Lost in Dialogue: A Review and Categorisation of Current Dialogue System Approaches and Technical Solutions

Hannes Kath[1,2(✉)], Bengt Lüers[1], Thiago S. Gouvêa[1(✉)], and Daniel Sonntag[1,2]

[1] German Research Center for Artificial Intelligence (DFKI), Oldenburg, Germany
{hannes_berthold.kath,bengt.lueers,thiago.gouvea,daniel.sonntag}@dfki.de
[2] University of Oldenburg, Applied Artificial Intelligence (AAI), Oldenburg, Germany

Abstract. Dialogue systems are an important and very active research area with many practical applications. However, researchers and practitioners new to the field may have difficulty with the categorisation, number and terminology of existing free and commercial systems. Our paper aims to achieve two main objectives. Firstly, based on our structured literature review, we provide a categorisation of dialogue systems according to the objective, modality, domain, architecture, and model, and provide information on the correlations among these categories. Secondly, we summarise and compare frameworks and applications of intelligent virtual assistants, commercial frameworks, research dialogue systems, and large language models according to these categories and provide system recommendations for researchers new to the field.

Keywords: Dialogue System · Conversational AI · Task-oriented · Natural Language Processing · Survey

1 Introduction

Major advances in natural language processing (NLP) through deep learning have tremendously strengthened research in dialogue systems [8,41]. However, the vast number of dialogue system descriptions and surveys lack standardised terminology and dialogue systems are mainly categorised by their objective [5,9,54], neglecting current research topics such as multi-modality [30]. That can be confusing and daunting for researchers and practitioners new to this field, and even experienced researchers could benefit from a structured review. The main goal of this paper is to facilitate researchers' entry into the field of dialogue systems. To achieve this, we first present a theoretical background that introduces the categories of modality, domain, architecture and model that we have derived from the literature, in addition to the objective. Then, we categorise applications and frameworks of intelligent virtual assistants, commercial frameworks, research dialogue systems, and large language models based on the derived categories and provide beginner-friendly system recommendations. The rest of the

D. Seipel and A. Steen (Eds.): KI 2023, LNAI 14236, pp. 98–113, 2023.
https://doi.org/10.1007/978-3-031-42608-7_9

paper is structured as follows: Sect. 2 outlines our structured literature review approach. Section 3 gives an overview of dialogue system categories derived from the literature and explains the relationships between them. Section 4 provides descriptions and recommendations for applications and frameworks suitable for different purposes.

2 Methods

The aim of our structured literature review is to provide a summary of categories, terminologies, applications and frameworks related to dialogue systems to assist researchers new to the field. We have manually reviewed contributions from the main technically oriented research and application venues for dialogue systems, in particular SIGdial[1] and Interspeech[2]. This resulted in a selection of 63 papers. To complement this selection with other technically sound survey papers, we extended the results by searching Scopus[3] (TITLE("dialog* system*" AND ("survey" OR "review")) AND PUBYEAR>2016) and ACM Digital Library[4] ([[Title:"dialog* systems"] OR [Title:"dialog* system"]] AND [[Title:survey] OR [Title:review]] AND [Publication Date: (01/01/2017 TO *)]). From 25 results we excluded duplicates (3), non-English articles (3), articles focusing on specific languages (2) and articles focusing on specific fields (medical domain) (3). The remaining 14 surveys cover the topics general knowledge [5,41], evaluation [7,9,14,29], deep learning [6,44], task-oriented dialogue system components (natural language understanding [31], dialogue state tracking [1]), empathy [36], corpora [35,54] and multi-modality [30]. While these surveys focus on specific topics in the field of dialogue systems, to our knowledge there is no elaboration that introduces newcomers to the topic and offers practical suggestions for applications and frameworks. The whole selection has 77 relevant papers. Acronyms used throughout the paper are listed in Table 1.

3 Dialogue Systems

A dialogue system, in literature also called conversational agent, virtual agent, (intelligent) virtual assistant, digital assistant, chat companion system, chatbot or chatterbot, is an interactive software system that engages in natural language conversations with humans. The communication is usually structured in *turns* (one or more utterances from one speaker), *exchanges* (two consecutive turns) and the *dialogue* (multiple exchanges) [9]. We present the criteria extracted from the structured literature review for categorising dialogue systems in the following subsections.

[1] https://www.sigdial.org/.
[2] https://www.interspeech2023.org/.
[3] https://www.scopus.com.
[4] https://dl.acm.org.

Table 1. List of acronyms

ASR	automatic speech recognition	LLM	large language model
CDS	conversational dialogue system	NLG	natural language generation
DM	dialogue manager	NLP	natural language processing
DRAS	dialogue response action selection	NLU	natural language understanding
DST	dialogue state tracking	QADS	question answering dialogue system
GUI	graphical user interface	TDS	task-oriented dialogue system
IVA	intelligent virtual assistant	TTS	text to speech

3.1 Objective

Most surveys divide dialogue systems into task-oriented dialogue systems (TDSs), conversational dialogue systems (CDSs) and question answering dialogue systems (QADSs) according to their objective. The terminology used in the literature is not consistent and occasionally ambiguous. A TDS is also referred to as task-specific, task-based, goal-driven or goal-oriented dialogue system and sometimes simply dialogue system [54] or conversational agent [41]. A CDS is also referred to as open-domain, chit-chat, non-goal-driven, social, non-task-orientated or chat-oriented dialogue system. The term chatbot is not clearly defined and used for TDSs [41], CDSs [7,36] or dialogue systems in general [44]. We use the term dialogue system as defined above and TDS, CDS and QADS as subcategories.

