (b) two characters; (c) three characters. The charts illustrate SR@K for context sizes of 30 s and 1, 5, and 10 min.

paired-samples t-tests also revealed significant differences in performance between the combined context condition using ideal transcription and the search context condition (p-values < 0.02, $d > 0.94$). These results suggest that the spoken context approach is generally good at suggesting query completions in the presence of limited user input. Again, the context size is important in improving QAC; we can observe better results with more context, notably with 5- and 10-min context sizes.

7 DISCUSSION

The idea of QAC is to predict the query that the user is typing in and complete it automatically. The benefits of this simple idea are manyfold. First, the search system can help the user by memorizing the right query vocabulary. Second, typing errors in the input can be minimized. Third, auto-completion speeds up the interaction by producing the query from fewer keystrokes than what would be required to type in the complete query.

In this research, we proposed a method for re-ranking QAC suggestions from the spoken conversational context. We reported a study seeking answers to the following research questions: (1) Does the use of spoken conversation as a context improve the ranking of query suggestions? (2) Does the spoken context help to reduce user effort in typing queries? (3) How does accuracy of speech recognition affect the ranking of query suggestions? Here, we answer these questions and reflect on the results and their impact for query suggestion, query auto-completion research, and search user interfaces in general.

7.1 Does the Use of Spoken Conversation as a Context Improve the Ranking of Query Suggestions?

Our approach shows significant improvements in ranking the correct query suggestion for all reported measures. The finding suggests the utility of spoken conversational input in query prediction and information retrieval. Similar findings have been reported in Mishne et al. [55]'s study; topic boundaries of multiple consecutive utterances could be identified and, segmented, and the topic models learned from the target segments can reveal the intended semantics required for constructing effective queries. In that study, users often preferred to type as little as possible to reach a query matching their search intention, only confirming the query suggestion [50]. Our results show that the improvement for the top-ranked suggestions at the character level in experimental conditions over QAC in the control condition was particularly significant. This indicates that our approach shows practically useful improvements for real-world scenarios considering users' usability preferences.

Compared to QAC in the control condition, that only models term similarities or query dependencies [8, 49], our approach helps to understand how the spoken conversational context occurring before users issue a query can be useful for improving QAC. Spoken conversational context can address the lack of data about user historical interactions in conventional context-aware methods [50, 74]. The only available data in these methods was the submitted queries. While the prefixes were simulated from all possible queries' prefixes, lack of associated information, such as the real-world context, prevents such methods from further improving their performance. Here, other contextual signals might need to be relied on, such as the spoken conversation and spoken content that should not remain as isolated information for long [60]. We can also incorporate more context information into the query prediction model. For example, with a user's explicit permission, it would be interesting to study the impact of combining spoken conversation with other personalized cues such as long-term historic context, social context, or location-based signals. There was no significant difference in QAC performance between the spoken context condition and the combined context condition, but the difference between the combined context condition and the search context condition was significant. This demonstrates that searches were dependent on the spoken context, but not dependent on browsing activity.

Our study suggests that the immediate pre-search voice context is not very helpful for predicting query suggestions; a longer context (between 1 and 5 min) is required. While this might be because of the types of tasks used, it indicates that modeling beyond simple detection of query terms is necessary to successfully predict query suggestions. This can further guide developers in designing

information access systems by providing an upper bound on the context length, which will make a significant impact on the performance.

7.2 Does the Spoken Context Help to Reduce User Effort in Typing Queries?

Compared to QACs in the control condition and search context condition, which do not benefit from the spoken conversational context, our approach uses additional contextual information to predict the ranking of query suggestions, but the query process still depends on the user inputting characters to initiate the search. We observed that the spoken context reduces the user's typing effort with an effect size implying practical user benefits. On average, QAC in the spoken context condition reduced the keystrokes up to 28.27% using ideal transcription, while QAC in the search context condition showed only a 4.93% improvement (ref. Table 6). By combining both the spoken and search contexts, we see a further improvement of up to a 29.60% reduction in keystrokes.

