DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Characteristics, Applications and Architectures of Conversational Search Systems: A Systematic Literature Review

Alina Dats

DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Characteristics, Applications and Architectures of Conversational Search Systems: A Systematic Literature Review

Charakteristika, Anwendungen und Architekturen von konversationsfähigen Suchsystemen: eine systematische Literaturrecherche

Author:	Alina Dats
Supervisor:	Prof. Dr. Florian Matthes
Advisor:	Phillip Schneider
Submission Date:	15.08.2022

I confirm that this master's thesis in informatics is my own work and I have documented all sources and material used.

Munich, 15.08.2022 Alina Dats

Abstract

Humans constantly seek information using digital devices to help them in their daily lives, and research shows that they prefer to interact naturally and conversationally. Therefore, conversational search is emerging as one of the key technologies in fulfilling that need. In conversational search, a system and a user interact over several semantically coherent turns on a search task through a natural language dialog. Such interactions enable the system to understand users' information goals or help users to clarify their needs by asking the appropriate questions directly. Conversational search thus aims to maximize the user's information gain by finding search results with maximum utility. An increase in natural language dialog between users and computer systems could even lead to conversational systems replacing the prevailing interaction model of one-time keyword queries due to their effectiveness and ease of use. Understanding the design and engineering of conversational search systems is an ongoing task, and academic researchers and developers have more work to do on the theory and practice of conversational search. This research gap opens the opportunity for researchers to explore this paradigm. However, there is a lack of publications that summarize existing work and consolidate findings in this area. In this master's thesis, we conduct a systematic literature review that covers 50 publications to investigate how conversational search systems can be conceptualized and designed based on observations from the academic literature. We approach the overall problem in four directions, formulated as research questions, and discuss the characteristic properties, the suitable application scenarios, the architectures for conversational search systems, and the dependency level between scenarios and architectures. Based on the findings, we provide a conceptualization of conversational search containing feasible characteristic properties for conversational search systems. We determine suitable modalities, application scenarios, and domains for conversational search. Finally, we present a reference architecture for conversational search based on six fundamental layers. We discuss the functionalities and techniques used to implement these layers to enable the possibility of practical integrations with the help of our reference architecture.

Contents

Abstract **iii**

1. Introduction **1**
1.1. Motivation . . . 1
1.2. Research goal . . . 5
1.3. Thesis overview . . . 6

2. Foundations **7**
2.1. Conversational user interface . . . 7
2.1.1. Rise of conversational interfaces . . . 7
2.1.2. Conversation process . . . 12
2.2. Information search . . . 17
2.2.1. Introduction to information search . . . 17
2.2.2. Interactive information retrieval . . . 18
2.2.3. Search user interface . . . 18
2.3. Conversational search system . . . 19

3. Related Work **21**
3.1. Conversational user interface . . . 21
3.2. Conversational question answering system . . . 22
3.3. Conversational recommendation system . . . 22
3.4. Conversational search system . . . 23

4. Method **24**
4.1. Review planning . . . 25
4.1.1. Objectives . . . 25
4.1.2. Research questions . . . 25
4.2. Conducting the review . . . 26
4.2.1. Search strategy . . . 26
4.2.2. Study selection . . . 28
4.2.3. Data extraction and classification . . . 30
4.2.4. Data synthesis . . . 33
4.3. Reporting the review . . . 33

5. Results **35**
5.1. Overview of included studies . . . 35

5.2. Research question 1 . . . 38
5.2.1. Definitions . . . 39
5.2.2. Characteristics . . . 41
5.2.3. Related concepts . . . 57
5.3. Research question 2 . . . 60
5.3.1. Modality . . . 60
5.3.2. Application scenarios . . . 64
5.4. Research questions 3 . . . 70
5.4.1. Architectures . . . 70
5.4.2. Reference architecture . . . 76
5.5. Research questions 4 . . . 88

6. Discussion **92**
6.1. Key findings . . . 92
6.1.1. General findings . . . 92
6.1.2. Research question 1 . . . 93
6.1.3. Research question 2 . . . 95
6.1.4. Research question 3 . . . 96
6.1.5. Research question 4 . . . 97
6.2. Limitations . . . 97

7. Conclusion **99**
7.1. Summary . . . 99
7.2. Future work . . . 101

A. Included studies **102**

List of Figures **107**

List of Tables **109**

Bibliography **110**

Studies included in the literature review **116**

1. Introduction

1.1. Motivation

Artificial intelligence (AI) is a field that is rapidly advancing in both academic and industrial research. Advances in computing power and AI methods enable powerful AI-based techniques that rapidly enter the market in the form of various everyday applications in areas such as transportation, healthcare, customer service, e-commerce, and others. Natural Language Processing (NLP) is an important area of interest in the field of AI that deals with the automatic processing of human language (also known as natural language) in the form of unstructured text or speech. Natural Language Understanding (NLU) is an essential and particularly challenging subfield of NLP. It aims to give computers a sense of language by enabling them to read and understand text or speech, considering the unique ambiguity of human language and the subtly different perceptions humans have of the meaning of words, phrases, and sentences [1]. NLP-based techniques already benefit various domains, including healthcare and the medical sector, e.g., automatic identification and extraction of information in radiology report [2], or the discovery of adverse drug reactions obtained from a large number of patient records [3].

There is a clear trend toward a more "natural" way of consuming information or interacting, such as pointing with a finger instead of using a mouse, speaking instead of typing, watching videos instead of reading text, and writing complete sentences instead of keywords [4]. However, this trend is not new. Designing a human-computer interface (HCI) is necessary for building any system where the system's users are humans. In the past, people entered commands in written language into the command-line console, and the computer executed those commands. A graphical user interface (GUI) helps to modernize this process. It allows users to open, move, and delete files and folders, for example, with "physical actions" of clicking. In addition, interaction via natural language interface (NLI) conveys more success in the development of "natural" systems. These improvements allow (non-technical) users to explore and consume information without relying on technical support. Thanks to advances in highly accurate AI models and natural language interfaces, conversational agents - agents that can interact naturally with humans, have found their place in modern technological advancements. Such systems are conversational in that they assist users through a dialog format, usually with a human-like vocabulary and in written or spoken form [5].

Today, people talk to their personal assistants on their smartphones, voice-controlled smart speakers, or voice-based smart car systems and ask them questions or give them commands. These intelligent assistants, a relevant type of conversational agent, have become

very popular in recent years. Some examples of such assistants include Amazon's Alexa[1], Apple's Siri[2], Google Assistant[3], and Microsoft's Cortana[4]. Such commercial conversational agents can fulfill various user needs, such as making food orders or suggesting vacation destinations. Example 1.1 provides an example dialog of a flight booking process with a conversational system called Mercury, proposed by researchers from the Massachusetts Institute of Technology (MIT) [6]. Personal digital assistants are usually embodied and co-located with users, allowing them to alter the dynamics of the environment. In addition, the system can engage in social behaviors with the user to promote anthropomorphism.

Example 1.1 | Interaction with Mercury [6]

agent:	How may I help you?
user:	I want to travel to Washington Dulles on November fourth.
agent:	From Boston to Washington Dulles on November 4. Can you provide the approximate departure time or airline preference?
user:	I need to arrive at around nine a.m.
agent:	Okay. Flights from Boston to Washington Dulles arriving at 9:00 am on November 4. One moment please. The best matching flight is a US Airways flight arriving at 8:59 a.m. I also have a United flight arriving at 8:36 a.m.
user:	What time does the United flight leave?
agent:	United flight 1819 departs at 7:00 am from Boston. Shall I book this flight for you?
user:	Yes please.
agent:	United flight 1819 has been added to your itinerary. You depart Boston at 7:00 am and arrive in Washington Dulles at 8:36 am. Where would you like to go next?
user:	I would like to return to Boston.
agent:	What date will you be returning on?
user:	The same day.

However, this technology is not only used widely for personal use. Companies from various sectors are incorporating conversational agents into their IT transformation strategy to increase operational efficiency and save costs[5] [7]. With a rise in conversational systems investments, global chatbot market revenue is expected to grow to 454.8 million USD by 2027. That's up from 40.9 million USD in 2018[6]. Example 1.2 is an example of how conversational agents can help with business decisions.

[1] Amazon's Alexa `https://developer.amazon.com/en-US/alexa`

[2] Apple's Siri `https://www.apple.com/siri/`

[3] Google Assistant `https://assistant.google.com/`

[4] Microsoft's Cortana `https://www.microsoft.com/en-us/cortana`

[5] "Chatbot Trends Report 2021"
`https://chatbotsjournal.com/chatbot-trends-report-2021-b15479c404e4`

[6] "Chatbot market revenue worldwide from 2018 to 2027"
`https://www.statista.com/statistics/1007392/worldwide-chatbot-market-size/`

Example 1.2 | A conversational dialog between human and agent during a process of making a business decision [8]

user:	Where are sales lagging behind our forecast?
agent:	The worst region is `[country]`, where sales are 15% below projections.
user:	Do you know why?
agent:	The forecast for `[product]` growth was overly optimistic.
user:	How can we turn this around?
agent:	Here are the 10 customers in `[country]` with the most growth potential, per our CRM model.
user:	Can you set up a meeting with the CTO of `[company]`?
agent:	Yes, I've set up a meeting with `[person name]` for next month when you are in `[location]`.
user:	Thanks.

A conversational user interface (CUI) replaces a graphical or command-line interface. CUI provides a natural language interface to mimic human conversation and allows users to interact with a system through text, touch, speech, and other input and output options. The meaning of the term "conversational" in CUIs can refer to two contexts: (1) providing a conversational style of interaction and (2) providing a naturally occurring conversation opportunity using natural human language [9]. The human ability to understand language is unique. In general, speech recognition is an essential function of conversational interfaces, as it enables anthropomorphic interactions with assistive technologies, thus promoting the socialization and perception of these devices as social actors [10]. Human conversation requires the ability to understand the meaning of spoken language, transfer that meaning into the context of the conversation, find a shared understanding and worldview among the interlocutors, and maintain logical and semantic coherence across multiple turns.

In academia and industry, several terms describe systems that implement conversational interfaces: conversational agent, dialog system, conversational AI bot, AI assistant, intelligent virtual assistant, virtual customer assistant, digital assistant, virtual agent, chatbot, or simply a bot. For these systems, different terms that do not always complement each other are used, as they may also have different levels of intelligence and different interaction levels. This research uses the term conversational agent to describe an "application system that provides a natural language user interface for human-computer interaction" [11]. Conversational agents rest upon the idea of communicating via natural language. Their purpose is to support the successful fulfillment of user requests and to develop enough knowledge about the user to improve the conversation and task fulfillment process [12]. These systems are directly related to the notion of virtual assistants. The term bot should not always be equated with a conversational agent, as bots are generally not conversational but an "automata used for background tasks" [13]. At a minimum, a conversational agent should have text or speech-based voice recognition and a system for synthesizing information in natural language form back to the user [14]. Conversational systems can be task-oriented, helping users complete

simple tasks ranging from scheduling appointments to planning vacations, and non-task-oriented, engaging in open dialog with a user and taking on the role of a chat companion or recommender [15, 8].

However, we can observe that traditional conversational systems are mainly task-oriented [15]. In practice, these systems typically only perform short, task-based interactions, such as answering simple questions, and are often unable to engage in longer, free-form conversations as human-like dialogs due to technical limitations and the complexities of human language. In this way, they relate to traditional command-line interfaces with the advance in some paraphrasing and speech recognition functions. Consequently, such systems are usually designed for one domain-specific task. The previously mentioned personal assistants, such as Apple's Siri, are composed of many of these single-task systems.

Non-task-oriented conversational systems (also called chatbots or chitchats) consolidate long-term and free open-domain conversations between an agent and a human. These systems should be able to engage in multi-turn, mixed-initiative, open-domain dialog without being tied to a specific topic or task. Chatbots should be simultaneously reactive and proactive, meaning they should quickly react to the user's requests and proactively intervene in the conversation to introduce new topics or take the initiative, just as it occurs in a natural human-to-human conversation [16]. The main problem with this functionality is that the open-domain problem is too complex and is not limited to a predefined set of domains or intents. Conversations can be informal, open-ended, and have a high vocabulary count, which can lead to technical limitations of the systems [17]. In natural conversations, several factors control the state of a conversation. Even if two people have similar backgrounds and expertise, their topic of conversation may change completely.

Users are constantly seeking information online to help them in their daily lives, and research has shown that users prefer natural and conversational behavior to keywords [19, 18]. This preference for natural interfaces leads to adaptations in the development of knowledge communication- and transfer mediums designed to facilitate the exchange of information via a dialog interface. Hence a new research stream on *Conversational Search (CS) systems* emerged. These systems interact with a user over multiple semantically coherent turns about a search task through natural language dialog. The system aims to understand the users' needs or help them clarify their needs by directly asking the appropriate questions. Vakulenko et al. [20] defined this paradigm as "the task of retrieving relevant information using a conversational interface is termed conversational search." Hence conversational search aims to find or recommend the most relevant information to the user through natural language conversations based on text or speech. An increase in natural language dialog between users and search systems could even lead to conversational search systems replacing the predominant interaction model of one-time keyword queries due to their effectiveness and ease of use [21]. The conversational search paradigm differs from both domain-open and task-oriented conversational systems because its goal primarily satisfies the user's information needs. It opens up new research possibilities [20]. Later in the thesis, we will briefly introduce open and task-based conversational systems.

1.2. Research goal

Understanding the conversational search paradigm is still a work in progress. Academic researchers and developers need to work more on the theory and practice of conversational search to gain consistency and insight at all systems' operation levels and contribute to technological improvement. This research gap leads us to the central question of our research:

How can conversational search systems be conceptualized and developed based on the academic literature?

In this master's thesis, we provide insights for researchers and practitioners about state-of-the-art research on the emerging paradigm of conversational search using a Systematic Literature Review (SLR). We approach the overall problem in four directions, formulated as research questions that we attempt to answer in this thesis.

This research seeks to address the following questions:

RQ1: *Which characteristics of conversational search systems are defined in the academic literature?*

One of our goals is to understand the distinctive properties that make the conversational search a paradigm of its own and distinguish it from information retrieval systems or dialog systems. In addition to the core features, we will explore the desired characteristics and overlapping aspects of conversational search and its related systems. The dialogic interaction between such a system and a user is also exciting to explore, as it involves peculiarities of behavior that can lead to human-machine debate or reasoning about the information needed. In addition, exploring common terminology and a possible typology will bring more structure to understanding the technology.

RQ2: *What application scenarios have been investigated for conversational search systems and why?*

Since conversational interfaces are not always suitable for information retrieval, we will explore different scenarios that invite conversational search and discuss their suitability. Defining whether a conversational search is the right approach may depend on several aspects, including the interaction modality of the system. Having a traditional computer within direct reach is not always possible, opening new conversational search scenarios. The degree of interaction in the task may also play a role. Hence the main aim of this part is to determine scenarios that apply conversational search and investigate which factors influence the acceptance and success of these scenarios.

RQ3: *What architectures have been proposed for conversational search systems?*

In addition, we want to ep. We seek to explore architectural elements, algorithms, and technologies that help design the system's architecture. A conversational search system requires more capabilities than traditional information retrieval approaches. Building such complex systems from scratch can be intimidating given the technological advances in this rapidly growing field.

RQ4: *To what extent do the system architectures depend on the scenarios?*

Finally, we focus on the dependencies between the scenarios that invite conversational

search and the architectures required to implement them. Our goal is to investigate how deeply this level of dependence can be defined using specific examples of the scenarios.

1.3. Thesis overview

Chapter 1 motivates the topic and outlines the research described in this master thesis. Chapter 2 describes the theoretical foundations of concepts closely related to the conversational search paradigm, such as conversational interfaces and information search. We also give a short introduction to the conversational search paradigm. Chapter 3 presents related work in conversational search research or closely related research areas, e.g., conversational recommender systems or question-answering systems. We also discuss the differences in the main contribution between related work publications and our research. Chapter 4 provides a detailed description of the research methodology. This chapter describes the SLR process, including the planning, reviewing, and reporting phases. Chapter 5 presents the SLR results for each defined research question. Additionally, we provide descriptive statistics of the studies we included in the research. Chapter 6 aims to discuss the presented results and comment on the limitations of the research study. Chapter 7 contains a summary of the master's thesis and an outlook on future work.

2. Foundations

This chapter introduces the main concepts related to conversational search. The main goal of this chapter is to provide the foundation for understanding the research and identifying the gaps. We provide a foundation that includes existing viewpoints and current research directions.

2.1. Conversational user interface

Recently, the research on the ability to have a natural conversation with an intelligent device started gaining widespread interest. A conversational interface offers this possibility and enables interaction between a smart device and a human using natural language - just like a human-to-human interaction [9]. A conversational interface addresses usability issues and information load during a conversation. During the interaction, people do not have to learn a system's operating vocabulary or output scheme and can actively engage in a conversation in a speech or multi-modal way, which generally describes the concept of natural interaction.

2.1.1. Rise of conversational interfaces

The idea of developing a technology that engages with a human in a natural human-like conversation tracks back to the 1950s. That is when Alan Turing proposed his groundbreaking "Imitation Game" [22], better known as the Turing Test, a test designed to determine whether a machine could give other humans the impression of being human itself. The Turing Test is still used today, for example, in the Loebner Prize, which annually awards a prize to the best computer system that pretends to be a human. One of the first examples of a system that communicated text-based and passed the Turing test was ELIZA, developed by Weizenbaum [23]. ELIZA chatbot imitated a Rogerian psychotherapist. ELIZA continues to inspire today and laid the foundation for the generation of new chatbots. Another instance of a system that passed the Turing test is ALICE (Artificial Linguistic Internet Computer Entity), developed by Wallace in 1995 [24]. Text-based interfaces were followed by the development of speech-based interfaces and embodied conversational agents in the 1980s [9].

Several research publications [15, 8, 9] mention the division of conversational interfaces into task-based and non-task-based (also called open-domain). However, the rapid changes in the expansion of conversational agents have implications on terminology and classification. Deriu et al. [25] divide dialog systems into question answering, task-based, and non-task-based systems. It is worth noting that the authors refer to non-task-based systems as conversational

agents. However, in academic research, the term conversational agent is often not restricted to the non-task-based goal and generally describes a system that provides a natural language user interface for human-machine interaction [16]. Vakulenko et al. [20] also introduce a third type - conversational search - to the task-based and non-task-based systems, which differs from both types because it serves a different main purpose, namely, satisfying users' information need.

Gnewuch et al. [26] propose a classification for conversational agents along two dimensions: (1) primary communication mode and (2) context. For primary communication mode, the authors distinguish between text-based and speech-based agents. At the same time, for the context, they propose two types similar to the above: general-purpose and domain-specific. Table 2.1 illustrates the dimensions along with the examples of conversational agents. The domain-specific context includes constraints of the domain, users, or tasks. Hill et al. [27] refer to text-based conversational agents as chatbots or chatterbots, while Shah et al. [28] refer to them as dialog systems. Chen et al. [29] also refer to the term dialog system as task-oriented and non-task-oriented systems. On the contrary, McTear [16] distinguish between task-based dialog systems and non-task-based systems (or chatbots) for social interactions. Consequently, both researchers and practitioners use slightly different definitions and conceptualizations when referring to conversational interfaces. What they all have in common, however, is that they provide a new kind of interface for interaction - a conversational user interface that enables interaction in a conversational manner, such as taking turns in a dialog [16, 9].

		Context	
		General-Purpose	**Domain-Specific**
Comm. mode	**Text-based**	Eliza, Cleverbot, etc.	Enterprise-class CAs, IKEA's Anna, etc.
	Speech-based	Apple's Siri, Amazon's Alexa, Google Now, Samsung's Bixby	SPECIES, in-car assistants, Mecedes-Benz Linguatronic etc.

Table 2.1.: Conversational agent types [26]

McTear [16] proposes three different types of conversational interfaces in terms of the type of interaction and control over the dialog between a user and a system:

(1) **User-controlled dialog**: A user initiates and controls the conversation. Typically, this occurs when the user interacts with a virtual assistant or smart speaker. The user initiates the conversation by asking a question or issuing a command, and the system responds.

(2) **System-controlled dialog**: A system takes control of the flow of the conversation. There are several distinctions here:

   (a) Proactive dialog initiation by a system, e.g., to remind the user of a task.

   (b) User-initiated dialog because the user asks for instructions, while the system later takes control, e.g., to provide instructions for cooking a meal.

   (c) User-initiated dialog due to seeking a service while the system later takes control, e.g.,

to help book a flight

(3) **Multi-turn open-domain dialogs**: Both the user and the system take the initiative for the dialog and engage in conversations as in a natural interaction between humans.

In the following, we discuss five different directions in the study of conversational interfaces, which historically evolved in the academic research and industrial area: (1) text-based and speech-based dialog systems, (2) voice user interfaces, (3) chatbots, (4) embodied conversational agents, and (5) social robots.

Text- and speech-based dialog system

The term dialog system refers to the systems that can interact text- or speech-based with humans and emerge from research laboratories or industry. The dialog systems in the 1960s or 1970s were still text-based [9]. For example, Green et al. [30] developed a text-based system called BASEBALL that could answer questions about baseball. Another example of a text-based dialog system from this period was GUS [31], which booked flights for users. It processed indirect speech acts and anaphoric references, i.e., the utterance "the next flight" was processed concerning the previous flight. It was not until the late 1980s that researchers introduced a speech-based interactive system, also known as Spoken Dialog System (SDS), that could reasonably process users' spoken input. Initially, they were primarily domain-specific, such as the Mercury flight reservation system developed by MIT researchers [6], and first had to elicit several relevant pieces of user information for the task to construct the request.

The architecture of traditional dialog systems rests upon slot filling, i.e., the dialog structure is predefined with the slots that the system fills out during the conversation with the user [15]. To stay with the examples on flight booking dialog systems: the slot `destination city` corresponds to the question *"To which city are you flying?"*. With today's significant advances in automatic speech recognition and ML algorithms, new architectures emerge for SDSs, such as end-to-end learning for task-based purposes. This learning method can be generalized across different domains [16].

Voice user interface

Advances in industrial and academic research on text and speech-based dialog systems have led to their adoption for commercial use and a new direction of Voice User Interfaces (VUIs). SDSs and VUIs use similar NLP techniques and algorithms but differ in their purpose. SDSs serve industrial and academic purposes so that researchers aim to contribute to the knowledge of new techniques and algorithms. At the same time, VUI developers focus on business requirements and contribute to improvements based on customer needs [16].

How May I Help You (HMIHY), developed by AT&T, is one of the first VUIs to be deployed commercially to improve call routing and information delivery to customers. Typically, HMIHY welcomes the callers with an open prompt that allows free user input in natural

language. The system then processes the unrestricted user input, confirms that it understands the request, and gathers additional information [32]. Nowadays, as these systems show increasing customer satisfaction, people still frequently encounter VUIs in customer service, such as when they call to access routine banking services, and an automated customer service VUI serves them.

Commercial use of VUIs is becoming increasingly popular [16]. Major companies such as Amazon, Google, and Apple have launched screenless devices such as Amazon's Echo, Google Home, and Apple's HomePod, which have entered people's homes and become part of their daily routine. Such a system helps people with various tasks, such as cooking, playing music, and retrieving news and information [10]. VUI devices allow hands-free and eyes-free interaction, bringing advantages in terms of flexibility and intuitiveness. This advantage, on the other hand, can lead to the VUIs' inefficiency since, in some scenarios, the information presented by the VUI in a speech-based manner is difficult to review or edit [33].

Chatbot

Originally, chatbots, also called chatterbots, were developed to simulate human conversations. Usually, a chatbot is a system that can engage with a user in a dialog on a range of topics [16]. As mentioned in this chapter, the ELIZA system [23] is considered the first chatbot. Developed with around 200 lines of code, ELIZA could apply a small set of simple strategies and mimic a psychiatrist by extracting simple context from sentences and applying rules to formulate the response. The system applied a priority number to a keyword in the input and then applied appropriate transformation by turning, e.g., words like I into you. For instance, to the input *"Well, my boyfriend made me come here"* ELIZA would respond *"Your boyfriend made you come here..."* and to the input *"I need help..."* the system would apply a pattern and respond *"What would it mean for you..."*. If ELIZA did not find a direct match to apply a rule, it urged the patient to continue, e.g., *"I could not go to the party"* was followed by *"That is interesting. Please continue"* or *"Tell me more"*. Due to its pattern-based or also called rule-based nature, ELIZA belongs to the type of pattern-based chatbots. Wahde et al. [34] distinguish between three types of chatbots:

(1) pattern-based chatbots,

(2) information-retrieval chatbots,

(3) generative chatbots.

For modern **pattern-based chatbots**, the Artificial Intelligence Markup Language (AIML)[1] was introduced to define the rules for template matching. Another scripting language used for chatbots is ChatScript[2]. A simple example where each pattern (user input) is associated with a template (system output) is shown below (Example 2.1). Mitsuku[3] is an AIML-based chatbot developed by Steve Worswick that implements interactive learning and won the

[1] `http://www.aiml.foundation/`

[2] `https://sourceforge.net/projects/chatscript/`

[3] `http://www.square-bear.co.uk/mitsuku/home.htm`

Loebner competition five times in the 2010s. The chatbot Rose and its two predecessors Suzette and Rosette have also won the Loebner competition in the 2010s.

Example 2.1 | Simple AIML specification [34]

```
<category>
<pattern> I feel * </pattern>
<template> What do you think makes you feel * </template>
</category>
```

Information-retrieval chatbots choose their answer from a large corpus of conversations, e.g., a large database [34]. Sentence embeddings compute the similarities between sentences and encode sentences as numerical vectors. For example, the TF-IDF (term frequency-inverse document frequency) approach converts each sentence into a vector whose length equals the number of words available in the dictionary [35]. Moreover, before calculating the length, the words are destemmed and lemmatized, i.e., converted into the other form, e.g., plural "dogs" converted into a singular "dog" and word "better" converted into "good". In the first part, the Term Frequency (TF) vector is simply the frequency of occurrence of each word normalized by the number of words in the sentence. However, this approach can lead to paying too much attention to simple words like "the" frequently occurring in the sentence and providing little context. In the second part, Inverse Document Frequency (IDF) reduces the weight of frequently occurring terms that are unlikely to contain relevant information about the sentence's content. Combining these two vectors with an element-wise product leads to a final vector. Further, this technique compares the final vector to the corresponding vector for all sentences in the corpus. TF-IDF comes with some limitations [34]. This technique does not consider: (1) the order in which words appear in a sentence, (2) that many words have synonyms, and (3) the context of the sentence, e.g., based on the previous sentences.

The last two categories rely on existing utterances in the form of given patterns or from a dialog corpus and fall into the category of interpretable systems. The third category of **generative chatbots** falls into either the interpretable systems category or the black box category if they utilize DNNs [34]. Generative chatbots generate responses using statistical models instead of existing utterances. DNN-based systems trained with large amounts of data currently dominate the academic research in chatbots [16]. Examples of current open-domain chatbots include Google's Meena [36] and Facebook's BlenderBot [37]. The Google Research Brain Team developed Meena - an open end-to-end neural chatbot. Moreover, BlenderBot is also a large open-domain chatbot model developed by the Facebook AI Research (FAIR) group. In end-to-end neural systems, an input is mapped directly to an output response without requiring processing by the modules of the modularized architecture. This approach uses DNNs within a Sequence-to-Sequence mapping (Seq2Seq) architecture, also noted as neural dialog. Seq2Seq-based systems have the highest human ratings for naturalness and quality. Developers implemented three models for the BlenderBot with 90 million, 2.7 billion, and 9.4 billion parameters, respectively. Transformers and a pre-training with 1.5 billion training examples from conversations on Reddit covering a wide range of topics served as

a basis. By comparison, Meena has 2.6 billion parameters and was trained using 341 GB of texts from social media conversations. Developers made BlenderBot an open-source project so that other innovators can experiment with it and create their chatbots [16, 34].

Embodied conversational agent and social robot

Advances in understanding human cognition have shown that our minds are not isolated from the body. Hence embodiment plays a vital role in the system's output, allowing it to produce gestures and behaviors that enhance the image it projects [16]. Embodied Conversational Agent (ECA) is a conversational agent that integrates facial expressions, body posture, gestures, or speech to provide a human-like, natural interaction. The ECA is a computer-generated virtual or screen-based character [9]. To endow ECAs, several standards, and annotation schemes have been developed, such as the Emotion Markup Language (EML)[4], the Behavior Markup Language (BML), and the Multi-modal Utterance Representation Language (MURL). An example of a 3D real-time ECA is Greta [38], which can communicate through verbal and nonverbal channels such as gaze, head and trunk movements, facial expressions, and gestures.

Social robots and artificial companions can also exist virtually and in physical form, e.g., digital robots or pets, to provide people with companionship and entertainment [16]. With social robots, people apply the patterns of social interaction and see robots as social companions with whom people can build stronger and more lasting relationships [9]. Examples of social robots include Socially Aware Robot Assistant (SARA) [39], developed at Carnegie Mellon University's ArticuLab, MultiModal Mall Entertainment Robot (MuMMER)[5], a European Union (EU) funded project, Furhat [40], and social robots Professor Einstein[6] and Leka[7]. SARA recognizes and displays emotions based on the conversation and the user's tone. MuMMER combines speech with nonverbal communication and human-centered navigation to interact socially with users, while Furhat specializes in head and neck movements to render facial expressions. Professor Einstein is a tutor for physics. And lastly, Leka is a companion for children with autism.

2.1.2. Conversation process

Despite advances in machine learning for conversational interfaces, the conversational part of human-machine communication is still in its preliminary phase due to its complexity [41]. Vakulenko [20] defines communication as *"a sequence of natural language expressions (utterances) made by several conversation participants in turns"*. The human ability to understand language is unique, and natural human language is an effective yet complex communication tool that can adapt to new domains and purposes [42].

[4] `https://www.w3.org/TR/emotionml/`
[5] `http://mummer-project.eu/`
[6] `https://www.hansonrobotics.com/professor-einstein/`
[7] `https://leka.io/`

Whether the communication is task-oriented or domain-open, modern systems use different techniques to understand natural human language.

Task-oriented system

Researchers often structure the communication between a system and a human for task or command purposes in the form of a pipeline architecture. Figure 2.1 provides an example of such a pipeline. Each component has a complex structure, and different machine learning models are used to increase the accuracy of each component.

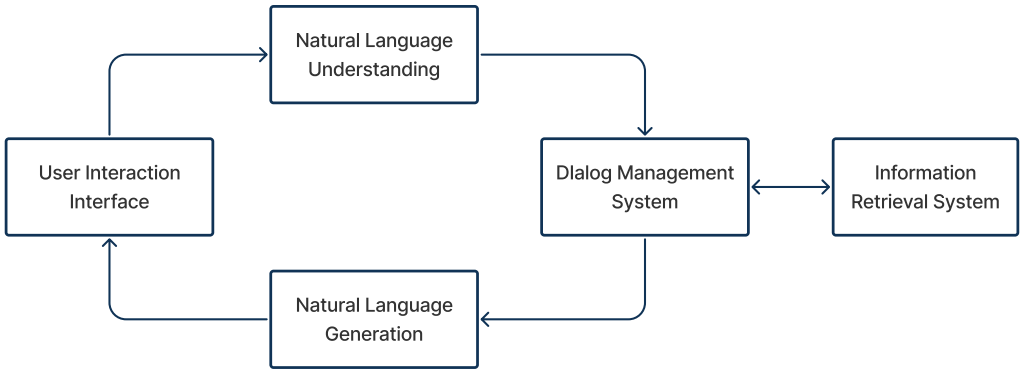

Figure 2.1.: Pipeline architecture for task-oriented conversational agents, adapted from [41].

To meet the challenge of understanding a user's input, NLP, in particular a subset of NLP - **NLU** - is responsible for understanding natural human language statistically by a computer. Two of the main tasks of NLU are Named Entity Recognition (NER) and Intent Classification (IC). The NER task is to identify the named entities of a sentence and classify them into different predefined classes. Consequently, a language system that applies NER attempts to define the "what" meaning of the user input. IC attempts to understand the user's actual intent behind the observed input [41]. Figure 2.2 illustrates an example of IC. In fact, most NLU models above the word level are usually designed for a specific task and have difficulty with data outside of the domain [43]. The human conversation contains many phenomena such as interruptions, affirmations, anaphora, or omissions. Some utterances can only be adequately understood within the context of a conversation or with domain knowledge. Furthermore, errors can also occur on the human side, such as mispronunciation or transposed words and letters, which lead to possible misunderstandings by a system during the conversation [14]. Despite frequent human errors and other difficulties in conversation, various strategies for error detection and correction, including various forms of confirmation, are already built into the systems. Since one of the main functions of conversational agents is to understand

the meaning of user input using NLU techniques to fulfill user requests successfully, these improvements are vital [17].

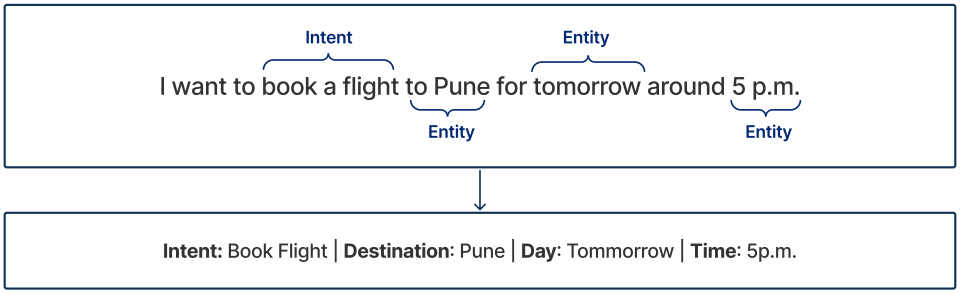

Figure 2.2.: Natural language understanding example [41].

A **Dialog Management (DM)** component further processes the semantic representation generated by the NLU. The goal of the DM component is to develop an interaction strategy and control the system's actions depending on the user's input. In addition, the DM component continuously manages the state of the conversation. The DM state tracker plays an essential role by estimating the user's goal at each turn of the dialog [16]. A simple approach to a state tracker is in the form of a switch statement. It simply has predefined actions for each input so that the specific intent of the NLU component triggers a unique response or action. The disadvantage of having a limited number of responses and actions is self-explanatory. In addition, the initiative for the conversation in this approach lies entirely with the user. When the set of possible inputs is limited, the DM component in the form of Finite State Machines (FSMs) guides through a finite number of states, and either the human or the system can change the direction of a conversation. This approach is inflexible but suitable for task-oriented goals.

A **statistical DM component** involves a probability distribution over a true dialog state. Researchers propose different approaches for dealing with probability distributions: robust sets of hand-crafted rules, conditional random fields, or maximum entropy models. The complex structure of the DM component includes integrating DL modules. DL modules' main idea is to predict the system's following action or reaction. DL is suitable for dealing with unstructured data. Based on DL, the state tracker can learn what action to perform next in two ways: reinforcement learning or belief tracking. In reinforcement learning-based approaches, through simple rewards and failures, the state tracker must learn to decide what actions to take based on observations of user input. The belief-based DM system is a modern approach that outputs a sequence of probability distributions over any number of possible values. Other domains can easily apply it too after training on one domain [41, 44, 29].

Finally, a **Natural Language Generation (NLG)** component generates a response in natural language based on an action decided by the DM component. NLG may involve sentence planning approaches in which the system converts the semantic symbols of the input into an intermediate form, e.g., in the form of a tree or template, and then converts this intermediate structure into the repose through surface realization. For a template form, such an approach

leads to identical responses due to the hand-crafted rules, which can lead to a lengthy conversation. Wen et al. [45] integrate a neural network (NN), specifically Recurrent Neural Network (RNN), with Long Short-Term Memory (LSTM) structure for NLG. To ensure that the response represents the intended meaning, the dialog action type and its slot value pairs can be converted into a vector and passed as additional input to NLG. The generator outperforms some baseline systems and produces higher quality and more natural responses. In the next extension of their work, Wen et al. [46] show that multiple domains can reapply this approach with less training data. Other researchers use an encoder-decoder LSTM-based structure to integrate the input information, the semantic slot value pairs, and the dialog action type to generate correct responses. Finally, Seq2Seq-based approaches help to adapt to the user's interaction behavior and thus became state-of-the-art approaches for NLG. Seq2Seq models use LSTM models by mapping an input to a vector. Then, the models sequentially predict tokens based on the previously obtained mapping [41, 29, 16].

The task-oriented pipeline architecture has limitations when adapting to new domains. One of these limitations is process independence, meaning that one component's input is independent of another's output. The solution would be to use an end-to-end model instead of a traditional pipeline. Such architecture consists of a single module that interacts with a structured external database. The system can be trained in a supervised manner to learn mappings from dialog context to system responses. In the next paragraph, we will discuss end-to-end NN approaches in more detail [29].

Open-domain system

Only recently have **end-to-end NN generative models** attracted attention and made progress in the development and drastically improved the field of conversational AI, especially for open domain purposes [29]. We already introduced generative models in the subsection about chatbots and will discuss them in more detail. A generative model is used in classification tasks in supervised ML. Compared to a discriminative model that distinguishes between classes, a generative model can produce a new object of either class.

