

Wizard of Oz Interface to Study System Initiative for Conversational Search

Sandeep Avula and Jaime Arguello
University of North Carolina at Chapel Hill
[asandeep,jarguell]@unc.edu

ABSTRACT

We describe a Wizard of Oz (WoZ) system, a WebApp, which we use to study how a conversational search system should take the initiative when engaging with users during collaborative search. This system integrates directly into Slack, a chat-messaging platform where users will collaborate. Through our system, the Wizard plays the role of a conversational search system that can search for information, send relevant web results, and message users. In our research, we study three Wizard conditions: *bot_info*, *bot_dialog*, and *bot_task*, which differ in terms of how the Wizard can intervene in a conversation. The intervention modes follow the mixed-initiative framework by Chu-Carroll and Brown [9] and provide us a foundation to study system initiative for conversational search. In this paper, we describe our design decisions and technical details on how we implemented the system.

KEYWORDS

Wizard of Oz; Mixed-Initiative Interactions; Conversational Search; Collaborative Search

ACM Reference Format:

Sandeep Avula and Jaime Arguello. 2020. Wizard of Oz Interface to Study System Initiative for Conversational Search. In *2020 Conference on Human Information Interaction and Retrieval (CHIIR '20)*, March 14–18, 2020, Vancouver, BC, Canada. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3343413.3377941>

1 INTRODUCTION

Conversational search as a research space has gained momentum in the last couple of years and is rapidly expanding. Recent work has focused on conceptualizing the capabilities of a conversational search system [22, 26], as well as implementing functions such as preference elicitation [8, 25] and information need clarification [1, 31]. In our own work, we have explored conversational search as a means to support collaborative search [3, 4]. One aspect of this research that has not received much attention, especially in Interactive Information Retrieval (IIR), is how a system should engage in a mixed-initiative interaction with users. With current Information Retrieval systems, the initiative is almost always one-sided, i.e., it either rests with the user when they issue search queries, or with the system when it is capable of asking clarification questions [1, 31] or eliciting preferences [8, 25]. Radlinski and Craswell [22], in their

framework of conversational search, identify this aspect of interaction (system taking initiative) and suggest that for a system to truly engage in a conversation, as humans do, it must be capable of dynamically exchanging initiative with the user. In this paper, we describe an interface to conduct a Wizard of Oz (WoZ) study to understand how a system should take the initiative when engaging with users over information-seeking tasks.

2 STUDY MOTIVATION

In our research, we investigate *how* and *when* a search system should take the initiative to support users during collaborative search. Traditionally, collaborative search, which happens when people work together on a shared information-seeking task, has been supported by making dedicated systems [6, 13, 20, 21, 23, 30]. These systems have been designed with the search engine as the centerpiece component, but include peripheral tools for collaborators to communicate, share information, and gain awareness of each other's activities. Though these systems have been found to provide benefits for users, stand-alone systems have not gained widespread adoption [19]. Research suggests that while people frequently engage in collaborative search, they do so using non-integrated tools—-independent search systems and communication tools such as instant messaging, email, and phone [7, 19]. Morris [19] and Hearst [15] highlighted these findings as a rationale to develop lightweight search tools that can be integrated directly into existing communication platforms.

In our own work, we followed Morris and Hearst's call to build search tools inside a communication platform. In Avula et al. [4], we explored a design space where a proactive search agent would intervene in conversations, based on strict protocols, to support collaborative search. In Avula et al. [3], we explored a design space for integrating search tools into a communication platform (Slack). We explored three conditions: (1) search tool outside Slack, (2) search tool inside Slack, and (3) search tool both inside and outside of Slack. While we observed advantages in both studies, there were also critical challenges. In Avula et al. [4], participants noted that the timing of the intervention was crucial. Participants reported avoiding the search agent when the intervention was too soon (before understanding the task), too late (after solving the task), or during periods when they were deeply engaged with other tasks. In Avula et al. [3], while participants noted that sharing the same search environment was beneficial for collaborative search, searching inside the chat platform led to less exploration and was also distracting at times.

Observations from both studies suggest that integrating search tools inside a communication platform is complex, requiring a deeper understanding of the *how* and *when* to intervene. In this work, we introduce a WoZ system that relaxes the constraints from our prior research to understand the fundamental question on *how*

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

CHIIR '20, March 14–18, 2020, Vancouver, BC, Canada

© 2020 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-6892-6/20/03.