The QAC model using ideal transcription shows significant improvement overall even the user inputs only a single character. However, we did not see any improvement when QAC in the spoken context condition using automatic transcription. In fact, there was a degradation in performance with 1-character prefix. We believe the degradation is due to the sparsity of information that speech recognition errors create. With two and three characters inputted, our QAC models in the spoken context condition, using either automatic or ideal transcription, are more efficient in predicting the correct query postfix at the highest rank. More important, our results show improved performance to rank the intended query at the top of the list (refer to Figure 8). This suggests that the spoken conversational context can lead to a highly practical reduction in user typing effort in several search scenarios. For example, call center agents can benefit from having efficient information retrieval capabilities by searching previous calls, which provide the solution to the caller's problem [51]. Our technique might also find useful application in other search situations where reliance on typed queries should be minimized [1, 45]. In such situations, the spoken context can be used to anticipate user information needs, which, in turn, can be used to perform proactive searches or propose query suggestions that are most relevant to the spoken context. This may create minimal effort systems that can even eliminate typing altogether from the searching process.

Context sizes such as 1, 5, and 10 min were the most effective in improving prediction quality, requiring the user to type fewer characters per query. When the context size was smaller than 30 seconds, improvements in all the measures were lower, because often only few useful utterances were available. This indicates that the most recent utterances may be less relevant to the query intents at the time of query submissions. This finding contrasts with prior work [55], suggesting short context such as the most recent one to three utterances are most effective to predict user intent. A possible explanation is that past research focused on domain-specific conversations dealing with software and hardware issues rather than open-domain topics. As a result, recent utterances tend to contain a high concentration of important words and topics that are not highly changeable within a short period of time.

7.3 How Does Speech Recognition Activity Affect Query Suggestion Rankings?

Compared to QAC based on the automatic method, using ideal transcription positively affects the query suggestion rankings. We found that speech recognition errors can greatly change the content and results of query suggestions, resulting a declining retrieval performance for individual queries. Typical errors made by automatic speech-to-text translation systems are misinterpretation of voice inputs, producing incorrect recognized words, and occasionally missing words [37]. While this is by no means new, speech recognition errors in spoken queries can cause a significant drop in retrieval performance [5]. In fact, our study revealed that only 61.96% of the keywords were correct detection in the automatic transcription (refer to Table 1). We observed that the

performance degraded when actual query content mentioned in a conversation was different from the transcribed query. This is not surprising: the query suggestions are ineffective when the transcribed texts are likely to be incorrect.

While a spoken conversation would be longer in duration compared to a spoken query [21], it may also include terms that are not central to the query topic. In conversations, people talk enthusiastically and excessively, which poses a new challenge for speech-enabled search engines. For example, our study revealed a 44.67% WER in automatic transcription. Despite such speech recognition errors, the topic modeling approach still performs well for QAC. This is consistent with Arguello et al. [5]'s earlier findings in which topic modeling was a useful approach to address speech recognition errors in document retrieval, the performance improved by reducing the spoken query to only topically coherent terms while omitting non-topical terms that automatic speech recognition likely misinterpreted.

However, if all the speech recognition errors are corrected, ideally the average success rate of hitting the right query in top suggestions can be significantly improved (refer to Table 5). An additional keystroke can also be saved with our QAC model if ideal transcription becomes available. This suggests future advancement in speech recognition backed by high-quality speech signal acquisition (e.g., in a smart speaker) can enable even higher quality search applications, in particular the QAC.

8 CONCLUSIONS

We set out to study how incorporating users' spoken context affects QAC performance and reported an experiment evaluating context-aware query prediction models over QAC with no contextualization.

8.1 Summary of Contributions

Our contributions are both methodological and empirical. First, we proposed a methodology using temporal topic models trained in using the spoken conversational context to re-rank query autocompletion suggestions. To our knowledge, we also report the first approach demonstrating the effect of spoken conversational input for query re-ranking.

We also report an empirical study with a unique dataset including search logs associated with the spoken conversational pre-search context. We evaluate the performance of our approach by using the QAC method Google uses as a control condition and examining the utility of the spoken context for query prediction in experimental conditions.

Our results using real-world speech data show that drawing on the spoken context can significantly improve QAC performance even with very little input from users, an improvement that stays consistent over various lengths of spoken context information. The results of overall performance and character-level input analysis revealed that comprehensive context information is preferable in improving QAC, while our approach can work even with only a single character input that users provided.

8.2 Limitations

While our work shows that the spoken conversational context can improve QAC in a Web search, it also has limitations. The main limitation is that our analysis relies on Google's implementation of the automatic speech recognizer, whereas there may be other systems that can produce more accurate transcription. However, we believe our experimental setting represents the current state-of-the-art technology and provides a reliable evaluation environment with a real-world query suggestion engine. In sum, our work contributes novel insights on the potential of using spoken

conversational context to improve Web searches and opens new opportunities for speech-based QAC in the near future.