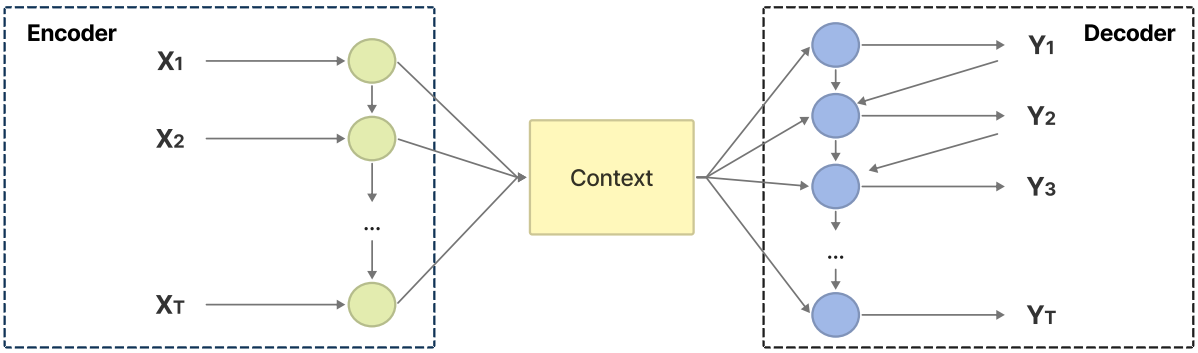

Figure 2.3.: Encoder-decoder model, adapted from [29]

As mentioned, end-to-end NN architecture involves learning mappings between input and

output utterances. This mapping describes the Seq2Seq mapping, which allows mapping from input in the source language to output in the target language without requiring intermediate processing. A Seq2Seq model uses an encoder-decoder structure [29, 41]. Figure 2.3 illustrates the encoder-decoder model. It must perform two main tasks: (1) encoding - to process and represent the input; and (2) decoding - to produce the output. An example from Figure 2.4 shows how the encoder-decoder network in the Seq2Seq framework models a conversation. The RNN reads one token at a time from a *"What are you doing tomorrow"* input. The context vector is the hidden state of a model. Finally, the trained RNN maps the input to the output *"I'm going to London"*, also one token at a time. RNNs operate on sequences of vectors and can capture information about past inputs, building a memory useful for sub-secondary processing.

However, such memory cannot penetrate far into past context, and the information encoded in the hidden states tends to remain local [16]. Consider a problem of predicting the next word in a modeling task in RNNs, described in Example 2.2.

Example 2.2 | Problem of predicting the next word in RNNs [16]

The flights the airline was canceling were full.

In this sentence, the subject-verb agreement is between the words "airline" and "was" and between the words "flights" and "were". Therefore, it is difficult to assign the correct probability to the word "were" because the word "flights" is far from "were" and the intervening context contains singular constituent entities.

Currently, mainly text-based dialogs apply the described architecture. In an SDS, the speech component must operate separately from the end-to-end NN architecture and provide the end-to-end architecture with input in text form. In the same way, the response in text form is passed to the Text-to-Speech (TTS) synthesis component to produce the spoken output [16].

Nevertheless, the end-to-end NN architecture offers advantages over the traditional pipeline architecture. For instance, if a user provides feedback that the system did not successfully respond to a particular input, how can we determine precisely which component failed? In the pipeline architecture, it could be a failed speech recognition or NLU, a failure of the DM component to select the correct action, or a failure of the NLG to produce an appropriate sentence from the output [16].

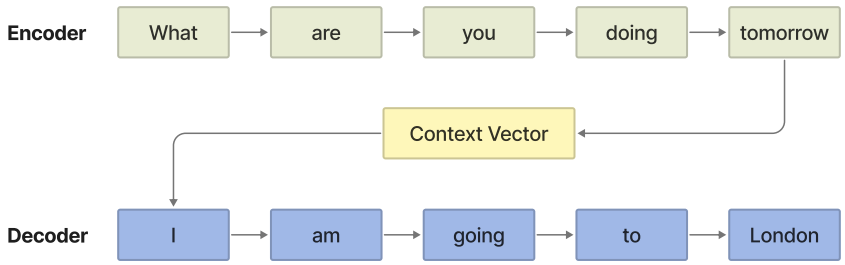

Figure 2.4.: Example using encoder-decoder in the Seq2Seq framework [16]

2.2. Information search

The cognitive process of information seeking is the primary activity in people's daily lives. Academic research on information science increases because researchers try to figure out how people search and use the information and the factors that encourage the search for information behavior [47].

2.2.1. Introduction to information search

Information search and information seeking are closely related in the field of information retrieval [47]. Marchionini [48] defines information seeking as:

"Information-seeking is a special case of problem-solving. It includes recognizing and interpreting the information problem, establishing a plan of search, conducting the search, evaluating the results, and if necessary, iterating through the process again."

In this context, Machionini [48] further outlines the concept of information search. He says:

"The term search is used to mean the behavioral manifestation of humans engaged in information seeking and also to describe the actions taken by computers to match and display information objects".

Moreover, Marchionini et al. [49] divide information search into three modes: (1) directed; (2) semi-directed; and (3) undirected. Directed information search describes a search process in which the searcher has a specific topic in mind so that the search behavior takes the form of a predetermined path. The semi-directed mode means that the searcher has an approximate idea of a topic. In the undirected mode, the searcher is interested in the topic and searches to get a better idea or to change it. Researchers may interchangeably use terms such as querying, searching, browsing, or navigating. Querying or searching refers to creating new and not previously collected sets of information in an ad hoc manner. On the other hand, navigating or browsing refers to the undirected exploration of navigational structures by following a chain of grouped links containing information [47].

Today, the search process should be able to support the user in accomplishing demanding tasks and facilitate search processes that extend over a longer period and take place on more than one device [50]. Bystöm et al. [51] distinguish several types of tasks in the information retrieval domain:

(1) work tasks,

(2) information-seeking tasks,

(3) information search tasks.

Each of these types is a subset of the others in sequential order. A work task refers to an activity people need to perform to fulfill a required responsibility assigned to them by others or by themselves. An information-seeking task is a subtask of a work task and a key component of information-intensive work tasks. Recognized need for information and a decision to act on it triggers an information-seeking task. An information-seeking task

can represent a series of successive smaller information search tasks. While information seeking focuses on satisfying the need for information, information search involves satisfying a definable portion of the information need through a single retrieval from one or more sources [51].

2.2.2. Interactive information retrieval

The view that IR systems are fundamentally interactive and should be evaluated from the user's perspective is not new, so researchers mentioning IR may consider IRR. Interactive information retrieval (IIR) examines user behavior and interactions with the search system and the information itself, while traditional IR excludes user behavior. IRR is closely related not only to traditional IR but also to library science, psychology, and HCI. The main question in evaluating IR and IRR systems can help find the main difference between these paradigms. For IR, it is - *does the system find relevant documents?* while for IRR, it is - *can people use this system to find relevant documents?* [52].

2.2.3. Search user interface

The worldwide spread of the Internet provided a historical change in the application of search user interfaces. Until then, only professionals and highly skilled individuals such as librarians, journalists, or paralegals could apply computerized IIR. From that point on, developing a successful search interface required a broader understanding of people's needs, including how they search for information and how they subsequently use it [50].

Guidelines	
Provide informative feedback	Support user control
Reduce the load on short-term memory	Provide shortcuts for experienced users
Provide simple error handling	Strive for consistency
Allow easy reversal of actions	Design for closure

Table 2.2.: Design guidelines for search user interfaces [53]

Searching is a mentally intensive task, and while searching for information, one is usually focused on this activity and cannot, for example, read and think about something else simultaneously, so the search interface should not interrupt the user [47]. Shneiderman et al. [53] suggest further design guidelines for search user interfaces (Table 2.2).

These guidelines are general recommendations and do not specify implementation. For example, to provide informative feedback, the researchers suggest showing the user some search results right after the initial query input [47]. Direct display of the results should help the system and the user ensure the search process is on the right track. In addition, the first search results can contain word suggestions that the user can use to refine her query.

When using web search engines, users usually specify their search in the form of keywords. An alternative method would be to enter questions or search instructions in complete sentences. There is evidence that this input method in natural language is more intuitive for humans. Therefore, there is much interest in applying the same method we use to seek information from other people to search systems. The main reason search engines did not support this in the past was the lack of technology available. Recently, however, systems that answer questions in complete sentences have become popular [47].

Question-answering system

Traditional question-answering systems do not generate new answers out of anything but link a query entered in natural language to the most relevant document, sentence, paragraph, or web page with existing information. This linking offers a user a rationale and explanation of where the answer came from. The length of the response also depends on the search context. The study by Lin et al. shows that people prefer long answers when searching for general information, while for fact-based queries, a phrase or one-sentence answer is more suitable for users.

Conversational question-answering system

The interest arises to revolutionize the human-machine interaction in question-answering systems further. The basic idea behind the conversational question-answering (CQA) system is the possibility to transform the simple question-answering process into a conversation with several turns when the user needs more detailed information about the posed question [54]. The concept of CQA is a concrete manifestation of the concept of conversational search, where in CQA, the system returns a single correct answer to the question entered by the user, rather than a list of relevant results [54, 55]. In the next section, we will introduce a conversational search system in more detail.

2.3. Conversational search system

According to Radlinski et al. [5], two aspects lend themselves particularly well to a search environment and the integration of conversational search systems. The first is when users do not know how to describe their information needs. Therefore, part of the conversation is the system's goal of discovering the user's actual need. Second, to have a conversation with multiple turns to determine a set of possible results based on the preferences entered by the user during search input.

In this context, Radlinski et al. [5] define a conversational search system that other researchers frequently cite:

"A conversational search system is a system for retrieving information that permits a mixed-initiative back and forth between a user and agent, where the agent's actions are chosen in response to a model of current user needs within the current conversation, using both short- and long-term knowledge of the user."

According to the typology of conversational search provided from Dagstuhl seminar [55] and shown in Figure 2.5, a conversational search system is either an IIR system with speech and language processing capabilities, a retrieval-based chatbot with task modeling capabilities, or an information-seeking dialog system with IR capabilities. Furthermore, Vakulenko [20] defines the task of retrieving relevant information through a conversational interface as a conversational search.

These are only some definitions of conversational search systems. Hence we can already state that there is no clear consensus on the definition yet. The characteristics of conversational search systems are also unclear and require further research, which we aim to accomplish in this research. Chapter 5 presents our results on this topic.

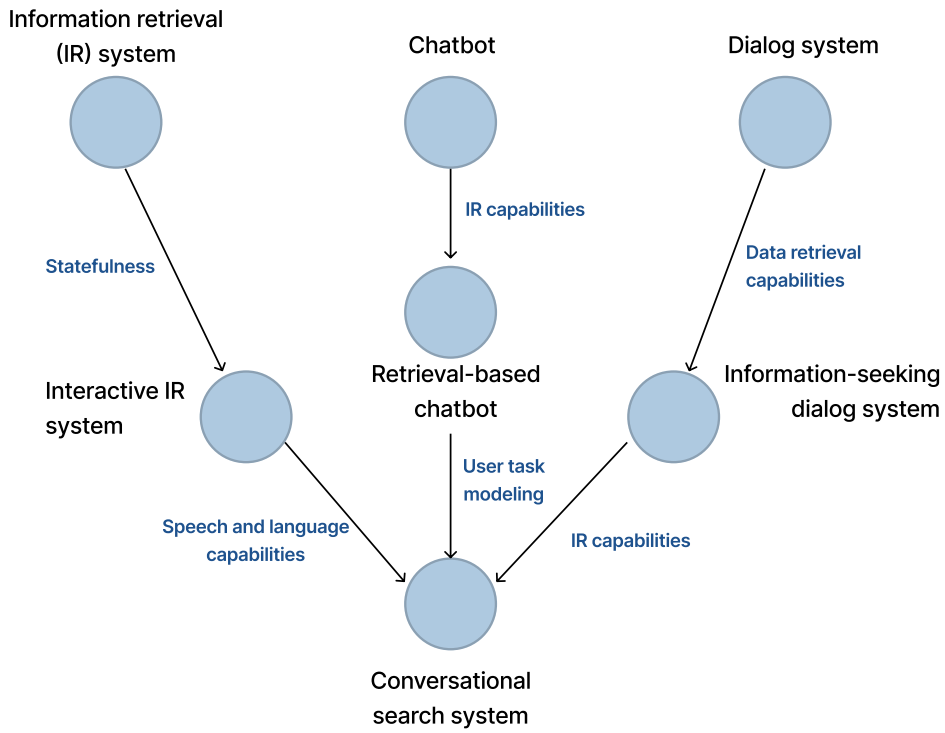

Figure 2.5.: Typology of Conversational Search defines conversational search systems via functional extensions of information retrieval systems, chatbots, and dialogue systems, adapted from [55]

3. Related Work

This chapter presents publications related to the context of our study. We first present what application domains the authors of conversational interfaces describe in terms of related work. We also found related work on conversational user interfaces. We identified related work on the question-answer paradigm closely related to conversational search. Finally, we discuss related work on conversational search, however, there are very few results that we can use.

3.1. Conversational user interface

In recent years, there has been an increasing interest in research of conversational user interfaces. Morger et al. [56] provided a holistic and detailed overview of the latest conversational agents. The authors provided an overview of the context and application areas of conversational agents in particular but also proposed a taxonomy of the main concepts of the systems. The research proposed several main goals for conversational agents: user support, information request, user engagement, action execution, user training, and information collection. Interestingly, the information request objective is what would probably partially describe the main objective for the conversational search system, but the authors mentioned Q&A chatbots or recommender systems as examples. These concepts are closely related to the conversational search paradigm, but the authors did not explicitly use the term conversational search in their work. This study was completed as a tertiary study, i.e., a systematic literature review, following the standard guidelines for this method.

Jaber et al. [57] surveyed conversational user interfaces in the context of mobile applications. In addition, the authors defined three domains for using conversational user interfaces on mobile devices: multi-modal, breakdown and recovery (resolving misunderstandings in human-agent interaction), and context (understanding the user's context). This survey used a systematic methodology with a synthesis of the themes, codes, and domains relevant to this research topic.

Zierau et al. [58] examined over one hundred empirical publications on conversational agents using a systematic literature review methodology. Based on the taxonomy of Feine et al. [59], the authors categorized human-computer interaction variables in aggregated dimensions. Examples of the variables include ease of use [60], quality of interaction [61], or perceived humanity [62]. As part of the verbal dimension, the authors also mentioned the variable use of contextual information, which is closely related to the characteristics of conversational search systems. This systematic investigation of the relationships between independent and dependent variables fills gaps in previously understudied yet promising

areas of user interaction with conversational agents.

Conversational agents find their benefit in various domains such as education, healthcare, e-commerce, and customer service. We identified systematic review research and surveys in healthcare [65, 66, 63, 64, 67], and education [69, 68] domains.

Bijan et al. [69] studied conversational agents in the education domain to identify implications for research and practice. The authors discussed "pedagogical CAs", a virtual educational companion in the form of an e-learning system, as one of the use cases in the educational scenario. The authors emphasized that during their study, they identified a research gap in long-term application in the education domain and concluded that this topic has not yet reached its full potential. Laranjo et al. [65] focused on the research of conversational agents in the healthcare domain. The authors stated that their distribution in healthcare is inevitable with the increased trustworthiness and competency in the systems. The research was conducted as a systematic literature review, considering only the publications where the agent processed any unconstrained natural language input.

3.2. Conversational question answering system

We found another work that explores concepts closely related to conversational search in information retrieval, such as conversational question answering and conversational recommender systems. Zaib et al. [54] examined over 80 publications in the conversational question-answering (QA) paradigm. The authors discussed the architecture and prevailing techniques contributing to context formulation and multiple-turn question answering. The authors also discussed the taxonomy of QA systems and described their characteristics and the general architecture these systems implement. Related to our research, Zaib et al. [54] stated that conversational question-answering systems are a simplified concept of conversational search settings.

Zaib et al. [44] investigated the applications of pre-trained language models for dialog systems with a particular focus on question-answering systems. The authors summarized pre-trained language modeling techniques for dialog systems, provided insights into these models' implementations, and discussed whether pre-trained language models could address challenges relevant to dialog systems. It is worth noting that this review did not address the core techniques, i.e., the text generation technique on the pre-trained language models. This survey is an extended version of the brief survey [70]. In contrast, Guo et al. [71] investigated in their survey conditional NLG technology and introduced such techniques as context-based text generation, topic-aware text generation, and knowledge-enhanced text generation.

3.3. Conversational recommendation system

Jannach et al. [72] surveyed Conversational Recommender Systems (CRS) based on recent advances in machine learning, particularly DL techniques and natural language-based in-

teraction. As defined by the authors, a CRS is a software agent that helps users achieve recommendation-based goals and find items that match their preferences and expectations through a multi-turn dialogue. The authors identified two categories for CRSs: NLP-based and form-based models. CRS based on the NLP model allows the user more freedom in deciding how to structure the dialogue. This approach is primarily user-driven or mixed-initiative, as the system usually still has some agenda on how to proceed with the conversation. In contrast, form-based models are mostly system-driven and consist of predefined and ambiguous actions based on forms (e.g., buttons, radio buttons) that can make such dialogues seem unnatural. Furthermore, the research highlighted the need to consider appropriate designs for CRS based on the scenarios and argued whether natural language interaction makes the recommendation process more efficient and effective. This study also suggested a relationship between CRS and conversational search systems, as these systems share several standard features. For example, one of the main tasks of both systems is to rank objects according to their assumed relevance, either for a given input as a query (search) or for the user's preferences (recommendation). The authors noted that it is difficult to find clear distinctions between personalized conversational search and recommendation systems. The following studies validated these assumptions [73, 74, 75].

3.4. Conversational search system

Furthermore, we identified related work based on the specific concept of conversational search. Disambiguation of ambiguous queries in conversational search systems was the main research topic of Keyvan et al. [76]. In practice, ideal solutions to the problem of identifying the user's true intent instead of insufficiently specified ambiguous queries do not yet exist. In their study, the authors propose integrating three successful techniques to address this problem: asking clarifying questions, suggesting questions, and reformulating queries. In addition, the authors provided a detailed overview of the characteristics of ambiguous queries to provide a comprehensive picture of the existing limitations.

To the best of our knowledge, no related work examines the conversational search paradigm, in general, using a systematic literature review method. Closely related, we could only locate one study on the topic of ambiguous queries in conversational search, published in the same year we wrote this paper. Our research uses the same methodology as several of the findings in the related work, a systematic literature review. We describe this methodology in the next chapter in detail. However, our research considers conversational search systems from properties, scenarios, and architecture perspectives. We hope this work will be a valuable resource for researchers interested in this area.

4. Method

We conducted this study as a systematic literature review based on the original guidelines proposed by Kitchenham et al. [77, 78]. In addition, we have adapted and applied the systematic mapping process proposed by Petersen K. et al. [79] to our study and defined separate process steps.

The essential steps of the process of our systematic literature review study are:

(1) Define research objectives,

(2) Define research questions,

(3) Define search strategy,

(4) Conduct search for relevant papers,

(5) Papers screening,

(6) Keywording of relevant papers,

(7) Data extraction and mapping process,

(8) Writing/reporting process.

Figure 4.1 models the proposed process flow for the research. Each process step has an outcome, and the final step outcome is the report of the SLR findings. Kitchenham et al. [78] and Keele et al. [77] propose to separate the review process into three stages: planning the review, conducting the review, and reporting the review. Steps 1-2 are associated with the planning phase, steps 3-7 with the review conduction, and step 8 represents the final phase. The PRISMA 2020 checklist helped in guiding the preparation of the SLR report. The following sections describe the steps of the SLR in detail.

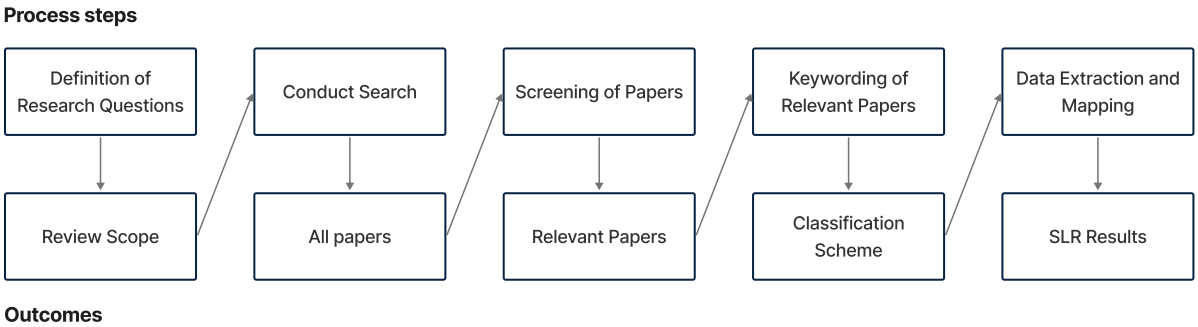

Figure 4.1.: Process for the current SLR, adapted from [79].

4.1. Review planning

The first phase of our approach is planning the review, as suggested by [78]. During planning, we define the objectives of our research using the SLR approach and based on our research questions.

4.1.1. Objectives

Conversational search is one of the key technologies to provide the most relevant information to the user through a natural language dialogue based on text, speech, or other modalities. A well-defined methodology in systematic literature reviews helps us study the conversational search paradigm unbiasedly. The SLR approach allows us to summarize the empirical evidence of the benefits and limitations of conversational search, identify the gaps in current research, and provide background information to support new research opportunities. The research questions identified in this study provide an overview of the current state of research in conversational search.

4.1.2. Research questions

The research questions addressed by this study are:

RQ1: Which characteristics of conversational search systems are defined in the academic literature?

RQ2: What application scenarios have been investigated for conversational search systems and why?

RQ3: What architectures have been proposed for conversational search systems?

RQ4: To what extent do the system architectures depend on the scenarios?

We defined RQ1-RQ3 based on the idea of exploring this paradigm through typical exploratory questions such as: *What does the system look like? Where is it used? How can it be built?*. RQ4 is a relationship-based question that examines the differences in architecture depending on the different scenarios, tasks, or modalities.

To address RQ1, we identified studies that described characteristics, properties, or aspects of conversational search. Researchers might not commonly use the term "conversational search" due to a lack of terminology for this paradigm. As a solution, we examined the characteristics of the proposed system and considered them to be the characteristics of related systems. We also examined the described HCI process or the desired features for such interaction.

For RQ2, we focused on the specific examples of conversational search scenarios based on two aspects: (1) the interaction modalities that the situation requires for use and (2) the task provided by the system. Moreover, we explored why these scenarios invite conversational search.

Concerning RQ3, we evaluated the descriptions of the architectural elements, techniques,

or algorithms that contribute to the implementation of conversational search systems. The results could contribute to whether it is a description of the entire architecture, which we hypothesize is rarely the case, or an architectural description of the individual components, e.g., the dialog manager component.

To address RQ4, we explored the dependency level between the proposed scenarios and architectures in the studies. We assumed that an explicit definition of the dependency level would rarely occur in the studies because researchers may mention the range of scenarios but not their architectural features. If the researchers mention architectural elements in a scenario context, their dependency level may not be explicitly described.

4.2. Conducting the review

We conduct review in the second step of our SLR approach. In this phase, we first define our search's strategy, then define the search string, select sources for the search, and define inclusion and exclusion criteria. In the following, we describe the whole process step by step.

4.2.1. Search strategy

We identified primary studies using search strings in scientific databases with specified search parameters and manually searching for relevant studies. The latter was accomplished using the forward and backward snowballing process.

Search string

We did not use keywords for the search string based on the proposed research questions because these keywords may not appear in the research's title, keywords, or abstract. The search string would increase significantly because we would have to include all possible synonyms, e.g., for the keywords such as "characteristics," which would result in biased research. Hence we wanted to get a broad overview of the research area. We also had to consider the close relationship between information-seeking dialogues and the concept of conversational search. In addition, the inconsistency of terminology led us to include terms that other researchers might use to describe the paradigm, such as "conversational information retrieval" or "conversational information-seeking." We manually tested whether these terms would bring more than zero results to our study collection. In this way, we could identify that the term "conversational answer retrieval" was irrelevant because it did not yield any additional results.

Furthermore, since our research is part of the NLP academic field, we wanted to include NLP keywords in our search string. Surprisingly, the manual sampling showed that we received a significantly lower number of research results. Although the conversational search paradigm is considered part of NLP and NLU, researchers do not consistently include these terms in titles, abstracts, or keywords.

We used Boolean operators to connect the keywords, hence the final search string is:

("conversational search" OR "conversational information retrieval" OR "conversational information-seeking" OR "information-seeking conversation" OR "information-seeking dialogue" OR "information-seeking dialog")

Sources

The following electronic databases were used to obtain the relevant studies: *ACL Anthology, SCOPUS, ACM Digital Library, Web of Science, ScienceDirect, IEEE Xplore*. We have limited the publication period to the last ten years to obtain state-of-the-art results. We performed the last database extraction at the end of the first quarter of 2022, so the publication period is (01/01/2012 - 03/31/2022). We applied our search string only for the study's title, abstract, and author keywords. Table 4.1 summarizes discussed search sources and and parameters.

Digital libraries	ACM Digital Library ACL Anthology SCOPUS Web of Science IEEE Xplore ScienceDirect
Publication period	January 2012 - March 2022
Search strategy	Title, abstract, author keywords

Table 4.1.: Search sources and parameters

Digital library	**Number of candidate papers**
ACM Digital Library	101
ACL Anthology	48
SCOPUS	46
Web of Science	9
IEEE Xplore	5
ScienceDirect	3
Sum	**212**

Table 4.2.: Results breakdown

We identified 431 candidate papers through the search process for the six predefined digital libraries. After that, we filtered out the duplicates and the documents by document type (which document types we discuss in detail below). As a result, we obtained 212 research papers. Table 4.2 shows the breakdown of results found for each database.

Snowballing

Software engineering lacks solid terminology, and software engineers tend to create new terms to describe new ideas that are closely related to each other [80]. It would complicate our search string to consider all possible related systems for conversational search. Instead, we used the forward and backward snowballing technique to reduce the complexity of the search string and obtain comprehensive results. Backward snowballing means that a search is based on the reference lists of papers known to be relevant (the included group by the search strategy). Forward snowballing identifies all papers that cite a known paper or a group of known papers [80].

Inclusion and exclusion criteria

We formulated inclusion and exclusion criteria relevant to the research questions for the study selection process. Table 4.3 shows the defined criteria, with the inclusion and exclusion criteria complementing each other.

Type	Exclusion criteria		Inclusion criteria	
Publication type	E1	Short papers, books, grey literature, doctoral symposium papers, lecture notes, editorials, summary of conference keynotes, comments, tutorials	I1	Peer-reviewed studies published in a journal or conference and early access articles
Language	E2	Not written in English	I2	Written in English
Access	E3	No access to full text	I3	Full text available
Quality	E4	With obvious factual errors and incorrect grammar or vocabulary	I4	With correct grammar and vocabulary from established journals and conferences
Topic	E5	Not related to the research questions	I5	Related to the research questions

Table 4.3.: Inclusion and exclusion criteria

4.2.2. Study selection

We exported the results as `.csv` or `.bib` files. There was no particular order for exporting or applying the search strategy to the databases, as we planned to remove duplicates in the subsequent steps. We used Python scripts to process the individual files containing the results. First, we read the raw `.csv` or `.bib` files into a separate raw data frame. Then, we harmonized the various column names and selected only the columns that were relevant to us. For example, some databases use the term *index keywords* instead of *author keywords*, or IEEE Xplore uses *IEEE terms*. We have grouped all these terms under the term *keywords*. We have also grouped terms such as *journal* and *proceedings title* under *publication title*. Generally, our relevant entries are *document type, title, keywords, abstract, publication title, source database, year, DOI, authors, URL, and affiliations*. The entries were saved in an Excel file with additional columns for further processing.

In addition, we preprocessed the entries, e.g., by lower-casing the title and DOI or removing various symbols to enable successful deduplication. We also double-checked with a script to ensure that the abstract, title, and keywords consisted of the search string to ensure that no errors occurred when exporting the publications. After this check, we removed no additional publications. The deduplication process was based on the title and DOI of the publication. The script detected 154 duplicate entries, which we removed from the list of eligible publications. In addition, the manual search detected another seven duplicate files. At that moment, we removed 161 publications and had 270 entries for further processing.

We automatically processed the E1 exclusion criterion with a Python script by defining the list of allowed types and removing those that do not meet the types. Hence, we automatically removed 58 papers that did not apply to the required publication type. Consequently, it resulted in 212 publications for further processing. We used the title, keywords, and abstract for the manual screening to evaluate the publications. If necessary, we also opened the full text of the paper to evaluate the inclusion and exclusion criteria. We defined separate columns for each of us in the Excel file where we entered our decisions to include or exclude the paper, and if we excluded it, by what criteria. We organized several rounds of discussion if we disagreed on including the paper in the preliminary results. First, we checked simple criteria, such as E2 and E3. Further, we used E5 to check whether the publication was eligible for a complete reading process. In fact, we never applied the E4 criteria, i.e., all results had no obvious factual errors or faulty grammar or vocabulary at first glance.

As a result, we included 47 publications in our primary set. Figure 4.4 summarizes the discussed search and selection process in a form of a flow diagram.

Quality assessment

In addition to the inclusion and exclusion criteria, we assessed the quality of the primary studies. As stated earlier, we defined but did not apply the criterion E4 to exclude the papers. There are two explanations for this: (1) we first checked whether the publication could answer at least one of the research questions we wanted to answer (I5). Hence, if the publication addressed E5, we excluded it first based on irrelevance according to E5, not E4, even if it contained grammatical errors, or (2) the publications did not contain obvious factual errors or incorrect grammar or vocabulary. If the publication was not excluded based on publication type, language, access, or topic (corresponding to E1, E2, E3, and E5), we verified that the authors of the publication answered the following questions:

QA1 Does the publication contain a clear statement of the research objectives?

QA2 Does the publication contain a detailed description of the proposed approach or solution?

Rather than numerically measuring these two quality criteria, we conducted a qualitative assessment. We found that all selected studies included clear objectives and a clear description of the proposed solution or approach.

In contrast, we numerically assessed the number of citations for the selected studies. We defined this quality criterion as follows:

QA3 Has the study been cited in other academic literature?

- **-1:** No. The study has zero citations.
- **0:** Partially. Between one and five scientific papers cited the study.
- **1:** Yes. More than five publications cited the study.

We found that 12% of the selected studies were not cited once by other authors, and the

majority of 64% were cited more than five times (Figure 4.2). By distinguishing the number of citations by year, we find that most not-cited publications were published either in the first quarter of 2022 or in 2021 (Figure 4.3).

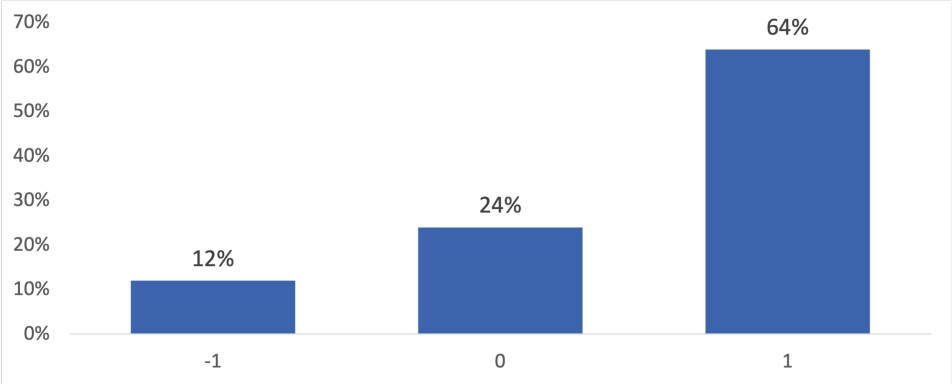

Figure 4.2.: QA3 criteria: citation count in percentage

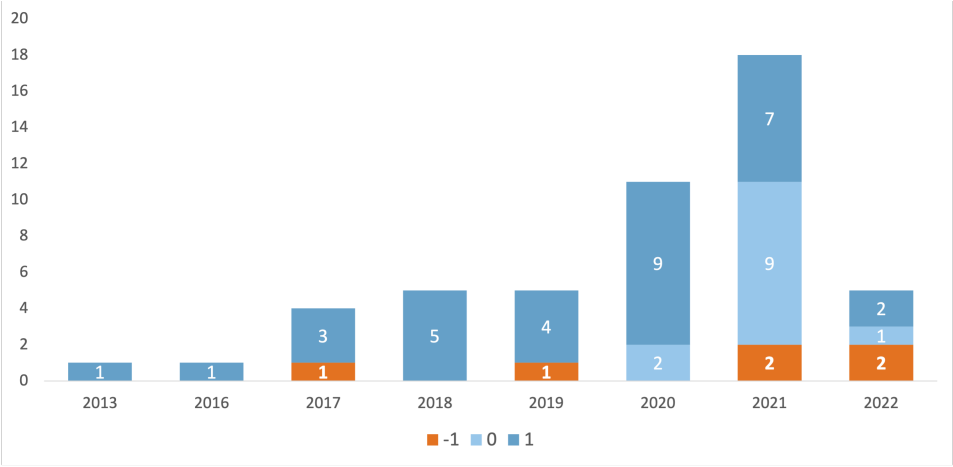

Figure 4.3.: QA3 criteria divided by year and citation count

4.2.3. Data extraction and classification

For the data extraction and classification step, we designed a data extraction form to collect the information from the primary studies. The form was designed during the research protocol development and aimed to collect important information to answer the research questions. We studied examples from Kitchenham [78, 80] and developed the extraction form which is summarized in Table 4.4.

(EF1) *Title* value was extracted from the primary research citation.

(EF2) *Year* value was extracted from the primary research citation.

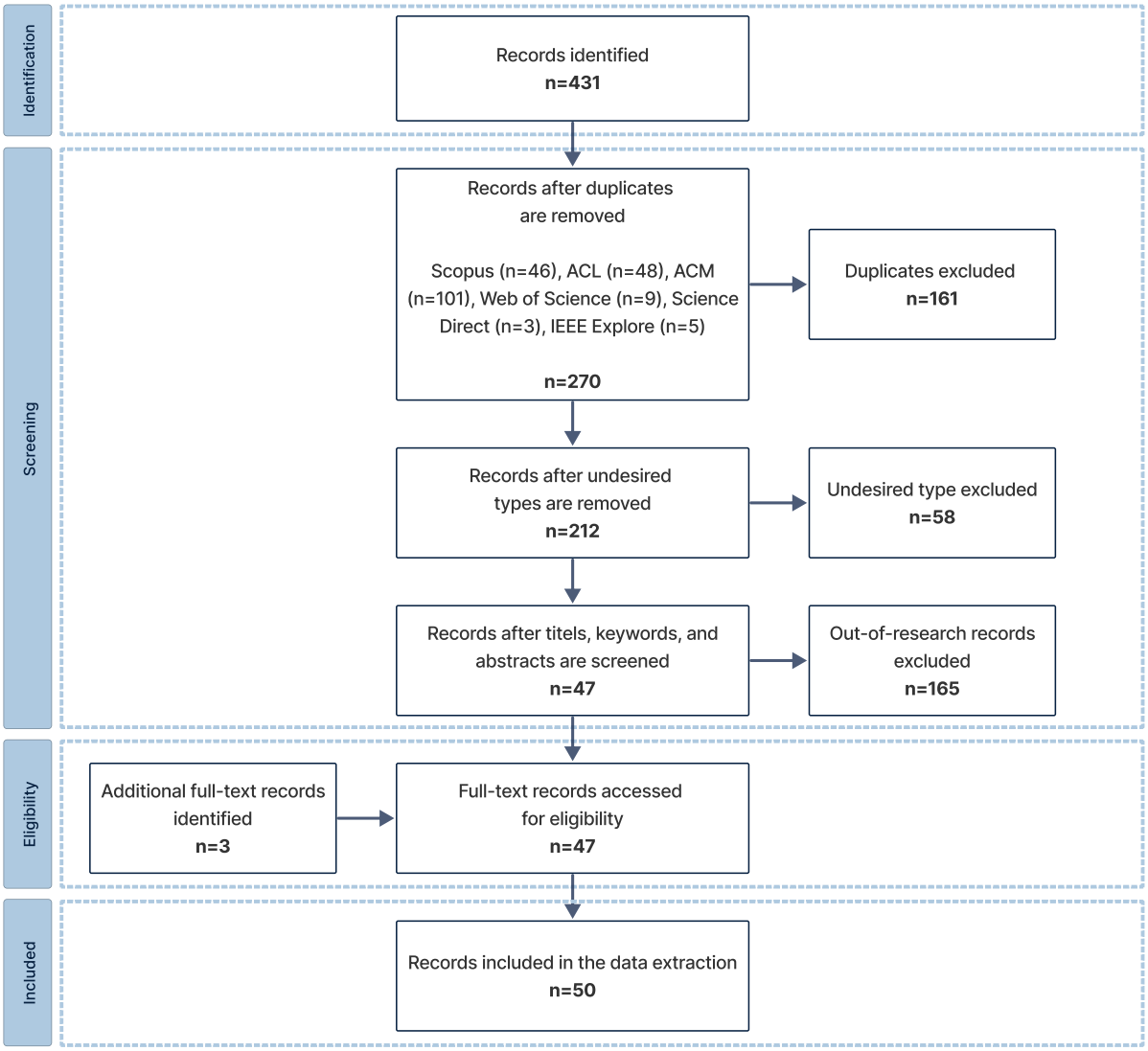

Figure 4.4.: Search and selection process

(EF3) *Country* value was extracted from the primary research citation.