Task-oriented Dialogue Systems (TDSs) are designed to help users complete specific tasks as efficiently as possible [5,9], such as providing information (e.g. timetable information) or carrying out actions (e.g. ordering food, booking a hotel). Due to the clearly defined goal the dialogue is highly structured. The initiative (party that initiates an exchange) is shared, as the user defines the target (e.g. ordering food) and the TDS requests the necessary information about the constraints (e.g. type of food) [9]. Requesting information allows multiple turns, but the dialogue is kept as short as possible. Evaluation metrics for TDSs are accuracy (correct result) and efficiency (number of turns). Evaluation methods include *user satisfaction modelling*, which derives objective properties (e.g. accuracy) from subjective user impressions using frameworks such as PARADISE [64], and *user simulation*, which mimics humans to assess comprehensibility, relevance of responses, user satisfaction and task performance. Most TDSs use the semantic output for agenda-based user simulation (ABUS) [51], while newer ones use the system output for neural user simulation (NUS) [24].

Conversational Dialogue Systems (CDSs) are designed to have long-term social conversations without solving a specific task [9,54]. Social conversations require extensive analysis (of content, user personality, emotions, mood, and background), system consistency (no inconsistencies in personality, language style or content) and interactivity, resulting in complex systems [20]. The dialogue is unstructured and aims to emulate natural human conversations. A

response generation engine computes the response $Y_t \in \Omega$ out of the response-space Ω from the current utterance X_t and the dialogue context C_t. In *retrieval-based methods*, Ω consists of a corpus of predefined utterances, and the response Y_t is produced by first ranking Ω based on X_t and C_t, and then selecting an element from some top subset. Ranking can be achieved with traditional learning-to-rank methods [32] or modern neural models [15,20,21,34] *Generation-based methods* use a corpus V of predefined words (dictionary). The response-space $\Omega = V^m$ is large, where m is the response length in words. These methods are mainly implemented by neural models [20] such as sequence-to-sequence models [55,57,58,63], conditional variational autoencoders [77], generative adversarial networks [28,72] or transformers [71,75]. *Hybrid methods* combine the advantages of retrieval-based methods (grammatically correct, well-structured responses of high quality) and generation-based methods (large response-space) [69,73]: a response selected from a corpus of predefined utterances is chosen and adapted using a corpus of predefined words [20].

As the aim of CDSs is to entertain the user, interactivity is required and the initiative is shared. Emulating social interactions leads to long dialogues [9].

There has been no agreement on how to evaluate CDSs due to the goal of user entertainment being vaguely defined [11]. Manual evaluation is time-consuming, costly and subjective [20]. Existing methods either use Turing test techniques [33] or evaluate the appropriateness of the generated responses [11].

Question Answering Dialogue Systems (QADSs) are designed to answer specific questions (e.g. extract information from an input sheet), often neglecting the naturalness of the answers generated [9]. The dialogue is unstructured but follows the question and answer style [9]. The initiative is user-centred, with the user posing a specific question.

The conversation is kept brief, with three distinct approaches being identified: *Single turn QADSs* respond without queries and are often used for simple tasks such as extracting information. While open QADSs use web pages or external corpora as knowledge sources [13], the more common reading comprehension extracts information from a document [9]. Modern approaches use pre-trained large language models such as BERT [10] or XLNet [74] (see Sect. 4.4). *Context QADSs* (also known as multi-turn or sequential QADSs) break down complex tasks into simple questions using follow-up questions and are used in reading comprehension to extract quotations. They often consist of single turn QADSs with extended input for dialogue flow [9]. *Interactive QADSs* combine context QADS with TDS and primarily coordinate constraints (e.g. pages to search). Few (many) constraints lead to many (few) results and are therefore added (removed) [9].

The evaluation metrics for QADS are accuracy and, less commonly, dialogue flow. Single turn QADSs are evaluated by mean average precision, mean reciprocal rank or F-Score, while context QADSs are mostly evaluated qualitatively by hand [9,53].

3.2 Modality

Modality refers to the channels of communication between humans and computers that can be text-based, speech-based, or multi-modal [30]. Multi-modal interfaces enable more expressive, efficient, and robust interaction by processing multimedia [45]. Dialogue systems can use different modalities for input and output.

3.3 Domain

A domain is the topic or specific area of knowledge that a dialogue system covers. *Single-domain* dialogue systems are restricted to a specific domain, such as a flight booking system. *Multi-domain* dialogue systems are restricted to several specific domains, e.g. weather, news, appointments and reminders. Single-domain and multi-domain dialogue systems are equipped with special knowledge bases tailored to the respective domains. *Open-domain* dialogue systems are not restricted to specific domains, but can in principle answer questions from any domain. On the other hand, they are usually unable to convey specific knowledge, tend to give ambiguous or incorrect answers due to incorrect semantic analysis, and are usually unable to handle complex, multi-step tasks such as booking a flight.

3.4 Architecture

The architecture of a dialogue system can be either modular (also called pipelined), where each task is performed by a separate module, or end-to-end, where intermediate results are not interpretable and outputs are generated directly from inputs [5,14,44]. In addition, the literature contains end-to-end modules, integrated systems that bypass interpretable intermediate results within the module.

Modular Dialogue Systems consist of different modules (see Fig. 1) [44,54], which are described in more detail below.