<https://doi.org/10.1145/3343413.3377941>

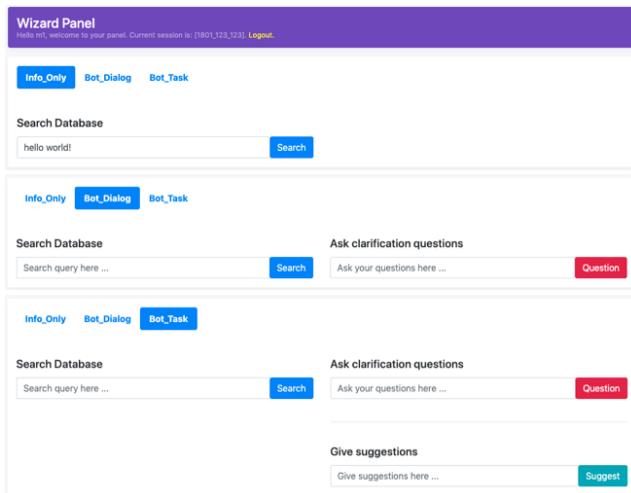


Figure 1: Wizard’s interface across the three conditions: *bot_info* (top), *bot_dialog* (middle), and *bot_task* (bottom) and when a search system should take the initiative to support collaborative search. Insights from our work can help advance research in collaborative search, especially the effect of system initiative on collaborations. Furthermore, our work also contributes to the field of conversational search, generally set in a single-user and single-agent setting, about single-agent multi-user interactions.

2.1 Why Wizard of Oz?

We conduct Wizard of Oz (WoZ) studies to mimic hypothetical systems that are not yet in existence or are bound by technical limitations such as imperfect automatic speech recognition and dialog management [16, 18, 24]. WoZ studies help us come up with frameworks within which the hypothetical systems may operate and also allow us to anticipate opportunities and challenges. Mixed-Initiative interactions too fall within this space of investigation, where we currently do not have systems that can accurately take the initiative to assist users with information seeking.

In this research, we investigate *how* and *when* a search system should take the initiative to support collaborative search. To study this, we follow Chu-Carroll and Brown’s [9] work to characterize and operationalize system initiative at two levels: dialog and task. At the dialog level (referred to as the *bot_dialog* condition), the Wizard can take the initiative by asking clarification questions to provide better search results in response to a user’s request. At the task level (referred to as the *bot_task* condition), the Wizard can take dialogue initiative, but also provide suggestions that may influence users to approach the task differently. In addition to these two conditions, we also set a baseline condition called *bot_info*. In this baseline condition, the Wizard behaves as a proxy to a search engine, by taking the users’ information request, run it against our search engine and forward them a relevant result. Overall, we summarize the different conditions as follows: (1) *bot_info*: search; (2) *bot_dialog*: search + clarification questions; and (3) *bot_task*: search + clarification questions + suggestions.

3 PRIOR DESIGN APPROACHES

In this section, we describe recent interface designs in WoZ studies. We focus on two aspects of these interfaces: (1) how the system

allows the Wizard to engage with users, and (2) how the system allows the Wizard to search for information.

3.1 Facilitating dialog engagement

There have been two primary approaches for Conversational Search research focusing on the dialog between a user seeking information, also referred to as a seeker, and a Wizard: *continuous turn-by-turn*, and *intermittent turn-by-turn*. These two approaches are similar in terms of their conversational structure, which are turn-by-turn. In a turn-by-turn structure, each participant produces an utterance alternatively, thereby giving away initiative after each turn. Coming to their differences, *continuous turn-by-turn*, and *intermittent turn-by-turn* dialogs diverge in terms of the continuity of the dialog and its contributors.

In the *continuous turn-by-turn* approach, pairs of users are recruited to engage over a messaging platform (chat or voice) for an information-seeking task [2, 11, 17, 27, 28]. One user plays the role of a seeker, and the other is the Wizard, who has access to a search system. Both engage in a turn-by-turn conversation to resolve the seeker’s information needs.

In *intermittent turn-by-turn* conversations, different users are recruited to contribute per turn to the conversation [5, 12, 29]. Rather than pairing users to engage over a messaging platform, each user is directed to a web page, where they are assigned a role, seeker or Wizard, and a transcript of an ongoing yet incomplete conversation. In this setting, the Wizard can study the conversation, without a time constraint, and make a meaningful contribution that advances the dialog to resolve the seeker’s information needs.