(EF4) *Solution proposal*: The publication proposes a new solution or significant improvement to a technique or method by demonstrating its benefits, relevance, and applicability through a small example

Validation research: The research evaluates the characteristics of proposed solutions that have not been put into practice yet.

Evaluation research: The research presents an empirical evaluation of an implemented technique or method that has been put into practice.

Opinion papers: The research does not rely on methods or related work and expresses a

ID	Extraction field	Value
EF1	Title	
EF2	Year	
EF3	Country	
EF4	Research method	Solution proposal Validation research Evaluation research Opinion papers Philosophical papers Experience papers Secondary research
EF5	Research contribution	Technique Method Tool Resource Guidelines
EF6	Focus topic	Themes and codes (see Table 4.5)
EF7	SLR RQ	RQ1, RQ2, RQ3, RQ4
EF8	Application domain	Health, business, engineering, education, information technology, miscellaneous, etc.

Table 4.4.: Data extraction form

personal opinion of the authors about technology assessment or instructions about specific things.

Philosophical papers: The research outlines a new way of looking at existing things by structuring the field in the form of taxonomy or conceptual framework.

Experience papers: The research describes a personal opinion or experience of the author about how certain things should be done.

Secondary research: The research summarizes the results of primary studies to provide a systematic analysis of a specific topic.

(EF5) *Technique*: Approach or algorithm for performing a concrete task.

Method: Set of procedures and techniques for performing a concrete task.

Tool: Complete application, software library, or prototype that helps accomplish a specific task and combines multiple techniques and methods.

Resource: Compiled data set that supports techniques, methods, or tools.

Guidelines: Descriptions in the form of advice or guidelines derived from a synthesis of research findings.

(EF6) Themes and codes are introduced separately in Table 4.5.

(EF7) This value includes at least one *research question* on which research could provide insights.

(EF8) This value describes the *application domain* for which research has provided findings and solutions.

The research method and research contribution schemes were adapted based on the suggestions of Wieringa et al. [81]. In addition, the scheme for classifying the domains was adapted based on the work of Glass et al [82].

4.2.4. Data synthesis

We applied a narrative method as a synthesis approach for the qualitative analysis [80]. Based on this approach, our process of synthesizing the data looked as follows:

- We identified textual elements (phrases, sentences, paragraphs, tables, or images) based on the predefined themes and codes in each primary study.
- We coded each text element and saved the codes in our data collection as an Excel file. In addition to the code information, we documented the relevance of our research questions. In general, the frequency of occurrence of each code provided information about which research question(s) the study addressed.
- The codes were checked for consistency across the primary studies. The coding process was consolidated among members, and differences were discussed.

4.3. Reporting the review

We present the results by describing the summary of studies and answering each research question. The description of each research question was based on the data extraction results and insights from the classification strategy. We used the PRISMA checklist[1] as a checklist for questions to be reported in the SLR.

[1] "PRISMA 2020 checklist" `https://prisma-statement.org/`

RQs	Theme	Codes	Description
RQ1	Definition and characterization	Definition	Definition and terminology of conversational search systems.
		Property	Distinguishing characteristic aspects of conversational search systems.
		Functionality	Functional descriptions of the activities that the system can perform.
		Interaction	Interactive behavior between a system and a user, including user behavior, user experience, dialog actions, and dialog roles.
		Related concept	Theoretical aspects related to conversational search systems.
RQ2	Scenarios	Suitability of the scenario	Explanations why conversational search and its scenarios are suitable for use and integration.
		Task	Examples of search tasks that invite conversational search.
		Modality	Communication channels between a system and a user that invite conversational search.
RQ3, RQ4	Architecture and techniques	NLU / NLG technique	Description of the techniques and algorithms that implement the input and output functions of the systems, i.e., NLU and NLG techniques and algorithms.
		Dialog management technique	Description of techniques and algorithms related to the system's architecture, including dialog management part, and the findings on the implementation of the knowledge base.
		Architecture	Architecture description of conversational search system.
		Related systems architecture	Description of the architecture of systems related to the conversational search system.
		Scenario-architecture dependency level	Dependency level between a concrete scenario and system's architecture

Table 4.5.: Themes and codes definition

5. Results

In the previous chapters, we introduced the research, presented background topics related to conversational search, discussed related work, and described our research methodology in detail. This chapter composes the central part of the thesis, which aims to answer four defined research questions. First, we begin with a brief overview of the studies' metadata. We then proceed question by question and report the main findings of the included studies.

5.1. Overview of included studies

This section provides an overview of the 50 included studies within the SLR. In general, we obtained results on metadata such as distribution of publications per year, Number of affiliated papers by county, research method and contribution types, distribution of application domains, and distribution of contribution to research questions. We report on each of these items below.

Distribution of publications per year

As mentioned earlier, we restricted the year of publication to the last ten years, so we obtained publications between 2012 and 2022. Figure 5.1 shows the distribution of the number of publications by year. Overall, there is an increase in publications. The lowest number is in 2013 and 2016, while 2021 has the highest number of publications. We constrained the export to March 2022, so the results for 2022 are incomplete.

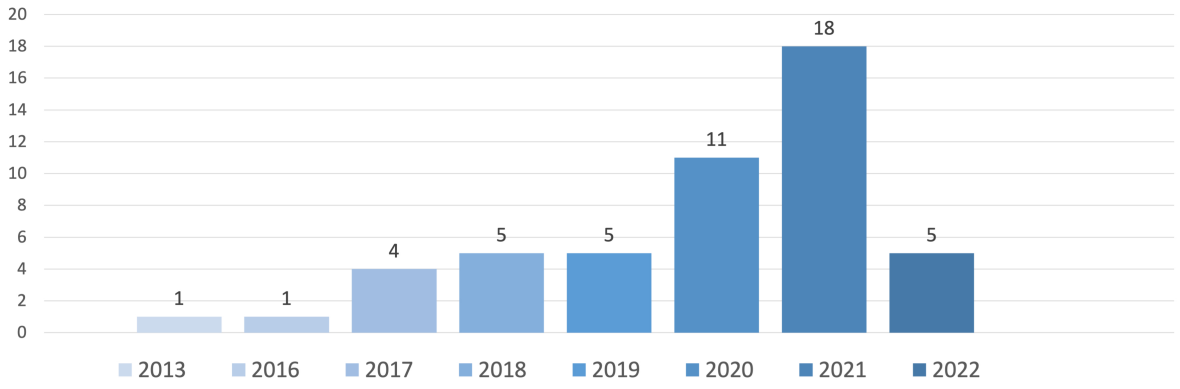

Figure 5.1.: Distribution of publications per year

Number of affiliated papers by county

We determined the geographic distribution of studies by assigning each study to one or more countries based on the geographic location of the research institutions. Several studies are assigned to multiple countries because some studies may be written collaboratively. Figure 5.2 shows the final geographic distribution. We can see that the United States accounts for the largest number of studies with 19, followed by the Netherlands and China. In general, 19 countries published at least one study on our research topic.

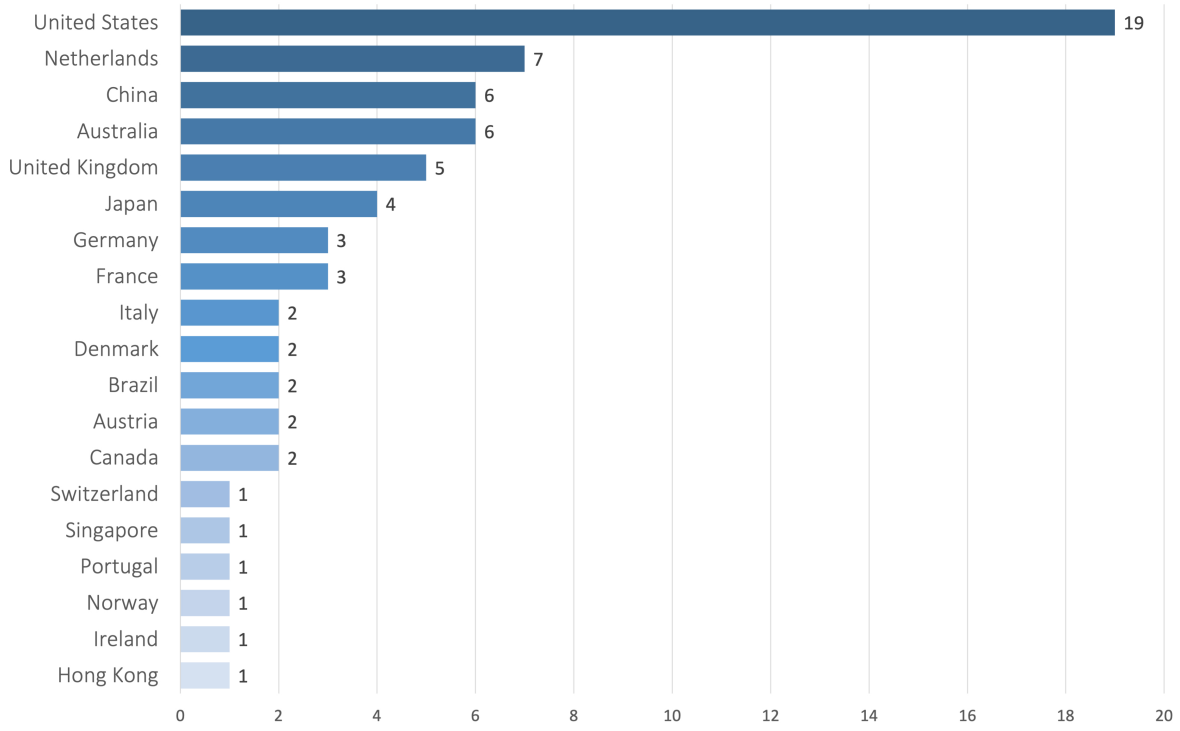

Figure 5.2.: Number of affiliated papers by county

Distribution of application domains

Researchers contributed seven application domains: health, tourism, business, law, aerospace, public sector, and miscellaneous. Only 11 of 50 papers described conversational search systems in a specific domain. Figure 5.3 shows the distribution within domains, with the business domain being the most popular. Figure 5.4 shows a general distribution of domains, including publications without domains. As can be observed, most publications did not conduct research for a specific domain.

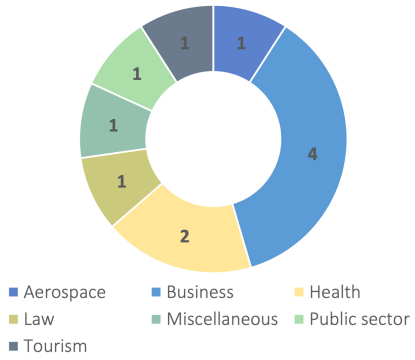

Figure 5.3.: Domains distribution

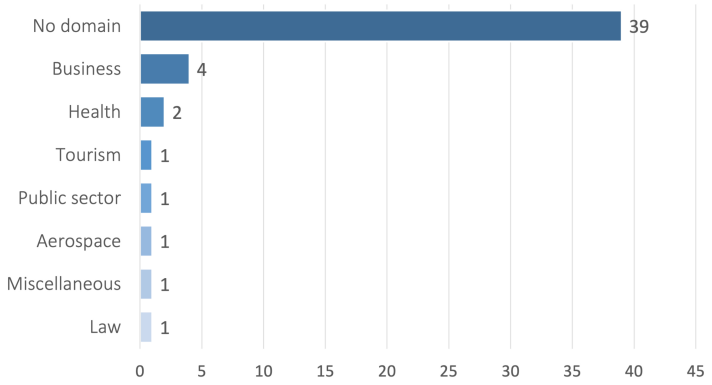

Figure 5.4.: Publications with and without domains

Research method and contribution

In the previous chapter, we presented our research method and contribution categories. The schemes were adapted based on the suggestions of Wieringa et al. [81]. Figure 5.5 and Figure 5.6 show their distribution for the included studies, although more than one method or contribution may apply to each study. We note that validation research was predominant between studies, followed by proposed solutions. Contributions are more distributed, with 39% tool suggestions and an almost even split between guidelines and methods.

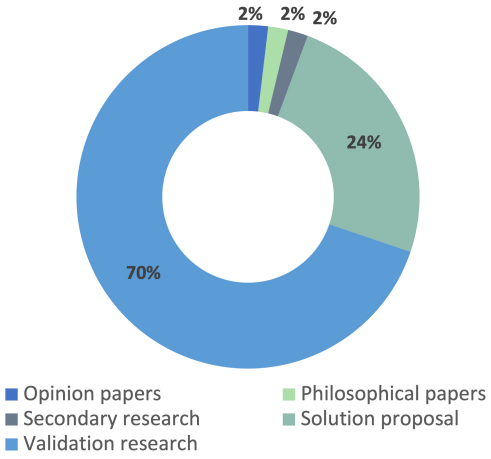

Figure 5.5.: Methods distribution

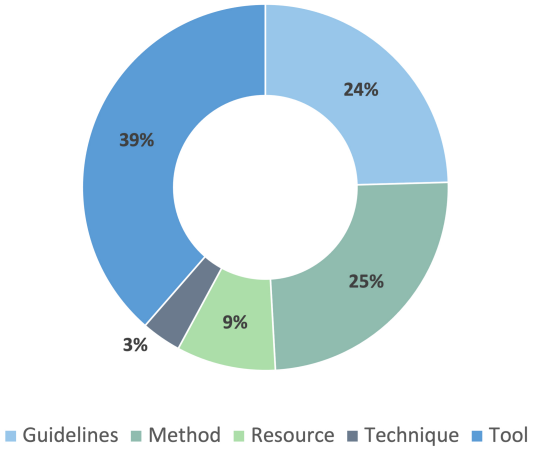

Figure 5.6.: Contribution distribution

5.2. Research question 1

Now that we have presented a brief overview of the included studies, this section aims to provide a deeper insight into the studies' content. We defined the first research question as:

RQ1: Which characteristics of conversational search systems are defined in the academic literature?

We use our coding structure to categorize our findings for the first research question. First, we discuss the proposed definitions of conversational search, its properties and functionalities, and finally, the related concepts for conversational search. We have also propose a conceptual framework for conversational search systems, shown in Figure 5.7. First, we show the primary goal of conversational search systems, then different categories for the definitions, and finally, we propose five characteristic properties.

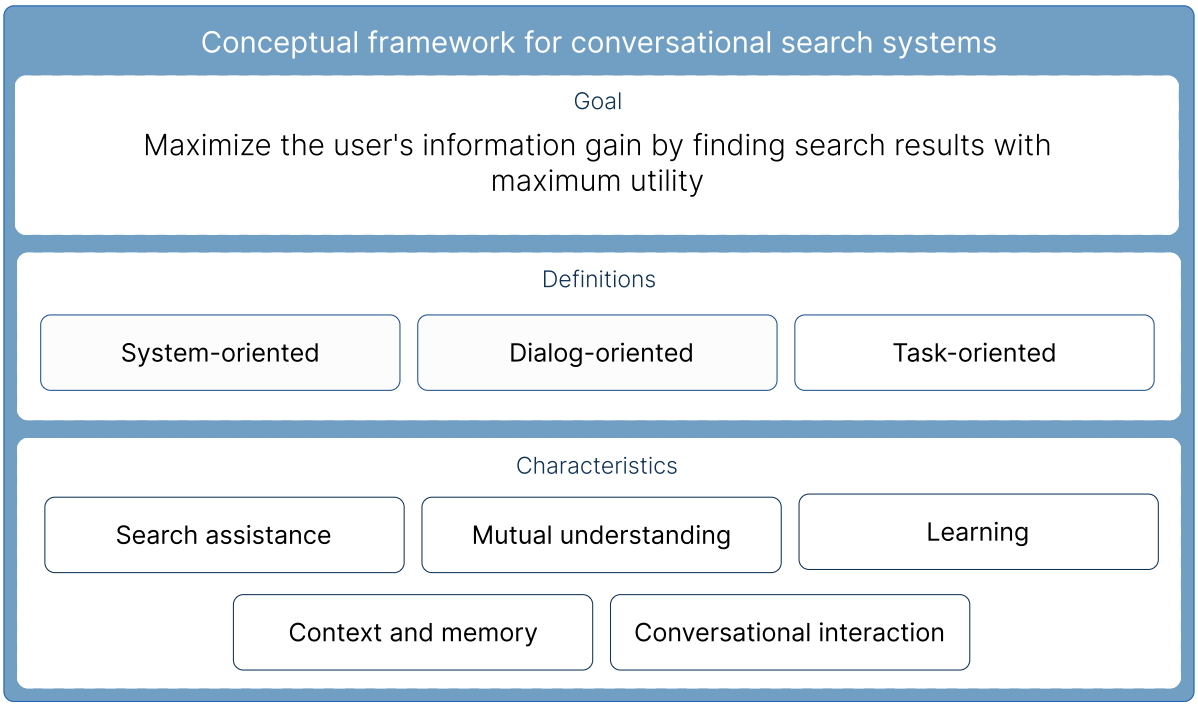

Figure 5.7.: Proposed conceptual framework for conversational search systems

The conversational search system's general goal is to maximize the user's information gain by finding search results with maximum utility. We divided the definitions into system-, dialog-, and task-oriented based on their primary focus. System-oriented definitions describe CS systems from the system perspective, dialog-oriented definitions describe the specifics of the interaction, and task-oriented definitions describe the tasks that the system needs to accomplish. In addition, we have proposed five characteristic properties that arise from the definitions: (1) conversational interaction, (2) context and memory, (3) mutual understanding, (4) search assistance, and (5) learning. We discussed these properties separately below.

5.2.1. Definitions

In this subsection, we aim to report all findings of how the studies' authors defined the emerging conversational search paradigm. We explicitly state the interpretations based on other included studies, as several studies described definitions of CS systems that were closely related. Table 5.1 summarizes the authors' proposed definitions for the paradigm based on one of the following types: (1) system-oriented, (2) dialog-oriented, and (3) task-oriented.

Definition	Source
System-oriented definitions	
"A *conversational search system* is a system for retrieving information that permits a mixed-initiative back and forth between a user and agent, where the agent's actions are chosen in response to a model of current user needs within the current conversation, using both short- and long-term knowledge of the user."	[S1]
"...the search performed by the users using a conversational IR system is a *conversational search*."	[S2]
"...*conversational search* is the development of systems that enable information retrieval using natural language interaction, i.e., a dialogue interface."	[S3]
Dialog-oriented definitions	
"*Conversational search* introduces dialog settings with flexible communication channels, such as where a screen or keyboard may be inconvenient or unavailable, where seekers may be unable to skim a search engine results page visually"	[S4]
"...such systems are conversational in that they assist users using a dialog interaction, be it in written or spoken form, usually with a rich human-like vocabulary."	[S1]
"*Conversational search* is an emerging area of research that aims to couch the information seeking process within a conversational format."	[S5]
Task-oriented definitions	
"*Conversational search* aims at finding or recommending the most relevant information (e.g., web pages, answers, movies, products) for users based on textual- or spoken dialogs, through which users can communicate with the system more efficiently using natural language conversations."	[S6]
"*Spoken Conversational Search* allows for progressing from an "action-response" search paradigm to a paradigm which has shared responsibilities between actors to succeed in the task. [...] users have to share their information need and ideally provide direct feedback to the system. Simultaneously the system will have to become more actively involved in deciding which results to present in a narrow audio channel."	[S7]
"*Conversational search* is a search paradigm where a user addresses information needs in a mix-initiative and multi-turn dialog."	[S8]
"One specific use case of chatbots is assisting with searching and retrieval of web content — a concept denoted as *conversational search*"	[S9]

Table 5.1.: Definitions of conversational search based on their type

First, we present basic term vital for the paradigm of conversational search, the definitions for conversation and information needs. Kiesel et al. [S4] defined the conversation specifically for the conversational search setting. They referred to it as *"an exchange of information between the seeker and the provider over a longer period of time."* Trippas et al. [S7] defined conversation as *"the natural mode for information exchange in daily life, a spoken conversational interaction for search input and output is a logical format for information seeking."* Deldjoo et al. [S10] named conversation *"an information exchange with more than two turns instead of commands such as setting a timer or turning on the light."* Also Vakulenko et al. [S11] defined conversation as *"a collaborative process that allows an information seeker to satisfy an information need."* While Shiga et al. [S12] defined an information need as *"a recognition that your knowledge is inadequate to satisfy a goal that you have."*

Next, we discussed the system-oriented definitions. Vakulenko et al. [S3] summarized definitions from various research studies, such as [S1, S7] and [20, 55], and defined the conversational search paradigm based on the type of dialog the system produces. The authors stated that CS *"is the development of systems that enable information retrieval using natural language interaction, i.e., a dialogue interface."* In this way, the CS system was defined in terms of the type of dialog it generates and is designed to automate, namely, information-seeking dialog. Radlinski and Craswell [S1] provided a popular definition for the CS system, which other authors have also referenced [S3, S8, S14, S13]. [S1] defined the CS system as *"a system for retrieving information that permits a mixed-initiative back and forth between a user and agent, where the agent's actions are chosen in response to a model of current user needs within the current conversation, using both short- and long-term knowledge of the user."* Sa and Yuan [S2] defined a conversational search as a *"search performed by the users using a conversational IR system."* Meanwhile, a conversational IR system becomes a conversational agent *"when natural language dialogs are supported in an IR system."*

Vakulenko et al. [S3] also mentioned the typology of the Dagstuhl Seminar [55], presented in the chapter 2, i.e., how CS systems have evolved from pre-existing systems by extending their capabilities. However, the authors explained that they examined the definition of CS systems not from the perspective of architectural design but the perspective of a dialog that the system produces. Unlike [S3], who defined conversational search by dialogue type, Aliannejadi et al. [S5] defined the paradigm based on several aspects: (1) the conversational strategy used (feedback-first (FF) or feedback-after (FA)), (2) the mixed-initiative approach (query clarification (QC) or query suggestion (QS)), (3) the number of feedback rounds, and (4) the number of document results presented.

In addition, we observed a pattern in which the definitions of conversational search systems are oriented based on dialog type and process. Aliannejadi et al. [S5] proposed that the conversational search paradigm *"aims to couch the information seeking process within a conversational format."* In the information-seeking dialogs, Deldjoo et al. [S10] described conversational understanding as *"a process of accurate representation of user information need in a multi-turn user–system conversation."* Kiesel et al. [S4] introduced conversational search as a *"dialog setting with flexible communication channels, such as where a screen or keyboard may*

be inconvenient or unavailable, where seekers may be unable to skim a search engine results page visually." Finally, [S1] described such systems *"conversational"* because *"they assist users using a dialog interaction, be it in written or spoken form, usually with a rich human-like vocabulary."*

We observe the last definition pattern in the directions of the system's tasks, goals, and purposes. Based on several works [S1, S14, S15], Xing et al. [S8] stated that a CS system is *"a search paradigm where a user addresses information needs in a mix-initiative and multi-turn dialog."* The authors reported that this search paradigm mimics conversational communication between a trained librarian and a customer. Kiesel et al. [S4] described a conversational search task as *"an exchange of messages: the seeker transmits one message to the provider, who replies with another message."* Trippas et al. [S7] defined Spoken Conversational Search (SCS) as a search paradigm based not on an "action-response" but instead on the task of establishment of *"shared responsibilities between actors to succeed in the task."* Zhang et al. [S6] stated a CS system *"aims at finding or recommending the most relevant information (e.g., web pages, answers, movies, products) for users based on textual- or spoken dialogs, through which users can communicate with the system more efficiently using natural language conversations."* Lastly, [S9] defined a concept of conversational search as a particular use case of a chatbot. The authors stated that *"one specific use case of chatbots is assisting with searching and retrieval of web content — a concept denoted as conversational search"*.

5.2.2. Characteristics

We discovered that the definitions in the included research papers often already contain some characteristics to describe the CS systems. In this subsection, we describe the main goals and popular properties relevant to CS systems in detail and propose our properties for conversational search.

One of the main goals of the CS system is to support and automate dialogs for information retrieval [S3]. Regarding information retrieval, [S1] defined the primary goal as the ability of the CS system to maximize user satisfaction by finding the results with maximum utility. This primary goal is similar to the primary goal we proposed for conversational search. However, we proposed to focus on maximizing the user's information gain rather than primarily maximizing user satisfaction. In addition, the system must keep track of the user's expectations, select an action in the current context, and interpret the user's responses concerning the previous context of a conversation. [S5] and [S14] also mentioned that the CS system's foremost goal is to maximize the user's profit while minimizing the cost. The system must constantly decide what response to give for each interaction, considering and estimating the user's satisfaction [S14]. [S2] described that the conversational search process must take the form of a dialogue to help users better identify and understand their information needs. The conversation's success is measured by the relevance value of the response or the answer accuracy in case the response is an answer to ground truth [S11]. [S16] claimed that due to the flexibility of mixed-initiative interactions, CS systems can capture the user's true intent and thus provide valuable information to a user with human-like responses. Liu et al. [S13] stated that one of the main goals of CS systems is to enable the delivery of information services in

an interactive style that resembles human-to-human information-seeking interaction.

Radlinski and Craswell's study [S1] made a significant contribution to research on conversational search by defining five properties desired for CS systems which are listed below. Several included studies [S17, S3, S5, S2, S18, S11] have mentioned or build up on these properties.

User Revealment: The system helps the user express (potentially discover) their true information need, and possibly also long-term preferences.

System Revealment: The system reveals to the user its capabilities and corpus, building the user's expectations of what it can and cannot do.

Mixed-Initiative: The system and user both can take initiative as appropriate.

Memory: The user can reference past statements, which implicitly also remain true unless contradicted.

Set Retrieval: The system can reason about the utility of sets of complementary items.

In a different approach, Trippas et al. [S7] suggested design recommendations for spoken conversational systems. They proposed integrating **search assistance**, **grounding as relevance feedback**, **visibility of system status**, and **navigational interactions**. Before describing our proposed properties (Figure 5.7), For instance, Radlinski's memory property is part of our **context and memory** property and user and system revealment are part of the **mutual understanding**. Radlinski's set retrieval is part of our **search assistance** property and grounding partially describes our **mutual understanding** property.

Conversational interaction

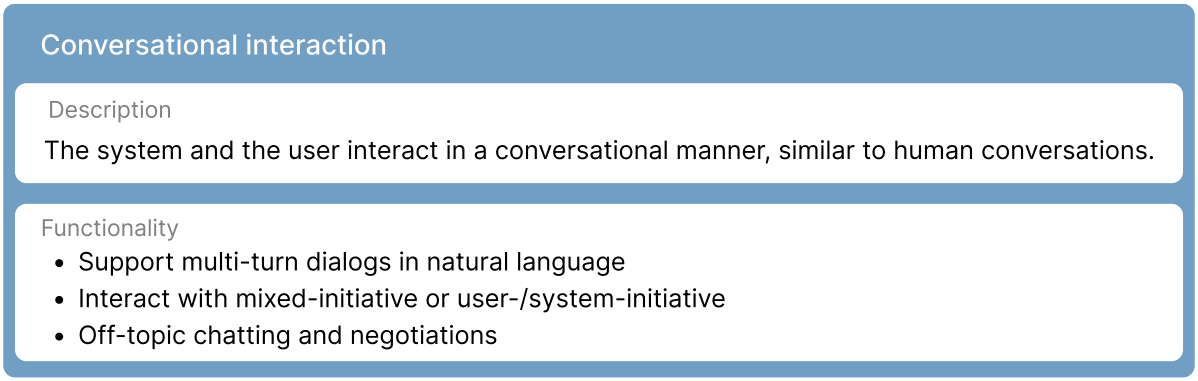

Figure 5.8.: Overview of conversational interaction characteristic

The first characteristic property we discuss is conversational interaction, i.e., the interaction between the system and the user in a conversational manner, similar to human conversations (Figure 5.8). We define that the CS system supports the following functionalities: multi-turn interaction, interaction with mixed-initiative between participants or user- or system-oriented

initiative, and the ability to have off-topic chats or negotiations. Multi-turn interaction is one of the fundamental features of human conversation, so we do not discuss this functionality separately. This research first gives a brief overview of how the authors of the included studies described the conversation process. We report findings about what action spaces a system and a user has and what behavior participants exhibit during a conversation. We also describe the specifics of utterances during the search and the associated challenges for the system. Finally, we separately discuss a mixed-initiative, and user- or system-focused initiative.

Shiga et al. [S12] stated that search interactions are becoming more conversational. CS systems improved both in terms of the richness of expressed information needs and person-to-person-like conversations [S4]. Kiesel et al. [S4] further referenced a set of maxims for cooperative conversation: say as much as you need, but no more; say what is true; say what is relevant; have good manners. Formally, [S19] described that in conversational search, users start the conversation with an initial information request q, with or without additional context description h. The system aims to provide an answer a to the request q. Before answering, the system may ask a particular number n of clarification questions cq_i to get more information about the user. [S20] assumed that the user specifies his information need or topic t in the initial information request. However, the system has yet to determine the true facet of the information need f_t. [S21] described the main stages of the conversational search process as (1) identifying the user's information need, (2) presenting the results, and (3) continuing the search process until the user is satisfied or abandons the process. Communication ends when the user successfully completes the search or abandons if unsuccessful. The success criteria of a conversational interaction are based on the information results, i.e., whether the search results satisfy the user's information needs and whether the interaction positively influenced the user's emotions [S15].

The researchers also proposed certain behaviors and actions that users and a CS system can perform during a conversational search. Azzopardi et al. [S22] summarized action spaces for users and CS systems, which are shown in Table 5.2.

In comparison to [S4] who described the number of user participants as unlimited, [S7] limited dialog participants to two actors and noted that the multi-turn dialog process goes beyond seeking outcomes and also includes feedback interaction or judgments. Trippas et al. [S7] proposed off-topic chatting and negotiating as additional functionalities for CS systems. The negotiation utterances help to bridge differences and reach agreements. Also, [S21] described various actions during the conversational search: clarification seeking of ambiguous queries, suggesting queries, and showing key details from the results set. [S19] limited the system's action space into either answering the query or asking clarification questions. [S23] proposed a meta-communication part with the following system actions: asks to repeat, query refinement offer, or actions confirmations. [S14] referred to asks to repeat actions as "repair" action, which has utterances like *"Sorry say that again"*, *"Can you repeat that please"*.

Users can also show certain behavior and perform various actions during a conversation. For instance, [S24] discussed flexible user action spaces, noting that users can freely interact

User			Agent		
	Query Formulation	**Reveal** Disclose, Non-disclose Revise Refine Expand	**Inquire** Extract Elicit Clarify	User Revealment	Memory
Set Retrieval	Result Exploration	**Inquire** List, Summarize, Compare, Subset, Similar	**Reveal** List, Summarize, Compare, Subset, Similar	System Revealment	
		Navigate Repeat, Back, More, ... , Note	**Traverse** Repeat, Back, More, ... , Record		
Mixed Initiative		**Interrupt** Interrupt	**Suggest** Recommend Hypothesize		
		Interrogate Understand, Explain	**Explain** Report, Reason		

Table 5.2.: Overview of user and system actions in conversational search [S22]

with a system through natural conversation, without constraints on vocabulary, grammar, or choice of intent, and freely navigate through any web page. Also, Trippas et al. [S7] asserted that users could formulate their information requests at any time to satisfy their information needs, whether in the form of a request for a document or a request for meta-information about a document or a clarification of search intent. [S25] and [S26] established questions of nine categories: What, When, Where, What, Who, Why, How, Yes/No, and Compare. Conversations in the dataset typically began with a "What" question and were likely followed by another "What" question. In contrast, the end of the conversation had no clear candidate. The authors found that the longer the conversation, the more likely it was to contain multiple topics, i.e., to elaborate further or to start a new topic. [S20] described the user's flexible options to (1) provide additional information for the search after the initial request or (2) not cooperate with the system. In the first case, the user can provide an informative answer by (1) leaking the intent partially, e.g., "No, not even close", or (2) leaking the intent completely, e.g., "No, I am looking for $*intent*". In the second case, if users consider clarification questions not worth answering, they usually leave the conversation [S19]. [S27] proposed the following categorization of user responses: (1) null action, (2) interruption or negotiation, (3) relevant response, (4) postpone, (5) criticism or clarifying response. The difference between null action and postpone action is that in null action, the user gives no response, and in postpone action, the user responds but asks to be reminded later of the conversation. Although several researchers stated that users should feel free to speak up at any point in a conversation, [S28] observed that users were often hesitant to interrupt a system's presentation of results and miss the flow. Unfortunately, after the presentation, they had difficulty remembering the questions they wanted to ask.

Several studies described user behavior during a conversation. Salle et al. [S20] proposed

two behavioral user metrics - cooperativeness and patience. The authors represented cooperativeness as a Bernoulli random variable based on the level of will to cooperate. For example, $cooperativeness = 0$ means that the user only gives yes/no answers, and $cooperativeness = 1$ means that the user always gives informative answers. The degree of patience provides information about the maximum effort the user is willing to put into the conversation. Also, Wang et al. [S19] described patience as a behavioral metric for a user's willingness to interact. Users have different expectations for CS efficiency measured by the number of questions the system asks before it finally provides the resulting answer. Additionally, the authors described tolerance τ to bad clarifying questions. τ is measured by the number of bad clarifying questions the user can tolerate before leaving the conversation.

Frummet et al. [S29] discussed the conversational interactions during the cooking process. The authors found that users frequently used discourse markers such as "Mhm", signaling that they understood the system's instructions and wanted to proceed with the cooking process (Example 5.1). From the systems perspective, they help link conversational units.

Example 5.1 | Discourse markers usage [S29]

Participant: OK. Good. (Alarm bell rings). Now we need to add zucchini and corn, right? Zucchini, corn and —
Assistant: Exactly. You need to mix corn, zucchini, and chickpeas.
Participant: **Mhm?**
Assistant: Fill it up with half a liter of water and add the tomato paste.

In their analysis of information-seeking dialogs, [S15] proposed to separate utterances into four classes (QRFA): for the user - *Query* and *Feedback*, and for the system - *Request* and *Answer*. Query provides context or input for the search process or prompts an action to perform. The request provides a system's pro-active request to gain more information or feedback from the user. The authors categorize feedback by the user into positive and negative. The answer is further categorized into (1) non-empty results set, (2) backchannel response, e.g., "One moment, I will look it up", or (3) empty result set. Figure 5.9 illustrates the theoretical model of information-seeking conversation proposed by [S15]. Moreover, Vakulenko et al. [S3] grouped dialog datasets into *search*, *sharing*, and *support* based on their structural pattern. According to this pattern, dialogs in which the user (seeker) asks most of the questions and the system gives most of the answers $(QA > RF)$ are search dialogs. When the system asks most of the questions $(QA < RF)$, it is a support dialog. The symmetry of the speaker roles indicates information-sharing dialog $(QA \approx RF)$. However, the authors could not find a structural difference between dialogs that are information-seeking and dialogs that are not information-seeking, such as task-oriented, knowledge-based dialogs.

Ren et al. [S16] have identified that conversations may contain more content than necessary due to grammatical and semantic integrity. When users have unclear information goals or needs or are unfamiliar with the topic, the utterances they create tend to be semantically non-compelling. Hence ellipses and anaphors are expected in such conversations. Salle et al. [S20] proposed the CoSearcher system in their research. They observed how ambiguous

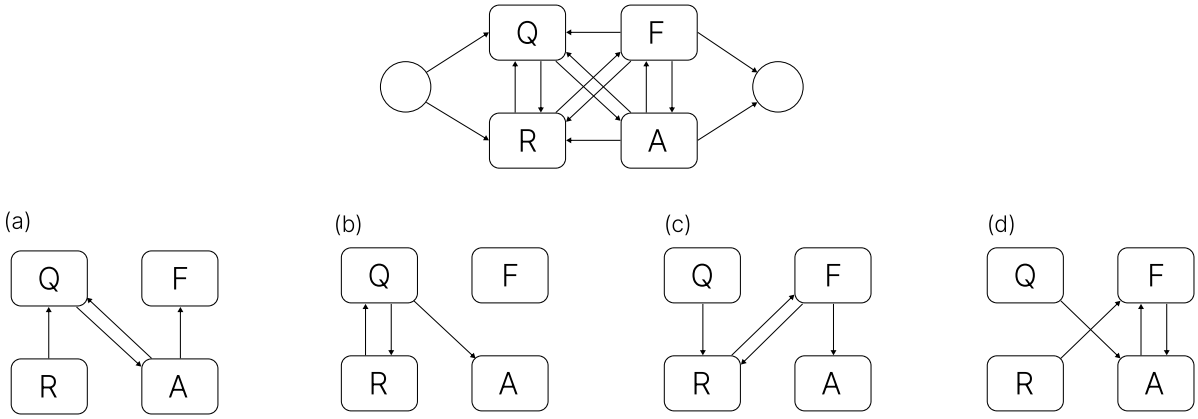

Figure 5.9.: Four cycles of information-seeking for QRFA model of conversational search: (a) question answering loop, (b) query refinement loop, (c) offer refinement loop, (d) answer refinement loop.

entities led to lower success rates across facet providers and emphasized the importance of the topic ambiguity. For instance, an entity with multiple senses is "iron" - a chemical element, clothing iron, or nutritional supplement. Another example of an ambiguous query is when it contains context terms that do not specify what should appear in the result document but rather describe the desired document. For example, a query that contains the term "pdf" may refer to the expected file type [S4].