Automatic speech recognition (ASR) transcribes the audio signal. As ASR is not part of the dialogue system, we will not go into further detail, but refer to [37] for a comprehensive overview of the state of the art.

Natural language understanding (NLU) extracts information from written text by performing intent and semantic analysis (see Table 2), also known as slot-filling [31,54]. The specific slots that need to be filled are not predetermined.

The *dialogue manager (DM)* is responsible for controlling the flow of the conversation, selecting appropriate responses based on the current context and user input, and maintaining the overall coherence and relevance of the dialogue.

Dialogue state tracking (DST) is the first module of the DM and computes the current dialogue state using the dialogue history [5,44] and either the output of NLU [17,18,70] or of ASR [19,22,66]. States are computed by slot-filling, but

Table 2. Extracted information for the example sentence *show restaurant at New York tomorrow* by the module natural language understanding [5]

Sentence	show	restaurant	at	New	York	tomorrow
Slots	O	O	O	B-desti	I-desti	B-date
Intent	Find Restaurant					
Category	Order					

unlike NLU, the slots are known in advance. States consist of target constraints (possible values: unimportant, not named or user defined), a list of requested slots and the search method (possible values: by constraints (user specifies the information about the requested slots), by alternatives (user wants an alternative for the requested slot), finished (user ends the dialogue)) [44].

Dialogue response action selection (DRAS) is the second module of the DM and selects for the dialogue state S_t at time t either an action from the action-space $A = \{a_1, ..., a_n\}(f : S_t \rightarrow a_i \in A)$ [44] or a request for missing information about necessary constraints to perform an action [5]. Recent DRAS implementations largely use reinforcement learning [16,44].

Natural language generation (NLG) generates a human-readable utterance from the action selected by DRAS. Pipeline structures of content planning, sentence planning and concrete realisation [50] have largely been replaced by end-to-end approaches [60,68,78].

Text to speech (TTS) converts the text output of NLG into an audio output. As TTS is not part of the dialogue system, we will not go into detail, but refer to [59] for a comprehensive overview of the state of the art.

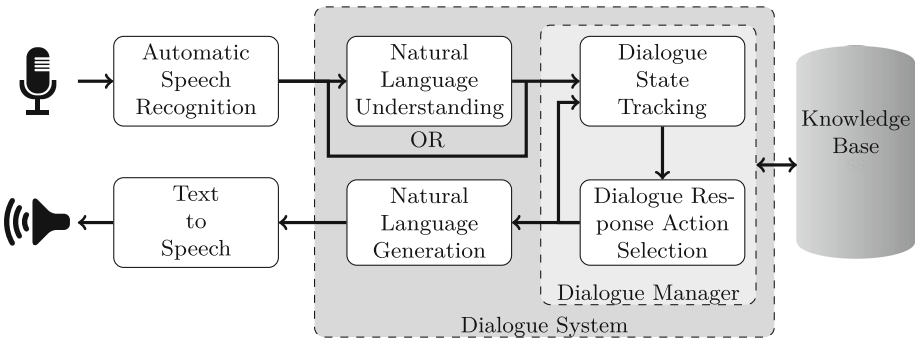

Fig. 1. Structure of a modular speech-based dialogue system [44,54]

End-to-end Dialogue Systems address the drawbacks of modular systems: modules must be well matched, improving individual modules may not improve

the whole dialogue system [5,44], and training the system with backpropagation requires all modules to be differentiable (which requires a gradient computation) [2,76]. Hence, current designs use either differentiable modules [26], facing the problem that knowledge retrieval is not differentiable [44], or end-to-end architectures that generate answers without interfaces/intermediate results from a discrete action-space or by statistical means, making the system more domain independent [54].

3.5 Model

Dialogue systems are usually created using artificial intelligence techniques. Rule-based models use symbolic artificial intelligence, where fixed sets of rules are implemented [5,41]. Statistical models using machine learning and neural models using deep learning, on the other hand, are data-driven approaches [6,44] that are trained on dialogue or speech corpora [35,54].

3.6 Correlations Among Categories

We have divided dialogue systems into categories, although it should be noted that the categories are partially correlated. To illustrate their relationships, Table 3 provides an overview of the main correlations, grouped by objectives.

Table 3. Correlations of the categories of dialogue systems, grouped by objective

Objective	Modality	Domain	Architecture	Model
TDS	text/speech/multi	single/multi	modular	rule/statistical/neural
CDS	text/speech/multi	open	end-to-end	neural
QADS	text	multi/open	modular/end-to-end	rule/statistical/neural

TDSs can be text-based, speech-based or multi-modal. They assist users in performing specific tasks within one or more domains, with a knowledge source providing the necessary information. The high degree of dialogue structure and the associated predetermined dialogue flow lead to modular architectures for most TDSs. They can be implemented using rule-based, statistical, neural or a combination of these methods.

CDSs can be text-based, speech-based or multi-modal. They engage in long-term conversations without preset topics and are therefore open-domain dialogue systems. Most CDSs are end-to-end approaches to generating domain-independent responses, but modular systems similar to Fig. 1 are also possible [20]. End-to-end approaches are most often implemented using neural models because of their ability to handle the high complexity of such systems.

QADSs are mostly text-based, but can also be speech-based or multi-modal. They are either tailored to specific domains or designed to handle questions from open domains. The architecture and method of implementing the functions of QADSs vary according to the specific purpose of the dialogue system.