3.2 Access to external resources

In terms of access to external resources, to search for information, recent work has taken two approaches to do this: *search-in-conversation*, and *search-out-conversation*.

In the *search-in-conversation* approach, the Wizard can look up information within the messaging platform itself [17]. In Li et al. [17], the Wizard could preface a message with a trigger such as the “@” symbol, which makes the system promptly suggest movie names based on the characters the Wizard enters following the trigger command. The Wizard can immediately select an option that is most suitable and push that suggestion into the messaging channel they share with the user. This approach is only applicable in *continuous turn-by-turn* interactions.

In the *search-out-conversation* approach, the Wizard has access to a search interface that is separate from the messaging platform. In *continuous turn-by-turn* interactions, this means that the Wizard has access to both a messaging platform and a search interface [2, 27, 28]. After finding relevant information, the Wizard has to either copy-paste the relevant web-link or extract a relevant passage and send that in the messaging platform. In *intermittent turn-by-turn* conversations, users are given their own search interfaces or lookup engines to search for information to contribute to the dialog [5, 12, 29]. Here, technically, the dialog and the search engine appear within the same layout, but since the conversation is not synchronous and persistent, we categorize them as being separated from the conversation.

4 GUIDELINES AND PROTOCOLS

Based on the design approaches in Section 3, we describe design guidelines that are suitable for our WoZ study. We also describe the Wizard's protocols.

4.1 Guidelines

We follow seven guidelines to build the Wizard's interface.

Separate conversations and search: Provide the Wizard with two screens, one to monitor the conversations and the other to search and interact with the users. The Wizard can only follow the conversation on the messaging platform and not interact in it. The Wizard interacts with the users from the second screen.

Make search prominent: Across the different intervention conditions, the Wizard has a search bar that is prominently displayed. The search results should appear right below the search bar, without any page reloads, along with options to paginate.

Feedback on relevant actions: Each time the Wizard performs an action that is critical to interacting with the user, there should be easy to comprehend visual feedback. The following actions fall in this category: (1) users' information request; (2) Wizard's ability to search; (3) pushing search results into the users' messaging channel; and (4) sending chat messages to the users.

Distinguish the Wizard from the users: Make it explicitly clear to the users that the Wizard is not one of them. Add visual markers to requests made to the Wizard and to the messages the Wizard sends to the users. In addition, remove cues that provide awareness of the Wizard's typing activity, for example, by removing the "typing dots" that appear when one person is typing out a message.

Facilitate collaborative information seeking dialogs: To encourage rich conversations between the users and the Wizard, limit the number of results that the Wizard can send them. To do this, do not provide a gateway for participants to search on their own, independently outside of the chat channel.

Ease of sending results: The Wizard should not have to copy-paste search results into the messaging platform. They should be able to effortlessly forward relevant results to the users.

Explicit division of interface elements: Every interface element should have its specific purpose and must not overflow into another. At the Wizard's end, this means that the ability to search, paginate, push a search result, and engage in the conversations should be visually and functionally separated.

4.2 Protocols

In this research, the Wizard operates in three intervention conditions, and to do this, we provide them with certain protocols.

bot_info: In this condition, users provide the Wizard with an information request by prefacing a message with "@max." Here, "@max" is the Wizard's handle inside Slack. The Wizard uses the search interface we provide to issue the users' query and sends them one relevant result.

bot_dialog: This condition builds on the *bot_info* condition by giving the Wizard the ability to ask follow-up questions. The Wizard may ask questions to elicit the users' preferences or clarify their information needs. The Wizard can only ask a question after the users' request for information. Once they send the users' a search

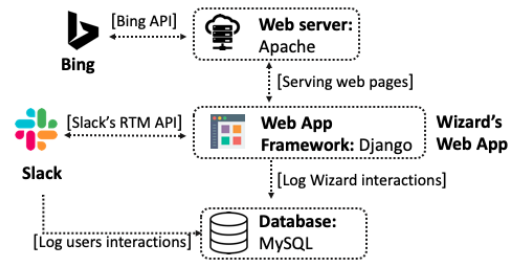


Figure 2: Architecture overview

result, they cannot interact with the users until the users request more information.

bot_task: This condition builds on the *bot_dialog* condition by giving the Wizard the ability to provide the users with suggestions. The Wizard may suggest to the users that the task direction they are pursuing may not be ideal or even suggest better alternatives. In this condition, the Wizard can choose to ask questions or provide suggestions at any point in the conversation.