Vakulenko et al. [S3] defined initiative as control over the direction of the multi-turn dialog flow. Several authors [S1, S17, S3, S11, S5, S2, S30, S27] proposed mixed-initiative as one of the key properties for conversational search. [S10] presented two viewpoints on multi-modal interaction: human-centered and system-oriented. We will discuss the mixed-initiative, user-focused, and system-focused initiative approaches separately.

[S1] defined conversational search as a mixed-initiative system since the information is exchanged and does not come only from one source, e.g., as in a lecture. [S5] suggested that the idea of mixed-initiative and system agency has led to the development of CS systems and increased user engagement and satisfaction. The degree of the initiative depends on the interaction capabilities with a system. If a system allows the user to return an unstructured text instead of structured preference-based input, the user can take the initiative at any time [S1]. Sa et al. [S2] made the same observation, noting that when the system takes the initiative, it offers multiple options to the user or asks for preferences. In contrast, the user has the flexibility to change the topic by entering free text. [S17] described this behavior for the system as *proactive* during the system's initiative and *reactive* during the user's initiative, while these states are constantly changing during the interaction. Conversations in practice use mixed-initiative interaction, where conversation control constantly shifts between participants through assertions, commands, questions, and prompts [S1]. [S3] described a study in which the researchers observed that the dialog pattern of a mixed-initiative interaction depends on the distribution of the participant's knowledge. For the mixed-initiative, multiple-agent

scenario, [S27] described a possibility for agents to negotiate mixed-initiative and coordinate who is better qualified to complete a task.

Zamani et al. [S30] noted that the CS system tends to take more leadership in the conversation to help users better express their information needs. [S6] proposed a "system asks - user responds" paradigm, where a user starts a conversation, but the system actively asks questions to understand the user's needs and provide competent results. [S27] focused their research on the system-initiative interactions for CIS systems and created a taxonomy based on three dimensions:

(1) initiation moment (when to initiate)

(2) initiation purpose (why to initiate)

(3) initiation means (how to initiate)

The initiation moment was divided into instant initiation and opportune moment initiation. To initiate a conversation immediately, the system must consider the user's situational context, such as the interaction's location, time, mood, or urgency. Further, the authors defined the following purposes for the system to initiate a conversation: information filtering, recommendation, follow-up on a past conversation, contribution to a multi-participant conversation, and feedback request. Example 5.2 illustrates the initiation of the conversation with the purpose of recommendation based on the initiation moments. The initiation depends on the system setting, e.g., in the multi-device setting, the system must decide on which device to initiate the conversation; in the multi-modal setting, the system must decide which interaction channel to use. [S27] concluded that the initiator's decision-making component is one of the essential components in CIS systems and stated that system-initiated conversations are a type of mixed-initiative CIS, hence categorizing this type of interaction as part of mixed-initiative interaction.

Example 5.2 | Streaming/filtering information based on user profile [S27]

Instant initiation: Events that may result in a security risk or danger to the user should be mentioned immediately.
Opportune moment initiation: A system-initiative agent can initiate a conversation at the right time to inform the user about different news topics based on the user's preferences.

Ren et al. [S31] emphasized the importance of mixed-initiative in CS systems. Nevertheless, they did not include it in their proposed system because their proposed dataset, built on the TREC CAsT dataset, is based on the opposite paradigm of mixed-initiative - "user asks - system responds." [S31] has explored this issue for a conversational question-answering scenario with search engines. We have already discussed the classification of search, exchange, and support for conversational dialogs presented by [S3]. According to the authors' definition, search dialogs are user-initiative since the user asks most of the questions. In the conclusion of their work, the authors suggested that conversational search systems should evolve from a type of *search* to a type of *share and support* by taking more initiative in a conversation.

Context and memory

The second characteristic we propose in our conceptual framework for CS systems is context and memory (Figure 5.10), i.e., the ability to preserve conversational context and store conversational history and user data so that both a user and a system can refer back to previous statements.

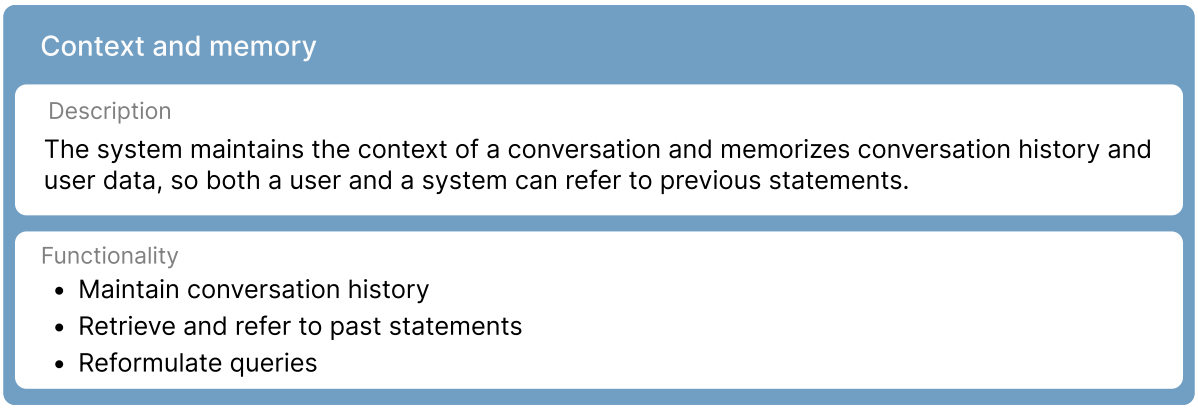

Figure 5.10.: Overview of context and memory characteristic

Conversational context-maintenance was proposed as one of the pivotal properties of CS system [S25, S32, S31, S2, S5]. Incorporating the conversational context during the interaction positively impacts the retrieval performance [S25] and reduces the system's complexity [S24]. [S2] and [S16] described the application of the previous search process in the current context as an essential feature of the CS system. This functionality allows the system to correctly interpret utterances such as "I would like to know more about *the second one*." or "Where is *it* located?". The authors defined three forms of context:

Visual context: the ability to refer to objects on the page.

Dialogic context: the ability to argue and perform complex interactions and co-reference resolutions.

Personal context: the ability to prioritize choices based on personal preferences.

[S25] defined an utterance as relevant to the current utterance if: (1) it clarifies the current utterance, or (2) it increases the information needs for the current utterance, or (3) it involves the current utterance. Mele et al. [S33] defined utterance rewriting as one of the most important mechanisms for resolving contextual dependencies in continuous conversations. The context, in this case, as defined by the authors, is a theme of an utterance in the multi-turn conversation. For utterance rewriting, the system takes the current utterance u_i and context keywords as input and rewrites the current utterance u_i according to one of the rewriting strategies defined by the system. For context detection, the system receives as input the current utterance u_i along with the sequence of previous utterances $u_1, ..., u_{i-1}$. Then, context detection tries to understand whether u_i is self-explanatory or needs to be enhanced with context obtained from the previous utterances of the conversation.

Deldjoo et al. [S10] discussed that research on co-reference and ellipsis resolution exists mainly for text-based conversations and should expand to multi-modal conversations. Xing et al. [S8] stated query rewriting shares similar functionality both for text-based and voice-based reformulation and divided the process into three steps: (1) specification, (2) rephrase, and (3) generalization. The context should be connected across modalities in multi-modal systems to enable successful query rewriting in a current context [S10]. Furthermore, Kumar et al. [S34] argued the effectiveness of co-reference resolution for context resolution functionality because co-reference models have difficulty resolving conversational input. Thus, an ineffective representation of the user's information need can lead to poor recall of information passages and thus the response. [S35] described the co-references resolution as a challenging functionality, significantly depending on the spoken language. In French, for example, possessive pronouns agree in gender and number with the noun they introduce. Consider an example of subsequent questions, *"Who is Michael Jackson?"*, *"What is his father's name?"*, *"And his mother's?"*. In English, "his" was used in all cases, but in French, it changes to *son père* (father), *sa mère* (mother), *ses frères* (brothers), which should be considered when resolving co-references. In addition, this example illustrates the resolution of elliptical questions in context, in that the question *"And his mother's?"* must be rewritten as *"What is his mother's name?"*.

The cognitive model has also been referred to as an essential component in CS systems [S14]. Trippas et al. defined the "remember" task for a CS system as *"retrieving, recognizing, and recalling relevant knowledge from long-term memory"*. This task was the least complex of the three tasks, remember, understand, and analyze. [S14] suggested that after the system creates a user model, the cognitive model should include information about (1) how the information should be presented to the user (form) and (2) what information should be presented to the user (content). [S4] described collective memory as a collection of messages that both a user and a system can refer to. In addition, for multiple interactions, the system can provide memoryless refinement for the user, i.e., the user can learn the correct terms to describe their information need when it has not yet been determined [S1]. [S1] and [S32] offered another advantage of the store for the user - the avoidance of remembering complex queries by collecting the user's personal data. Thus, memory functionality leads to non-scripted conversation adapted to the current context [S1].

For the emerging paradigm of conversational search, the memory property plays a key role by (1) retrieving past statements and (2) referring to past statements, for example, to contradict them [S1]. Radlinski et al. [S1] claimed that the first functionality is provided by default because conversational search is a continuous process, and the second is addressed by the requirement to allow the user to enter free text. For the system, the ability to query previous statements may be associated with the need for clarification, e.g., "What I meant was...", and for the users, the need to reconsider or change the input made in previous statements because they incorrectly provided an answer. Moreover, the traceable history provides further transparency and explainability of the system [S7].

Aliannejadi et al. [S5] described memory integration for the case to provide only unseen results. Continuing, [S14] proposed to memorize seen results and make them distinguishable,

e.g., by highlighting them with a different color, like in the traditional SERP results. Finally, the authors argued that it would be difficult to establish the same in an audio-only environment because of the lack of visual feedback.

Search assistance

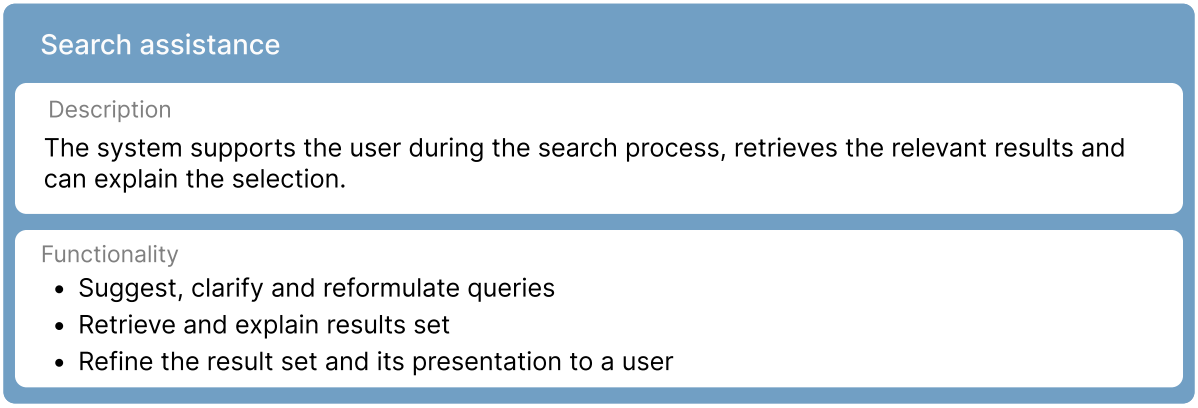

Figure 5.11.: Overview of search assistance characteristic

The third distinguishing characteristic we propose is search assistance (Figure 5.11). We define the primary goal of search assistance as assisting the user during the search process, retrieving the relevant results, and explaining the selected items. Hence, the CS system has a functionality that allows manipulating user queries, for instance, making suggestions, clarifying, or reformulating. Further, the CS system can retrieve and explain the results and, if necessary, refine the results and their presentation.

Several authors described the system's ability to support and assist the user. Trippas et al. [S7] proposed a capability of the CS to provide search assistance, where the system assists the user in searching by providing search suggestions, advice, or relevance estimations. Both actors can initiate the assistance process. The CS system additionally takes the form of a personal assistant that works alongside the user [S21]. [S21] proposed the user can interact directly with the search engine while the assistant helps them formulate their query and interact with the content they find. Navigation capability allows users to maneuver in the (multi-dimensional) information space [S7].

Functionalities to suggest, clarify and reformulate the user's search queries are essential for search assistance. During a conversation, the system elicits the user's preferences and constructs a model of the user's information needs [S1]. One of the strategies for revealing user preferences is to help refine their information and priorities, for example, by presenting multiple alternatives within the range of available items. It is easier for the user to clarify their needs by having a clear choice, rather than expecting them to come up with independent choices [S1]. In addition, selection options help point out relevant product features when the user is unfamiliar with the domain, for example, when a parent buys a stroller for the first

time [S1]. [S5] proposed several strategies for improving users' information needs in a mixed-initiative context: (1) query suggestions or refinements, or (2) or query clarifications. Which strategy is most appropriate depends on the interchange between the relative costs of query and feedback, the performance of the original query, and the total amount of gain obtained. [S18] mentioned that there are already functionalities that try to address the problem of user's information need understanding, such as search query auto-completion and search query suggestion techniques. Trippas et al. [S14] observed three stages in the search session: (1) query formulation, (2) search result exploration, and (3) query reformulation. Salle et al. [S20] suggested that CS systems can help users identify their information needs through a series of questions. Moreover, the system can ask for confirmations, request a rating or a critique, or display a partial items selection [S1].

We consider the property of set retrieval as a part of the search assistance. Radlinski and Craswell [S1] defined set retrieval as the system's ability to present a set of high utility elements as proposed solutions to a user and after presenting the set of complementary elements as alternative solutions based, for example, on the user's criticism. For the search of the set items, the system determines a combined utility function of the set, the form of which depends explicitly on the type of information need that the user addressed. [S5] represented set retrieval as one of the essential properties of the CS system. The authors described this property as the system's ability to operate with, manipulate, and draw conclusions about a set of objects retrieved for the user based on the conversational context. [S18] indicated that difficulties in result retrieval, such as retrieving unreliable and low-quality results, are caused by the user's vague information requests, as the users may not have a clear idea of what they want to search. Papenmeier et al. [S17] called narrowing down a broad range of specific items to clarify the user's information needs as a funnel strategy. Finally, Kiesel et al. [S4] referenced various behavioral, affective, and physiological strategies for improving the relevance of the result set, such as dwell time or tracking the user's facial expressions and measuring the user's heart rate or neural activity.

Rosset et al. [S36] discussed the relevance of valuable information in the response set. They argued that "click-bait" suggestions are not always valuable for the user, and the CS system should not be susceptible to such responses. Research into implicit relevance feedback has attempted to use these post-retrieval aspects. [S25] proposed an extension of the Query Likelihood Model, a linear interpolation of the language model based on user input and the language model based on user feedback, as a method for retrieving a ranked list of items. Aliannejadi et al. [S5] defined two strategies for soliciting feedback from a user: (1) feedback-first (FF) and (2) feedback-after (FA). The user performs a query-feedback loop in the FF before evaluating the items. The CS system with this strategy is like a librarian or a booking process. The system first makes suggestions to refine the user's information needs or asks clarification questions before presenting the final response to the query. For the FA, the user performs evaluation loops where they provide feedback after evaluating several items and repeating the process several more times. This approach is more exploratory in that the user first learns about the topic and then provides feedback to the agent so that it can move its search in the topic space to advance its search. The authors emphasized that in practice,

the mixture of these two strategies will result in an effective CS system.

Zamani et al. and Kiesel et al. [S30, S4] identified result set explanation as a significant research area and valuable function for complex search tasks. [S22] suggested that a system must be able to report on its understanding, justify, and reason why it has chosen a particular course of action or made a specific proposal. The authors provided the following dialogue excerpt as an example.

Example 5.3 | Results explanation example [S22]

User: Why did you recommend going to Tuscany?
Assistant: Tuscany is a beautiful region of Italy known for hot days and warm nights, and has a variety of interesting sites to visit with cultural significance.

Further on the topic of the results explanation, [S7] suggested that visibility of system status would provide transparency and control over the system's actions and outcomes. Thus, under this condition, the system must disclose the decision explanation to present the selected outcome, especially since it is challenging in a purely voice-based manner. [S4] claimed meta-information about the provider's actions, stored as conversational context, can facilitate the process of reasoning about the results set. [S2] stated that the system must check information sources. In text-based search, the titles of the snippets must contain a hyperlink so the user can click on it and review the source results. In voice-based search, the system reads out the results' titles, and the user can select a specific result and ask to go to it for further analysis. [S32] integrated hyperlinks with the results' source into their proposed system not by default but upon request, suggesting that supporting the system's credibility with trusted sources is essential. [S4] suggested that the domain knowledge of both the user and the system can improve the reasoning process of results utility, rank the results, and provide appropriate suggestions.

Trippas et al. [S7] discussed that CS systems can read, interpret, or provide an overview of the results set. The system can refine the result set or its presentation to the user to give an overview of the results. [S2] proposed to refine the search results to a specific category, e.g., news or images, which were confirmed by a user, before presenting them to the user. The authors proposed three ways to present the results in conversational search: (1) by default ranking, (2) by the source of the result, and (3) by the keywords, i.e., the terms in the query after removing the stop terms displayed in the results. Synthesis of summaries from the obtained results was recognized as one of the CS functionalities [S4]. Zhang et al. [S37] proposed a concept of summarizing results from tabular data. The authors argued that such functionality leads to the system driving the conversation - the summary gives users hints that can give them an idea of what to ask next. Hence, [S37] claimed that a summary of results must meet the following characteristics: (1) linguistic quality: must be concise and easy to read; (2) relevance; and (3) conversational appeal. [S32] experimented with human participants and found that some users would like the system to be able of opinion aggregation, i.e., to aggregate opinions from the response set and present these opinions in summary form. [S30] proposed to integrate "People Also Ask" (PAA) functionality that is

also used by commercial search engines.

In order to appropriately limit the level of detail or scope of the conversation, the CS system needs to know how it should present the relevant results [S4]. For instance, for the user's request, *"Could you give me details about the Spanish flu? But I don't have much time."*, the response *"Sure, here is an in-depth article about the historical relevance of the Spanish flu"* would be inappropriate [S4]. Multi-modal CS systems should thus handle the presentation of responses in multiple modalities, e.g., displaying an image in addition to a text or voice description of that image [S10]. In addition, [S32] discovered during their experiment that there are two groups of users: those who prefer extended responses with a broader context and those who prefer direct responses. The first group justified their preference for extended responses to ensure that the system understood their genuine information need.

Mutual Understanding

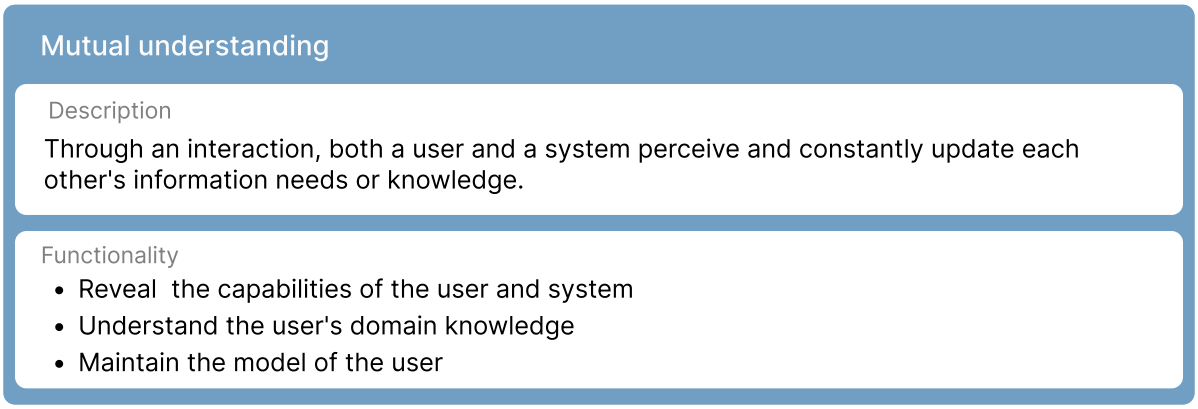

Figure 5.12.: Overview of mutual understanding characteristic

We consider Radlinski's properties [S1] of user and system revealment as part of the mutual understanding we proposed for CS systems (Figure 5.12). The authors described mutual understanding as the system knowing what the user needs and the user knowing the system's knowledge base. [S7] described this ability as grounding, which establishes mutual understandings between conversation participants and involves sharing information, beliefs, and values. Furthermore, discourse management property must ensure that it has understood the user's information requests and back this up with confirmations or prompts to repeat actions [S7]. The misunderstandings can occur during the conversation and, in extreme cases, can lead to a double illusion of transparency where each participant believes that mutual understanding has occurred, however, it has not [S4].

The CS system should help users to formulate their information needs appropriately [S1]. Information need as defined by Belkin [83] was mentioned by several researchers [S4, S31, S10, S7]. [83] defined it as an anomalous state of knowledge, where a user is *"faced with a problem, recognizes that their state of knowledge is inadequate for resolving that problem, and decides that*

obtaining information about the problem area and its circumstances is an appropriate means toward its resolution". Taylor's model, which consists of four levels of information needs: visceral needs, conscious needs, formalized needs, and compromised needs, was referenced by multiple researchers [S12, S27, S29, S10, S7]. The visceral need is an actual but unexpressed, vague type of nonverbal expression. The conscious need is an ambiguous and digressive expression that may eventually evolve into formalized need, a qualified and rational expression of need. Moreover, the compromised need is a short and unspecified question to the search engine. [S29] distinguishes between fact-based and competence-oriented information needs. The former refers to the need to check some facts, e.g., time or quantity. Competency-oriented asks for a description of a process, e.g., how to cook or fix something. [S29] discovered that fact-based needs included more information need-discriminating words, and skill-oriented needs required conversational context.

The system must adapt to changes in users' domain knowledge. In practice, the system can only estimate the user's knowledge and select results based on what the system assumes the user understands [S4]. [S7] claimed that users should feel unrestricted to tell the system that they have identified the information gap before formalizing the request. Users can provide atomic information, "information nuggets," about users' domain knowledge [S4]. Knowing the user's needs and personal data also improves personalization for future conversations and long-term preferences [S1]. However, [S27] asserted that because of the risk of personal data exposure, CS systems must implement secure protocols for information retrieval and data sharing to ensure that user data remains secure.

The CS system should reveal to the user aspects of the available search space and knowledge about the characteristics of the items in the corpus [S1]. While displaying, for instance, partial items selection, the system already demonstrates its capabilities and how it can partition or refine the search space [S1]. For a successful system unveiling, according to [S4], the system should promote and exemplify its capabilities. Also, [S4] stated that disclosing result set meta-information can help searchers get an overview, learn how the system's knowledge is structured, and discover indications for more targeted queries. Besides, the system can also reflect the user's understanding of the input through automatic actions and detect user interaction comprehension level [S17]. [S2] suggested that for systems that allow both speech and text-based interaction, the system should reveal its ability to speak the response aloud when interacting via text. However, the user must decide on switching the primary channel. [S14] described the process by which the user builds a system model during the conversation, i.e., the system reveals its capabilities to the user during the interaction, and the user remembers them for future interactions. For example, after retrieving results, the system asks the searcher if they would like to open a link in a new tab and shows the user that it has that option. In the subsequent interactions, users can automatically indicate in the input request that they want the results to open in new tabs.

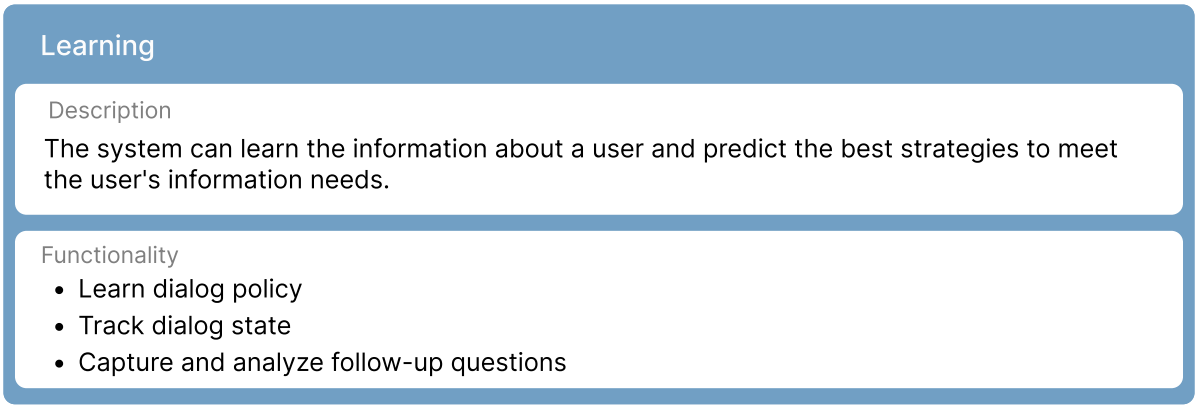

Figure 5.13.: Overview of learning characteristic

Learning

The final characteristic we identify as essential to CS systems is the ability to *learn*, with a summary provided in Figure 5.13. With this capability, a system can train using ML models to extract the relevant information for future purposes, providing long-term benefits at the expense of the current query. Consequently, during an interaction, the system obtains relevant results and improves the user model to enhance personalization [S1]. Dialog policy determines the system's next action in a current state. In addition, the system can learn the situational context and, for example, assess whether an irrelevant conversation started by the user is a suitable opportunity to start a conversation to provide information [S27]. These functionalities generally describe the system's capabilities not only to learn but also to *understand* and *analyze* what also Trippas et al. [S23] described in their work as CS system tasks. Understanding describes inferring meaning from oral, written, and graphic communications by interpreting, illustrating, classifying, summarizing, inferring, comparing, and explaining. While analyzing means decomposing material into its constituent parts and determining how the parts relate to each other and to an overall structure or purpose by differentiating, organizing, and classifying [S23]. In the included studies, researchers often discussed the ability of CS systems to estimate their next strategy, such as asking clarification questions, providing feedback, or deciding to show the result. As a result, we discuss these functionalities in more detail below.

The system must be able to receive feedback from a user, including negative feedback. Multiple user feedback interactions can improve the system's understanding of information needs [S1, S32, S14]. Bi et al. [S38] defined positive feedback as *inclusive* information, meaning that not all positive feedback values should have the same property. For example, telling the system that the user likes a red color may also mean that the user may be happy with pink. In contrast, negative feedback provides *exclusive* information. If the user provides information that she or he dislikes red, the system must exclude all products with this color from the search results. [S38] emphasized the relevance of negative feedback, as it simplifies the collection of detailed information about the irrelevant items for the system. The system narrows down the list of results in a product search scenario by collecting the irrelevant items

and iterating with the user over multiple passes. [S1] proposed a strategy to explicitly reduce the likelihood of unexpected feedback by modeling frequent conversation outcomes.

Next, we report on the CS system's ability to learn how to ask clarification questions. Zamani et al. [S30] defined clarification and preference elicitation as significant and necessary. At the same time, [S6] referred to the ability to ask clarifying questions as one of the most distinguishable and vital functionalities of CS systems. These functionalities increase the user's chance to elicit better results [S19]. [S6] claimed clarifications lead to accurate and higher-quality results because the system better understands the user's information needs. [S20] defined three possible actions a user could take after being asked a clarification question: (1) respond cooperatively, (2) respond lazily (yes or no statements only), or (3) do not respond at all. A poor clarification question or a wrong answer leads to user dissatisfaction or may cause the user to leave the conversation [S19, S18]. [S19] advised against considering a clarifying question as an alternative to answering the user's query by default, as poor clarifications can be even worse than the suboptimal answer. In addition, [S36] discussed the challenge that user feedback is inherently biased. Therefore, the model trained based on only biased feedback may get stuck in a local optimum.

Wang et al. [S18, S19] proposed a risk-aware CS agent that implements strategies for deciding between showing an answer and asking a clarification question based on reinforcement learning. The system's re-ranker rates each round's answer and clarification question. Based on the decision of the re-ranker, the system provides the corresponding output. The system can ask as many clarification questions as possible if they are relevant to the topic. If the users have infinite tolerance, they will answer every question. Otherwise, they may decide to end the conversation. In this case, the authors proposed to set a Mean Reciprocal Rank (MRR) equal to zero if the user leaves the conversation due to a lousy clarification question, penalizing the model.

Salle et al. [S20] suggested a function to identify the desired topic during the result search by asking clarification questions. The system selects the candidate facet $c_i \in C$ and asks the clarification question, "Are you looking for c_i?". The user must answer either "Yes" or "No." If the user confirms, c_i is the best estimate of the user's information need. Otherwise, the agent selects the next candidate facet $c_{i+1} \in C$ and repeats the process. If the user gives an extended answer, e.g., "No, I am looking for...", the system updates the current context and re-ranks the candidate facets, and the process is repeated.

Theoretical models

Finally, we have identified several descriptions of theoretical models for conversational search. One of such was proposed by Radlinski and Craswell [S1]. A user performs a search by looking for an element that may be ambiguous i in a predefined corpus C , i.e., $i \in C$. i represents an information need of the user. During a conversation, the system aims to estimate the utility of such elements, u_i. Based on long-term memory, the system can make a preliminary estimate about i, $\widehat{u_i}$ before the user has provided additional information. In

addition, the system can estimate the uncertainty of the utility, δ_i. Based on the action a taken by the system and the subsequent user's response r, the system predicts the update of the item's utility and uncertainty. The utility of the system's action represents an expected reduction in uncertainty about which items have the highest utility when calculated by summing over all items and possible user responses. In some cases, however, the distribution of user responses is unknown, so the utility must be estimated based on previous observations of the system.

Compared to Radlinski's action space model, Kiesel et al. [S4] proposed a meta-information-centered conversational search model. Meta information can be elicited and used in various functions. For instance, meta-information may help the user to explore the documents from the results set. Instead of directly presenting the results, the system may first provide meta-information on the documents, e.g., by stating, "Found two entities, nine web pages, eight related search queries, ten videos, and ten Wikipedia articles."

Finally, Ferreira et al. [S39] provided a formal description of the CS system based on its task. A conversational search task is defined by a sequence of natural language conversational turns for a topic T and queries q. Each conversation consists of n conversation turns, resulting in $T = q_1, ..., q_n$. The goal of the conversational search task is to find relevant passages p_k for each query q_i, satisfying the user's information need based on the conversational context.

5.2.3. Related concepts

In this section, we report the results of comparisons of conversational search with other systems related to this paradigm. Examples include comparisons with classical IR systems, chatbots, and recommender systems. We also provided an overview of how researchers have classified conversational search in their work.

Classification

In their study, Vakulenko et al. [S3] mentioned the typical categorization of conversational systems into questioning, task-oriented, and social chatbots based on their tasks. [S11] provided a classification scheme by dividing systems into question answering, task-oriented, and chitchats. [S13] referenced dialogue systems classification into task-oriented dialogue systems, conversational agents, and interactive question-answering. [S29] referred to studies that position conversational search in the class of information-seeking dialogs. Also, [S2] defined conversational search as a search performed with a conversational IR system. [S5] categorized conversational search as a particular case of IIR because the dialog is based on conversational turns, and the system supports mixed-initiative interactions. [S30] described conversational search, question answering, conversational recommendation, and joint search and recommendation as separate tasks. Similarly, [S27] described the conversational search as part of conversational information-seeking systems, along with conversational recommendation and question-answering systems (Figure 5.14).

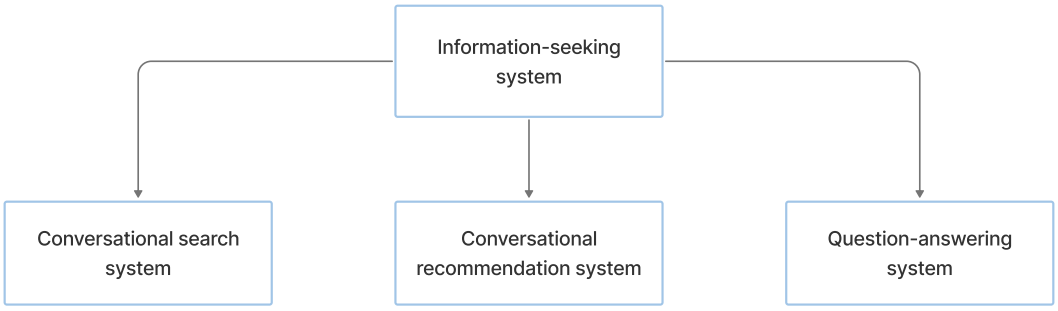

Figure 5.14.: Conversational search system is part of an information-seeking system, as proposed by [S27]

Comparison to classical information retrieval

Kiesel et al. [S4] described the classical IR system as query-centric to understand the information needs of the searcher, while the IIR system is user-centric. In comparison, in conversational search, the authors further equated the roles of the system and the user by defining dynamic multi-turn conversations as a requirement, as in typical conversations between people seeking information. [S14] described CS systems as more complex than classical IR systems and argued that CS systems focus more on the "user's search process as a whole". CS system has an advantage over classical IR by actively asking users questions to clarify their intents [S6]. Compared to classical IR, which retrieves relevant documents based on a self-explanatory query, conversational search enriches queries with contextual information [S33]. It provides more agency to satisfy the user's information need [S5].

The search process with the traditional search engine may require many sessions to complete the task [S1], but the user has complete control over the interaction and decisions [S14]. The CS system, in contrast, essentially fulfills the long-term purpose [S1]. [S2] explained that the current web search engines are not CS systems because they do not respond with natural language, although the user can enter search queries in natural language. On the contrary, [S36] stated that current search engines have improved beyond the "ten blue links" paradigm and now offer more natural language answers, summaries, and knowledge graphs on the SERP. [S10] classified traditional browser-based search as unimodal information search because of both text-based input and output. In contrast, the authors believed CS systems could integrate multi-modal interactions and provide ubiquitous search interactions compared to traditional desktop search.

Comparison to chatbots and commercial agents

Ferreira et al. [S39] emphasized that a different type of dialogue has evolved that aims to satisfy the user's information needs and where co-referencing, ambiguity, and subtopic switching may also occur, compared to chitchat and task-oriented dialogs. [S3] stated that the primary goal of social chatbots is to entertain the user. However, spoken conversation systems such as Google Home or Apple Homepod are not yet capable of performing complex information search tasks, i.e., supporting multiple turns, reformulating queries, or proactively recommending various search strategies [S7]. [S1] argued that open-domain agents rely on a traditional search results page for IR in many scenarios. Finally, [S11] highlighted research that showed that chat systems tend to seize control of the interaction and ask too many questions while ignoring the user's initiative. [S9] used a questionnaire to compare website SUIs and chatbots and found that participants could hardly define any added value that the chatbot could bring to their search process. One of the participants even stated that "a chatbot can add value where social interaction with a human need to be replaced...and search is not a social interaction."

Comparison to conversational recommendation

Vakulenko et al. [S3] noted that conversational search is similar to conversational recommendation and that the boundaries between these tasks are not explicitly defined in research. [S6] also observed the technical similarities between conversational search and recommendation in the e-commerce domain.

5.3. Research question 2

We defined the second research question as follows:

RQ2: What application scenarios have been investigated for conversational search systems and why?

To answer this research question, we look at the modalities, tasks, and scenarios that researchers from the included studies have proposed. First, we report modalities that are valuable to the conversational search paradigm. Next, we describe application scenarios, including specific tasks, that invite or do not invite conversational search, and explain why. Finally, we demonstrate several domains that the authors have explored in the context of conversational search.

5.3.1. Modality

In this part, we reported the results on different modalities for conversational search systems. We reported separately on different modalities researchers proposed, such as uni-modal (text-based or speech-based) or multi-modal systems. We also discussed the modality for CS systems from different perspectives, such as interaction channel or presentation mode (Figure 5.15). As interaction channels, we observed text-, speech-based or hybrid systems. We observed two hybrid forms - a combination of speech and text and a combination of speech and gestures. Presentation mode was described as either verbal or non-verbal. Table 5.3 shows publications that describe or build upon a particular modality or presentation mode.