4 Applications and Frameworks

Dialogue systems have become ubiquitous, e.g. in the form of emotional care robots, website guides and telephone assistants [31]. While applications are ready-to-use systems, frameworks provide a development environment for creating applications. In this section, we provide a detailed description of the four main categories and recommend applications or frameworks for each. *Intelligent virtual assistants (IVAs)* are applications (either virtual embodied agents or voice assistants), typically developed by large companies, and provide personalised answers or actions for various problems in real time. *Commercial frameworks* refer to frameworks used by companies to develop their own applications and integrate them into their business. *Research dialogue systems* are applications or frameworks that are developed for the purpose of advancing technology, investigating new functionalities, and addressing existing challenges. Table 4 gives an overview of these three categories. *Large language models (LLMs)* are models, sometimes applications, that can represent the semantics and syntax of human language more accurately than traditional machine learning models. Besides dialogue systems, LLMs are used in NLP for a variety of other tasks. They do not fully fit into the categories outlined in Table 4 and have therefore been excluded.

4.1 Intelligent Virtual Assistants (IVAs)

IVAs, also called voice assistants, are usually freely available in multiple languages, not open source TDSs and designed to provide users with a quick and effective way to interact with technology through a text or speech interface. Challenges include recognising wake words (e.g. "Hey Siri"), assisting users with a variety of tasks and providing instant information retrieval with little effort [9], simplifying daily tasks and activities. The top part of Table 4 contains a selection of the most commonly used IVAs. Amazon's Alexa, Samsung's Bixby, Microsoft's Cortana, Google's Assistant and Apple's Siri differ mainly in the platforms and integrations they can be used on, which are based on each company's devices. Due to the similarity of these systems, we recommend using the IVA that best matches the researcher's existing devices to get a first insight into their functions. XiaoIce is different in that it is a Chinese CDS optimised for long-term user engagement [79].

4.2 Commercial Frameworks

Commercial frameworks (middle part of Table 4) are designed to easily integrate an interactive interface into business applications to elicit responses or perform actions. The diversity of customer companies leads to a low-code or no-code policy for all frameworks except Rasa. A free trial is available for all frameworks except Dragon and SemVox. The ontology used in commercial frameworks to create TDSs consists of *Intents* and *Entities* (also called *Concepts*), which are learned through examples. An intent is the goal or purpose behind a user's input, while entities are lists of specific information relevant to fulfilling that

Table 4. Selected intelligent virtual assistants, commercial frameworks and research dialogue systems categorised by objective, modality, domain, architecture and model. (•) means it applies, (-) not applicable, (x) no information available.

		Objective			Modality			Domain			Architecture		Model				
	System name	Task-oriented	Conversational	Question Answering	Text	Speech	Multi-modal	Single	Multi	Open	Modular	End-to-end	Rule-based	Statistical	Neural	Open source	Specifications
Intelligent Virtual Assistants	Amazon Alexa[a]	•	-	-	•	•	-	-	•	-	-	•	•	•	•	-	
	Bixby[b]	•	-	-	•	•	-	-	•	-	x	x	•	•	•	-	
	Cortana [48]	•	-	-	•	•	-	-	•	-	•	-	•	•	•	-	
	Google Assistant[c]	•	-	-	•	•	-	-	•	-	x	x	•	•	•	-	
	Siri[d]	•	-	-	•	•	-	-	•	-	x	x	•	•	•	-	
	XiaoIce [79]	-	•	-	•	•	-	-	-	•	•	-	•	-	•	-	Emotional module, Chinese
Commercial Frameworks	Cerence Studio[e]	•	-	-	•	•	-	-	•	-	•	-	-	•	•	-	
	Conversational AI[f]	•	-	-	•	•	-	-	•	-	-	•	x	x	x	-	
	Dialogflow[g]	•	-	-	•	•	-	-	•	-	-	•	•	-	•	-	
	Dragon[h]	-	-	-	-	•	-	-	•	•	x	x	-	-	•	-	ASR (dictation software)
	LUIS[i]	•	-	-	•	•	-	-	•	-	x	x	-	-	•	-	
	Nuance Mix[j]	•	-	•	•	•	-	-	•	•	•	-	•	-	•	-	
	Rasa[k]	•	-	-	•	-	-	-	•	-	•	-	-	•	•	•	NLU&DM, also research system
	SemVox [3]	•	-	-	-	•	•	-	•	-	•	-	x	x	x	-	also IVA
	Watson Assistant[l]	•	-	•	•	•	-	-	•	-	•	-	-	-	•	-	
Research Dialogue Systems	DenSPI [52]	-	-	•	•	-	-	-	-	•	-	•	-	-	•	•	
	DrQA [4]	-	-	•	•	-	-	-	-	•	•	-	•	-	•	•	
	R^3 [65]	-	-	•	•	-	-	-	-	•	•	-	-	-	•	•	
	SmartWeb [56]	-	-	•	•	•	•	-	•	-	•	-	•	-	-	•	
	ELIZA [67]	-	•	-	•	-	-	-	-	•	•	-	•	-	-	•	First dialogue system (1966)
	ConvLab [27]	•	-	-	•	-	-	-	•	-	•	•	-	-	•	•	GUI included
	DialogOS [23]	•	-	-	-	•	-	-	•	-	•	-	•	-	-	•	GUI included
	Nemo [25]	•	•	•	-	•	-	-	-	•	•	-	-	-	•	•	
	ParlAI [40]	•	•	•	•	-	•	-	-	•	-	•	-	-	•	•	
	Plato [47]	•	-	-	•	-	-	-	•	-	•	•	•	•	-	•	Speech interface integrable
	PyDial [61]	•	-	-	•	•	-	-	•	-	•	-	-	•	-	•	
	ReTiCo [38,39]	•	-	-	•	•	-	-	•	-	•	-	-	•	•	•	DM based on Rasa
	Siam-dp [42]	•	-	•	-	•	•	-	•	-	•	-	•	-	-	•	