Another limitation is using Google QAC as a control condition. There could be other machine learning models that perform better than the Google service employs. Our aim was not to propose a new model in which it would need comparison with other state-of-the-art approaches, rather it was to study whether leveraging the spoken context can improve QAC. Here, we are more interested in the effects of the different data inputs instead of a particular model's performance. Therefore, every other variable (e.g., model choices or the impact of long-term user behavior prior to the experiments) was kept the same in all conditions. Comparing different machine learning models is a subject for future work.

The third limitation is the small size of our dataset. However, this kind of data is difficult to collect on a large scale due to privacy issues. We addressed this challenge by conducting our data collection experiment in a controlled laboratory environment; people who had given their informed consent could safely share their conversations and search logs for research purposes. We believe this makes our data and results highly valuable regardless of the size limitation.

Last, our approach relies on re-ranking a set of query suggestions originating from a commercial search engine provider. This means that we did not have control over the underlying query suggestion generation methods. However, showing improvements over a popular commercial search engine suggests that our results can challenge a strong real-world query prediction approach and thus have both theoretical and practical value.

8.3 Future Work

Our research leaves room for future work in several areas. First, although our results were extremely promising, we did not explore how interactive search systems, including support for speech-based QAC, may affect the user experience for people searching during conversations. Users learn to trust query suggestions and while our approach seems to be highly beneficial, learned usage patterns that users may be accustomed to can change as a result of utilizing the spoken context. Future research could also be conducted to investigate the opportunity to integrate our technique with other types of user models, contexts, and modalities to capture user preferences and context. For example, a spoken context contribution with more comprehensive data utilization from users' behavioral history could reveal additional insights about user interests and preferences that go beyond the context available shortly before searching [77]. More direct modalities, such as physiological interfaces [9], could be explored to reveal preferences that have not yet manifested as spoken conversation. Additionally, we have shown the number of keystrokes to enter a query decreases by showing only one query completion. Displaying more suggestions is more appropriate for query completion, but user-perceived improvement might degrade if more than one completion is shown. Users may take more time to find and select the desired query completion as the interface will become more cluttered. Quantifying the tradeoff of more suggestions and cognitive load is an interesting avenue for future research. Future research could also consider utilizing the spoken context to improve other search tasks. For example, intelligent systems, such as chat bots or voice assistants embedded in smart speakers, may suggest content and queries that are most relevant to the conversational context even before observing any explicit user request or input. This may lead to the creation of new types of proactive methods that provide opportunities to eliminate much of the laborious query input and reformulation that currently sets the burden on users instead of proactive intelligence embedded in the systems serving users [47]. Finally, obtaining user information behavior in spoken conversations while still protecting users' privacy can also be interesting research. For example, although most existing speech-to-text systems aim to ensure privacy while processing users' voice in the cloud, it is still a difficult endeavor

for current technologies. With computing resources becoming cheaper and more powerful, spoken data can be processed locally, where a transcript is generated, rather than being transmitted to a centralized data processing center. This may help reduce the current dependency on Internet connectivity, while it also can mitigate the necessity of putting sensitive data into the cloud.

REFERENCES

[1] Salvatore Andolina, Khalil Klouche, Tuukka Ruotsalo, Patrik Floréen, and Giulio Jacucci. 2018. Querytogether: Enabling entity-centric exploration in multi-device collaborative search. *Info. Process. Manage.* 54, 6 (2018), 1182–1202. DOI: https://doi.org/10.1016/j.ipm.2018.04.005

[2] Salvatore Andolina, Valeria Orso, Hendrik Schneider, Khalil Klouche, Tuukka Ruotsalo, Luciano Gamberini, and Giulio Jacucci. 2018. Investigating proactive search support in conversations. In *Proceedings of the Designing Interactive Systems Conference (DIS'18)*. ACM, New York, NY, 1295–1307. DOI: https://doi.org/10.1145/3196709.3196734

[3] Salvatore Andolina, Valeria Orso, Hendrik Schneider, Khalil Klouche, Tuukka Ruotsalo, Luciano Gamberini, and Giulio Jacucci. 2018. SearchBot: Supporting voice conversations with proactive search. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work and Social Computing*. 9–12.