5 ARCHITECTURE OVERVIEW

As shown in Figure 2, interactions happen in two main components: Slack, where messaging between the users and the Wizard takes place; and the Web App, used by the Wizard to search and interact with users. Below, we describe the architecture of these two components.

5.1 Wizard's Web App

The Wizard interacts with users through a WebApp. This App was built using Django, a high-level python based web framework. Through this App, the Wizard can do three things: (1) search for information; (2) send specific results to the Slack channel; and (3) message the users in the Slack channel.

To search for information, we created a Bing API-based search engine, as shown in Figure 3. When the Wizard issues a query, we retrieve the top 50 results using the Bing Web API and paginate with ten results per page. In this setup, we followed the guidelines in Section 4.1, and put a *send* button beside each retrieved result. This button facilitates the Wizard to *send* a specific result to the users' Slack channel instantly. We believe this to be better than having the Wizard manually copy paste a URL or a snippet from a webpage into the Slack channel. Next, as described in Section 4.2, the Wizard can intervene in the users' Slack channel based on the different intervention conditions. To facilitate this, based on their intervention condition, the Wizard can type a message or search query into an input cell (Figure 1).

In the *bot_info* condition, they can only search for the users. Therefore, they only have access to a search bar where they can type their search queries. In the *bot_dialog* condition, in addition to searching for the users, the Wizard can ask follow-up questions. For this purpose, they have a search bar and an input cell to enter their questions. Their submitted questions appear as a message from the Wizard in the users' Slack channel (Figure 4). Finally, in the *bot_task* condition, in addition to the capabilities in the *bot_dialog* condition, the Wizard can also provide suggestions. Therefore, they have a search bar and two more input cells, one for asking questions and the other to provide suggestions. In this condition too, the Wizard's suggestions and questions once submitted appear as a message

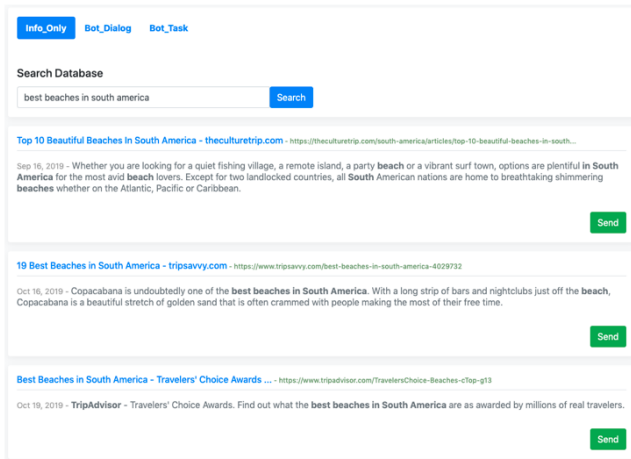


Figure 3: Wizard’s search results page

from the Wizard in the users’ Slack channel (Figure 4). We connect the input messages on the WebApp to Slack using Slack’s Real Time Messaging (RTM) API.

The WebApp uses Slack’s RTM API to send messages and web results to the users’ Slack channel. We use Apache and MySQL to host the WebApp and store the interactions.

5.2 Users @ Slack

While on Slack, users can do three things: (1) engage in a text-based conversation; (2) request information or provide feedback to the Wizard; and (3) explore the search results sent by the Wizard.

We use Slack as our messaging platform to have the users engage in a dialog with themselves and the Wizard. We chose Slack as it allows for easy integrations for custom third-party applications, which in our case, is the WebApp we made for the Wizard. To request information from the Wizard or to provide feedback from the Wizard, users use Slack’s “@” functionality (Figure 4). This functionality creates quick visual feedback for users and the Wizard about when users are exchanging messages between themselves versus interacting with the Wizard, to request information or respond to a Wizard-initiated question/suggestion.

Next, the search result the Wizard pushes to the users appears as a Slack attachment in the users’ channel. This attachment has the display-url on the top, followed by the title, snippet, and finally date of document retrieved (Figure 4). When users click on the search result, it opens in a default browser. We use Slack’s RTM API to log the users chat messages. User clicks on the search results are also logged and stored in the same MySQL database we use to log the Wizard’s search behaviors.