[a] https://developer.amazon.com/alexa
[b] https://bixbydevelopers.com
[c] https://developers.google.com/assistant
[d] https://developer.apple.com/siri
[e] https://developer.cerence.com/landing
[f] https://cai.tools.sap
[g] https://cloud.google.com/dialogflow
[h] https://www.nuance.com/dragon.html
[i] https://www.luis.ai
[j] https://docs.nuance.com/mix
[k] https://rasa.com/
[l] https://www.ibm.com/products/watson-assistant

intent. The example sentence "I want a coffee" could be mapped to the intent ORDER_DRINK with the entity DRINK_TYPE and it's value (from a predefined list) set to COFFEE.

The nine commercial frameworks can be summarised as follows: SAP's Conversational AI, Google's Dialogflow, Microsoft's LUIS (next version: CLU) and IBM's Watson Assistant allow easy integration of dialogue systems into applications such as Twitter, Facebook, Slack, etc., do not offer separate access to intermediate results, and the last three frameworks require (even to use the free trial) further personal data such as phone number or payment option. Cerence Studio, Dragon and Nuance Mix were originally developed by Nuance Communications, a leading provider of speech processing solutions for businesses and consumers. Dragon is not a framework for dialogue systems, but one of the leading dictation software. Cerence Studio and Nuance Mix use similar graphical user interfaces (GUIs) and workflows, both providing separate access to each module of the dialogue system. SemVox allows multi-modal interfaces and also uses the ASR and TTS modules from Cerence Studio. Unlike the other commercial frameworks, Rasa can be used offline and is open source. Rasa is very popular in the research community and is used for many projects such as ReTiCo [38,39].

To help developers and researchers get started with commercial frameworks, we recommend Cerence Studio because it offers a state-of-the-art demo version that allows the retrieval of intermediate results from any dialogue system module, has an intuitive GUI, and provides straightforward tutorials. It is also free to use and does not require any personal information other than name and email address. The workflow is divided into two steps, using pre-trained ASR and TTS modules. For each intent of the NLU module, the user creates example phrases with associated concepts in the *.nlu* tab to train the model. The DST, DRAS and NLG modules are trained in the *.dialog* tab. The user creates a table with example sentences, collected concepts, actions, and answers/requests. Deploying an application generates an app_id, an app_key and a context_tag, which are used to access the services via a WebSocket connection (a GUI-enabled JavaScript client is provided for testing).

4.3 Research Dialogue Systems

Research dialogue systems are usually open source and serve either as frameworks for research environments or as applications developed to present research results. The bottom part of Table 4 provides a brief summary of the research dialogue systems identified during our structured literature review. Table 5 compares these systems in terms of features relevant to implementation and use.

Large companies have become interested in researching dialogue systems as the technology has been integrated into various fields such as healthcare, education, transport and communication: DrQA and ParlAI were developed by Facebook, R^3 by IBM, ConvLab by Microsoft, Nemo by Nvidia and Plato by Uber. DialogOS was developed by the smaller company CLT Sprachtechnologie GmbH. The pioneering ELIZA dialogue system (1966) and ReTiCo were developed by

Table 5. Properties relevant for implementation of research dialogue systems: (•) means applicable, (-) not applicable, as of status 04/2023

System	Application	Framework	Code Availability	Programming Language	Windows	MacOS	Linux	Tutorial	Demonstration	Last Activity (Releases, Commits, Issues)	Purpose
DenSPI	•	-	•	Python	•	•	•	-	•	06/2022	Real-time, Wikipedia-based
DrQA	•	-	•	Python	-	•	•	-	•	11/2022	Machine reading at scale
R^3	•	-	•	Lua, Python	•	•	•	-	-	04/2018	Reinforcement learning
SmartWeb	•	-	-	Java	•	•	•	-	-	2014	Multi-modal access to Web
ELIZA	•	•	•	Python	•	•	•	-	•	1966	Simulate interlocutors
ConvLab	-	•	•	Python	-	•	•	•	-	04/2023	Reusable experimental setup
DialogOS	-	•	•	Java	•	•	•	-	-	12/2022	Student projects
Nemo	-	•	•	Python	•	•	•	•	-	04/2023	Reuse code and models
ParlAI	-	•	•	Python	-	•	•	•	•	04/2023	Share, train and test systems
Plato	-	•	•	Python	•	•	•	•	-	09/2020	Multi-agent setting
PyDial	-	•	•	HTML	•	•	•	•	•	04/2022	Research on modules
ReTiCo	-	•	•	Python	•	•	•	-	-	04/2023	Real time, incremental
Siam-dp	-	•	•	Java	•	•	•	-	-	02/2017	Multi-modal development

a single researcher, while DenSPI, SmartWeb, PyDial and Siam-dp were developed by research institutions. Typically, research focused on dialogue systems is designed to investigate specific research questions, often resulting in small, specialised projects that may still be in the development phase when published and become inactive shortly afterwards. For a first insight into research, we recommend using the DrQA application and the NeMo framework. Both are state of the art, fully functional, user friendly, actively used and provide tutorials or demonstrations.