[4] Charles E. Antoniak. 1974. Mixtures of Dirichlet processes with applications to bayesian nonparametric problems. *Ann. Statist.* 2, 6 (11 1974), 1152–1174. DOI: https://doi.org/10.1214/aos/1176342871

[5] Jaime Arguello, Sandeep Avula, and Fernando Diaz. 2017. Using query performance predictors to reduce spoken queries. In *Advances in Information Retrieval*. Springer International Publishing, Cham, 27–39. DOI: https://doi.org/10.1007/978-3-319-56608-5_3

[6] Amos Azaria and Jason Hong. 2016. Recommender systems with personality. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys'16)*. ACM, New York, NY, 207–210. DOI: https://doi.org/10.1145/2959100.2959138

[7] Ziv Bar-Yossef and Naama Kraus. 2011. Context-sensitive query auto-completion. In *Proceedings of the 20th International Conference on World Wide Web (WWW'11)*. ACM, New York, NY, 107–116. DOI: https://doi.org/10.1145/1963405.1963424

[8] Ziv Bar-Yossef and Naama Kraus. 2011. Context-sensitive query auto-completion. In *Proceedings of the 20th International Conference on World Wide Web (WWW'11)*. ACM, New York, NY, 107–116. DOI: https://doi.org/10.1145/1963405.1963424

[9] Oswald Barral, Ilkka Kosunen, Tuukka Ruotsalo, Michiel M. Spapé, Manuel J. A. Eugster, Niklas Ravaja, Samuel Kaski, and Giulio Jacucci. 2016. Extracting relevance and affect information from physiological text annotation. *User Model. User-Adapt. Interact.* 26, 5 (2016), 493–520.

[10] Holger Bast, Debapriyo Majumdar, and Ingmar Weber. 2007. Efficient interactive query expansion with complete search. In *Proceedings of the 16th ACM Conference on Conference on Information and Knowledge Management (CIKM'07)*. Association for Computing Machinery, New York, NY, 857–860. DOI: https://doi.org/10.1145/1321440.1321560

[11] Holger Bast and Ingmar Weber. 2006. Type less, find more: Fast auto-completion search with a succinct index. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'06)*. Association for Computing Machinery, New York, NY, 364–371. DOI: https://doi.org/10.1145/1148170.1148234

[12] N. J. Belkin, H. M. Brooks, and P. J. Daniels. 1987. Knowledge elicitation using discourse analysis. *Int. J. Man-Mach. Studies* 27, 2 (1987), 127–144. DOI: https://doi.org/10.1016/S0020-7373(87)80047-0

[13] Sumit Bhatia, Debapriyo Majumdar, and Prasenjit Mitra. 2011. Query suggestions in the absence of query logs. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'11)*. ACM, New York, NY, 795–804. DOI: https://doi.org/10.1145/2009916.2010023

[14] Steffen Bickel, Peter Haider, and Tobias Scheffer. 2005. Learning to complete sentences. In *Machine Learning: ECML 2005*, João Gama, Rui Camacho, Pavel B. Brazdil, Alípio Mário Jorge, and Luís Torgo (Eds.). Springer, Berlin, 497–504.

[15] Barry Brown, Moira McGregor, and Donald McMillan. 2015. Searchable objects: Search in everyday conversation. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work and Social Computing (CSCW'15)*. ACM, New York, NY, 508–517. DOI: https://doi.org/10.1145/2675133.2675206

[16] Fei Cai and Maarten de Rijke. 2016. Selectively personalizing query auto-completion. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'16)*. ACM, New York, NY, 993–996. DOI: https://doi.org/10.1145/2911451.2914686

[17] Fei Cai, Maarten De Rijke et al. 2016. A survey of query auto-completion in information retrieval. *Found. Trends Info. Retriev.* 10, 4 (2016), 273–363.