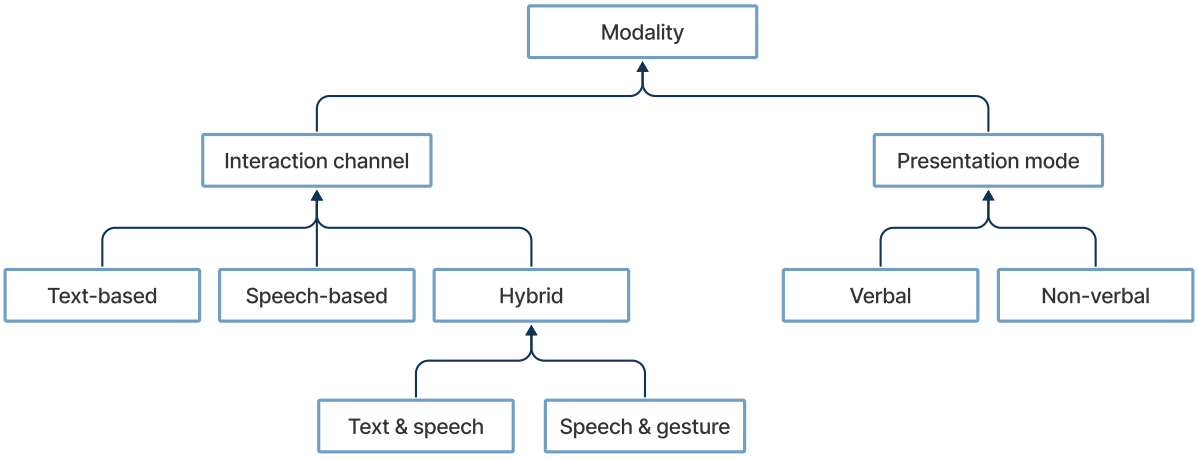

Figure 5.15.: Observed modalities in the included studies

Liao et al. [S43] observed several trends for conversational search interaction: shifting from textual modality to cross-modality, supporting multiple domains, and focusing on search and recommendation scenarios for task-based dialog systems. [S5] claimed that the CS modality and interface affect the user's gain from the conversation. [S5] discussed the CS system can be from the interaction and presentation perspectives: (1) speech-only CS system often via VUI,

Modality	
Interaction channel	
Text-based	[S40], [S9], [S41], [S5], [S32]
Speech-based	[S17], [S14], [S23], [S8]
Hybrid	[S42], [S4], [S6], [S2], [S1], [S21], [S24]
Presentation mode	
Verbal	[S40], [S9], [S41], [S5], [S32], [S17], [S14], [S23], [S8], [S6], [S2], [S1], [S21]
Non-verbal	[S42], [S4], [S24]

Table 5.3.: Observed interaction channels and presentation modes in the included studies.

(2) chat-based CS system via, e.g., Slack or Telegram, (3) augmented search engine interface, or (4) multi-modal virtual assistant.

Xing et al. [S8] examined age-related differences in conversational search behavior and found that interaction modality influences how older adults allocate their attention and time to search. For example, the authors observed that older adults rephrased fewer queries and spent more time examining results in text-based interactions than in voice-based interactions. [S2] observed general differences in user behavior between text-based and voice-based CS systems.

Uni-modality

Systems with only one communication channel, e.g., voice or text, are called **uni-modal** [S10]. [S9] proposed a text-based conversational search interface and referred to it as a chatbot. The authors argued that chatbot interfaces are convenient for searching web pages and the conversational modality helps to search the database and present the results in a chat window. However, they also pointed out the difficulties when navigating the search results. [S5] also proposed a chat-based CS system. [S21] proposed a conversational search system with an enhanced standard graphical search interface. The user communicates with the proposed system through a chat box. [S2] observed that users prefer text input over speech input if they would disturb someone. However, if the user makes a text input, a system usually will respond in text form [S2]. In addition, the text output must not simply be a transcription of the spoken output and vice versa [S2, S23].

Trippas et al. [S23, S7] and Jung et al. [S28] discussed a purely speech-based search approach. [S14] focused on a purely speech-based approach that excludes multi-modal or visual interactions and argued that spelling is an important feature because typing is impossible for users in some scenarios. [S8] pointed out that speech-based interaction eliminates the need to acquire experience and skills in technology. [S14] stated that the CS system should become more active and participate more in the search process to overcome the difficulties of speech-only modality. [S4] discussed that voice-only interaction is a viable

approach for multi-participant conversations, i.e., when multiple users interact with a system and use the same CUI. However, in this case, the CS system must recognize and separate users based on their voices. [S14] suggested that multi-document summarization could benefit speech-only conversations. Additionally, [S10] emphasized that presenting results in a speech-only manner can overwhelm users.

Interestingly, [S2] noted that speech-based CS systems should be purely speech-based only when their interaction channel has no screen. Also, Trippas et al. [S23, S7] discussed screenless speech-based conversational interaction with the search system. [S1] discussed the importance of speech-based CS systems with small or no screens to adapt the result representation from a large to one or a small number of responses. Modern devices already present this practice, so concise, conversational responses are appropriate. However, in their analysis, [S9] argued that people might have difficulty getting used to conversational search because most search interfaces are currently GUI-based.

Multi-modality

Liao et al. [S43] stated conversational search paradigm has been evolving from unidirectional and text-based modality to **multi-modality**. [S24] defined simultaneous combination of two modes of intent expression as multi-modal intent. Deldjoo et al. [S10] defined three dimensions for multi-modal CIS: (1) processing modality in conversation C, (2) multi-modality in user-system interaction I, and (3) multi-modality in processing and accessing information items D. Therefore, the multi-modality is defined as $MMCIS = C + I + D$.

In their research, Deldjoo et al. [S10] further discussed two views of multi-modal interaction: (1) the human-centered view and (2) the systems-oriented view. The modality of the human-centered view refers to the five primary human senses: sight, hearing, touch, taste, and smell. Consequently, human-centered multi-modal interaction refers to the human's ability to receive, process, and transmit information. In contrast, systems-centered interaction can accept many different inputs in combinations. In addition, the authors modeled a conceptual design for multi-modal systems, depicted in Figure 5.16.

Moreover, [S10] suggested analyzing multi-modality in terms of communication/interaction channel, processing modalities, presentation mode, or a combination.

An **interaction channel** is a bidirectional path between a user and a system with an input (for the system) and an output (for a human). *Input channels* represent pointing, touch, speech, or gestures, while *output channels* are based on audio output or visual text on the screen or other options.

A **processing modality** describes the system's processes and the data representation of information elements. [S10] suggested the following processing modalities: *visual, text, audio, touch, and others*.

A **presentation mode** describes a presented stimulus to a human in either *verbal* or *non-verbal* form. Verbal communication consists of sending and receiving messages through the

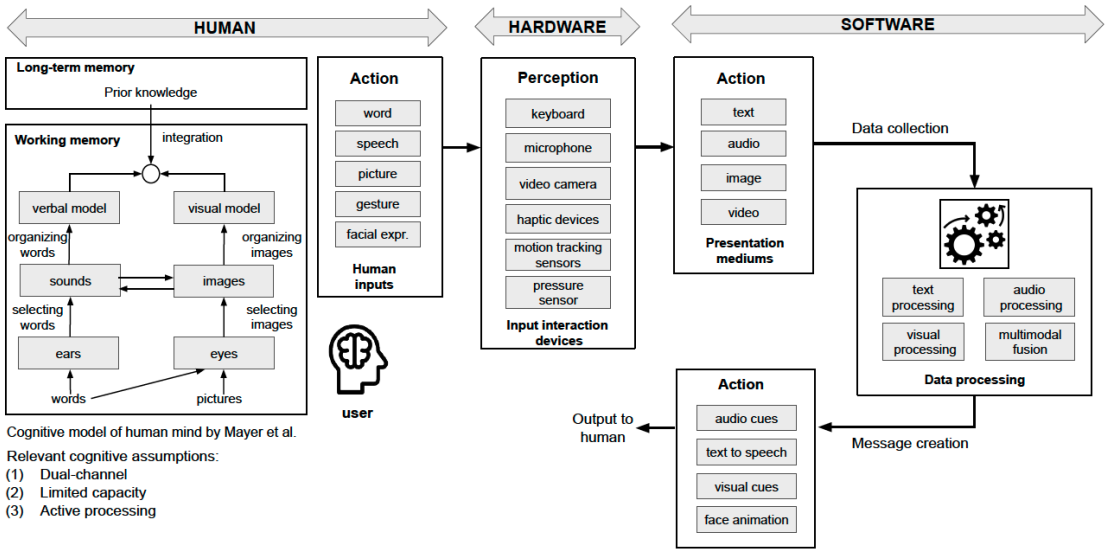

Figure 5.16.: A conceptual design of a multi-modal system and its elements [S10]

use of words only. In contrast, non-verbal communication provides wordless communication using, e.g., body language, gesture, posture, or facial expressions. [S10] stated that multi-modality in presentation mode, i.e., using both verbal and nonverbal channels, improves the quality of multimedia learning.

Several authors [S1] proposed integrating both written and spoken forms for conversational search systems. In addition to a text query, a user can drag an image into an input box and extend the query. Furthermore, [S24] focused their research on multi-modal conversational search systems and discussed dynamic contextual adaptation of speech recognition and understanding models using visual context. In their research, users interacted with gestures and speech through the browser, web page interfaces, and page elements such as links, dropdown menus, and forms as an interaction channel. [S2] proposed users can switch between communication channels, e.g., continue to interact by voice even though a system has a screen, speak and read a screen simultaneously, or refer to a screen from time to time. [S30] stated CS system response messages could be multi-modal and consist of text, speech, link, and options list. [S43] pointed out the popularity of combining visual modality and speech interaction, e.g., image captioning or visual question answering (VQA). [S30] developed a multi-modal conversational search system called Macaw that supports various interfaces such as command line interface, mobile, desktop, and web applications. Also, [S10] pointed out that changing the modality of the retrieved or generated response should be one of the functions of the multi-modal system. Modality change can be done, for example, when converting text to speech or generating text from pictures and diagrams and vice versa. However, it is also important to consider which modalities are available in a multi-modal system or desirable by a user at any given moment in a conversation [S4].

Heck et al. [S24] suggested going beyond speech-only settings and using visual constraints such as gaze, touch, and gesture. The authors argued that pointing gestures could constrain the focus of attention to a subset of the visual presentation. An example would be selecting an object, pointing to it, and saying "that one" simultaneously. In their study, the authors concluded that people could make precise gestures in the direction of the selected item. Although they did so for only 30% of the total interaction time, other, more consistent modalities (e.g., gaze) could be utilized to improve meaning over speech alone.

Bickmore et al. [S42] proposed a combined modality for an embodied conversational agent with non-verbal behavior recognition. [S4] suggested adding hand gestures or voice tone to the text and verbal forms of interaction. The authors argued that detecting a positive or negative tone of voice improves query interpretation and thus retrieval. In addition, [S4] suggested integrating additional meta-information as user's facial expressions so that a system can recognize, for instance, that a conversation is about to break down and clarification is needed. [S10] suggested CS modalities should go beyond spoken language as users can express information needs in more than keywords. Despite the interest in using nonverbal communication in CS systems, [S24] have observed that when hand signals are captured, false gestures can also be captured, such as when users raise their arm to reach something on the table.

5.3.2. Application scenarios

Radlinski et al. [S1] discussed the complexity of CS tasks and argued that there might be scenarios that do not have a fixed schema or where the given data is not structured as a database. Hence, the slot filling approach is not appropriate in such cases. Instead, tasks involving a free-form conversation with multiple turns, clarifications, reasoning, and feedback may be applicable for conversational search. Ren et al. [S16] argued that conversational search is more effective when users have complex and compound information needs or when users' information needs are unclear or exploratory. Hence, there are application scenarios or concrete tasks that are not suitable for conversational search. We first describe suitable scenarios for conversational search in detail and shortly report at the end which scenarios have been suggested as less suitable.

Radlinski et al. [S1] proposed several scenarios for conversational search: basic information retrieval and personal information search. Deldjoo et al. [S10] identified three example scenarios for conversational search in a multi-modal context: classic information retrieval, on-the-go and longitudinal information seeking, and multi-party or collaborative search. Liu et al. [S40] characterized the usage of the conversational search paradigm in search scenarios like web search, product search, and academic search. Based on the scenarios described, we proposed four different suitable scenario types, depending on their focus: (1) complexity-based, (2) task-based, (3) context-based, and (4) domain-based. Figure 5.17 illustrates the categorization in detail. We found that complexity-based scenarios belong to the focus of CS systems, task-based and context-based to the focus of users, and domain-based describe the focus of industry and research.

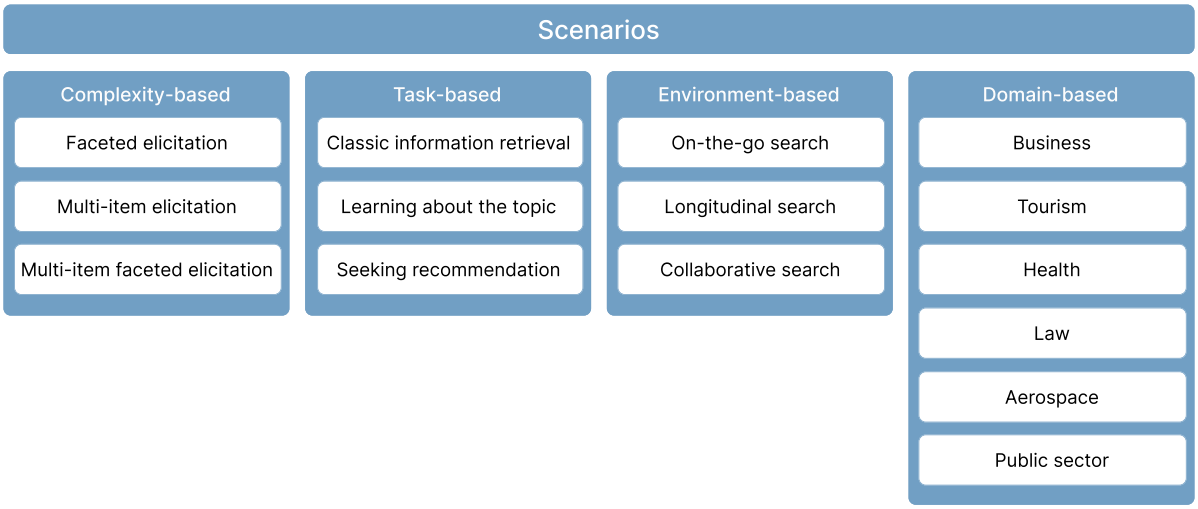

Figure 5.17.: Proposed scenarios and their categorization

Complexity-based scenarios

The first group of scenarios we identified is complexity-based scenarios. They focus more on the system's perception of the complexity of a search task. Radlinski et al. [S1] suggested three complex scenarios for which CS might be appropriate: (1) faceted elicitation, (2) multi-item elicitation, and (3) multi-item faceted elicitation. The scenarios are listed from least to the most complex.

Faceted elicitation. This scenario describes a search task for an item with complex individual attributes provided piecewise to a system, as users may identify relevant aspects first during a conversation. An example request for faceted elicitation is: *"I am looking for an email that contains a link to a research paper that I got from a student who emailed me right after SIGIR last year. I cannot remember the student's name, but I had never heard from her before."* However, [S6] pointed out that faceted search usually requires structured knowledge.

Multi-item elicitation. This scenario describes a search task for a single item depending on a set of other items that must also be found first. An example of such a query: *"I am looking for a photo Alice took of me right after I took her picture a few months ago."*

Multi-item faceted elicitation. The last scenario describes a combination of earlier scenarios where a system retrieves a set of items. In this scenario, a system must estimate the utility of the individual items and the combined utility to obtain an overall score for the entire set. An example would be a vacation planning set consisting of hotels, travel arrangements, restaurant plans, and excursion destinations. A system elicits user preferences on various subtopics, combines the utilities, and returns a relevant set of elements for vacation planning.

Task-based scenarios

The next group of scenarios we identified is task-based scenarios. They describe scenarios of a task that the user wants to accomplish with the help of the CS system, so this scenario type is focused on the user.

Classic information retrieval. Fergencs et al. [S9] defined the main task of the conversational search concept as a chatbot that helps search and retrieve web content. However, this scenario consists not only of a single short query but of a multi-turn exchange with memory and context maintenance to refer to past steps when needed, and finally provide results to the input query based on some topic [S1]. However some authors criticized query-based inputs. [S16, S6] stated that traditional keyword-based search is unsuitable for conversational search. [S1] found that a scenario with the standard search setting of a single query and the user's expectation of receiving relevant results in one turn is also not suitable for conversational search. Classic information retrieval should have moderate or advanced complexity rather than just one keyword query. A CS system can proactively convert to narrow or specify the search and use multiple modalities to present the results [S10]. [S13] separated search tasks according to their complexity. Complex tasks required a high level of problem understanding and cognitive thinking to answer questions, whereas simple tasks generally involved fact-checking. Another example is a search over a user's personal information. This task considers a search over heterogeneous with rich metadata personal user information. A user requires sophisticated assistance from a system [S10] during this task.

Learning about the topic. In this scenario, the user is usually curious to learn about a particular topic. Such an interaction can be initiated by both a system and a user and has an undirected, open-ended flow. The system's goal is not only to answer questions about a topic the user knows little about but also to provide information that stimulates meaningful follow-up questions [S44]. The example illustrates such a conversation. In addition, [S9] described the "library chatbot" - a chatbot that helps in cases where information needs are challenging to articulate, allowing users to define complex search queries. Finally, [S28] discussed conversational search in a news scenario and found that this case increased user curiosity. Thus, users went from simply listening to actively searching for information to understand the news.

Seeking recommendation To some extent, search, and recommendation tasks are similar. In this scenario, users are engaged in the process of receiving a recommendation or advice while expanding their knowledge about a specific recommendation topic [S6]. A user can initiate the scenario, but usually, the system guides the user and takes the initiative to recommend the best options based on their information needs [S6].

Environment-based scenarios

The third user-oriented group of scenarios we identified is context-based scenarios. They ensure that the CS system adapts to the user's current context and situation, e.g. when the user is in a hurry and pressed for time, when the user is on the go, or when the user wants to

search for information with friends. Therefore, based on the results of the included studies, we identified three scenarios: on-the-go search, longitudinal search, and collaborative search.

On-the-go search. Multi-modal integrations can improve a system's assistance during a user's search by utilizing, for example, different sensors or GPS. [S10] provided the following example: the MMCIS system follows a user throughout the day and knows the user's plans from presenting a budget at work, taking on a new client, picking up the kids after school, and cooking dinner. The user is riding her bike down the road and sees a plant on the side of the road. She quickly stops, takes a picture, and continues her way. Meanwhile, she asks her earphones what kind of plant it is and if it is edible. The MMCIS system considers the user's GPS location, the photo, and the input query. The system measures heart rate and recognizes that the user is cycling and therefore does not provide visual information but presents the result via speech into the earphones. Next, the user can ask for information about working on the budget presentation. The system adapts to the user's information needs, assesses which modalities are most appropriate, and offers an appropriate conversation.

Longitudinal search. During such a scenario, the user may make numerous requests, possibly having limited access to the device (e.g., while cycling and making requests by voice) [S10]. Therefore, particular information needs may not be met immediately but may be queued by a CS system, allowing the user to gather and retrieve information over an extended period.

Collaborative search. Collaborative search enables synchronous and asynchronous search with multiple participants simultaneously and continuous modality updates during the conversation. Such curated conversation can reduce the participants' cognitive load in information search [S10].

Domain-based scenarios

We observed seven separate domains described in the included studies and summarized them in Table 5.4. Below, we describe each domain in detail.

Application domain	Mentions
Business	[S17], [S38], [S6], [S45]
Tourism	[S43]
Health	[S9], [S42]
Law	[S40]
Aerospace	[S13]
Public sector	[S41]
Miscellaneous	[S29], [S43]

Table 5.4.: Observed domains in included studies.

Business. Zhang et al. [S6] studied product search and recommendation and encountered that these two tasks are technically very similar in the e-commerce domain. The authors also explained that personalization is essential for product search because users may have different preferences for the same product and prefer different items even when having the same conversation. [S38] proposed a conversational paradigm for product search. The method retrieves one result suggestion but can also be extended to multi-item retrieval. The authors emphasized the importance of eliciting user preferences through multiple rounds of conversations. For example, when a user needs a new mobile phone, a CS system asks for their preferences regarding certain features such as screen size, brand, and others. [S1] suggested product recommendation task for conversational search would be more appropriate for a complex scenario where the goal is to obtain recommendations based on the user's vague information needs, for example, because of the user's unfamiliarity with a topic. [S4] mentioned complex recommendations, such as in a multi-participant context, retrieving results based on speaker recognition and after querying the user's personal information, as a valuable task for conversational search. Finally, [S45] proposed a CS system for the e-commerce scenario, specifically for online shopping. The researchers argued that it is tempting for users to navigate through products while conversing with the virtual shopping assistant, as in traditional in-store shopping.

Tourism. Shiga et al. [S12] described travel planning as a "highly information-intensive task with many decisions to make based on outcomes of searching". [S1] described travel planning as a concrete example task for multi-item faceted elicitation, which involves estimating the combined utility of heterogeneous elements that have their utility value estimated by a user. [S43] focused on several tasks suitable for conversational search in travel scenarios: venue search based on an image, concept recognition based on an image, venue search based on preferences, cross-domain venue recommendation, subsequent venue replacement, venue comparison, and search for a specific store in a shopping mall. Venues can be from one of the following domains: food, hotel, nightlife, shopping mall, or sightseeing. However, in examining selected corpora, the authors found that none of them provided a comprehensive basis for studying various multi-modal search tasks in conversations. Likewise, [S12] described tasks for a CS system in a travel planning scenario: finding exciting places to visit, comparing flight schedules, and deciding about a hotel.

Health. Traditional web-based search engines may be unusable for individuals with low health literacy. Thus [S42] developed a conversational search engine that enables people with limited health and computer skills to find and learn about clinical trials on the Internet. The search tasks were based on the clinical trial search process, i.e., finding one or more groups of cancer-related trials for which the user is eligible. Considering that users have limited health and literacy, the CS system guides the user in the search and displays the information about the studies step by step, showing only the details that the user needs for the evaluation at each step. The authors defined this task as "multifaceted" because users are never prompted to retrieve and enter text, but the CS system always assists them with various input options. The authors' system provided access to at least one-third of the users, and all authors were more satisfied with the conversational approach than the traditional one. During the research

experiment, users in the [S8] study performed a complex search task on depression and antidepressant medication. The authors argued whether conversational search tasks could influence older adults' search behavior and performance. The authors could not provide a detailed answer to this issue because of the limited quality and variety of search tasks.

Law. [S40] proposed to adopt conversational search in the legal case retrieval scenario. In this case, the CS paradigm improves search accuracy, and users save the effort of formulating complex queries and examining the results. A conversational query for legal case retrieval consists of the following steps: (1) a user asks a legal issue question in natural language, (2) a system clarifies the user's information need, (3) the system submits the search queries to the legal case query system and displays the selected SERP results to the user, and (4) the user reviews the results and either terminates the search based on success or failure or repeats the process by updating the query. [S40] observed that people examined cases more carefully and patiently in the conversational search setting. Finally, the authors showed that using CS search was more successful than the traditional approach, especially when users did not have sufficient knowledge.

Aerospace. The aerospace industry relies on extensive collections of documents with system descriptions, manuals, or procedures [S13]. Therefore [S13] assessed the usefulness of conversational search for cockpit documentation. The authors conducted a user experiment to determine the relationship between search tasks in typical flight operations scenarios and the system's perceived usefulness in task performance. Consequently, the authors concluded that when the goal is to improve user search performance, perceived system usefulness and system responses relevance are better predictors than user satisfaction with the system.

Public sector. [S41] developed a system to improve government transparency and citizen participation. Specifically, the goal was to create a system that allows non-expert people to find and explore open government data through conversation.

Miscellaneous. [S29] proposed conversational search in a home cooking scenario, describing the task as highly procedural, goal-oriented, and "involves completing a sequence of individual steps to achieve a larger goal." The authors compared cooking to tasks requiring sequential processes, such as maintaining bicycles, fixing a blown tire, or assembling flat-pack furniture.

Remaining scenarios

Finally, we identified a few scenarios and tasks that we could not assign to a specific category. [S27] described possible tasks of CIS systems with a broader impact than just information retrieval. For example, a CIS system can warn users of potential harm and danger to the user's health and safety. In addition, the CIS system can inform users of potential misinformation or abusive content. With the help of augmented reality devices, CIS systems can allow users to experience a virtual environment while guiding them in their information search in parallel. Finally, [S10] were also interested in research about integrating augmented reality or eye-tracking as part of multi-modal interaction in conversational search.

5.4. Research questions 3

We defined the third and fourth research questions as follows:

RQ3: What architectures have been proposed for conversational search systems?

So far, we have only discussed theoretical models of CS systems that described the desired characteristics but no concrete proposal for their technical implementation. Therefore, now we investigate proposed architectures for CS systems and how the researchers developed them. First, we describe and depict different architectures proposed for CS and related systems. Based on these findings, we propose a reference architecture for CS systems. After, we show in detail how our high-level architecture evolved from the findings and what specific technical aspects the authors of the included studies proposed under individual reference architecture components.

5.4.1. Architectures

Eighteen discrete architectures were proposed by the authors of the included studies, all of which had modularized architectures, a system architecture divided into smaller, manageable modules. Modularized systems have proven to be stable and provide the flexibility to design each module independently. Several from the proposed had a pipeline structure [S35, S41, S47, S48, S34, S30, S31, S46, S33] and a few had a end-to-end neural architecture [S16, S39, S45]. We have covered in detail some architectures in order of complexity proposed between 2020 and 2021, starting with 2020. We have briefly presented the rest.

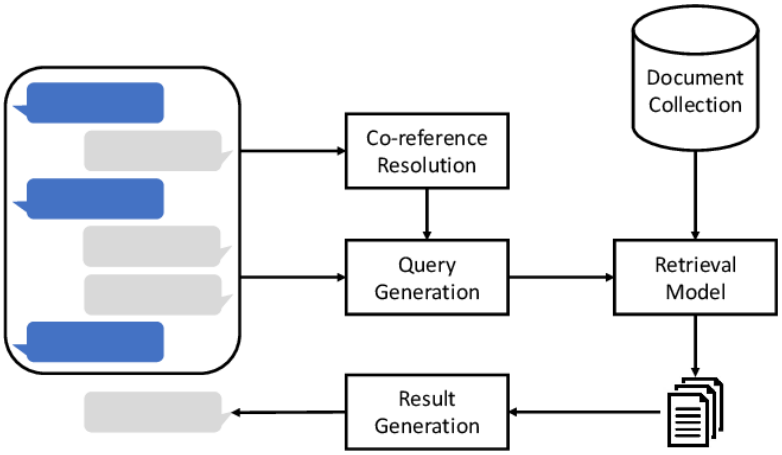

Figure 5.18.: An overview of retrieval and question answering in Macaw [S30]

Zamani et al. [S30] proposed an open-source framework for CIS, called Macaw, based on the Model-View-Controller architecture. Macaw supports multi-modal interaction with text, speech, image, or click input messages. Output messages can be text, speech, link, or a list of options. Macaw has been implemented in Python, and for the ML models, authors

used PyTorch[1], Scikit-learn[2], and TensorFlow[3]. Macaw is based on four components: (1) **co-reference resolution**, (2) **query generation**, (3) **retrieval model**, and lastly, (4) **result generation** component. Figure 5.18 depicts the proposed components. Macaw has a memory and stores all interactions in an "interaction database". Macaw looks for the last interactions between the user and the system, both the user's responses and the system's responses, to create a conversation list that contains a list of messages from the interactions. However, Macaw does not support the generation of clarification questions because the authors were unaware of a published solution for generating clarification questions based on public resources.

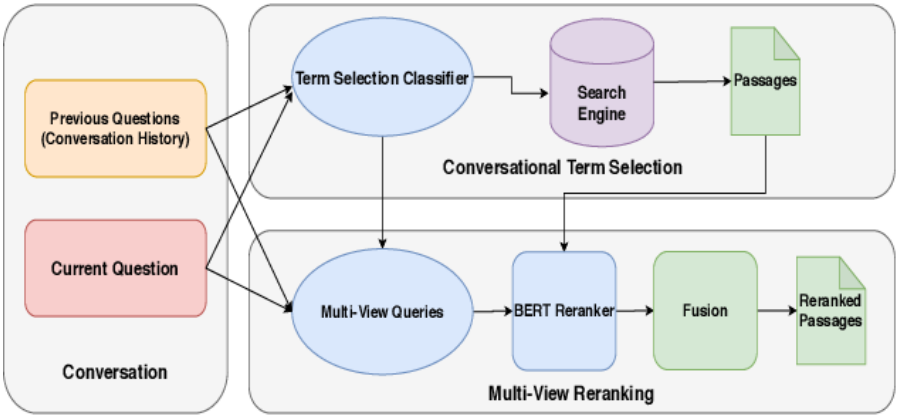

Figure 5.19.: Pipeline architecture proposed by [S34]

Further, [S34] proposed a modular pipeline architecture for effective passage retrieval for conversational search. The pipeline consists of two major components: (1) **Conversational Term Selection (CTS)** and (2) **Multi-View Reranking (MVR)**, depicted in Figure 5.19. CTS handles the first-stage passage retrieval. MVR undertakes the re-ranking process, i.e., it re-ranks received from the CTS passages. More precisely, the search engine looks for passages based on the selected terms and the entered question. The fusion step aims to merge the rankings created from the previous components, which is done by simply aggregating the scores created for a passage. The remaining components of the system are described later in the section.

Zhang et al. [S46] developed a framework for conversational search, Chatty Goose, via a standard multi-stage ranking pipeline. For their development, the authors integrated open-source libraries such as Transformers from HuggingFace[4] and ParlAI from Facebook [84], which enabled direct interaction with users through an existing Facebook Messenger

[1] https://pytorch.org/
[2] https://scikit-learn.org/
[3] http://tensorflow.org/
[4] https://huggingface.co/castorini/t5-base-canard

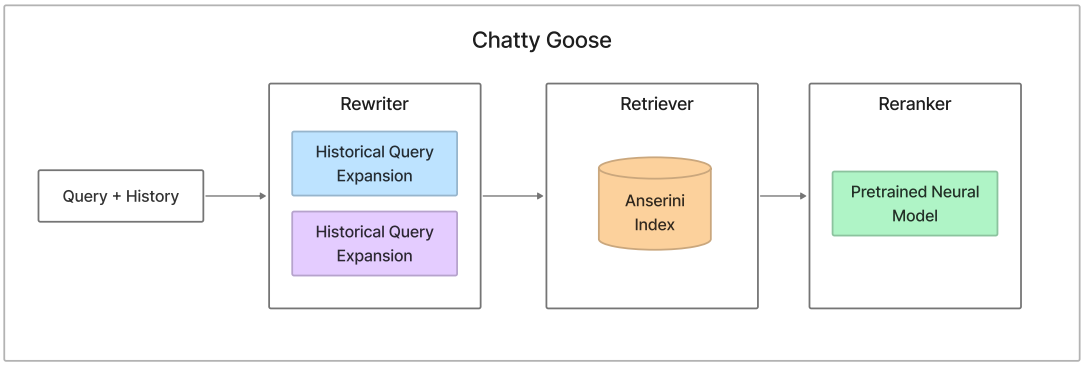

Figure 5.20.: Architecture of the conversational passage retrieval pipeline for Chatty Goose [S46]

platform and self-developed components, including Pyserini IR toolkit [85] and PyGaggle[5]. Their passage retrieval pipeline consists of three blocks: (1) **rewriter**, which converts natural language queries into self-explanatory queries; (2) **retriever**, which retrieves candidate passages from a collection of documents; and (3) **re-ranker**, which re-orders the output of the retriever to produce better results for the document collection. Figure 5.20 depicts the Chatty Goose architecture. These three blocks are similar to the three-stage pipeline architecture of CS systems proposed by [S33]: (1) **utterance understanding**, (2) **first-stage top retrieval**, and (3) **second-stage neural re-ranker**. For re-ranking, [S33] used the 2019 TREC Conversational Assistant Track (CAsT) framework.

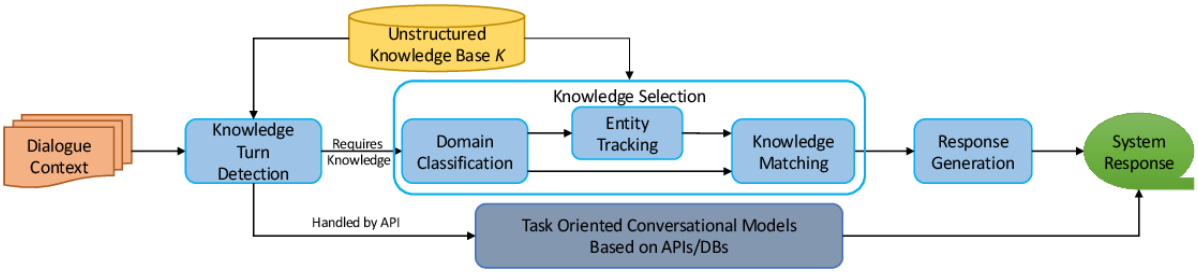

Figure 5.21.: Architecture of the knowledge-grounded dialog system proposed by [S47]

In their work, [S47] focused on responding to user needs beyond API coverage by incorporating external, unstructured knowledge sources. By API coverage, the authors meant the limited domain API coverage for task-oriented conversational systems. Hence, [S47] proposed a pipeline architecture for a task-oriented knowledge-based dialog system with unstructured knowledge. The system consisted of three components: (1) **knowledge-seeking**

[5] `https://pypi.org/project/pygaggle`

turn detection, (2) **knowledge selection**, and (3) **response generation**, depicted in Figure 5.21. The first component identifies user requests that are outside the API coverage. In the second component, knowledge selection finds the most appropriate knowledge that answers user queries based on the identified queries from the previous component. Finally, the response generation generates a response based on a dialog history and the retrieved knowledge. The authors, like [S48], used data augmentation methods for the first two steps and showed that using information from the dialog context improves knowledge selection and end-to-end performance.

Moreover, we have identified a few modularized end-to-end neural architectures. The design of each component in modularized form still requires the creation of rules and labels with domain-specific expert knowledge. This makes it very difficult to adapt these modules to new domains. Neural modules, on the other hand, learn independently and are then combined into a pipeline architecture for an end-to-end system, providing flexibility and good quality. [S26] concluded that pre-trained transformer variants are currently the best performing models on the task of conversational interaction.

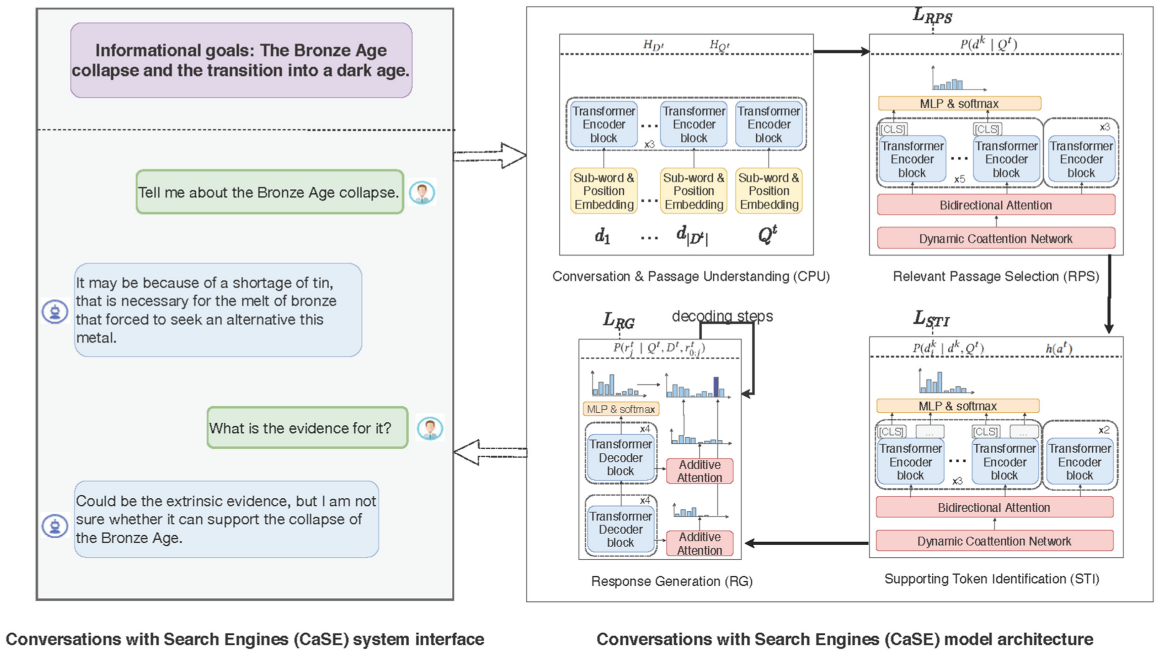

Figure 5.22.: An overview of Conversations with Search Engines (CaSE) proposed by [S31]

Ren et al. [S31] developed a transformer-based pipeline for conversations with search engines, Conversations with Search Engines Engines (CaSE), based on their Search as a Conversation (SaaC) dataset. The CaSE pipeline combines four subtasks: (1) **conversation and passage understanding (CPU)** , (2) **relevant passage selection (RPS)**, (3) **supporting token identification (STI)**, and (4) **response generation (RG)**. The complete architecture is shown in Figure 5.22. The CPU module aims at understanding and encoding conversations

and passages. The authors used a transformer model for CPU that relies on self-attention to extract meaningful information for representing conversations and passages. After, the RPS module selects the relevant passages by estimating the relevance probability for each passage in the candidate pool based on the query and passage representations from the previous module. Next, STI identifies the supporting tokens that contribute to the final answer. Hence, STI estimates the probability that each passage is a supporting token. As a result, the RG module generates the response token by token based on the results of the previous three modules. The authors described SaaC as suitable for search engine conversations because (1) there are multi-turn conversations, which require modeling context from historical turns, and (2) the responses are more abstract and conversational, which is closer to real conversation scenarios.