DrQA is an open-domain QADS consisting of a *Document Retriever*, which selects relevant documents from a large unstructured knowledge source (e.g. Wikipedia), and a *Document Reader*, which searches them to answer the given question. The installation process includes the following steps: (1) clone repository, (2) set up virtual environment, (3) install requirements, (4) configure CoreNLP tokeniser, (5) install Document Retriever, (6) install Document Reader, (7) install Wikipedia snapshot, (8) download evaluation datasets. We provide a Docker image[5] to set up DrQA with a single command. Users can combine, train and test different tokenisers, Document Retrievers, and Document Readers on supplied datasets via the command line. DrQA supports the development of new Document Retriever and Document Reader models, accompanied by documentation.

[5] `docker exec -it bengt/drqa venv/bin/python scripts/pipeline/interactive.py`.

NeMo is a framework for building dialogue system applications through reusability, abstraction, and composition. Detailed installation instructions are provided with the software, and numerous Google Colab tutorials are available to help users get started. The question-answering tutorial for example provides step-by-step instructions from the installation to the usage of LLMs to extract or generate answers. Using the datasets SQuAD [49] and MS MARCO [43], the pre-trained LLMs are fine-tuned and evaluated. NeMo supports the creation of experiments and applications by using different models, varying hyperparameters, and implementing other components.

4.4 Large Language Models (LLMs)

While there is no formal definition, LLMs typically refer to pre-trained deep transformer models [62] with a vast number of parameters that can generate human-like language. They can perform a range of language-related tasks, including text summarisation, sentiment analysis and translation. Training LLMs from scratch is costly due to the vast number of parameters and the large text datasets, so it is primarily suitable for large companies. Figure 2 shows recently developed LLMs along with their model size and developer. To get a practical insight into the performance of LLMs in the field of dialogue systems, we suggest ChatGPT[6], which is free to use, although it requires users to provide an email address and a phone number. LLM research currently focuses on multi-modality: Google devel-

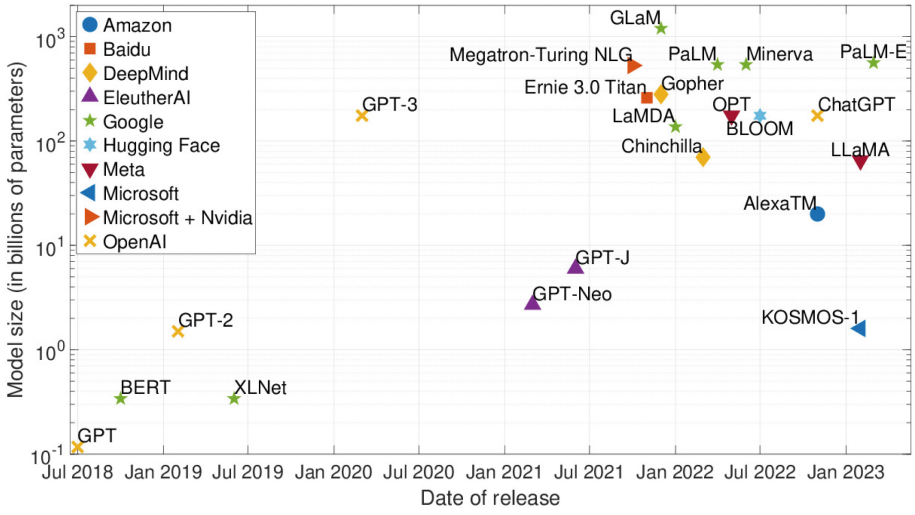

Fig. 2. Comparison of selected large language models developed since 2018, with their model size and developer. GPT-4, released in March 2023, was excluded due to an unknown number of parameters.

[6] https://openai.com/blog/chatgpt.

oped PaLM-E [12] by integrating vision transformers with PaLM, and OpenAI extended GPT-3 to include vision data processing in GPT-4 [46].

5 Conclusion and Future Work

The aim of this paper is to provide researchers with easy access to the field of dialogue systems. We presented a structured literature review and provided a list of relevant papers, articles and books covering all relevant technical topics. The main findings of our literature review have been presented through the clarification of terminology and the derived categories of objective, modality, domain, architecture and model and their main correlations. To facilitate practical entry into dialogue systems research, we have described the four main application and framework categories and recommended dialogue systems for each category: Cerence Studio as a commercial framework, DrQA and NeMo as research dialogue systems, and ChatGPT as an large language model. For intelligent virtual assistants, our recommendation depends on the device used.

Future work includes extending the structured literature review to other knowledge bases and extending the search terms to include synonyms identified in the literature. In addition, a performance comparison based on a benchmark dataset will allow a more accurate comparison of the dialogue systems.