[18] Fei Cai, Ridho Reinanda, and Maarten De Rijke. 2016. Diversifying query auto-completion. *ACM Trans. Info. Syst.* 34, 4, Article 25 (June 2016), 33 pages. DOI: https://doi.org/10.1145/2910579

[19] Giuseppe Carenini, Jocelyin Smith, and David Poole. 2003. Towards more conversational and collaborative recommender systems. In *Proceedings of the 8th International Conference on Intelligent User Interfaces (IUI'03)*. ACM, New York, NY, 12–18. DOI: https://doi.org/10.1145/604045.604052

[20] Surajit Chaudhuri and Raghav Kaushik. 2009. Extending auto-completion to tolerate errors. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD'09)*. Association for Computing Machinery, New York, NY, 707–718. DOI: https://doi.org/10.1145/1559845.1559919

[21] Fabio Crestani and Heather Du. 2006. Written versus spoken queries: A qualitative and quantitative comparative analysis. *J. Amer. Soc. Info. Sci. Technol.* 57, 7 (2006), 881–890. DOI: https://doi.org/10.1002/asi.20350 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/asi.20350

[22] Jeffrey Dalton, Victor Ajayi, and Richard Main. 2018. Vote goat: Conversational movie recommendation. In *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'18)*. Association for Computing Machinery, New York, NY, 1285–1288. DOI: https://doi.org/10.1145/3209978.3210168

[23] Nan Du, Mehrdad Farajtabar, Amr Ahmed, Alexander J. Smola, and Le Song. 2015. Dirichlet-hawkes processes with applications to clustering continuous-time document streams. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'15)*. ACM, New York, NY, 219–228. DOI: https://doi.org/10.1145/2783258.2783411

[24] J. Fan, H. Wu, G. Li, and L. Zhou. 2010. Suggesting topic-based query terms as you type. In *Proceedings of the 12th International Asia-Pacific Web Conference*. 61–67. DOI: https://doi.org/10.1109/APWeb.2010.13

[25] Henry Feild and James Allan. 2013. Task-aware query recommendation. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'13)*. Association for Computing Machinery, New York, NY, 83–92. DOI: https://doi.org/10.1145/2484028.2484069

[26] Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A Deep Semantic Natural Language Processing Platform. Retrieved from https://arxiv:cs.CL/1803.07640.

[27] Korinna Grabski and Tobias Scheffer. 2004. Sentence completion. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'04)*. Association for Computing Machinery, New York, NY, 433–439. DOI: https://doi.org/10.1145/1008992.1009066

[28] Ido Guy. 2016. Searching by talking: Analysis of voice queries on mobile web search. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'16)*. Association for Computing Machinery, New York, NY, 35–44. DOI: https://doi.org/10.1145/2911451.2911525

[29] Ido Guy. 2018. The characteristics of voice search: Comparing spoken with typed-in mobile web search queries. *ACM Trans. Info. Syst.* 36, 3, Article 30 (Mar. 2018), 28 pages. DOI: https://doi.org/10.1145/3182163

[30] Jacek Gwizdka. 2010. Distribution of cognitive load in Web search. *J. Amer. Soc. Info. Sci. Technol.* 61, 11 (2010), 2167–2187. DOI: https://doi.org/10.1002/asi.21385 Retrieved from arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/asi.21385.

[31] Ahmed Hassan Awadallah, Ranjitha Gurunath Kulkarni, Umut Ozertem, and Rosie Jones. 2015. Characterizing and predicting voice query reformulation. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (CIKM'15)*. Association for Computing Machinery, New York, NY, 543–552. DOI: https://doi.org/10.1145/2806416.2806491

[32] Alan G. Hawkes. 1971. Spectra of some self-exciting and mutually exciting point processes. *Biometrika* 58, 1 (1971), 83–90. Retrieved from http://www.jstor.org/stable/2334319.

[33] Marti A. Hearst. 2011. "Natural" search user interfaces. *Commun. ACM* 54, 11 (Nov. 2011), 60–67. DOI: https://doi.org/10.1145/2018396.2018414

[34] Larry Heck, Dilek Hakkani-Tür, Madhu Chinthakunta, Gokhan Tur, Rukmini Iyer, Partha Parthasacarthy, Lisa Stifelman, Elizabeth Shriberg, and Ashley Fidler. 2013. Multimodal conversational search and browse. IEEE Workshop on Speech, Language and Audio in Multimedia. Retrieved from https://www.microsoft.com/en-us/research/publication/multimodal-conversational-search-and-browse/.