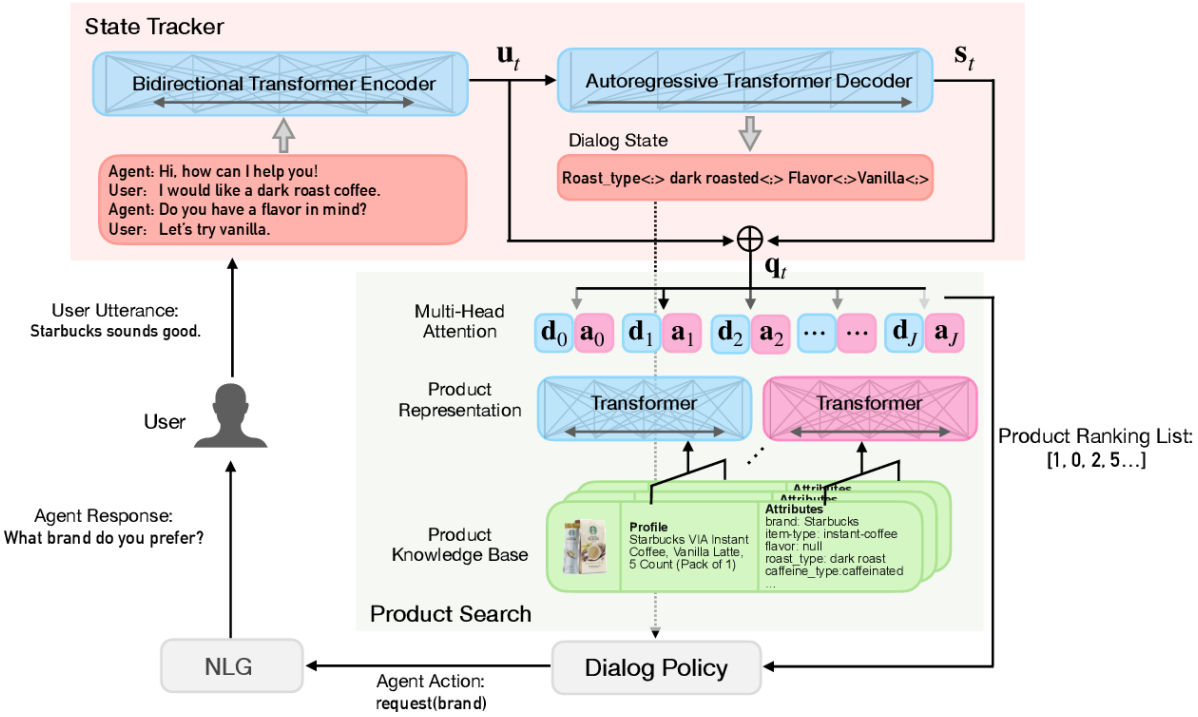

Figure 5.23.: Conversational end-to-end search system proposed by [S45]

Figure 5.23 depicts ConvSearch, an end-to-end conversational online shopping search system proposed by Xiao et al. [S45]. The architecture consists of four modules: (1) **State Tracker**, (2) **Product Search**, (3) **Dialog Policy**, and (4) **Natural Language Generation**. The State Tracker module interprets the dialog's content and outputs the user's intent along with the product attributes in which the user is interested. The Product Search module outputs a list of products that match the user's desired attributes. Based on the output from State Tracker, the dialog policy processes the agent's response according to the user's intent and the result of the candidate search. The NLG module converts the response into natural language, which is presented to the user. This research addresses two significant challenges in conversational search in online shopping: the imperfect entity attributes with

multi-turn utterances in conversations and the lack of in-domain annotations for training due to long-tail entities. ConvSearch addressed the first challenge and the proposed dialog generation method, M2M-UT, addressed the second problem by (1) generating utterances from existing dialogs of similar domains and (2) creating dialog outlines from e-commerce search behavior data.

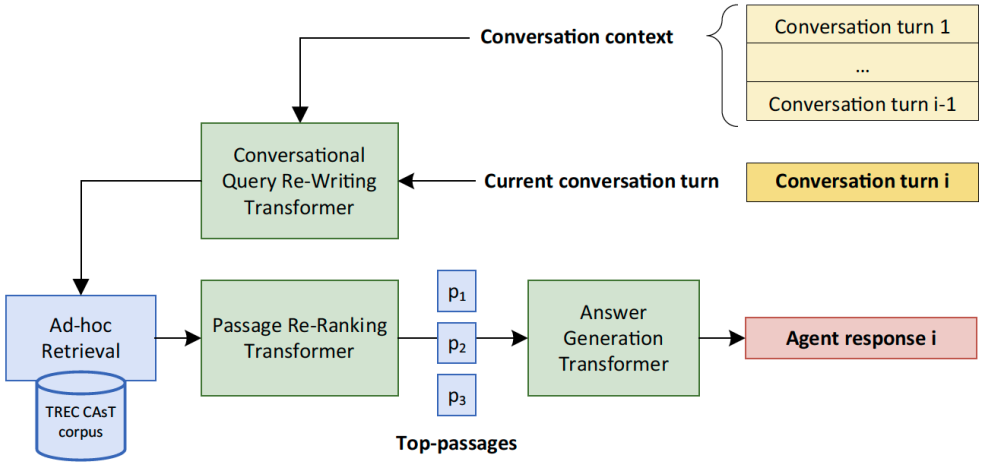

Figure 5.24.: Transformer-based conversational search system proposed by [S39]

[S39] proposed a transformer-based conversational search assistant that uses a four-stage architecture: (1) **context tracking**, (2) **retrieval**, (3) **re-ranking**, and (4) **answer generation**. The conversational query rewriting task is to perform co-reference resolution and incorporate context from previous rounds. The authors integrated T5, which they fine-tuned to reformulate conversational queries, by providing the sequence of queries and passages as input and the rewritten query as output. The authors created the input sequence at each query-passage pair instead of each utterance to delimiter each turn. For the first retrieval stage, [S39] retrieve the top-n passages and send them to the re-ranking model. In this stage, the BM25 term matching model, the language model with Dirichlet (LMD) and the Jelinek-Mercer smoothing (LMJM) [86] are used to retrieve a small set of passages from millions of available. In addition, the passage re-ranking transformer uses the pre-trained neural language model BERT to obtain contextual embeddings for a sentence and each of its tokens. These embeddings are used as input to perform re-ranking. Finally, the abstract search-response transformer creates a response based on the candidate passages. The transformer model generates a natural language answer by summarizing the passages using text-to-text approaches trained on large and comprehensive collections. Such approaches are very effective nowadays and can even understand different topics. For the T5 text-to-text transformer, pre-training is performed using supervised and self-supervised training.

Subsequent studies have also proposed architectures for CS systems [S33, S38, S48, S18, S24, S21, S41, S35, S49, S16, S19]. For example, Wang and Ai [S18] proposed a risk-aware, context-sensitive paradigm for retrieving responses. The authors proposed the **risk decision**

module component, which is unique compared to other components. It aims to balance the risk of insufficient clarification questions and immature results. Moreover, Wang and Ai [S19] revised this model in their follow-up research by integrating more datasets, extending with Ubuntu Dialog Corpus and OpenDialKG, and creating more user settings with different patient levels for clarification questions. [S38] proposed an aspect-value likelihood model to incorporate positive and negative feedback for the conversational product search. [S49] presented a conversational movie recommendation system, Vote Goat, developed using Google's DialogFlow framework. In addition, [S35] presented a spoken question answering system that can answer questions about common topics in French.

A relatively early study in 2013 [S24] explored two components for conversational search (1) dynamic contextual adaptation of speech recognition and understanding models using visual context, and (2) fusion of user speech and gesture input to understand the user's intentions and associated arguments. Also, [S48] described three steps for information-seeking dialog systems, already described in other architectures: (1) **user utterances understanding**, (2) **relevant knowledge retrieval**, and (3) **agent responses generation**. [S21] developed a conversational search application that converses with users to support their search activities. The system's workflow was divided into **conversation management** and **search management**, while the first component integrated the RASA toolkit[6]. [S41] also integrated the RASA toolkit for their architecture. They developed a pipeline architecture consisting of **message interpretation** and **dialog management** modules for the proposed chatbot.

Finally, we identified only one modularized end-to-end neural architecture for CIS proposed by [S16], which consists of six subtasks: (1) **intent detection**, (2) **key phrase extraction**, (3) **action prediction**, (4) **query selection**, (5) **passage selection**, and (6) **response generation**. The neural system can train and evaluate the subtasks jointly and separately and develop a pre-training/fine-tune learning scheme. The authors proposed to develop the model based on transformer encoders and decoders, with two layers (self-attention and position-wise feed-forward layers) and three layers (output-attention, input-attention, and position-wise feed-forwards layers), respectively.

5.4.2. Reference architecture

Therefore, based on the architectures in the included studies, we propose a high-level architecture for conversational search systems, depicted in Figure 5.25. It contains six layers: (1) user interface, (2) NLU, (3) dialog management, (4) search, (5) knowledge, and (6) NLG. All layers are typical of dialog systems, while the search layer is specific to conversational search and is used to retrieve users' information needs from the system's knowledge. The study of [S47] gave us the idea to integrate a component responsible for turn detection when external API models are used. In the following, we give an overview of each layer and component and describe how the authors of the included studies implemented them.

[6] `https://rasa.com/`

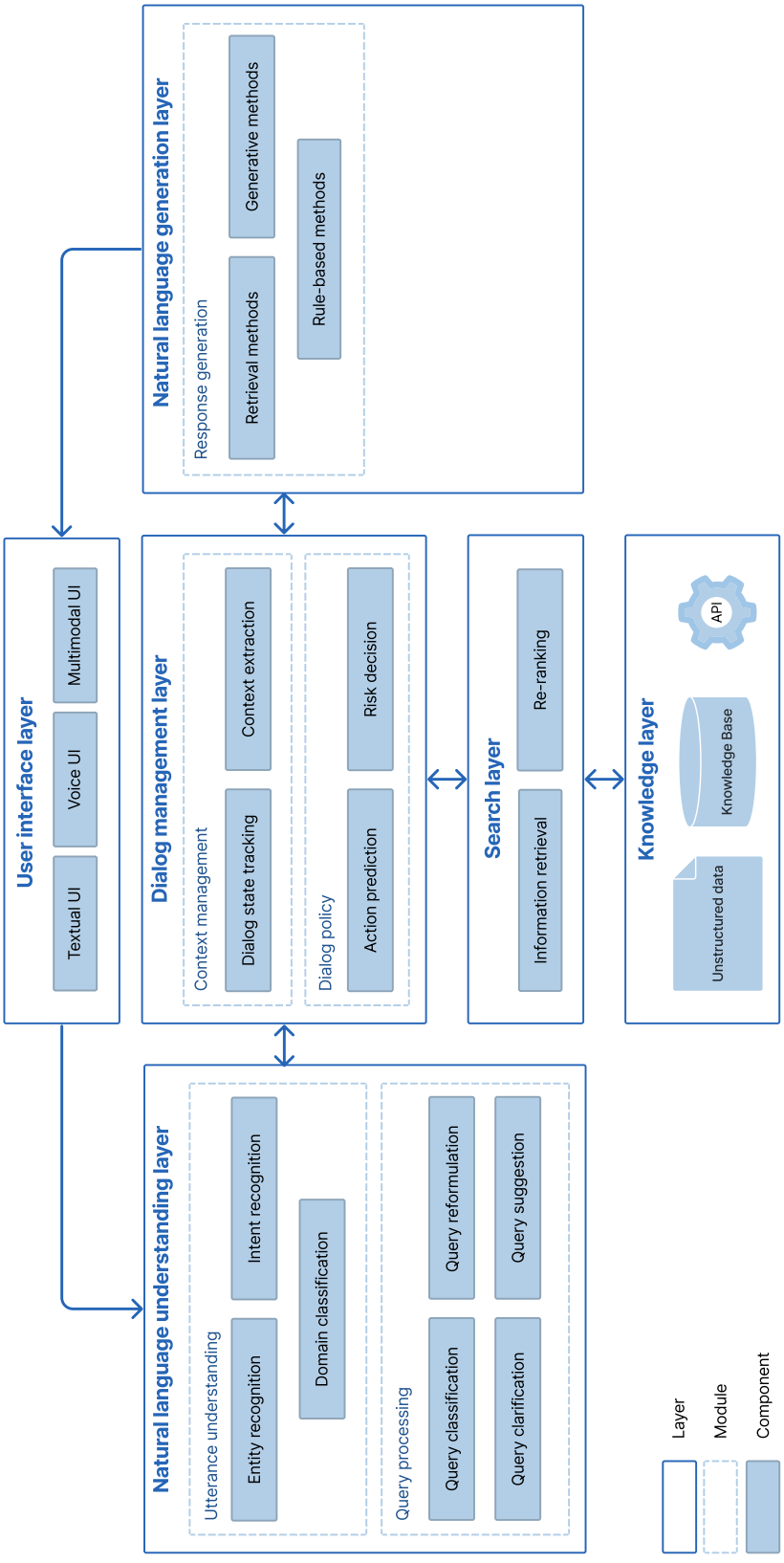

Figure 5.25.: Reference architecture for conversational search systems

UI layer

The user interface layer represents the interface that establishes the interaction channel between a system and a user. We have proposed three components for this layer based on the modalities discussed in section 5.3: textual, voice, and multimodal.

NLU layer

The NLU layer takes the results from the automatic speech recognition and creates a representation of the meaning of the user's request. NLU requires methods to resolve contextual dependencies to understand the conversation correctly. We have not included speech or text synthesis in this component. Therefore, we identified two modules for this layer - utterance understanding and query processing. The modules consist of several components that describe the functionality of the modules in more detail. We proposed entity recognition, domain classification, and intent extraction as components for utterance understanding. Query processing includes classification, reformulation, clarification, and suggestion components.

For instance, Ren et al. [S31] referred to the complete NLU layer in their proposed architecture as conversation and passage understanding. They implemented it based on the transformer model, which encodes and preserves hidden representations for each query.

Utterance understanding. Understanding utterances performs a linguistic analysis and classification of the utterance and its context to identify and extract the intentions, entities, and domains contained in the utterance.

Vakulenko et al. [S3] proposed a vocabulary generator, which generates from a predefined vocabulary set of utterances, where H, "Hi", and B, "Bye", stand for utterances expressing greetings and farewells. Type I represents initiative and is represented by: (1) request utterances, e.g., "Please help me find...", (2) information need utterances, e.g., "I am looking for...", (3) questions asked by either party, e.g., "Are you looking for...?". Other types belong to N. In their study, [S3] proposed a dialog analysis framework, ConversationShape, that includes the following functionalities: (1) *fingerprinting*, an approach to the representation of dialogs; (2) a *model of the dialog flow* that shows regular patterns of initiative dynamics; and (3) a *set of asymmetry metrics* that reflect the distribution of initiative among dialog participants. The authors defined fingerprinting as a matrix of size $n_i \times m_i$, where n_i is the number of utterances in the dialog i, and m_i is the number of features representing the dialog i.

The dialog is encoded in a sequence in which exactly five features represent each utterance: $[A-H-4-0-0, A-I-41-1-0, S-N-42-0-0, ..., A-N-32-0-1]$. Such a representation respects privacy since there is no way to recover the dialog's contents from its fingerprint.

Entity recognition is one of the utterance understanding components, and it handles information extraction to locate entities mentioned in unstructured text and classify them into predefined categories. In the included studies, [S41] proposed a message interpretation component that includes ML modules for entity recognition and intent classification. The

entity recognition method recognized two types of entities - topical keywords and geographic entities. However, the researchers incorporated a different unsupervised approach to entity mention extraction with a list of geo-entities because the pre-trained model often could not extract geo-entities. [S29] suggested BERT-based models due to their success in NLU problems, such as named entity recognition and text classification.

An essential task during each turn is determining the users' intents based on their utterances [S22]. That is, as soon as a user makes a new input, the conversational search system should be able to detect the intention the user is trying to achieve. [S16] defined the formal goal of intent extraction as learning a mapping: $\{C^\tau, X^\tau\} \rightarrow I$, where C^τ is a context, X^τ is a current utterance, and the goal is to predict the user intent i. The authors used transformers and linear classifiers to extract the intents. They concatenated C^τ and X^τ in reverse order of utterance turns and put the unique token $[T1]$ at the beginning to receive input: $S_{T1} = [[T1], X^\tau, C^\tau]$. This sequence is forwarded to a token and positional embedding layer to get E_c. Then the embedding sequence is forwarded into a stack of transformer encoders to get a hidden representation for all tokens $HT_1 = TEncoder(EC)$. Lastly, the hidden representations w.r.t. $[T1]$ are obtained, and a linear classifier with softmax is used to get intent IDs.

Keyner et al. [S41] trained intention classification component with a support vector machine (SVM) classifier to detect nine intentions: greeting, good-bye, add a keyword, add location, search, explore, thank you, affirm, deny. [S16] defined user intents: reveal, revise, interpret, request-phrase, chitchat. Also, [S29] referred to several related works that tried to predict generic user intents with the help of dialogue acts or focused on predicting latent intents in text-based product queries. [S29] mentioned that intent classification models are also used in conversational domains and, e.g., in generic agents. However, [S16] argued that the proposed taxonomies for user intents are either too general or do not represent user behavior well.

Domain classification aims to reduce the search space if the domain is known so that only domain-specific data can be used during the search process. The domain classification component was proposed by [S47]. The authors implemented a domain classifier by fine-tuning the RoBERTa-Large model, which takes the whole dialog context and outputs the domain label.

Query processing. Query processing means a system handles the query by classifying, reformulating, or clarifying the given query given the context of the conversation or generating a query suggestion for a user.

Preprocessing of queries means their simple modification before other NLU components further process them. It can contain such methods as capitalization, stemming, lemmatization, tokenization, and stop word removal. The capitalization technique changes the capitalization of the words. The stemming technique reduces inflected or derived words to their word stem or root form. Compared to stemming, lemmatization goes beyond word reduction and considers the entire vocabulary of a language to perform morphological analysis of words. Stop word reduction filters out common words with irrelevant information. For example, [S35] integrated the UDpipe library for query preprocessing. UDpipe provides essential NLP functions like language-independent tokenization, tagging, lemmatization, and dependency

parsing of raw text. Also, [S25] performed stop word removal and tokenization for the user query.

Query reformulation aims to reformulate a query into self-explanatory based on the conversation context. Recognizing and dealing with such linguistic features as anaphora, i.e., words that explicitly refer to previous conversational turns, or ellipsis, i.e., words that are omitted from a conversation yet understood in the context of the remaining elements, are essential in the context of query or utterance reformulation techniques.

Mele et al. [S33] described an utterance rewriting problem. It takes u_i and T_ctx, the context generated by the context generator component, and rewrites u_i based on one of the rewriting strategies. Authors classified queries as *self-explanatory* (SE), utterances referring to the *first topic* in the conversation (FT), and utterances referring to the *previous topic* (PT). Moreover, the authors suggested five strategies for utterance rewriting: *standard, enriched, last SE, first and last SE, and first or last SE*, presented in Table 5.5. As a comparison, [S25] suggested seven utterance reformulation techniques in their work.

Name	**Description**
Standard:	If u_i is classified as FT, then it is enriched with context extracted from the first utterance. If as PT, then the context is extracted from the previous utterance.
Enriched:	Similar to the standard approach, but if u_i is classified as PT, then the context is extracted from the previous enriched utterance.
Last SE:	Regardless of the current utterance label (FT or PT), this method always propagates the context extracted from the last identified SE utterance.
First and last SE:	Similar to the last SE method, but the utterance is rewritten with the context from both the last seen SE utterance and the first utterance.
First or last SE:	If u_i is classified as FT, it is enriched with the context extracted from the first utterance. If as PT, it is extracted from the last seen SE instead of the previous utterance. For instance, the utterance *"Where are they banned to minors?"* becomes "Where are energy drinks banned to minors?" As energy drinks are the last seen topic.

Table 5.5.: Strategies for utterance rewriting [S33]

Both [S39] and [S46] have integrated a pre-trained text-to-text transformer (T5) for query rewriting, which can be fine-tuned to reformulate conversational queries by providing the query sequences and passages as input and receiving the rewritten query as output. Moreover, [S33] extensively researched related work for query or utterance rewriting techniques. The authors referred to the studies that proposed a Seq2Seq model or a generative approach for contextual query rewriting. In addition, the authors mentioned a study in which a neural utterance relevance model based on BERT was introduced to identify utterances relevant to a particular turn. The unidirectional transformer decoder was used for question rewriting in another related work. Also, a binary term classification was used for query rewriting in conversational search, where a classifier decides for each term in previous rounds whether to add it to the current turn or not. The authors pointed out several related works focused on

sequence generation and term classification models, while [S33] used heuristics for general utterance rewriting.

For co-reference (anaphors) resolution [S25] used the AllenNLP [87] tool. This toolkit has many different features, such as reading comprehension, entity recognition, and co-reference resolution. [S30] defined a co-reference resolution component that implemented co-reference resolution techniques for effective result retrieval but has not provided detailed information on the techniques.

During the query reformulation process, [S25] performed an additional step on the utterances in BERT-CUR to ensure that in the final query, they maintain the same order as in the conversation. This procedure ensures that BERT also learns from the relative order of the questions.

Query expansion describes techniques for adding new terms to a query. [S46] integrated historical query expansion (HQE) component in the Chatty Goose system. The HQE extracts keywords for query expansion using the BM25 scoring function. [S30] also proposed a query generation component that generates a query based on the past user-system interactions. The query generation component may use co-reference resolution for query expansion or re-writing. d [S31] followed an approach by Voskarides et al. [88] to extend the current query by extracting words that contain relevant information to introduce more complex passages while reaching a higher recall rate. Voskarides et al. [88] proposed a query expansion model based on an assumption of word centrality (the most important words) and word recency (words that appear in the most recent turns).

The need for clarification typically arises when a user query is ambiguous, and a system should take the initiative to resolve this ambiguity [S3]. Thus, [S21] found that their system can resolve ambiguous queries and suggest words for use in revised queries. However, the authors did not detail the implementation details of these actions. In contrast, a study dealing with ambiguous queries in conversational search [76], which is not included in our list of included studies, provides a detailed summary of selection and ranking techniques for clarification. For instance, one approach to asking clarifying questions is finding the most relevant question for the given query from a question bank. They referred to a technique that uses a recurrent neural network (RNN) using GloVe word embedding for ranking clarifying questions. In addition, [76] reported on the study that created a taxonomy for clarification requests. The classification included: check, more information, general, selection, confirmation, experience, and others.

Query suggestions provide an opportunity to reflect information needs, capture user intent from feedback, and if the user's information needs are not fully covered, query suggestions can help clarify them. Rosset et al. [S36] proposed a new evaluation metric, usefulness, which measures whether suggestions provide valuable content for the user's following action. The metric included six labels: misses intent, too specific, prequel, duplicate with a query, duplicate with an answer, and useful. They also introduced a question suggestion framework with two conversation approaches: ranking and generating suggestions. Their system for ranking suggestions called DeepSuggest, based on a pre-trained BERT ranker, and their

system for generating suggestions called DeepSuggest-NLG, a fine-tuned GPT-2, were both trained on collected Bing search logs.

Dialog Management

Dialog Management (DM) describes a central component that processes interpreted input from the NLU layer, estimates and decides on the next actions to conduct for the user, control the risks, and generally controls the flow of dialog. We subdivided DM component into *Dialog Decision* and *Dialog Context* sub-layers. Dialog decision has the main task of deciding what action to take next, given the user's input and the current state of the dialog. The dialog context sub-layer is responsible for the conversation's context, state, and history and records information relevant to the conversation to support the dialog management process.

[S41] defined dialog management in their architecture as a component that retrieves the entities and intents from the previous component and selects the following action from a predefined set. Behind the selection process was a trained neural network model.

Dialog policy. The goal of a policy is to estimate and predict the next action for the conversational search system with the highest rating or reward. The agent must decide what actions to take to provide a valuable and meaningful response that advances the conversation, considering the user's goals, the previous conversation, and the user's request or response [S22].

[S16] defined the primary goal of an action prediction functionality is to learn a mapping $\{C^\tau, X^\tau, Q^\tau, D^\tau\} \rightarrow a$, where C^τ is a given context, X^τ is a current user utterance, Q^t are candidate queries, and D^τ are candidate passages. Similar to the intent extraction process described by the authors, [S16] retrieves hidden representations corresponding to the action prediction task based on embedding layers of query and passage candidates. Then, hidden representations for each query and passage candidate, q and d respectfully, are combined with a max pooling to get a single hidden representation, on top of which a linear classifier with softmax is applied to predict the next action. Dialog Policy component from [S45] takes inputs from S_t, intent, and ranked list of products and decides the responses. The authors used written agent templates for the responses. For instance, the decision action request(brand) corresponds to the question, "Do you have a brand in mind?".

To compare, [S18] proposed to let risk-aware decision-making policy learn through reinforcement learning. As a decision-making policy, a Deep Q Network (DQN) was used to decide whether to answer the question or ask a clarification question. The DQN first uses a BERT-based encoder to encode the initial query q, the context history h, the top-k clarification questions, and the answers, $\{cq_1, ..., cq_k\}$ and $\{a_1, ..., a_k\}$ respectfully. It reads re-ranking scores of the top-k questions and answers and answers the output of the re-ranker. DQN is a binary classification network that outputs a 2×1 decision vector $y_pred = \{r_{ans}, r_{cq}\}$ of predicted rewards. There was no annotated data to train the DQN, so the DQN was trained by reinforcement learning from the reward or penalty. The goal of DQN training is to predict the reward of an action (response or clarification question) based on an input state.

Context management. It represents all aspects of the interaction relevant for the system to select the next action. It updates the dialog state based on a context or new observations or other events relevant to the dialog.

Xiao et al. [S45] proposed an integrated state tracker component in their architecture. They treated state tracking as a Seq2Seq problem that takes utterances to predict dialog state S_t using transformers. The state tracker interprets the dialog's content and outputs the user's intent along with the product attributes the user is interested in during the product search. Therefore, in their example, the authors did not use an NLU component to process user utterances but instead integrated a bidirectional transformer encoder into the state tracker component for processing. In general, the authors focused on an end-to-end conversational search system.

Context identification recognizes a set of previous utterances relevant to the current utterance context.[S33] described the main aim of the component to convert the current utterance u_i in natural language into a structured query q_i expressed in an IR-specific query language that can be effectively processed further by an IR system. Thus, the context identification component identifies a set of utterances with their context classification information U_{ctx}, which is necessary for rewriting u_i.

[S25] describes the goal of this component as selecting related utterances so that the current utterance better expresses the user's information needs. Finding a related utterance among multiple utterances during a conversation, therefore, requires joint modeling of the utterance language and the position in the conversation. Therefore, [S25] uses the BERT model to learn these representation functions. Moreover, [S29] cited a work that found that contextually relevant utterances are located at a position in the conversation close to the current utterance, which influences the system's design of the memory and context property.

Context extraction extracts keywords from previous utterances required for the current utterance context. [S33] described this component as context generator component. The component obtains context information U_{ctx} for an utterance u_i. It parses the utterances in U_{ctx} and extracts context information in T_{ctx}, which is passed to the next component. For an utterance u_i, the context extracted from the utterance is c_i, which in turn consists of a list of tokens. Each of these context c_i is paired with the corresponding utterance label y_i, so T_{ctx} is defined as a list of (c_i, y_i) pairs depending on U_{ctx}.

Further, [S33] introduced different methodologies exploiting distinct linguistic features for the context generation component.

- *Context extraction* - identifies significant noun chunks using dependency parsing. For example, the utterance "Is Red Bull bad for you?" extracts the noun chunk "Red Bull".
- *Context on cue* is based on keywords and extracts the current context based on dependency parsing. For example, for the utterance "Tell me about taurine," the noun "taurine" is returned.
- *Context binder* captures all previous noun blocks that are either subject or object to enrich the conversation context up to the current turn.

Search layer

The search layer is responsible for the search process and retrieves the top search results to display to the user directly or for further processing by other components.

Dinan et al. [S50] assumed a large knowledge base consisting of hierarchically placed documents consisting of paragraphs and sentences $M = \{m_1, ..., m_N\}$. For their knowledge retrieval module, they used a standard IR techniques ($c = IR(x, m)$) to first return a set of candidates $m_{c1}, ..., m_{cK}$ for further fine-grained retrieval.

Jin et al. [S47] proposed their knowledge selection component. During this process, the system predicts the relevance between a particular dialog context and each candidate in the entire knowledge base. This process may be time-consuming due to the size of the knowledge base. Therefore, the authors proposed a hierarchical filtering method to narrow the candidate search space. Knowledge selection was divided into domain classification, entity tracking, and knowledge matching. A fine-tuned RoBERTa-Large model was proposed for the domain classifier, which took the dialog context and outputted a domain label for it. The entity tracking module detected the entities mentioned in the dialog context and matched them with the entity-level candidates in the knowledge base. An unsupervised approach based on fuzzy N-gram matching was used for this purpose. Knowledge matching relates to the re-ranking component proposed in our high-level architecture.

Information retrieval. This component handles the information, e.g., documents or passages, retrieval from the knowledge layer.

Formally, for passage retrieval, given a series of natural language utterances or questions $U = \{u_1, u_2, ..., u_n\}$ based on a conversational topic T, the task is to retrieve relevant passages P_i for each utterance u_i. Furthermore, u_i can be conditioned on utterances or questions prior to it, i.e, $\{u_1, u_2, ..., u_{i-1}\}$ [S34].

Several authors described and performed a first-stage retrieval step [S33, S39]. The first-stage retrieval is the initial step of retrieving information from the dataset. This process must retrieve relevant results fast and effectively. Although this first step usually does not provide the best possible order, it can find a certain number or set of results, on the order of millions, in a short time. In the first-stage retrieval by [S39], the BM25 term-matching retrieval model, and the language models with Dirichlet (LMD), and the Jelinek-Mercer smoothing (LMJM) were used to retrieve a small set of passages from millions of available.

CTS component proposed by [S34] also handles the first-stage retrieval process. CTS uses BERT in conjunction with a linear classifier to perform a binary classification over the terms stored in the dialog memory. After the result set is retrieved, it is concatenated with the input question and sent to the search engine to retrieve the passages. The authors used the Indri[7] search engine for the passage retrieval process. The authors found it difficult to develop an effective classifier for CTS because of the limited training data available in CAsT. Therefore,

[7]Indri is an open-source search engine originally implemented to support language models in information retrieval as part of the Lemur Project `https://lemurproject.org/`. For more information on Indri, visit `https://lemurproject.org/indri.php/`

they used weak supervision training by additionally using dialogs from other task-oriented dialog datasets. [S30] and [S33] also like [S34] integrated Indri python interface to retrieve documents from an arbitrary document collection. In addition, Macaw supports web search via the Bing Web Search API. The Retriever component by [S46] retrieves documents via Pyserini, a Python-based toolkit, which implements Python bindings for the Anserini IR toolkit.

[S48] used Knowledge Identification (KI) component to retrieve knowledge from documents. For that, the authors leveraged the pre-trained language model RoBERTa-large and utilized data augmentation methods to learn the general pattern for the information-seeking task. [S45] described results retrieval in the product search module, which outputs a list of products that match the user's desired attributes.

Finally, similar to its previous components, [S16] used transformer encoders for the query selection process to obtain hidden representations, on top of which a binary classifier with sigmoid is used to predict whether a query q should be selected or not. The authors used a similar method for passage retrieval.

Re-ranker. This component receives a set of results obtained through the search and re-ranks them based on the predefined algorithms.

BERT can be fine-tuned in a passage re-ranking with good results, which several researchers have also suggested in their studies [S33, S39, S34]. In the system proposed by [S39], the passage re-ranking transformer uses the pre-trained neural language model BERT to obtain contextual embeddings for a sentence and each of its tokens. These embeddings are used as input to perform re-ranking. The Re-ranker component in Chatty Goose [S46] takes as input the results of the first stage of the search and re-ranks them using pre-trained neural modules provided by PyGaggle, a neural text ranking library. [S38] defined two modules for their model whose main goals were to (1) select aspect-value pairs to ask for feedback and (2) rank based on the fine-grained feedback. Thus for the ranking module, the authors used the aspect-value likelihood embedding model (AVLEM), which could rank items with and without feedback. Finally, [S47] described that the knowledge matching component in their system obtained a list of knowledge candidates and ranked them according to their relevance to the input dialog context. For this purpose, the authors chose hinge loss for model training instead of cross-entropy loss, which is usually used in baseline systems, as it reportedly performed better in re-ranking.

[S34] proposed that the re-ranker component in the MVR module creates different views of information needs embedded in an input query. The re-ranker creates three views, each of which is a query with a different type of extracted contextual information. The first view is created with the terms from the dialog history. The second view is created based on the terms in the retrieved passages. The third view is the reformulation of one of the input questions. Finally, MVR re-ranks each passage using BERT in each view and performs a fusion over the created rankings. BERT was explicitly tuned for re-ranking passages. Therefore, even if the automatic query reformulations are ideal, their overall performance in retrieving passages will have an upper limit equal to the ground truth that can be achieved. Moreover, current

automatic methods do not aim to adapt the re-ranker to the conversational situation.

Knowledge layer

In general, developers are unlimited in selecting knowledge components for CS systems. APIs, unstructured text or corpora, or various knowledge bases can be integrated into the knowledge layer.

There are two well-established datasets for conversational IR - CAsT 2019 and ConvQuestions. The CAsT 2019 provides a dataset with 80 multi-turn conversations, having each from 8 to 12 utterances (748 utterances in total). ConvQuestions (ConvQ) dataset consists of 350 conversations [S3].

Furthermore, [S48] used the Doc2Dial dataset, which addresses the challenge of modeling different dialogue scenes with documents, providing free-form answers, and allowing follow-up questions. In addition, the authors described the MRQA 2019 Shared Task dataset, a collection of multiple reading comprehension datasets for evaluating the generalization ability of question-answering models. Wizard-of-Wikipedia (WoW) is a commonly used knowledge-grounded dialogue dataset that aims to provide content-full responses to user utterances based on Wikipedia documents. [S47] used an unstructured knowledge base, while the questions in the knowledge base were augmented for the training set. For example, all questions that contained an entity name were duplicated with "it" in place of the entity name. [S39] used TREC CAsT corpus as their dataset in the system's implementation. Finally, [S44] developed and used a dataset for information-seeking and curiosity-driven scenarios by crowd-sourcing and released 14K dialogs (181K utterances).

NLG layer

This is the last layer in our reference architecture, which converts the response into a natural language format presented to the user.

Response generation. The response generation module generates a natural language response for the user from the context and result set. The response generation module describes three techniques for generating answers in our reference architecture: retrieval methods, generative methods, and rule-based methods. Rule-based methods use rules to map user sentences into system responses. Given a corpus, retrieval methods retrieve an answer from a corpus based on the conversational context. Generative methods, which also consider conversational context, generate responses based on encoder-decoder or language models.

Dinan et al. [S50] developed two classes of models that can retrieve and generate knowledge: (1) retrieval models that generate an output from a set of answer choices, i.e., from the set of utterances from the training set; and (2) generative models that generate word by word. The authors used the attention mechanism to perform fine-grained retrieval of knowledge sentences. Each sentence in the system's memory $m_{c1}, ..., m_{cK}$ and dialog context x is independently encoded in the same transformer. Then, dot-product attention is performed

between the memory candidates and the dialog context. The final input encoding is computed with dot product attention over $enc(m_{c1}), ..., enc(m_{cK})$ and the resulting weighted sum of these vectors is added to $enc(x)$ to obtain the representation $repLHS(m_{c1}, ..., m_{cK}, x)$.

For the generative transformer approach [S50] considered two ways: two-stage and end-to-end generation. The goal is to elicit the most relevant part of knowledge m_{best}. After, the system has to perform encoding by concatenating m_{best} with dialog context x. In their research, generative models employed BPE encoding. For the two-stage version, the authors used two separately trained models for knowledge selection and utterance prediction tasks. For the end-to-end version, a common transformer encoder encodes all candidates m_{ci} and the dialog history. The complete flow is depicted in Figure 5.26.

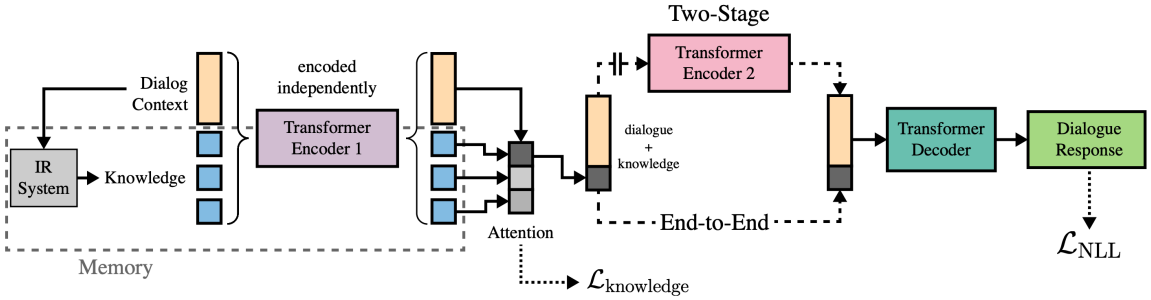

Figure 5.26.: Generative Transformer Memory Network, proposed by [S50]. An IR system provides knowledge candidates. Dialog context and knowledge are encoded using a common encoder. In the two-stage model, the dialog and knowledge are re-encoded after knowledge selection.