References

1. Balaraman, V., et al.: Recent neural methods on dialogue state tracking for task-oriented dialogue systems: a survey. In: SIGdial. pp. 239–251. ACL (2021)
2. Bordes, A., et al.: Learning end-to-end goal-oriented dialog. In: ICLR. OpenReview.net (2017)
3. Bruss, M., Pfalzgraf, A.: Proaktive assistenzfunktionen für hmis durch künstliche intelligenz. ATZ Automobiltechnische Zeitschrift **118**, 42–47 (2016)
4. Chen, D., et al.: Reading Wikipedia to answer open-domain questions. In: ACL, pp. 1870–1879. ACL (2017)
5. Chen, H., et al.: A survey on dialogue systems: Recent advances and new frontiers. SIGKDD Explor. **19**(2), 25–35 (2017)
6. Cui, F., et al.: A survey on learning-based approaches for modeling and classification of human-machine dialog systems. IEEE Trans. Neural Netw. Learn. Syst. **32**(4), 1418–1432 (2021)
7. Curry, A.C., et al.: A review of evaluation techniques for social dialogue systems. In: SIGCHI, pp. 25–26. ACM (2017)
8. Deng, L., Liu, Y.: Deep Learning in Natural Language Processing. Springer, Singapore (2018). https://doi.org/10.1007/978-981-10-5209-5
9. Deriu, J., et al.: Survey on evaluation methods for dialogue systems. Artif. Intell. Rev. **54**(1), 755–810 (2021)
10. Devlin, J., et al.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT, pp. 4171–4186. ACL (2019)
11. Dinan, E., et al.: The second conversational intelligence challenge (convai2). CoRR abs/1902.00098 (2019)

12. Driess, D., et al.: PaLM-E: an embodied multimodal language model. CoRR abs/2303.03378 (2023)
13. Fader, A., et al.: Paraphrase-driven learning for open question answering. In: ACL, pp. 1608–1618. ACL (2013)
14. Fan, Y., Luo, X.: A survey of dialogue system evaluation. In: 32nd IEEE, ICTAI, pp. 1202–1209. IEEE (2020)
15. Fan, Y., et al.: MatchZoo: a toolkit for deep text matching. CoRR abs/1707.07270 (2017)
16. Henderson, J., et al.: Hybrid reinforcement/supervised learning of dialogue policies from fixed data sets. Comput. Linguist. **34**(4), 487–511 (2008)
17. Henderson, M., et al.: The second dialog state tracking challenge. In: SIGDIAL, pp. 263–272 (2014)
18. Henderson, M., et al.: The third dialog state tracking challenge. In: SLT, pp. 324–329. IEEE (2014)
19. Hu, J., et al.: SAS: dialogue state tracking via slot attention and slot information sharing. In: ACL, pp. 6366–6375. ACL (2020)
20. Huang, M., et al.: Challenges in building intelligent open-domain dialog systems. ACM Trans. Inf. Syst. **38**(3), 21:1-21:32 (2020)
21. Huang, P., et al.: Learning deep structured semantic models for web search using clickthrough data. In: ACM, pp. 2333–2338. ACM (2013)
22. Kim, S., et al.: Efficient dialogue state tracking by selectively overwriting memory. In: ACL, pp. 567–582. ACL (2020)
23. Koller, A., et al.: DialogOS: simple and extensible dialogue modeling. In: Interspeech, pp. 167–168. ISCA (2018)
24. Kreyssig, F., et al.: Neural user simulation for corpus-based policy optimisation of spoken dialogue systems. In: SIGdial, pp. 60–69. ACL (2018)
25. Kuchaiev, O., et al.: Nemo: a toolkit for building AI applications using neural modules. CoRR abs/1909.09577 (2019)
26. Le, H., et al.: Uniconv: a unified conversational neural architecture for multi-domain task-oriented dialogues. In: EMNLP, pp. 1860–1877. ACL (2020)
27. Lee, S., et al.: ConvLab: multi-domain end-to-end dialog system platform. In: ACL, pp. 64–69. ACL (2019)
28. Li, J., et al.: Adversarial learning for neural dialogue generation. In: EMNLP, pp. 2157–2169. ACL (2017)
29. Li, X., et al.: A review of quality assurance research of dialogue systems. In: AITest, pp. 87–94. IEEE (2022)
30. Liu, G., et al.: A survey on multimodal dialogue systems: recent advances and new frontiers. In: AEMCSE, pp. 845–853 (2022)
31. Liu, J., et al.: Review of intent detection methods in the human-machine dialogue system. J. Phys. Conf. Ser. **1267**(1), 012059 (2019)
32. Liu, T.: Learning to rank for information retrieval. In: SIGIR, p. 904. ACM (2010)
33. Lowe, R., et al.: Towards an automatic turing test: learning to evaluate dialogue responses. In: ACL, pp. 1116–1126. ACL (2017)
34. Lu, Z., Li, H.: A deep architecture for matching short texts. In: NeurIPS, pp. 1367–1375 (2013)
35. Ma, L., et al.: Unstructured text enhanced open-domain dialogue system: a systematic survey. ACM Trans. Inf. Syst. **40**(1), 9:1-9:44 (2022)
36. Ma, Y., et al.: A survey on empathetic dialogue systems. Inf. Fus. **64**, 50–70 (2020)
37. Malik, M., et al.: Automatic speech recognition: a survey. Multim. Tools Appl. **80**(6), 9411–9457 (2021)