[35] Bo-June (Paul) Hsu and Giuseppe Ottaviano. 2013. Space-efficient data structures for top-k completion. In *Proceedings of the 22nd International Conference on World Wide Web (WWW'13)*. Association for Computing Machinery, New York, NY, 583–594. DOI: https://doi.org/10.1145/2488388.2488440

[36] Jeff Huang and Efthimis N. Efthimiadis. 2009. Analyzing and evaluating query reformulation strategies in web search logs. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM'09)*. Association for Computing Machinery, New York, NY, 77–86. DOI: https://doi.org/10.1145/1645953.1645966

[37] Jiepu Jiang, Wei Jeng, and Daqing He. 2013. How do users respond to voice input errors? Lexical and phonetic query reformulation in voice search. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'13)*. Association for Computing Machinery, New York, NY, 143–152. DOI: https://doi.org/10.1145/2484028.2484092

[38] Jyun-Yu Jiang, Yen-Yu Ke, Pao-Yu Chien, and Pu-Jen Cheng. 2014. Learning user reformulation behavior for query auto-completion. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'14)*. ACM, New York, NY, 445–454. DOI: https://doi.org/10.1145/2600428.2609614

[39] Jyun-Yu Jiang, Yen-Yu Ke, Pao-Yu Chien, and Pu-Jen Cheng. 2014. Learning user reformulation behavior for query auto-completion. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'14)*. ACM, New York, NY, 445–454. DOI: https://doi.org/10.1145/2600428.2609614

[40] Michael Johnston, Srinivas Bangalore, Gunaranjan Vasireddy, Amanda Stent, Patrick Ehlen, Marilyn Walker, Steve Whittaker, and Preetam Maloor. 2002. MATCH: An architecture for multimodal dialogue systems. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Philadelphia, Pennsylvania, 376–383. DOI: https://doi.org/10.3115/1073083.1073146

[41] Maryam Kamvar and Shumeet Baluja. 2007. The role of context in query input: Using contextual signals to complete queries on mobile devices. In *Proceedings of the 9th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI'07)*. ACM, New York, NY, 405–412. DOI: https://doi.org/10.1145/1377999.1378046

[42] Lauri Kangassalo, Michiel Spapé, Giulio Jacucci, and Tuukka Ruotsalo. 2019. Why do users issue good queries? Neural correlates of term specificity. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 375–384.

[43] Lauri Kangassalo, Michiel Spapé, Niklas Ravaja, and Tuukka Ruotsalo. 2020. information gain modulates brain activity evoked by reading. *Sci. Rep.* 10, 1 (2020), 1–10.

[44] Julia Kiseleva, Kyle Williams, Ahmed Hassan Awadallah, Aidan C. Crook, Imed Zitouni, and Tasos Anastasakos. 2016. Predicting user satisfaction with intelligent assistants. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'16)*. Association for Computing Machinery, New York, NY, 45–54. DOI: https://doi.org/10.1145/2911451.2911521

[45] Khalil Klouche, Tuukka Ruotsalo, Diogo Cabral, Salvatore Andolina, Andrea Bellucci, and Giulio Jacucci. 2015. Designing for exploratory search on touch devices. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI'15)*. ACM, New York, NY, 4189–4198. DOI: https://doi.org/10.1145/2702123.2702489

[46] Weize Kong, Rui Li, Jie Luo, Aston Zhang, Yi Chang, and James Allan. 2015. Predicting search intent based on pre-search context. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'15)*. ACM, New York, NY, 503–512. DOI: https://doi.org/10.1145/2766462.2767757

[47] Markus Koskela, Petri Luukkonen, Tuukka Ruotsalo, Mats Sjoberg, and Patrik Floréen. 2018. Proactive information retrieval by capturing search intent from primary task context. *ACM Trans. Interact. Intell. Syst.* 8, 3 (2018), 1–25.

[48] Unni Krishnan, Alistair Moffat, and Justin Zobel. 2017. A taxonomy of query auto-completion modes. In *Proceedings of the 22Nd Australasian Document Computing Symposium (ADCS'17)*. ACM, New York, NY, Article 6, 8 pages. DOI: https://doi.org/10.1145/3166072.3166081

[49] Liangda Li, Hongbo Deng, Jianhui Chen, and Yi Chang. 2017. Learning parametric models for context-aware query auto-completion via Hawkes processes. In *Proceedings of the 10th ACM International Conference on Web Search and Data Mining (WSDM'17)*. ACM, New York, NY, 131–139. DOI: https://doi.org/10.1145/3018661.3018698