6 WHAT CAN YOU DO WITH OUR SYSTEM?

We will make the code public, so you can use the system as-is to research problems related to system initiative. Furthermore, the system is integrated into Slack in a manner where multiple Wizards can take part in a conversation with any number of users. So, studying system initiative in a multi-user multi-agent scenario is possible. We also encourage interested people to use annotated data sets such as [1, 10, 14], which have questions aligned with different information needs. These data sets could be used with our

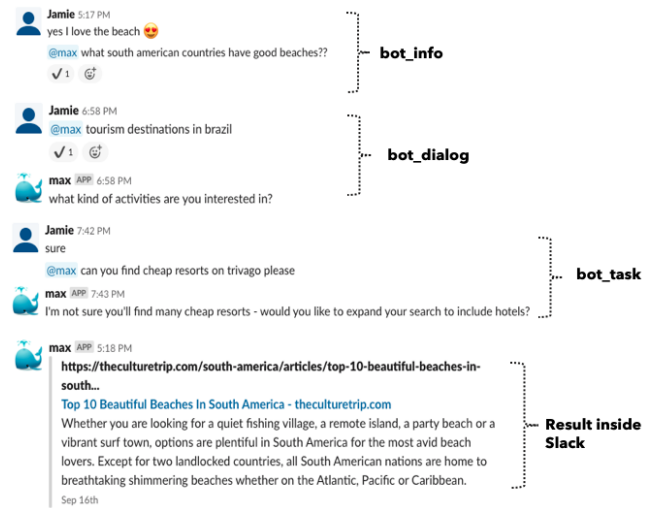


Figure 4: Example of conversations and result presentation in Slack across the three intervention conditions system to expand the Wizard’s functionalities for studying system initiative.

7 DEMONSTRATION PLAN

In the demonstration, one of the authors will play the role of the Wizard across the three conditions. We will ask the conference attendees to collaborate on Slack, and show them how the Wizard intervenes across the three conditions. After an initial run of the three conditions, we will describe the Wizard protocols across the intervention conditions. Next, we will explain the different interface elements and why we built them. Finally, we will give participants first-hand experience in playing the role of the Wizard.

8 CONCLUSION

In this paper, we introduce a Wizard of Oz (WoZ) system to study how a search system should take the initiative when engaging with users. We use Chu-Carroll and Brown’s [9] framework on Mixed-Initiative interactions to design the interaction space for the Wizard. Next, we describe recent designs to conduct WoZ studies and take lessons from them to come up with guidelines and protocols for our study. After this, we describe the technical details of our system, which includes a description of our Web App and how we integrate it into Slack. Following this, we discuss how others can use and expand on our system to research mixed-initiative interactions for conversational search. Finally, we describe our demonstration plan.

Acknowledgement: This work was supported in part by NSF grant IIS-1451668. Any opinions, findings, conclusions, and recommendations expressed in this paper are the authors’ and do not necessarily reflect those of the sponsor.

REFERENCES

- [1] Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W Bruce Croft. 2019. Asking clarifying questions in open-domain information-seeking conversations. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 475–484.
- [2] Layla El Asri, Hannes Schulz, Shikhar Sharma, Jeremie Zumer, Justin Harris, Emery Fine, Rahul Mehrotra, and Kaheer Suleman. 2017. Frames: A corpus for adding memory to goal-oriented dialogue systems. *arXiv preprint*