38. Michael, T.: ReTiCo: an incremental framework for spoken dialogue systems. In: SIGdial, pp. 49–52. ACL (2020)
39. Michael, T., Möller, S.: ReTiCo: an open-source framework for modeling real-time conversations in spoken dialogue systems. In: ESSV, pp. 134–140 (2019)
40. Miller, A.H., et al.: ParlAI: a dialog research software platform. In: EMNLP, pp. 79–84. ACL (2017)
41. Motger, Q., et al.: Software-based dialogue systems: survey, taxonomy, and challenges. ACM Comput. Surv. **55**(5), 1–42 (2022)
42. Nesselrath, R., Feld, M.: SiAM-dp: a platform for the model-based development of context-aware multimodal dialogue applications. In: IE, pp. 162–169. IEEE (2014)
43. Nguyen, T., et al.: MS MARCO: a human generated machine reading comprehension dataset. In: NeurIPS, vol. 1773. CEUR-WS.org (2016)
44. Ni, J., et al.: Recent advances in deep learning based dialogue systems: a systematic survey. CoRR abs/2105.04387 (2021)
45. Obrenovic, Z., Starcevic, D.: Modeling multimodal human-computer interaction. Computer **37**(9), 65–72 (2004)
46. OpenAI: GPT-4 technical report. CoRR abs/2303.08774 (2023)
47. Papangelis, A., et al.: Plato dialogue system: a flexible conversational AI research platform. CoRR abs/2001.06463 (2020)
48. Paul, Z.: Cortana-intelligent personal digital assistant: a review. Int. J. Adv. Res. Comput. Sci. **8**, 55–57 (2017)
49. Rajpurkar, P., et al.: SQuAD: 100,000+ questions for machine comprehension of text. In: EMNLP, pp. 2383–2392. ACL (2016)
50. Reiter, E.: Has a consensus NL generation architecture appeared, and is it psycholinguistically plausible? In: INLG (1994)
51. Schatzmann, J., et al.: Agenda-based user simulation for bootstrapping a POMDP dialogue system. In: NAACL HLT, pp. 149–152. ACL (2007)
52. Seo, M.J., et al.: Real-time open-domain question answering with dense-sparse phrase index. In: ACL, pp. 4430–4441. ACL (2019)
53. Serban, I.V., et al.: A hierarchical latent variable encoder-decoder model for generating dialogues. In: AAAI, pp. 3295–3301. AAAI Press (2017)
54. Serban, I.V., et al.: A survey of available corpora for building data-driven dialogue systems: the journal version. Dialogue Discourse **9**(1), 1–49 (2018)
55. Shang, L., et al.: Neural responding machine for short-text conversation. In: ACL, pp. 1577–1586. ACL (2015)
56. Sonntag, D.: Ontologies and Adaptivity in Dialogue for Question Answering, Studies on the Semantic Web, vol. 4. IOS Press (2010)
57. Sordoni, A., et al.: A neural network approach to context-sensitive generation of conversational responses. In: NAACL HLT, pp. 196–205. ACL (2015)
58. Sutskever, I., et al.: Sequence to sequence learning with neural networks. In: NeurIPS, pp. 3104–3112 (2014)
59. Tan, X., et al.: A survey on neural speech synthesis. CoRR abs/2106.15561 (2021)
60. Tran, V.-K., Nguyen, L.-M.: Semantic refinement GRU-based neural language generation for spoken dialogue systems. In: Hasida, K., Pa, W.P. (eds.) PACLING 2017. CCIS, vol. 781, pp. 63–75. Springer, Singapore (2018). https://doi.org/10.1007/978-981-10-8438-6_6
61. Ultes, S., et al.: PyDial: a multi-domain statistical dialogue system toolkit. In: ACL, pp. 73–78. ACL (2017)
62. Vaswani, A., et al.: Attention is all you need. In: NeurIPS, pp. 5998–6008 (2017)
63. Vinyals, O., Le, Q.: A neural conversational model. CoRR abs/1506.05869 (2015)

64. Walker, M.A., et al.: PARADISE: a framework for evaluating spoken dialogue agents. In: ACL, pp. 271–280. ACL (1997)
65. Wang, S., et al.: R^3: reinforced ranker-reader for open-domain question answering. In: AAAI, pp. 5981–5988. AAAI Press (2018)
66. Wang, Y., et al.: Slot attention with value normalization for multi-domain dialogue state tracking. In: EMNLP, pp. 3019–3028. ACL (2020)
67. Weizenbaum, J.: ELIZA - a computer program for the study of natural language communication between man and machine. Commun. ACM **9**(1), 36–45 (1966)
68. Wen, T., et al.: Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking. In: SIGDIAL, pp. 275–284. ACL (2015)
69. Weston, J., et al.: Retrieve and refine: improved sequence generation models for dialogue. In: SCAI, pp. 87–92. ACL (2018)
70. Williams, J.D., et al.: The dialog state tracking challenge. In: SIGDIAL, pp. 404–413. ACL (2013)
71. Wolf, T., et al.: TransferTransfo: a transfer learning approach for neural network based conversational agents. CoRR abs/1901.08149 (2019)
72. Xu, J., et al.: Diversity-promoting GAN: a cross-entropy based generative adversarial network for diversified text generation. In: EMNLP, pp. 3940–3949. ACL (2018)
73. Yang, L., et al.: A hybrid retrieval-generation neural conversation model. In: CIKM, pp. 1341–1350. ACM (2019)
74. Yang, Z., et al.: XLNet: generalized autoregressive pretraining for language understanding. In: NeurIPS, pp. 5754–5764 (2019)
75. Zhang, Y., et al.: DIALOGPT: large-scale generative pre-training for conversational response generation. In: ACL, pp. 270–278. ACL (2020)
76. Zhao, T., Eskénazi, M.: Towards end-to-end learning for dialog state tracking and management using deep reinforcement learning. In: SIGDIAL, pp. 1–10. ACL (2016)
77. Zhao, T., et al.: Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In: ACL, pp. 654–664. ACL (2017)
78. Zhou, H., et al.: Context-aware natural language generation for spoken dialogue systems. In: COLING, pp. 2032–2041. ACL (2016)
79. Zhou, L., et al.: The design and implementation of xiaoice, an empathetic social chatbot. Comput. Linguist. **46**(1), 53–93 (2020)