[50] Yanen Li, Anlei Dong, Hongning Wang, Hongbo Deng, Yi Chang, and ChengXiang Zhai. 2014. A two-dimensional click model for query auto-completion. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'14)*. ACM, New York, NY, 455–464. DOI: https://doi.org/10.1145/2600428.2609571

[51] Jonathan Mamou, David Carmel, and Ron Hoory. 2006. Spoken document retrieval from call-center conversations. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'06)*. Association for Computing Machinery, New York, NY, USA, 51–58. https://doi.org/10.1145/1148170.1148183

[52] Gary Marchionini and Ryen White. 2007. Find what you need, understand what you find. *Int. J. Hum.–Comput. Interact.* 23, 3 (2007), 205–237. DOI: https://doi.org/10.1080/10447310701702352 arXiv:https://doi.org/10.1080/10447310701702352

[53] Donald McMillan, Antoine Loriette, and Barry Brown. 2015. Repurposing conversation: Experiments with the continuous speech stream. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI'15)*. ACM, New York, NY, 3953–3962. DOI: https://doi.org/10.1145/2702123.2702532

[54] Michael F. McTear. 2002. Spoken dialogue technology: Enabling the conversational user interface. *ACM Comput. Surveys* 34, 1 (2002), 90–169.

[55] Gilad Mishne, David Carmel, Ron Hoory, Alexey Roytman, and Aya Soffer. 2005. Automatic analysis of call-center conversations. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management (CIKM'05)*. Association for Computing Machinery, New York, NY, 453–459. DOI: https://doi.org/10.1145/1099554.1099684

[56] A. Moreno-Daniel, S. Parthasarathy, B. H. Juang, and J. G. Wilpon. 2007. Spoken query processing for information retrieval. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'07)*, Vol. 4. IV–121–IV–124. DOI: https://doi.org/10.1109/ICASSP.2007.367178

[57] Andrew Morris. 2002. An information theoretic measure of sequence recognition performance. http://infoscience.epfl.ch/record/82766.

[58] Arnab Nandi and H. V. Jagadish. 2007. Effective phrase prediction. In *Proceedings of the 33rd International Conference on Very Large Data Bases (VLDB'07)*. VLDB Endowment, 219–230.

[59] Matteo Negri, Marco Turchi, José G. C. de Souza, and Daniele Falavigna. 2014. Quality estimation for automatic speech recognition. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING'14)*. 1813–1823. Retrieved from http://aclweb.org/anthology/C/C14/C14-1171.pdf.

[60] Douglas W. Oard. 2012. Query by babbling: A research agenda. In *Proceedings of the 1st Workshop on Information and Knowledge Management for Developing Region (IKM4DR'12)*. Association for Computing Machinery, New York, NY, 17–22. DOI: https://doi.org/10.1145/2389776.2389781

[61] Florian Pecune, Shruti Murali, Vivian Tsai, Yoichi Matsuyama, and Justine Cassell. 2019. A model of social explanations for a conversational movie recommendation system. In *Proceedings of the 7th International Conference on Human-Agent Interaction*. ACM, 135–143.

[62] Matthew E. Peters and Dan Lecocq. 2013. Content extraction using diverse feature sets. In *Proceedings of the 22nd International Conference on World Wide Web (WWW'13)*. Association for Computing Machinery, New York, NY, 89–90. DOI: https://doi.org/10.1145/2487788.2487828

[63] Chen Qu, Liu Yang, W. Bruce Croft, Yongfeng Zhang, Johanne R. Trippas, and Minghui Qiu. 2019. User intent prediction in information-seeking conversations. In *Proceedings of the Conference on Human Information Interaction and Retrieval (CHIIR'19)*. Association for Computing Machinery, New York, NY, 25–33. DOI: https://doi.org/10.1145/3295750.3298924

[64] Filip Radlinski and Nick Craswell. 2017. A theoretical framework for conversational search. In *Proceedings of the Conference on Conference Human Information Interaction and Retrieval (CHIIR'17)*. ACM, New York, NY, 117–126. DOI: https://doi.org/10.1145/3020165.3020183

[65] Jinfeng Rao, Ferhan Ture, and Jimmy Lin. 2018. What do viewers say to their TVs? An analysis of voice queries to entertainment systems. In *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'18)*. Association for Computing Machinery, New York, NY, 1213–1216. DOI: https://doi.org/10.1145/3209978.3210140

[66] Soo Young Rieh and Hong (Iris) Xie. 2006. Analysis of multiple query reformulations on the web: The interactive information retrieval context. *Info. Process. Manage.* 42, 3 (May 2006), 751–768. DOI: https://doi.org/10.1016/j.ipm.2005.05.005

[67] Tuukka Ruotsalo, Giulio Jacucci, Petri Myllymäki, and Samuel Kaski. 2014. Interactive intent modeling: Information discovery beyond search. *Commun. ACM* 58, 1 (2014), 86–92.