The response generation component by [S48] generated responses based on the concatenation of knowledge evidence and dialog context and utilized data augmentation techniques for pre-training and fine-tuning. The authors argued that data augmentation methods are simple to integrate, thus promising in practical use. [S31] generated the responses token by token based on the results of the previous three modules. The proposed CaSE system used the Prior-aware Pointer Generator (PPG) to help generate more accurate responses. PPG can generate tokens from a predefined vocabulary and copy tokens from queries and passages. The idea is that each token can be generated in three modes, i.e., (1) vocabulary generator that generates from a predefined vocabulary, (2) query pointer generator that copies from queries, and (3) prior-aware passage pointer generator that copies from passages. For response generation, [S47] tested three pre-trained Seq2Seq models: T5-Base, BART-Large, and Pegasus-Large. These models take the concatenated sequence of the entire dialog context and knowledge response as input and output of the natural language response. [S2] used template-based (rule-based) techniques for the response generation.

Moreover, if a text is too long, it is important to summarize and shorten the answer for a better user experience. [S39] described the abstract search-response transformer that creates a response based on the candidate passages. The transformer model generates a natural

language answer by summarizing the passages using text-to-text approaches trained on extensive and comprehensive collections. Such approaches are very effective nowadays and can even understand different topics. For the T5 text-to-text transformer, pre-training is performed using supervised and self-supervised training. In self-supervised training, a corrupted sentence is created with 15% sentence tokens removed and replaced. An encoder receives such a sentence as an input, and the original sentence is given to the decoder so that the model trains to predict the replaced tokens. The Pre-training with Extracted Gap-sentences for Abstractive Summarization Sequence-to-sequence model (PEGASUS) model is another response generation approach that specializes precisely in summarization. It has an encoder-decoder architecture and performs the tasks of masked language modeling and gap sentence generation.

Zhang et al. [S37] described CopyNet, an LSTM-based encoder-decoder model that incorporates the copy mechanism. During training, the model encodes the input table using a layer of bidirectional LSTMs and attempts to decode it into the human-written summary. The copying mechanism selects partial sequences from the input and places them in the correct locations in the output. In addition, GPT-2, a large-scale language generation model, was trained with 40 GB of text data from the Internet. Unlike CopyNet, which must be learned from scratch, GPT-2 already learns general language patterns and requires less task-specific training data. The model of GPT-2 consists of the decoder part of the transformer.

5.5. Research questions 4

Finally, in this section, we answer the fourth research question by reporting the results of the scenario-architecture dependency level of CS systems. We defined the fourth research question as follows:

RQ4: To what extent do the system architectures depend on the scenarios?

Here we have looked at the publications that have studied application scenarios and have reported on the architectural specifics for the particular scenario or modality. We report architectural differences concerning the proposed reference architecture presented in Figure 5.25. The results for this research question are the most succinct because not all studies have simultaneously described architectural elements of CS systems for the particular application scenario or modality.

First, we observe the architectural dependence on CS multi-modality. Deldjoo et al. [S10] presented research on the architectures of CS systems considering multi-modality (Figure 5.27). In the proposed architecture, the authors presented the *request dispatcher* component that takes input from the multi-modal conversational understanding component and forwards the processed query to multiple multi-modal information-seeking processes responsible for retrieving or generating responses. Another component presented was *output modality selection*. Unfortunately, the authors did not provide detailed information about the request dispatcher and output modality selection components.

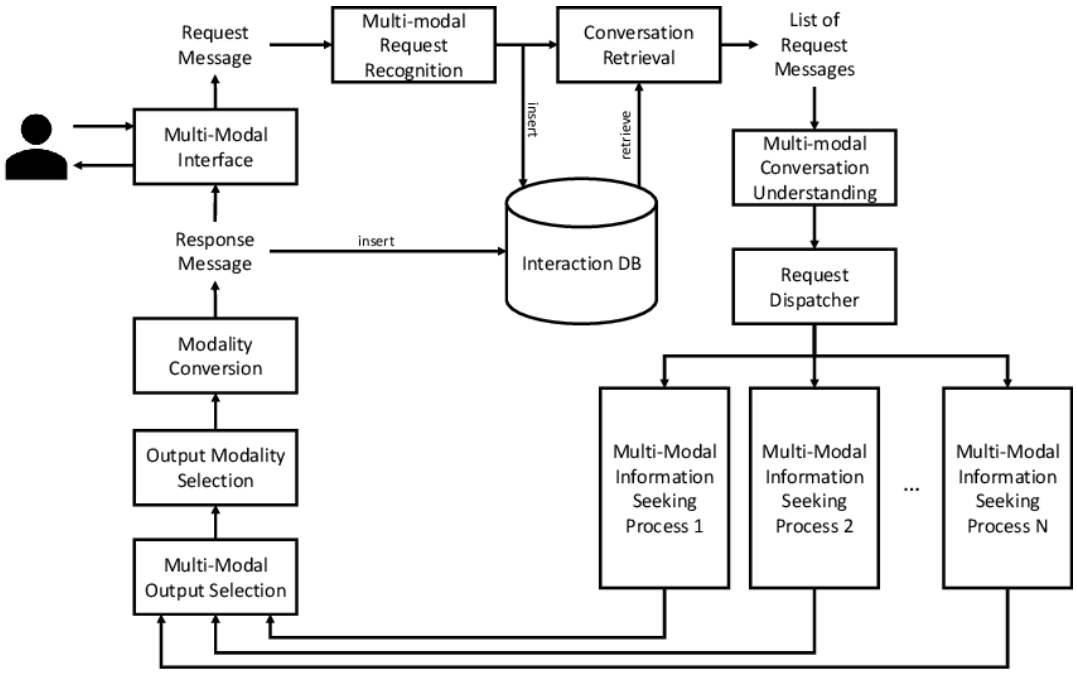

Figure 5.27.: A high-level architecture for the Macaw-MMCIS platform [S10]

Moreover, we observed several scenarios-architecture dependencies. [S13] proposed a conversational search system architecture for document exploration for pilots. The system was built around three main components (Figure 5.28):

(1) Dialog engine based on the RASA toolkit handles the conversation and identifies the user's intentions.

(2) Solr-based search engine, where the document collection is ranked according to the BM25F relevance framework.

(3) QA engine based on a BERT-Large model that was fine-tuned using the FARM framework. The engine performed two tasks: the classical QA task, e.g., recognizing the text segments, and the classification task, i.e., determining whether the answer to the question is in the document excerpt.

The architecture contains several modules from our reference architecture, e.g., the dialog policy, the search layer, or the NLU layer. For the user interface, the authors decided to provide a speech-only interface and thus a separate *linguistic layer* for text-to-speech and speech-to-text synthesis, which is not included in our reference architecture and not part of this research.

Unlike [S13], who proposed interaction through a speech-only interface, [S42] proposed a multi-modal CS system for clinical trials exploration for people with low literacy with a web-based interface and an ability to speak aloud the results on the screen. The user's input is limited to a multiple-choice selection of utterances from a dynamically updated list as

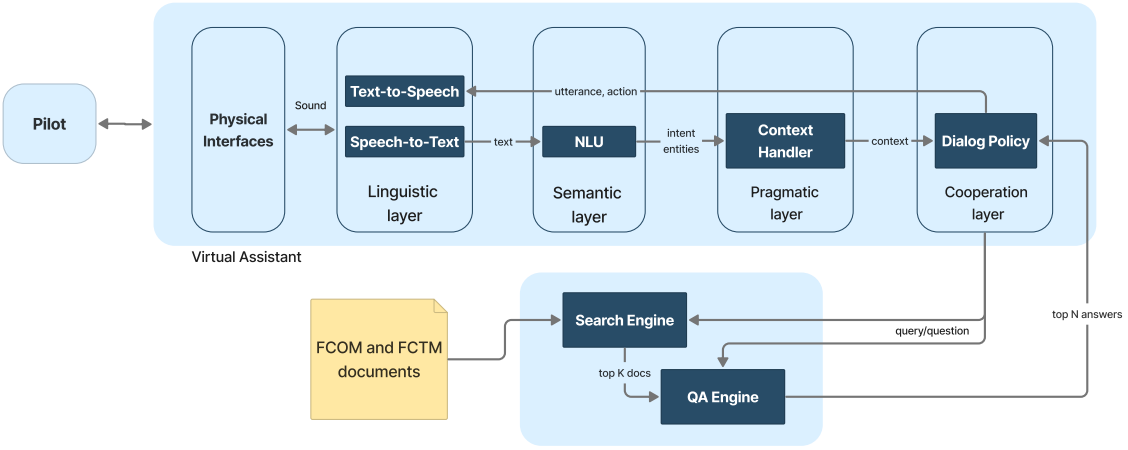

Figure 5.28.: Adapted figure of prototype architecture proposed by [S13]

the interaction progresses. The authors also discussed several components of their system implementation. For example, education modules offered explanations of various concepts underlying clinical trials. Periodically, the system displayed search criteria to confirm or allow users to revise their selections. The system was solely responsible for refining the search. If the search did not yield results, the system suggested other multiple-choice choices to modify the search criteria. As a result, the CS system receives a predefined set of user inputs from multi-choice options, which means that the NLU layer may not be of high complexity.

Kaushik et al. [S21] developed a prototype of a conversational search interface with the ability to perform an exploratory online search. The user could use both traditional web search and conversational search. In their research, the authors suggested allowing the user to disable the conversational search agent and use only web search. The authors did not specify how to implement this part. In the context of online search, the system proposed by [S21] worked as an additional assistant in an exploratory online search setting.

The authors of the included studies also talked about extending their prototypical or developed systems to other application scenarios. [S41] developed a prototype of a conversational search application for government or publicly available datasets. They proposed a pipeline architecture consisting of message interpretation and dialog management modules for the proposed chatbot. The first module includes ML modules for entity recognition and intent classification. The entity extraction method recognized two types of entities - topical keywords and geographic entities. However, the researchers incorporated an additional unsupervised approach to entity mention extraction with a list of geo-entities because the pre-trained model often could not extract geo-entities. Furthermore, the dialog management component retrieves the entities and intents from the previous component and selects the following action from a predefined set. Behind the selection process is a trained neural network model. Based on this architecture, the authors noted that their design approach is not limited to

their specific scenario. They suggested that it can be integrated into other scenarios, such as e-commerce or cultural heritage. The authors suggested that this design may be beneficial in scenarios where users can benefit from more compact and engaging interfaces for search and discovery.

We obtained a similar finding at [S6], where the authors proposed the "system asks - user answers" paradigm for conversational search and recommendation in a product search scenario. During the conversation, the system performs a search with a search component based on the query entered and the items under consideration. Suppose the system is not satisfied with the results. In that case, it generates a question to ask based on a question component in its architecture that also considers the user's query and item representations. The authors stated that their proposed paradigm is not limited to product search and can be extended to scenarios such as conversational academic search, legal search, medical search, or general web search. In addition, it may be helpful in scenarios beyond search and recommendation, such as intelligent device question answering.

6. Discussion

In this chapter, we discuss our key findings from the content and metadata of the studies included in our SLR approach. Although we adopted a rigorous research design and paid significant attention to the selection and review of published studies, the SLR has limitations that may have undercut its efficiency. Below we discuss these limitations.

6.1. Key findings

Despite the diversity of views on conversational search, we found common characteristics, application scenarios, and proposed architectures. We begin with our findings on the general observations from the publications and their metadata and then discuss the results for each research question separately.

6.1.1. General findings

Through the reported statistic, we observed a significant interest increase in the research on the conversational search paradigm in recent years. We almost did not observe studies any studies dated ten years back. However, the ones we studied, for instance, [S24], brought valuable insights into our research, specifically on the modalities. Hence, we would not reduce the research year interval.

Moreover, we did observe how the term conversational search, consequently the interest to study the paradigm, appears around the globe, with 19 countries writing at least one publication on conversational search. Interestingly, the University of Amsterdam in the Netherlands significantly contributes to research on this topic.

We also found that validation research was the primary research method in the included studies, while researchers primarily contributed with a tool and secondarily with guidelines or methods. While 39% presented a tool in their research, not all presented an architecture for it. 36% (18 of 50) of the publications presented a distinct architecture for the CS system. This means that not all who described tools have delivered an architecture as the architecture could also be provided in a method contribution. Interestingly, we have almost omitted publications that present techniques since our goal is to study the system as a whole rather than individual techniques.

Finally, we have obtained interesting insights and specific trends by mapping the research questions. For example, we can see that a publication contributes to RQ4 only in combination

with RQ2 or RQ3, which alone contribute to the application scenarios or architectures. Also, publications that contribute to RQ3 can contribute to RQ1, likely to the functionalities or interaction of a system. Authors describing characteristic properties of CS systems are more likely to provide insights into architectures or application scenarios. Finally, RQ2 provides more independent insights in comparison to other research questions.

6.1.2. Research question 1

In relation to the first research question, many researchers discussed the characteristic properties of conversational search. 87% of the publications mentioned in some way properties, definitions, related concepts, or interaction processes of conversational search. Some proposed entire theoretical or conceptual frameworks [S1, S7] that sufficiently influenced our conceptual framework for conversational search, outlined in Figure 5.7. In our conceptual framework, we divided the proposed definitions into system-based, task-based, and dialog-based. We proposed five characteristic properties of conversational interaction, context and memory, mutual understanding, search assistance, and learning. Conversational interaction and memory and context properties were mentioned several times in the analysis. This is also comprehensible since conversational interaction is a fundamental property of conversational user interfaces and is of general interest to the field of NLP. Moreover, the ability to refer to previous statements and expect the agent to store users' information is expected from modern conversational agents. The concept of mutual understanding may be expected in other conversational UIs but is complex to implement in real-world scenarios, and current VUIs are more likely to result in errors [16]. On the contrary, the ability to learn users' proper information needs while intelligently assisting them during the search process makes conversational search stand out.

In general, we observed several key results for the first research question. Until now, there is no unambiguous definition for a conversational search system. We observed researchers citing several popular studies on the conversational search and proposing definitions. It was possible to divide the described definitions into system-based, task-based, or dialog-based categories (Table 5.1). However, the categories and the definitions are too general, making it challenging to find just a single definition for conversational search. Radlinski et al. [S1] and Trippas et al. [S7], for example, provided independent definitions for conversational search. Nevertheless, Radlinski's [S1] definition did not describe information needs, but user needs in general, which could also imply some user tasks or commands. Trippas et al. [S7] focused only on speech-based conversational search in several publications and provided a stand-alone definition for an SCS. However, we did not observe the requirement that conversational search is exclusively speech-based. Instead, we considered the CS system from its goal viewpoint. We defined that the system's main goal is to maximize the user's information gain by finding search results with maximum utility. In addition, the definitions described gave us an apparent reference to what characteristic properties CS might have and provided insights into how we might design our own.

One of the main goals of conversational search is to elicit genuine information needs of

the user. This goal of the CS paradigm was mentioned several times by the authors of the included studies [S3, S14, S2] and is also related to the general goal of the CS system that we defined. When defining the CS paradigm or describing its goals, the researchers mentioned that the user's informational needs could be achieved through mixed-initiative or conversational dialogue, asking clarifying questions, or giving feedback. This goal also led to the definition of a characteristic property of mutual understanding, where the system knows what the user needs and the user understands what the system's knowledge base is. Consequently, we identify information needs elicitation as a separate objective compared to task-based or open-domain conversational interface goals. This observation leads us to the following important finding.

Conversational search is a distinct paradigm from task-oriented or domain-open dialogs. Since CS describes the same goal as the notion of information-seeking, namely to identify and satisfy users' information needs, it is reasonable to hypothesize that conversational search belongs to a distinct paradigm as task-based and open domain-based systems. Not all authors concluded the distinct notion of conversational search from task-oriented and domain-open systems. However, they described the goals and objectives of CS from an information-seeking perspective. Also, [S2] stated that conversational search is a search performed with a conversational IR system.

Next, a clear trend exists to understand better and conceptualize the conversational search paradigm. As mentioned earlier, some authors have proposed a conceptual framework for CS. For example, Radlinski et al. [S1] presented a theoretical framework for information interaction in a textual setting for CS that emphasizes the need for multi-turn interactions. Azzopardi et al. [S5] proposed a conceptual framework that outlines the actions and intentions of users and agents to enable users to explore the search space and meet their information needs. Kiesel et al. [S4] described many CS properties from a metadata perspective. However, since these papers are relatively recently published, their proposed properties are not aligned, but some overlap could be found that helped us design our own characteristic properties.

In most cases, the conversational search system was described as a mixed-initiative system, but the possibility that one of the participants may take more initiative should not be ruled out. At some level, the notion of conversational interaction can lead to the assumption that conversational search systems must have a mixed-initiative interaction with an equal distribution of initiators on both sides. However, in our research, we observed several studies in which the conversational search process was clearly defined as having the initiative focus on either the system or the user [S31, S27, S31]. A more user-focused initiative leads to the notion of a system answering questions, while a focus on the system's initiative describes more of a setting for conversational recommendations. Nevertheless, several authors explicitly pointed out that the boundaries between conversational search and recommendation tasks are unclear [S3, S6], while one study used the terms search and recommendation interchangeably [S27].

6.1.3. Research question 2

For RQ2, we identified several modalities and application scenarios for conversational search and presented them in Figure 5.15 and Figure 5.17, respectively. About 59% of all included studies pointed to a specific modality or scenario for the CS system. The taxonomy of mentioned modalities represents a summary of the mentioned ways to interact with a CS system. They were divided by interaction channel and presentation mode. The interaction channel can be text-based, speech-based, or hybrid, and the presentation mode can be either verbal or nonverbal. This taxonomy is only a proposal based on the included studies. Other researchers may identify other modalities, especially hybrid ones. The same observation ceased for the application scenarios. We identified application scenarios in four domains: complexity-based, task-based, context-based, and domain-based. The application scenarios we identified describe a wide variability. Nonetheless, we expected to find more application scenarios with different overlaps in their description. However, in some areas, the scenarios describe proposals from only one study, e.g., the scenarios in the complexity domain were proposed by Radlinski et al. [S1] only, while the scenarios in the environment domain were from Deldjoo et al. [S10]. Nevertheless, we could identify different domains, observe how researchers integrate CS systems into this domain and observe their benefits. Hence, we observed two additional findings regarding the second research question.

In theory, conversational search systems pursue multi-modality, but existing implementations are mainly text-based. It was interesting to observe that researchers have indicated in theory that CS systems should integrate multi-modality or hybrid forms of interaction and incorporate nonverbal forms of interaction, such as finger pointing. Some studies did integrate such concepts, e.g., [S42] introduced an avatar that animated nonverbal behaviors (e.g., hand gestures, facial expressions, gaze) in synchrony with voice. However, in most cases, the practical development of CS systems was text-based with a purely verbal presentation mode [S40, S9, S41, S32]. However, this also means an increasing interest in integrating more complex modalities into the system. In future scenarios, the authors tended not to limit the modality possibilities [S10]. In some way, researchers described a conversational search system as an AI that can connect with the user at any time, with various modalities, and under various conditions. Therefore, we expect that practical solutions will complement the theoretical proposals.

Conventional web search remains an important conversational search scenario. Many studies have consistently described the web search process, also referred to as browsing, as one of the tasks of conversational search. These were primarily chat-based solutions that used chat interfaces for free-text conversation with a user and displayed SERP results [S41, S21]. [S21] even proposed to allow the user to disable the CS agent and use only web search. Thus, conversational search should not be automatically comprehended as cutting-edge technology but rather depending on the scenarios and system capabilities. In some cases, conversational search exceeds a simple web search, for instance, when it is difficult for a user to formulate a web search query (e.g., when searching for legal cases [S40]). Hence a CS system helps users meet their information needs in the area where they have limited knowledge.

6.1.4. Research question 3

For RQ3, eighteen publications with several other papers discussing various techniques gave us architectural insights into conversational search and helped us project our reference architecture for CS systems, outlined in Figure 5.25. The reference architecture consists of six layers: (1) user interface, (2) NLU, (3) dialog management, (4) search, (5) knowledge, and (6) NLG. We have described the reference architecture in detail in section 5.4. The collaboration of NLU, DM, search, and NLG layers functioning for the goal of information retrieval highlights the architectural properties and a general notion of conversational search. We can map the proposed characteristic properties to the architectural layers and find that, for example, conversational interaction affects the NLU layer, search assistance addresses the search layer, the learning property addresses the DM layer, and mutual understanding addresses multiple layers from NLU to DM. Moreover, it is interesting to note that all the proposed architectures had a modular structure, and several studies developed end-to-end neural modules [S16, S39, S45]. In most cases, the authors of the included studies did not refer to a reference architecture, which prompted us to construct our own.

Throughout our research of RQ3, we observed several key findings.There is an apparent discrepancy between theoretical concepts and practical implementation. Compared to how much work has already been published on a conversational search, with current technology, it is not yet clear what a conversational search system should perform. Researchers also lack a broader understanding of how users interact with these highly interactive search systems and what components might play a contributing role. For example, including positive or negative feedback was an essential functionality from a theoretical perspective, but only a few studies described which technology implements it [S18, S38]. There is an attempt to provide an open-source, easy-to-use, flexible, and reproducible base system, which some have done [S30, S46]. However, large number of components required to build a system makes it difficult for researchers to build on the work of others.

A major technical challenge is anticipating users' information needs or learning their preferences over time. To accomplish this challenge, a CS system requires sufficient user modeling algorithms. In addition, depending on the application and the technology used, the design and implementation of an intent database, e.g., using Google's DialogFlow engine, can lead to a significant manual effort and require the participation of professional writers to achieve a certain degree of naturalness and richness of the conversation. At the same time, the rule-based modeling approach ("if-this-then-that"), as implemented in the extensive studies, can easily lead to large knowledge bases that are difficult to maintain, creating a need for alternative modeling solutions. Furthermore, to this date, available datasets are generally limited in their linguistic variability, lack multi-domain and multi-modal use cases, or lack annotations [S10]. It is achievable to develop the models with only a handful of examples. However, exploring other scalable approaches that can help extend these models incrementally beyond the initial built-in assumptions about the dialog structure is vital.

6.1.5. Research question 4

We observed the dependencies between CS system architectures and application scenarios or modalities for the last research question. At the beginning of our study, we hypothesized that we might not find profound insights because not all studies would report on the scenario - or modality-architecture dependency in their work. In general, the authors almost did not provide such information, instead, we had to conclude it ourselves, considering architectures and scenarios from the remaining studies. Based on our observations, we can make the following statements.

The architectural characteristics depend on the application scenarios, but not very severely. We observed that the architectural elements were described or introduced differently depending on the scenario. For example, the existence of request dispatcher and output modality selection components is unique in a multi-modal architecture [S13], and we have not observed this in other proposed architectures. In addition, in this thesis, we have discussed how conversational search incorporates the naturalness of interaction, as in human conversations. However, a CS system proposed by [S42] only allows multiple-choice user input as a use case for people with low literacy skills. We assume that such architectural variation may occur mainly in exceptional use cases like the one presented above. However, this still demonstrates that the architecture of conversational search depends on the scenarios.

Finally, several researchers stated that their CS system architecture could be extended to other domains. [S41] suggested an extension to e-commerce or cultural heritage scenarios, while [S6] suggested an extension to conversational, academic search, legal search, medical search, or general web search. The authors did not provide further details on whether there would be changes to the architecture or whether this would result in limitations. However, we conclude that the architectural changes are almost unnecessary if the different tasks have similarities and can be combined under a general scenario. In the example of [S6], web search, medical search, legal search, and others can be categorized under the standard information retrieval scenario that we proposed in section 5.3. If these search tasks require the same user interface, the only difference may be in the retrieval techniques, depending on what the user wants as output. If the goal is to retrieve documents, e.g., some legal files, patient records, or others, the system may treat this as a similar retrieval task, but one that spans multiple domains.

6.2. Limitations

There are several limitations to this study. Firstly, some critical publications may have been overseen, which could affect the incompleteness of our results. We conducted our method according to a set protocol, performed a rigorous search, used multiple databases, and applied a snowballing approach to reduce the risk of missing studies. We executed Python scripts to remove duplicates and unsolicited studies. However, we took the risk that the core context of the paper should be included in the title, abstract, and keywords. Therefore, there might be

some missing or newly published but not included publications that could contribute to our research.

Another limitation that may have occurred is the definition of an incomplete search string. We tested several search terms, and the last one was the most optimal for finding a sufficient number of studies. For example, we tried to include NLP-related terms in the search term because we presumed that the researchers state conversational search paradigm is related to NLP. However, the number of papers found was sufficiently reduced because the researchers had not explicitly stated it in the title, keywords, or abstract. We included several related domains in the notion of conversational search. However, we also observed a tendency to refer to text-based conversational search systems as chatbots, yet the chatbot term is not included in the string.

Another limitation that may have occurred in performing the SLR approach is that the data we collected from the publications may be flawed. Because the data extraction process for many studies is complex, we cannot exclude the possibility of some errors in the data.

Finally, there is a possibility that we have not identified the architectures based on the related concepts of conversational search. Question answering, conversational recommendation, or other closely related paradigms could provide insights into the high-level architecture of conversational search. However, if our search string did not identify them, we could not detect them and may have missed some insights about architectural concepts.

7. Conclusion

In our final chapter, we aim to provide a comprehensive summary of our research by presenting the results and key findings for each research question. Finally, there are opportunities to explore conversational search systems further from diverse directions, including those we have focused on in this thesis. Below, we discuss possible future directions of this research.

7.1. Summary

In this thesis, we conducted a systematic literature review on the conversational search systems. We explored how it can be conceptualized and developed based on the current academic research literature. We approached the overall problem in four directions, formulated as research questions. First, we analyzed the characteristics of conversational search systems as defined in the academic literature. In doing so, we identified a variety of definitions, characteristics, and related concepts of conversational search. Based on these findings, we developed a conceptual framework for conversational search. In our conceptual framework, we categorized the proposed definitions of CS into system-based, task-based, and conversational. We also proposed five characteristic properties of conversational interaction, context and memory, mutual understanding, search assistance, and learning. Conversational interaction allows a user and a CS system to interact in a conversational manner, similar to a human conversation. With the context and memory property, the CS system maintains the conversation context and stores the conversation history and user data. Both the system and the user can thus refer to past statements. With the search assistance property, a CS system assists a user in searching, retrieves relevant results, and can explain the selection of the results. Mutual understanding describes the ability of a user and a system to perceive and continuously update each other's information needs or knowledge. Finally, a system can learn the information about a user and predict the best strategies to satisfy the user's information needs.

We described each characteristic property and proposed a set of functionalities that these properties can entail in chapter 5. We also presented how researchers describe these properties and functionalities. In addition to the conceptual framework, we also discussed the differences between CS and its related concepts. In general, we found that the primary goal of a conversational search system is to maximize the user's information gain by finding search results with maximum utility. This goal leads us to conclude that conversational search is a distinct paradigm from domain-open or task-based dialog systems because it pursues the notion of information-seeking - satisfying the user's need for information.

Second, we examined the application scenarios proposed for the conversational search paradigm. Many studies that described and discussed the conversational search paradigm did so for a specific modality or application scenario. In general, we identified three channels of interaction: text-based, speech-based, or hybrid, consisting of speech and text or speech and gestures. In addition, the CS system can have two presentation modes - verbal and nonverbal. We observed that researchers tend to embrace multi-modal conversational search systems because they use richer context, avoid errors, and improve accessibility. However, the implementations proposed in the included studies for practical use cases were mainly text-based. We also identified application scenarios for which the new CS paradigm is suitable. We identified four different types of application scenarios: complexity-based, task-based, environment-based, and domain-based, and described the proposed scenarios for each class. We identified six different domains: business, tourism, health, law, aerospace, and the public sector, for which researchers have proposed different scenarios for integrating conversational search. The researchers generally described various scenarios, from future perspective scenarios to the tasks in the classic information retrieval scenarios. As a result, we also concluded that the traditional web search scenario is still considered suitable for the emerging CS paradigm.

In our third research question, we explored architectures for the conversational search paradigm to go beyond theoretical observations. We identified various architectural elements, techniques, and algorithms that compile the CS architecture. During conducting the research, we identified the need for a reference architecture for CS systems. Hence based on the findings of architectural elements and general knowledge of dialog system architecture, we proposed a reference architecture for CS systems consisting of six main layers. The user interface layer represents the interface that establishes the interaction channel between a system and a user. The natural language understanding layer represents the meaning of the user's request. The dialog management layer describes a central component that estimates and decides the subsequent system actions and controls the risks and the flow of dialog. The search layer is responsible for the results retrieval from a knowledge layer. The knowledge layer can represent various data types, from unstructured data to knowledge bases and APIs. The last layer of natural language generation converts the response into a natural language format and presents it to the user.

In this work, we presented in detail each layer, its components, and the proposed components' techniques. Generally, the ensemble of NLU, DM, Search, and NLG layers working for the goal of information retrieval highlights the architectural characteristics of conversational search. We can map the proposed characteristic properties to the architectural layers and find that, for example, conversational interaction affects the NLU layer, search assistance addresses the search layer, learning property impacts the DM layer, and mutual understanding impacts several layers from NLU to DM. However, this mapping is very ambiguous, which confirms our observation during the research that there is an apparent discrepancy between theoretical concepts and practical implementation.

For the last research question, we focused on the dependencies between the scenarios

suitable for CS and its architectural elements. Our goal was to investigate how deeply this level of dependence can be defined. We concluded that development for a particular application scenario might introduce components relevant to the scenario but generally result only in partial changes to the architecture.

7.2. Future work

As for future work, understanding the emerging paradigm of conversational search is still an ongoing process, so the possibilities for future work are open-ended. One could go into more detail in exploring the four proposed research questions. As new literature on this topic is constantly published, researchers can introduce new characteristic properties. As an extension of this study, one can observe the changes to the proposed by us conceptual framework. New techniques could be introduced for complex multi-modal interactions.

New architectures could also be introduced over time, which is also exciting to investigate as an extension of this study. One can also compare the efficiency of the architectures or evaluate the divergence between end-to-end and pipeline architectures. We already stated that a significant technical challenge would be anticipating users' information needs or learning their preferences over time. Hence, one can explore the existing techniques and the steps to improve the technology in this direction.

It is also possible to identify new domains for CS systems and research their suitability for the paradigm and what applications are used for the specific purpose of the domain. For example, we have not identified any research for the search for pharmacy or biomedicine scenarios in health domain, but these scenarios are applicable for conversational search.

Finally, we believe validating the SLR results from the academic literature through expert interviews is crucial. Business-driven industry and enterprise players can understand the conversational search paradigm on a different level from a practical perspective and provide valuable insights into the scenarios in a business domain and technological advances for CS systems.

A. Included studies

The table below shows the 50 studies we considered for our research. You can read about how we conducted our SLR approach in chapter 4.

We only provide the summary of the data extraction fields here to allow for a better understanding when reviewing the table.

ID	Extraction field	Value
EF1	Title	
EF2	Year	
EF3	Country	
EF4	Research method	Solution proposal Validation research Evaluation research Opinion papers Philosophical papers Experience papers Secondary research
EF5	Research contribution	Technique Method Tool Resource Guidelines
EF6	Focus topic	Themes and codes (see Table 4.5)
EF7	SLR RQ	RQ1, RQ2, RQ3, RQ4
EF8	Application domain	Health, business, engineering, education, information technology, miscellaneous, etc.

Table A.1.: Data extraction form

Title	Year	Countries	Research method	Contribution	Focus topic	SLR RQ	Application domain
“Mhm…” – Conversational Strategies For Product Search Assistants	2022	Germany	Validation research	Guidelines	Interaction, Modality	RQ1, RQ2	Business
“What Can I Cook with these Ingredients?” - Understanding Cooking-Related Information Needs in Conversational Search	2022	Germany	Validation research	Resource, Guidelines	Interaction	RQ1	Miscellaneous
A large-scale analysis of mixed initiative in information-seeking dialogues for conversational search	2021	Netherlands	Validation research	Tool, Guidelines	Interaction, NLU/NLG technique	RQ1, RQ3	
A theoretical framework for conversational search	2017	United Kingdom	Solution proposal	Method	Definition, Property, Functionality, Tasks	RQ1, RQ2	
Adaptive utterance rewriting for conversational search	2021	Italy, United States	Validation research	Tool, Resource	Functionality, Interaction, NLU/NLG technique	RQ1, RQ3	
Age-related difference in conversational search behavior: preliminary findings	2022	United States	Solution proposal	Guidelines	Definition, Interaction, Modality, Task	RQ1, RQ2	
An analysis of mixed initiative and collaboration in information-seeking dialogues	2020	Netherlands	Validation research	Method	Definition, Interaction, Related concepts	RQ1	
Analysing mixed initiatives and search strategies during conversational search	2021	Australia, United Kingdom, Netherlands, United States	Validation research	Method	Definition, Property, Modality	RQ1, RQ2	
Caire in dialdoc21: data augmentation for information seeking dialogue system	2021	Hong Kong	Validation research	Tool, Resource	Architecture	RQ3	
Can i be of further assistance using unstructured knowledge access to improve task-oriented conversational modeling	2021	United States	Validation research	Method	Architecture	RQ3	
Challenges in conversational search: improving the system capabilities and guiding the search process	2020	United States	Opinion papers, Solution proposal	Method, Guidelines	Definition, Property, Modality	RQ1, RQ2	
Chatty goose: a python framework for conversational search	2021	Canada	Validation research	Tool	Architecture	RQ3	

Conceptualizing agent-human interactions during the conversational search process	2018	United Kingdom	Solution proposal	Guidelines	Interaction	RQ1	
Controlling the risk of conversational search via reinforcement learning	2021	United States	Validation research	Tool	Functionality, Interaction, DM technique	RQ1, RQ3	
Conversational product search based on negative feedback	2019	United States	Validation research	Tool	Modality, Architecture	RQ2, RQ3	Business
Conversational vs traditional: comparing search behavior and outcome in legal case retrieval	2021	China	Validation research	Guidelines	Interaction, Tasks	RQ2	Law
Conversations with Search Engines: SERP-based Conversational Response Generation	2021	China, Netherlands	Validation research	Tool	Architecture	RQ3	
End-to-end conversational search for online shopping with utterance transfer	2021	China	Validation research	Method	Architecture	RQ2, RQ3	Business
Engagement and Usability of Conversational Search – A Study of a Medical Resource Center Chatbot	2021	Denmark	Validation research	Guidelines	Suitability of scenarios, Tasks, Modality	RQ2	Health
Evaluation of conversational agents for aerospace domain	2020	Denmark, France	Validation research	Tool	Architecture, S-A dep level	RQ3, RQ4	Aerospace
Exploring Conversational Search With Humans, Assistants, and Wizards	2017	Canada, United States	Validation research	Guidelines	Property, Interaction, Functionality	RQ1	
Harnessing Evolution of Multi-Turn Conversations for Effective Answer Retrieval	2020	Switzerland, Netherlands	Validation research	Tool	Property, Interaction, NLU/NLG technique	RQ1, RQ3	
How Do People Interact in Conversational Speech-Only Search Tasks: A Preliminary Analysis	2017	Australia	Validation research	Technique	Interaction, Modality, Tasks	RQ1, RQ2	
Improving Access to Online Health Information With Conversational Agents:A Randomized Controlled Experiment	2016	United States	Validation research	Tool	Modality, S-A dep level	RQ2, RQ4	Health

Information seeking in the spirit of learning: a dataset for conversational curiosity	2020		Validation research	Tool, Resource	Architecture	RQ3	
Informing the Design of Spoken Conversational Search	2018	Australia, Japan	Validation research	Guidelines	Interaction, Modality, Tasks	RQ1, RQ2	
Leading conversational search by suggesting useful questions	2020	United States	Validation research	Tool	Interaction, Functionality	RQ1	
Macaw: An Extensible Conversational Information Seeking Platform	2020	United States	Solution proposal	Tool	Architecture	RQ3	
Making information seeking easier: an improved pipeline for conversational search	2020	United States	Validation research	Tool	NLU/NLG technique, DM technique	RQ3	
Meta-information in conversational search	2021	Germany	Validation research	Guidelines	Property, Functionality, Interaction	RQ1	
Mmconv: an environment for multimodal conversational search across multiple domains	2021	China, Singapore	Validation research	Tool	Modality, S-A dep level	RQ2, RQ4	Tourism, miscellaneous
Modelling information needs in collaborative search conversations	2017	Australia, Japan	Validation research	Method	Interaction, Functionality	RQ1	
Multi-modal conversational search and browse	2013	United States	Validation research	Method	Interaction, Modality, Tasks	RQ1, RQ2	
Multi-view conversational search interface using a dialogue-based agent	2021	France, Ireland	Solution proposal	Tool	DM technique, S-A dep level	RQ3, RQ4	
Open data chatbot	2019	Austria	Solution proposal	Tool	Architecture, S-A dep level	RQ3, RQ4	Public sector
Open-domain conversational search assistants: the transformer is all you need	2022	Portugal	Validation research	Tool	Functionality, Architecture	RQ3	
Qrfa: a data-driven model of information-seeking dialogues	2019	Austria, Brazil, Netherlands	Solution proposal	Method	DM technnique, Interaction	RQ1, RQ3	
Simulating and modeling the risk of conversational search	2022	United States	Validation research	Tool	Functionality, Interaction, Architecture	RQ1, RQ3	
Spoken conversational search for general knowledge	2019	France	Solution proposal	Method	Architecture	RQ3	
Studying the effectiveness of conversational search refinement through user simulation	2021	Brazil, United States	Validation research	Method	Functionality, Interaction, Architecture	RQ1, RQ3	

Summarizing and exploring tabular data in conversational search	2020	United Kingdom, Norway, United States	Solution proposal	Technique	Functionality, NLU/NLG technique	RQ1, RQ3	
Tell me more: understanding user interaction of smart speaker news powered by conversational search	2019	Japan	Validation research	Guidelines	Functionality, Interaction	RQ1	
The second conversational intelligence challenge (convai2)	2020		Solution proposal, Secondary research	Method, Resource	DM technique	RQ3	
Towards a model for spoken conversational search	2020	Australia, Japan	Validation research	Guidelines	Property, Functionality, Modality	RQ1, RQ2	
Towards conversational search and recommendation: system ask, user respond	2018	China, United States	Validation research	Method	Definition, Functionality	RQ1	Business
Towards multi-modal conversational information seeking	2021	Australia, Italy, United States	Solution proposal, Opinion papers	Method, Guidelines	Tasks, Modality, S-A dep level	RQ2, RQ4	
Towards system-initiative conversational information seeking	2021	United States	Philosophical papers	Guidelines	Interaction, Tasks	RQ1, RQ2	
Vote goat: conversational movie recommendation	2018	United Kingdom	Solution proposal	Tool	Architecture	RQ3	
Wizard of search engine access to information through conversations with search engines	2021	China, Netherlands	Validation research	Tool	Functionality, Architecture	RQ3	
Wizard of wikipedia: Knowledge-powered conversational agents	2018		Validation research	Tool	DM technique	RQ3	

List of Figures

2.1. Pipeline architecture for task-oriented conversational agents, adapted from [41]. 13
2.2. Natural language understanding example [41]. 14
2.3. Encoder-decoder model, adapted from [29] 15
2.4. Example using encoder-decoder in the Seq2Seq framework [16] 16
2.5. Typology of Conversational Search defines conversational search systems via functional extensions of information retrieval systems, chatbots, and dialogue systems, adapted from [55] 20

4.1. Process for the current SLR, adapted from [79]. 24
4.2. QA3 criteria: citation count in percentage 30
4.3. QA3 criteria divided by year and citation count 30
4.4. Search and selection process 31

5.1. Distribution of publications per year 35
5.2. Number of affiliated papers by county 36
5.3. Domains distribution 37
5.4. Publications with and without domains 37
5.5. Methods distribution 37
5.6. Contribution distribution 37
5.7. Proposed conceptual framework for conversational search systems 38
5.8. Overview of conversational interaction characteristic 42
5.9. Four cycles of information-seeking for QRFA model of conversational search: (a) question answering loop, (b) query refinement loop, (c) offer refinement loop, (d) answer refinement loop. 46
5.10. Overview of context and memory characteristic 48
5.11. Overview of search assistance characteristic 50
5.12. Overview of mutual understanding characteristic 53
5.13. Overview of learning characteristic 55
5.14. Conversational search system is part of an information-seeking system, as proposed by [S27] 58
5.15. Observed modalities in the included studies 60
5.16. A conceptual design of a multi-modal system and its elements [S10] 63
5.17. Proposed scenarios and their categorization 65
5.18. An overview of retrieval and question answering in Macaw [S30] 70
5.19. Pipeline architecture proposed by [S34] 71

5.20. Architecture of the conversational passage retrieval pipeline for Chatty Goose [S46] 72
5.21. Architecture of the knowledge-grounded dialog system proposed by [S47] . . 72
5.22. An overview of Conversations with Search Engines (CaSE) proposed by [S31] 73
5.23. Conversational end-to-end search system proposed by [S45] 74
5.24. Transformer-based conversational search system proposed by [S39] 75
5.25. Reference architecture for conversational search systems 77
5.26. Generative Transformer Memory Network, proposed by [S50]. An IR system provides knowledge candidates. Dialog context and knowledge are encoded using a common encoder. In the two-stage model, the dialog and knowledge are re-encoded after knowledge selection. 87
5.27. A high-level architecture for the Macaw-MMCIS platform [S10] 89
5.28. Adapted figure of prototype architecture proposed by [S13] 90

List of Tables

2.1. Conversational agent types [26] 8
2.2. Design guidelines for search user interfaces [53] 18

4.1. Search sources and parameters 27
4.2. Results breakdown 27
4.3. Inclusion and exclusion criteria 28
4.4. Data extraction form 32
4.5. Themes and codes definition 34

5.1. Definitions of conversational search based on their type 39
5.2. Overview of user and system actions in conversational search [S22] 44
5.3. Observed interaction channels and presentation modes in the included studies. 61
5.4. Observed domains in included studies. 67
5.5. Strategies for utterance rewriting [S33] 80

Bibliography

[1] M. Bates. "Models of natural language understanding." In: *Proceedings of the National Academy of Sciences* 92.22 (1995), pp. 9977–9982.