- arXiv:1704.00057* (2017).
- [3] Sandeep Avula, Jaime Arguello, Robert Capra, Jordan Dodson, Yuhui Huang, and Filip Radlinski. 2019. Embedding Search into a Conversational Platform to Support Collaborative Search. In *Proceedings of the 2019 Conference on Human Information Interaction and Retrieval*. ACM, 15–23.
 - [4] Sandeep Avula, Gordon Chadwick, Jaime Arguello, and Robert Capra. 2018. SearchBots: User Engagement with ChatBots During Collaborative Search. In *Proceedings of the 2018 Conference on Human Information Interaction & Retrieval*. ACM, 52–61.
 - [5] Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Inigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. *arXiv preprint arXiv:1810.00278* (2018).
 - [6] Robert Capra, Annie T Chen, Katie Hawthorne, Jaime Arguello, Lee Shaw, and Gary Marchionini. 2012. Design and evaluation of a system to support collaborative search. *Proceedings of the American Society for Information Science and Technology* 49, 1 (2012), 1–10.
 - [7] Robert Capra, Gary Marchionini, Javier Velasco-Martin, and Katrina Muller. 2010. Tools-at-hand and learning in multi-session, collaborative search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 951–960.
 - [8] Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. 2016. Towards conversational recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, 815–824.
 - [9] Jennifer Chu-Carroll and Michael K Brown. 1997. Tracking initiative in collaborative dialogue interactions. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 262–270.
 - [10] Laura Dietz, Ben Gamari, Jeff Dalton, and Nick Craswell. 2018. TREC Complex Answer Retrieval Overview. In *TREC*.
 - [11] Mateusz Dubiel, Martin Halvey, Leif Azzopardi, and Sylvain Daronnat. 2018. Investigating how conversational search agents affect user’s behaviour, performance and search experience. In *The Second International Workshop on Conversational Approaches to Information Retrieval*.
 - [12] Mihail Eric and Christopher D Manning. 2017. Key-value retrieval networks for task-oriented dialogue. *arXiv preprint arXiv:1705.05414* (2017).
 - [13] Gene Golovchinsky, Abdigani Diriye, and Jeremy Pickens. 2011. Designing for collaboration in information seeking. *Proc. HCIR* (2011).
 - [14] Helia Hashemi, Mohammad Aliannejadi, Hamed Zamani, and W Bruce Croft. 2019. ANTIQUE: A Non-Factoid Question Answering Benchmark. *arXiv preprint arXiv:1905.08957* (2019).
 - [15] Marti A Hearst. 2014. What’s missing from collaborative search? *Computer* 47, 3 (2014), 58–61.
 - [16] Diane Kelly et al. 2009. Methods for evaluating interactive information retrieval systems with users. *Foundations and Trends® in Information Retrieval* 3, 1–2 (2009), 1–224.
 - [17] Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards deep conversational recommendations. In *Advances in Neural Information Processing Systems*. 9725–9735.
 - [18] Q Vera Liao, Mas-ud Hussain, Praveen Chandar, Matthew Davis, Yasaman Khazani, Marco Patricio Crasso, Dakuo Wang, Michael Muller, N Sadat Shami, Werner Geyer, et al. 2018. All Work and No Play?. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 3.
 - [19] Meredith Ringel Morris. 2013. Collaborative search revisited. In *Proceedings of the 2013 conference on Computer supported cooperative work*. ACM, 1181–1192.
 - [20] Meredith Ringel Morris and Eric Horvitz. 2007. SearchTogether: an interface for collaborative web search. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*. ACM, 3–12.
 - [21] Sharoda A Paul and Meredith Ringel Morris. 2009. CoSense: enhancing sense-making for collaborative web search. In *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 1771–1780.
 - [22] Filip Radlinski and Nick Craswell. 2017. A theoretical framework for conversational search. In *Proceedings of the 2017 conference on conference human information interaction and retrieval*. ACM, 117–126.
 - [23] Chirag Shah. 2010. Coagmento-a collaborative information seeking, synthesis and sense-making framework. *Integrated demo at CSCW 2010* (2010).
 - [24] Ameneh Shamekhi, Q Vera Liao, Dakuo Wang, Rachel KE Bellamy, and Thomas Erickson. 2018. Face Value? Exploring the effects of embodiment for a group facilitation agent. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 391.
 - [25] Yueming Sun and Yi Zhang. 2018. Conversational recommender system. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 235–244.
 - [26] 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 2018 Conference on Human Information Interaction & Retrieval*. ACM, 32–41.
 - [27] Johanne R Trippas, Damiano Spina, Lawrence Cavedon, and Mark Sanderson. 2017. How do people interact in conversational speech-only search tasks: A preliminary analysis. In *Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval*. ACM, 325–328.
 - [28] Alexandra Vtyurina, Denis Savenkov, Eugene Agichtein, and Charles LA Clarke. 2017. Exploring conversational search with humans, assistants, and wizards. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, 2187–2193.
 - [29] Tsung-Hsien Wen, David Vandyke, Nikola Mrksic, Milica Gasic, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2016. A network-based end-to-end trainable task-oriented dialogue system. *arXiv preprint arXiv:1604.04562* (2016).
 - [30] Zhen Yue, Shuguang Han, and Daqing He. 2012. An investigation of search processes in collaborative exploratory web search. *Proceedings of the American Society for Information Science and Technology* 49, 1 (2012), 1–4.
 - [31] Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and W Bruce Croft. 2018. Towards conversational search and recommendation: System ask, user respond. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 177–186.