[68] Tuukka Ruotsalo, Jaakko Peltonen, Manuel J. A. Eugster, Dorota Głowacka, Patrik Floréen, Petri Myllymäki, Giulio Jacucci, and Samuel Kaski. 2018. Interactive intent modeling for exploratory search. *ACM Trans. Info. Syst.* 36, 4, Article 44 (Oct. 2018), 46 pages. DOI: https://doi.org/10.1145/3231593

[69] Ning Sa. 2016. Improving query reformulation in voice search system. In *Proceedings of the ACM on Conference on Human Information Interaction and Retrieval (CHIIR'16)*. Association for Computing Machinery, New York, NY, 365–367. DOI: https://doi.org/10.1145/2854946.2854951

[70] Johan Schalkwyk, Doug Beeferman, Françoise Beaufays, Bill Byrne, Ciprian Chelba, Mike Cohen, Maryam Kamvar, and Brian Strope. 2010. *"Your Word is My Command": Google Search by Voice: A Case Study*. Springer US, Boston, MA, 61–90. DOI: https://doi.org/10.1007/978-1-4419-5951-5_4

[71] Glenn Shafer. 2016. Dempster's rule of combination. *Int. J. Approx. Reason.* 79 (2016), 26–40. DOI: https://doi.org/10.1016/j.ijar.2015.12.009 40 years of Research on Dempster-Shafer Theory.

[72] J. P. Shaffer. 1995. Multiple hypothesis testing. *Annu. Rev. Psychol.* 46, 1 (1995), 561–584. DOI: https://doi.org/10.1146/annurev.ps.46.020195.003021 arXiv:https://doi.org/10.1146/annurev.ps.46.020195.003021

[73] Sosuke Shiga, Hideo Joho, Roi Blanco, Johanne R. Trippas, and Mark Sanderson. 2017. Modelling information needs in collaborative search conversations. In *Proceedings of the 40th International ACM SIGIR Conference on Research*

and Development in Information Retrieval (SIGIR'17). Association for Computing Machinery, New York, NY, 715–724. DOI: https://doi.org/10.1145/3077136.3080787

[74] Milad Shokouhi. 2013. Learning to personalize query auto-completion. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'13)*. ACM, New York, NY, 103–112. DOI: https://doi.org/10.1145/2484028.2484076

[75] Milad Shokouhi and Qi Guo. 2015. From queries to cards: Re-ranking proactive card recommendations based on reactive search history. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'15)*. Association for Computing Machinery, New York, NY, 695–704. DOI: https://doi.org/10.1145/2766462.2767705

[76] Milad Shokouhi, Rosie Jones, Umut Ozertem, Karthik Raghunathan, and Fernando Diaz. 2014. Mobile query reformulations. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'14)*. Association for Computing Machinery, New York, NY, 1011–1014. DOI: https://doi.org/10.1145/2600428.2609497

[77] 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 on Conference on Information and Knowledge Management*. 553–562.

[78] Johanne R. Trippas, Damiano Spina, Lawrence Cavedon, Hideo Joho, and Mark Sanderson. 2018. Informing the design of spoken conversational search: Perspective paper. In *Proceedings of the Conference on Human Information Interaction and Retrieval (CHIIR'18)*. ACM, New York, NY, 32–41. DOI: https://doi.org/10.1145/3176349.3176387

[79] Tung Vuong, Miamaria Saastamoinen, Giulio Jacucci, and Tuukka Ruotsalo. 2019. Understanding user behavior in naturalistic information search tasks. *J. Assoc. Info. Sci. Technol.* 70, 11 (2019), 1248–1261.

[80] Wolfgang Wahlster. 2006. *SmartKom: Foundations of Multimodal Dialogue Systems (Cognitive Technologies)*. Springer-Verlag, Berlin.

Received March 2020; revised December 2020; accepted January 2021