[2] E. Pons, L. M. Braun, M. M. Hunink, and J. A. Kors. "Natural language processing in radiology: a systematic review". In: *Radiology* 279.2 (2016), pp. 329–343.

[3] X. Wang, G. Hripcsak, M. Markatou, and C. Friedman. "Active computerized pharmacovigilance using natural language processing, statistics, and electronic health records: a feasibility study". In: *Journal of the American Medical Informatics Association* 16.3 (2009), pp. 328–337.

[4] M. A. Hearst. "'Natural'search user interfaces". In: *Communications of the ACM* 54.11 (2011), pp. 60–67.

[5] F. Radlinski and N. Craswell. "A theoretical framework for conversational search". In: *Proceedings of the 2017 conference on conference human information interaction and retrieval*. 2017, pp. 117–126.

[6] S. Seneff and J. Polifroni. "Dialogue management in the Mercury flight reservation system". In: *ANLP-NAACL 2000 Workshop: Conversational Systems*. 2000.

[7] T. Lewandowski, J. Delling, C. Grotherr, and T. Böhmann. "State-of-the-Art Analysis of Adopting AI-based Conversational Agents in Organizations: A Systematic Literature Review." In: *PACIS* (2021), p. 167.

[8] J. Gao, M. Galley, L. Li, et al. "Neural approaches to conversational ai". In: *Foundations and trends® in information retrieval* 13.2-3 (2019), pp. 127–298.

[9] M. McTear, Z. Callejas, and D. Griol. "The Conversational Interface: Talking to Smart Devices: Springer International Publishing". In: *Doi: https://doi.org/10.1007/978-3-319-32967-3* (2016).

[10] M. Porcheron, J. E. Fischer, S. Reeves, and S. Sharples. "Voice interfaces in everyday life". In: *proceedings of the 2018 CHI conference on human factors in computing systems*. 2018, pp. 1–12.

[11] R. Meyer von Wolff, S. Hobert, and M. Schumann. "How may i help you?–state of the art and open research questions for chatbots at the digital workplace". In: *Proceedings of the 52nd Hawaii international conference on system sciences*. 2019.

[12] R. Dale. "The return of the chatbots". In: *Natural Language Engineering* 22.5 (2016), pp. 811–817.

[13] J. Grudin and R. Jacques. "Chatbots, humbots, and the quest for artificial general intelligence". In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 2019, pp. 1–11.

[14] J. Glass. "Challenges for spoken dialogue systems". In: *Proceedings of the 1999 IEEE ASRU Workshop*. Vol. 696. MIT Laboratory fot Computer Science Cambridge, MA, USA. 1999.

[15] Z. Yu, A. W. Black, and A. I. Rudnicky. "Learning conversational systems that interleave task and non-task content". In: *arXiv preprint arXiv:1703.00099* (2017).

[16] M. McTear. "Conversational AI: Dialogue systems, conversational agents, and chatbots". In: *Synthesis Lectures on Human Language Technologies* 13.3 (2020), pp. 1–251.

[17] C. Khatri, A. Venkatesh, B. Hedayatnia, R. Gabriel, A. Ram, and R. Prasad. "Alexa prize—state of the art in conversational ai". In: *AI Magazine* 39.3 (2018), pp. 40–55.

[18] E. Luger and A. Sellen. "'"Like Having a Really Bad PA" The Gulf between User Expectation and Experience of Conversational Agents". In: *Proceedings of the 2016 CHI conference on human factors in computing systems*. 2016, pp. 5286–5297.

[19] A. Thatcher. "Information-seeking behaviours and cognitive search strategies in different search tasks on the WWW". In: *International journal of industrial ergonomics* 36.12 (2006), pp. 1055–1068.

[20] S. Vakulenko. "Knowledge-based Conversational Search". In: *arXiv preprint arXiv:1912.06859* (2019).

[21] M. Dubiel, M. Halvey, L. Azzopardi, and S. Daronnat. "Investigating how conversational search agents affect user's behaviour, performance and search experience". In: *The second international workshop on conversational approaches to information retrieval*. 2018.

[22] A. M. Turing. "Computing machinery and intelligence". In: *Parsing the turing test*. Springer, 2009, pp. 23–65.

[23] J. Weizenbaum. "ELIZA—a computer program for the study of natural language communication between man and machine". In: *Communications of the ACM* 9.1 (1966), pp. 36–45.

[24] R. Wallace. "The elements of AIML style". In: *Alice AI Foundation* 139 (2003).

[25] J. Deriu, A. Rodrigo, A. Otegi, G. Echegoyen, S. Rosset, E. Agirre, and M. Cieliebak. "Survey on evaluation methods for dialogue systems". In: *Artificial Intelligence Review* 54.1 (2021), pp. 755–810.

[26] U. Gnewuch, S. Morana, and A. Maedche. "Towards Designing Cooperative and Social Conversational Agents for Customer Service." In: *ICIS*. 2017.

[27] J. Hill, W. R. Ford, and I. G. Farreras. "Real conversations with artificial intelligence: A comparison between human–human online conversations and human–chatbot conversations". In: *Computers in human behavior* 49 (2015), pp. 245–250.

[28] H. Shah, K. Warwick, J. Vallverdú, and D. Wu. "Can machines talk? Comparison of Eliza with modern dialogue systems". In: *Computers in Human Behavior* 58 (2016), pp. 278–295.

[29] H. Chen, X. Liu, D. Yin, and J. Tang. "A survey on dialogue systems: Recent advances and new frontiers". In: *Acm Sigkdd Explorations Newsletter* 19.2 (2017), pp. 25–35.

[30] B. F. Green Jr, A. K. Wolf, C. Chomsky, and K. Laughery. "Baseball: an automatic question-answerer". In: *Papers presented at the May 9-11, 1961, western joint IRE-AIEE-ACM computer conference*. 1961, pp. 219–224.

[31] D. G. Bobrow, R. M. Kaplan, M. Kay, D. A. Norman, H. Thompson, and T. Winograd. "GUS, a frame-driven dialog system". In: *Artificial intelligence* 8.2 (1977), pp. 155–173.

[32] A. L. Gorin, G. Riccardi, and J. H. Wright. "How may I help you?" In: *Speech communication* 23.1-2 (1997), pp. 113–127.

[33] C. Myers, A. Furqan, J. Nebolsky, K. Caro, and J. Zhu. "Patterns for how users overcome obstacles in voice user interfaces". In: *Proceedings of the 2018 CHI conference on human factors in computing systems*. 2018, pp. 1–7.

[34] M. Wahde and M. Virgolin. "Conversational agents: Theory and applications". In: *arXiv preprint arXiv:2202.03164* (2022).

[35] G. Salton and C. Buckley. "Term-weighting approaches in automatic text retrieval". In: *Information processing & management* 24.5 (1988), pp. 513–523.

[36] D. Adiwardana, M.-T. Luong, D. R. So, J. Hall, N. Fiedel, R. Thoppilan, Z. Yang, A. Kulshreshtha, G. Nemade, Y. Lu, et al. "Towards a human-like open-domain chatbot". In: *arXiv preprint arXiv:2001.09977* (2020).

[37] S. Roller, E. Dinan, N. Goyal, D. Ju, M. Williamson, Y. Liu, J. Xu, M. Ott, K. Shuster, E. M. Smith, et al. "Recipes for building an open-domain chatbot". In: *arXiv preprint arXiv:2004.13637* (2020).

[38] R. Niewiadomski, E. Bevacqua, M. Mancini, and C. Pelachaud. "Greta: an interactive expressive ECA system". In: *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*. Citeseer. 2009, pp. 1399–1400.

[39] Y. Matsuyama, A. Bhardwaj, R. Zhao, O. Romeo, S. Akoju, and J. Cassell. "Socially-aware animated intelligent personal assistant agent". In: *Proceedings of the 17th annual meeting of the special interest group on discourse and dialogue*. 2016, pp. 224–227.

[40] S. Al Moubayed, J. Beskow, G. Skantze, and B. Granström. "Furhat: a back-projected human-like robot head for multiparty human-machine interaction". In: *Cognitive behavioural systems*. Springer, 2012, pp. 114–130.

[41] P. Kulkarni, A. Mahabaleshwarkar, M. Kulkarni, N. Sirsikar, and K. Gadgil. "Conversational AI: An Overview of Methodologies, Applications & Future Scope". In: *2019 5th International Conference On Computing, Communication, Control And Automation (ICCUBEA)*. IEEE. 2019, pp. 1–7.

[42] W. T. Fitch, L. Huber, and T. Bugnyar. "Social cognition and the evolution of language: constructing cognitive phylogenies". In: *Neuron* 65.6 (2010), pp. 795–814.

[43] D. Khurana, A. Koli, K. Khatter, and S. Singh. "Natural language processing: State of the art, current trends and challenges". In: *arXiv preprint arXiv:1708.05148* (2017).

[44] M. Zaib, Q. Z. Sheng, and W. Emma Zhang. "A short survey of pre-trained language models for conversational ai-a new age in nlp". In: *Proceedings of the Australasian Computer Science Week Multiconference*. 2020, pp. 1–4.

[45] T.-H. Wen, M. Gasic, N. Mrksic, P.-H. Su, D. Vandyke, and S. Young. "Semantically conditioned lstm-based natural language generation for spoken dialogue systems". In: *arXiv preprint arXiv:1508.01745* (2015).

[46] T.-H. Wen, M. Gasic, N. Mrksic, L. M. Rojas-Barahona, P.-H. Su, D. Vandyke, and S. Young. "Multi-domain neural network language generation for spoken dialogue systems". In: *arXiv preprint arXiv:1603.01232* (2016).

[47] M. Hearst. *Search user interfaces*. Cambridge university press, 2009.

[48] G. Marchionini. "Information-seeking strategies of novices using a full-text electronic encyclopedia". In: *Journal of the american society for Information science* 40.1 (1989), pp. 54–66.

[49] G. Marchionini. *Information seeking in electronic environments*. 9. Cambridge university press, 1997.

[50] C. Liu, Y.-H. Liu, J. Liu, R. Bierig, et al. "Search interface design and evaluation". In: *Foundations and Trends® in Information Retrieval* 15.3-4 (2021), pp. 243–416.

[51] K. Byström and P. Hansen. "Conceptual framework for tasks in information studies". In: *Journal of the American Society for Information science and Technology* 56.10 (2005), pp. 1050–1061.

[52] D. Kelly et al. "Methods for evaluating interactive information retrieval systems with users". In: *Foundations and Trends® in Information Retrieval* 3.1–2 (2009), pp. 1–224.

[53] B. Shneiderman, D. Byrd, and W. B. Croft. *Clarifying search: A user-interface framework for text searches*. 1997.

[54] M. Zaib, W. E. Zhang, Q. Z. Sheng, A. Mahmood, and Y. Zhang. "Conversational question answering: A survey". In: *arXiv preprint arXiv:2106.00874* (2021).

[55] A. Anand, L. Cavedon, H. Joho, M. Sanderson, and B. Stein. "Conversational search (dagstuhl seminar 19461)". In: *Dagstuhl Reports*. Vol. 9. 11. Schloss Dagstuhl-Leibniz-Zentrum für Informatik. 2020.

[56] Q. Motger, X. Franch, and J. Marco. "Software-Based Dialogue Systems: Survey, Taxonomy and Challenges". In: *ACM Computing Surveys (CSUR)* (2022).

[57] R. Jaber and D. McMillan. "Conversational user interfaces on mobile devices: Survey". In: *Proceedings of the 2nd Conference on Conversational User Interfaces*. 2020, pp. 1–11.

[58] N. Zierau, E. Elshan, C. Visini, and A. Janson. "A review of the empirical literature on conversational agents and future research directions". In: *International Conference on Information Systems (ICIS)*. 2020.

[59] J. Feine, U. Gnewuch, S. Morana, and A. Maedche. "A taxonomy of social cues for conversational agents". In: *International Journal of Human-Computer Studies* 132 (2019), pp. 138–161.

[60] I. Van Es, D. Heylen, B. van Dijk, and A. Nijholt. "Gaze behavior of talking faces makes a difference". In: *CHI'02 Extended Abstracts on Human Factors in Computing Systems*. 2002, pp. 734–735.

[61] Z. Ashktorab, M. Jain, Q. V. Liao, and J. D. Weisz. "Resilient chatbots: Repair strategy preferences for conversational breakdowns". In: *Proceedings of the 2019 CHI conference on human factors in computing systems*. 2019, pp. 1–12.

[62] H. Candello, C. Pinhanez, and F. Figueiredo. "Typefaces and the perception of humanness in natural language chatbots". In: *Proceedings of the 2017 chi conference on human factors in computing systems*. 2017, pp. 3476–3487.

[63] C. Bérubé, T. Schachner, R. Keller, E. Fleisch, F. v Wangenheim, F. Barata, T. Kowatsch, et al. "Voice-based conversational agents for the prevention and management of chronic and mental health conditions: systematic literature review". In: *Journal of medical Internet research* 23.3 (2021), e25933.

[64] Q. Jin, Z. Yuan, G. Xiong, Q. Yu, H. Ying, C. Tan, M. Chen, S. Huang, X. Liu, and S. Yu. "Biomedical question answering: A survey of approaches and challenges". In: *ACM Computing Surveys (CSUR)* 55.2 (2022), pp. 1–36.

[65] L. Laranjo, A. G. Dunn, H. L. Tong, A. B. Kocaballi, J. Chen, R. Bashir, D. Surian, B. Gallego, F. Magrabi, A. Y. Lau, et al. "Conversational agents in healthcare: a systematic review". In: *Journal of the American Medical Informatics Association* 25.9 (2018), pp. 1248–1258.

[66] J. L. Z. Montenegro, C. A. da Costa, and R. da Rosa Righi. "Survey of conversational agents in health". In: *Expert Systems with Applications* 129 (2019), pp. 56–67.

[67] T. Schachner, R. Keller, F. Von Wangenheim, et al. "Artificial intelligence-based conversational agents for chronic conditions: systematic literature review". In: *Journal of medical Internet research* 22.9 (2020), e20701.

[68] S. Hobert and R. Meyer von Wolff. "Say hello to your new automated tutor–a structured literature review on pedagogical conversational agents". In: (2019).

[69] B. Khosrawi-Rad, H. Rinn, R. Schlimbach, P. Gebbing, X. Yang, and C. Lattemann. "Conversational Agents in Education–A Systematic Literature Review". In: (2022).

[70] J. Li, T. Tang, W. X. Zhao, and J.-R. Wen. "Pretrained language models for text generation: A survey". In: *arXiv preprint arXiv:2105.10311* (2021).

[71] B. Guo, H. Wang, Y. Ding, W. Wu, S. Hao, Y. Sun, and Z. Yu. "Conditional text generation for harmonious human-machine interaction". In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 12.2 (2021), pp. 1–50.

[72] D. Jannach, A. Manzoor, W. Cai, and L. Chen. "A survey on conversational recommender systems". In: *ACM Computing Surveys (CSUR)* 54.5 (2021), pp. 1–36.

[73] T. Mahmood and F. Ricci. "Improving Recommender Systems with Adaptive Conversational Strategies". In: *Proceedings of the 20th ACM Conference on Hypertext and Hypermedia*. HT '09. Torino, Italy: Association for Computing Machinery, 2009, pp. 73–82. ISBN: 9781605584867. DOI: 10.1145/1557914.1557930. URL: https://doi.org/10.1145/1557914.1557930.

[74] Y. Sun and Y. Zhang. "Conversational recommender system". In: *The 41st international acm sigir conference on research & development in information retrieval*. 2018, pp. 235–244.

[75] Y. Zhang, X. Chen, Q. Ai, L. Yang, and W. B. Croft. "Towards Conversational Search and Recommendation: System Ask, User Respond". In: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. CIKM '18. Torino, Italy: Association for Computing Machinery, 2018, pp. 177–186. ISBN: 9781450360142. DOI: 10.1145/3269206.3271776. URL: https://doi.org/10.1145/3269206.3271776.

[76] K. Keyvan and J. X. Huang. "How to Approach Ambiguous Queries in Conversational Search? A Survey of Techniques, Approaches, Tools and Challenges". In: *ACM Computing Surveys (CSUR)* (2022).

[77] S. Keele et al. *Guidelines for performing systematic literature reviews in software engineering*. Tech. rep. Technical report, ver. 2.3 ebse technical report. ebse, 2007.

[78] B. Kitchenham. "Procedures for performing systematic reviews". In: *Keele, UK, Keele University* 33.2004 (2004), pp. 1–26.

[79] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson. "Systematic mapping studies in software engineering". In: *12th International Conference on Evaluation and Assessment in Software Engineering (EASE) 12*. 2008, pp. 1–10.

[80] B. Kitchenham, D. Budgen, and P. Brereton. *Evidence-Based Software Engineering and Systematic Reviews*. Nov. 2015. ISBN: 9780429157653. DOI: 10.1201/b19467.

[81] R. Wieringa, N. Maiden, N. Mead, and C. Rolland. "Requirements engineering paper classification and evaluation criteria: a proposal and a discussion". In: *Requirements engineering* 11.1 (2006), pp. 102–107.

[82] R. L. Glass and I. Vessey. "Contemporary application-domain taxonomies". In: *IEEE software* 12.4 (1995), pp. 63–76.

[83] N. J. Belkin. "Anomalous states of knowledge as a basis for information retrieval". In: *Canadian journal of information science* 5.1 (1980), pp. 133–143.

[84] A. H. Miller, W. Feng, A. Fisch, J. Lu, D. Batra, A. Bordes, D. Parikh, and J. Weston. "Parlai: A dialog research software platform". In: *arXiv preprint arXiv:1705.06476* (2017).

[85] J. Lin, X. Ma, S.-C. Lin, J.-H. Yang, R. Pradeep, and R. Nogueira. "Pyserini: A Python toolkit for reproducible information retrieval research with sparse and dense representations". In: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2021, pp. 2356–2362.

[86] C. Zhai and J. Lafferty. "A study of smoothing methods for language models applied to ad hoc information retrieval". In: *ACM SIGIR Forum*. Vol. 51. 2. ACM New York, NY, USA. 2017, pp. 268–276.

[87] M. Gardner, J. Grus, M. Neumann, O. Tafjord, P. Dasigi, N. Liu, M. Peters, M. Schmitz, and L. Zettlemoyer. "Allennlp: A deep semantic natural language processing platform". In: *arXiv preprint arXiv:1803.07640* (2018).

[88] N. Voskarides, D. Li, A. Panteli, and P. Ren. "ILPS at TREC 2019 Conversational Assistant Track." In: *TREC*. 2019.

[89] A. Purington, J. G. Taft, S. Sannon, N. N. Bazarova, and S. H. Taylor. "'"Alexa is My New BFF": Social Roles, User Satisfaction, and Personification of the Amazon Echo". In: *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. CHI EA '17. Denver, Colorado, USA: Association for Computing Machinery, 2017, pp. 2853–2859. ISBN: 9781450346566. DOI: 10.1145/3027063.3053246. URL: https://doi.org/10.1145/3027063.3053246.

Studies included in the literature review

[S1] F. Radlinski and N. Craswell. "A theoretical framework for conversational search". In: Association for Computing Machinery, Inc, 2017, pp. 117–126. DOI: 10.1145/3020165.3020183.

[S2] N. Sa and X. Yuan. "Challenges in conversational search: Improving the system capabilities and guiding the search process". In: vol. 3. International Institute of Informatics and Systemics, IIIS, 2020, pp. 37–42.

[S3] S. Vakulenko, E. Kanoulas, and M. De Rijke. "A Large-scale Analysis of Mixed Initiative in Information-Seeking Dialogues for Conversational Search". In: 39.4 (2021). DOI: 10.1145/3466796.

[S4] J. Kiesel, L. Meyer, M. Potthast, and B. Stein. "Meta-Information in Conversational Search". In: 39.4 (2021). DOI: 10.1145/3468868.

[S5] M. Aliannejadi, L. Azzopardi, H. Zamani, E. Kanoulas, P. Thomas, and N. Craswell. "Analysing Mixed Initiatives and Search Strategies during Conversational Search". In: Association for Computing Machinery, 2021, pp. 16–26. DOI: 10.1145/3459637.3482231.

[S6] Y. Zhang, X. Chen, Q. Ai, L. Yang, and W. Bruce Croft. "Towards conversational search and recommendation: System Ask, user respond". In: Association for Computing Machinery, 2018, pp. 177–186. DOI: 10.1145/3269206.3271776.

[S7] J. Trippas, D. Spina, P. Thomas, M. Sanderson, H. Joho, and L. Cavedon. "Towards a model for spoken conversational search". In: 57.2 (2020). DOI: 10.1016/j.ipm.2019.102162.

[S8] Z. Xing, X. Yuan, and J. Mostafa. "Age-Related Difference in Conversational Search Behavior: Preliminary Findings". In: *ACM SIGIR Conference on Human Information Interaction and Retrieval*. Association for Computing Machinery, 2022, pp. 259–265. DOI: 10.1145/3498366.3505830.

[S9] T. Fergencs and F. Meier. "Engagement and Usability of Conversational Search – A Study of a Medical Resource Center Chatbot". In: 12645 LNCS (2021), pp. 328–345. DOI: 10.1007/978-3-030-71292-1_26.

[S10] Y. Deldjoo, J. Trippas, and H. Zamani. "Towards Multi-Modal Conversational Information Seeking". In: Association for Computing Machinery, Inc, 2021, pp. 1577–1587. DOI: 10.1145/3404835.3462806.

[S11] S. Vakulenko, E. Kanoulas, and M. De Rijke. "An Analysis of Mixed Initiative and Collaboration in Information-Seeking Dialogues". In: Association for Computing Machinery, Inc, 2020, pp. 2085–2088. DOI: 10.1145/3397271.3401297.

[S12] S. Shiga, H. Joho, R. Blanco, J. Trippas, and M. Sanderson. "Modelling information needs in collaborative search conversations". In: Association for Computing Machinery, Inc, 2017, pp. 715–724. DOI: 10.1145/3077136.3080787.

[S13] Y.-H. Liu, A. Arnold, G. Dupont, C. Kobus, and F. Lancelot. "Evaluation of conversational agents for aerospace domain". In: vol. 2621. CEUR-WS, 2020.

[S14] J. Trippas, D. Spina, L. Cavedon, H. Joho, and M. Sanderson. "Informing the design of spoken conversational search". In: vol. 2018-March. Association for Computing Machinery, Inc, 2018, pp. 32–41. DOI: 10.1145/3176349.3176387.

[S15] S. Vakulenko, K. Revoredo, C. Di Ciccio, and M. de Rijke. "QRFA: A data-driven model of information-seeking dialogues". In: 11437 LNCS (2019), pp. 541–557. DOI: 10.1007/978-3-030-15712-8_35.

[S16] P. Ren, Z. Liu, X. Song, H. Tian, Z. Chen, Z. Ren, and M. de Rijke. "Wizard of Search Engine: Access to Information Through Conversations with Search Engines". In: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, 2021, pp. 533–543. DOI: 10.1145/3404835.3462897.

[S17] A. Papenmeier, A. Frummet, and D. Kern. "'"mhm.." conversational strategies for product search assistants". In: Association for Computing Machinery, Inc, 2022, pp. 36–46. DOI: 10.1145/3498366.3505809.

[S18] Z. Wang and Q. Ai. "Controlling the risk of conversational search via reinforcement learning". In: Association for Computing Machinery, Inc, 2021, pp. 1968–1977. DOI: 10.1145/3442381.3449893.

[S19] Z. Wang and Q. Ai. "Simulating and Modeling the Risk of Conversational Search". In: *ACM Trans. Inf. Syst.* 40.4 (2022). DOI: 10.1145/3507357.

[S20] A. Salle, S. Malmasi, O. Rokhlenko, and E. Agichtein. "Studying the Effectiveness of Conversational Search Refinement Through User Simulation". In: 12656 LNCS (2021), pp. 587–602. DOI: 10.1007/978-3-030-72113-8_39.

[S21] A. Kaushik, N. Loir, and G. Jones. "Multi-view Conversational Search Interface Using a Dialogue-Based Agent". In: 12657 LNCS (2021), pp. 520–524. DOI: 10.1007/978-3-030-72240-1_58.

[S22] L. Azzopardi, M. Dubiel, M. Halvey, and J. Dalton. "Conceptualizing agent-human interactions during the conversational search process". In: *The second international workshop on conversational approaches to information retrieval*. 2018.

[S23] J. Trippas, D. Spina, L. Cavedon, and M. Sanderson. "How do people interact in conversational speech-only search tasks: A preliminary analysis". In: Association for Computing Machinery, Inc, 2017, pp. 325–328. DOI: 10.1145/3020165.3022144.

[S24] L. Heck, D. Hakkani-Tür, M. Chinthakunta, G. Tur, R. Iyer, P. Parthasarathy, L. Stifelman, E. Shriberg, and A. Fidler. "Multi-modal conversational search and browse". In: vol. 1012. CEUR-WS, 2013, pp. 96–101.

[S25] M. Aliannejadi, M. Chakraborty, E. Ríssola, and F. Crestani. "Harnessing evolution of multi-turn conversations for effective answer retrieval". In: Association for Computing Machinery, Inc, 2020, pp. 33–42. DOI: 10.1145/3343413.3377968.

[S26] E. Dinan, V. Logacheva, V. Malykh, A. Miller, K. Shuster, J. Urbanek, D. Kiela, A. Szlam, I. Serban, R. Lowe, et al. "The second conversational intelligence challenge (convai2)". In: *The NeurIPS'18 Competition*. Springer, 2020, pp. 187–208.

[S27] S. Wadhwa and H. Zamani. "Towards system-initiative conversational information seeking". In: vol. 2950. CEUR-WS, 2021, pp. 102–116.

[S28] H. Jung, G. Hwang, J. Lee, C. Oh, C. Oh, and B. Suh. "Tell me more: Understanding user interaction of smart speaker news powered by conversational search". In: Association for Computing Machinery, 2019. DOI: 10.1145/3290607.3312979.

[S29] A. Frummet, D. Elsweiler, and B. Ludwig. ""What Can I Cook with These Ingredients?" - Understanding Cooking-Related Information Needs in Conversational Search". In: *ACM Trans. Inf. Syst.* 40.4 (2022). DOI: 10.1145/3498330.

[S30] H. Zamani and N. Craswell. "Macaw: An Extensible Conversational Information Seeking Platform". In: Association for Computing Machinery, Inc, 2020, pp. 2193–2196. DOI: 10.1145/3397271.3401415.

[S31] P. Ren, Z. Chen, Z. Ren, E. Kanoulas, C. Monz, and M. De Rijke. "Conversations with Search Engines: SERP-based Conversational Response Generation". In: 39.4 (2021). DOI: 10.1145/3432726.

[S32] A. Vtyurina, D. Savenkov, E. Agichtein, and C. Clarke. "Exploring conversational search with humans, assistants, and wizards". In: vol. Part F127655. Association for Computing Machinery, 2017, pp. 2187–2193. DOI: 10.1145/3027063.3053175.

[S33] I. Mele, C. Muntean, F. Nardini, R. Perego, N. Tonellotto, and O. Frieder. "Adaptive utterance rewriting for conversational search". In: 58.6 (2021). DOI: 10.1016/j.ipm.2021.102682.

[S34] V. Kumar and J. Callan. "Making information seeking easier: An improved pipeline for conversational search". In: Association for Computational Linguistics (ACL), 2020, pp. 3971–3980.

[S35] L. Rojas-Barahona, P. Bellec, B. Besset, M. Dos-Santos, J. Heinecke, M. Asadullah, O. Le-Blouch, J. Lancien, G. Damnati, E. Mory, and F. Herledan. "Spoken conversational search for general knowledge". In: Association for Computational Linguistics (ACL), 2019, pp. 110–113.

[S36] C. Rosset, C. Xiong, X. Song, D. Campos, N. Craswell, S. Tiwary, and P. Bennett. "Leading Conversational Search by Suggesting Useful Questions". In: Association for Computing Machinery, Inc, 2020, pp. 1160–1170. DOI: 10.1145/3366423.3380193.

[S37] S. Zhang, Z. Dai, K. Balog, and J. Callan. "Summarizing and Exploring Tabular Data in Conversational Search". In: Association for Computing Machinery, Inc, 2020, pp. 1537–1540. DOI: 10.1145/3397271.3401205.

[S38] K. Bi, Q. Ai, Y. Zhang, and W. Bruce Croft. "Conversational product search based on negative feedback". In: Association for Computing Machinery, 2019, pp. 359–368. DOI: 10.1145/3357384.3357939.

[S39] R. Ferreira, M. Leite, D. Semedo, and J. Magalhaes. "Open-domain conversational search assistants: the Transformer is all you need". In: 25.2 (2022), pp. 123–148. DOI: 10.1007/s10791-022-09403-0.

[S40] B. Liu, Y. Wu, Y. Liu, F. Zhang, Y. Shao, C. Li, M. Zhang, and S. Ma. "Conversational vs Traditional: Comparing Search Behavior and Outcome in Legal Case Retrieval". In: Association for Computing Machinery, Inc, 2021, pp. 1622–1626. DOI: 10.1145/3404835.3463064.

[S41] S. Keyner, V. Savenkov, and S. Vakulenko. "Open Data Chatbot". In: 11762 LNCS (2019), pp. 111–115. DOI: 10.1007/978-3-030-32327-1_22.

[S42] T. Bickmore, D. Utami, R. Matsuyama, and M. Paasche-Orlow. "Improving access to online health information with conversational agents: A randomized controlled experiment". In: 18.1 (2016). DOI: 10.2196/JMIR.5239.

[S43] L. Liao, L. Long, Z. Zhang, M. Huang, and T.-S. Chua. "MMConv: An Environment for Multimodal Conversational Search across Multiple Domains". In: Association for Computing Machinery, Inc, 2021, pp. 675–684. DOI: 10.1145/3404835.3462970.

[S44] P. Rodriguez, P. Crook, S. Moon, and Z. Wang. "Information Seeking in the Spirit of Learning: A Dataset for Conversational Curiosity". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2020, pp. 8153–8172. DOI: 10.18653/v1/2020.emnlp-main.655.

[S45] L. Xiao, J. Ma, X. L. Dong, P. Martinez-Gomez, N. Zalmout, W. Chen, T. Zhao, H. He, and Y. Jin. "End-to-End Conversational Search for Online Shopping with Utterance Transfer". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2021, pp. 3477–3486. DOI: 10.18653/v1/2021.emnlp-main.280.

[S46] E. Zhang, S.-C. Lin, J.-H. Yang, R. Pradeep, R. Nogueira, and J. Lin. "Chatty Goose: A Python Framework for Conversational Search". In: Association for Computing Machinery, Inc, 2021, pp. 2521–2525. DOI: 10.1145/3404835.3462782.

[S47] D. Jin, S. Kim, and D. Hakkani-Tur. "Can I be of further assistance? using unstructured knowledge access to improve task-oriented conversational modeling". In: (2021), pp. 119–127. DOI: 10.18653/v1/2021.dialdoc-1.16.

[S48] Y. Xu, E. Ishii, G. I. Winata, Z. Lin, A. Madotto, Z. Liu, P. Xu, and P. Fung. "CAiRE in DialDoc21: Data Augmentation for Information Seeking Dialogue System". In: *Proceedings of the 1st Workshop on Document-grounded Dialogue and Conversational Question Answering (DialDoc 2021)*. Association for Computational Linguistics, 2021. DOI: 10.18653/v1/2021.dialdoc-1.6.

[S49] J. Dalton, V. Ajayi, and R. Main. "Vote goat: Conversational movie recommendation". In: Association for Computing Machinery, Inc, 2018, pp. 1285–1288. DOI: 10.1145/3209978.3210168.

[S50] E. Dinan, S. Roller, K. Shuster, A. Fan, M. Auli, and J. Weston. "Wizard of wikipedia: Knowledge-powered conversational agents". In: *arXiv* (2018).