

Interactive Question Answering Systems: Literature Review

GIOVANNI MARIA BIANCOFIORE*, YASHAR DELDJOO, TOMMASO DI NOIA, EUGENIO DI SCIASCIO, and FEDELUCIO NARDUCCI, Polytechnic University of Bari

Question answering systems are recognized as popular and frequently effective means of information seeking on the web. In such systems, information seekers can receive a concise response to their query by presenting their questions in natural language. Interactive question answering is a recently proposed and increasingly popular solution that resides at the intersection of *question answering* and *dialogue systems*. On the one hand, the user can ask questions in normal language and locate the actual response to her inquiry; on the other hand, the system can prolong the question-answering session into a dialogue if there are multiple probable replies, very few, or ambiguities in the initial request. By permitting the user to ask more questions, interactive question answering enables users to dynamically interact with the system and receive more precise results.

This survey offers a detailed overview of the interactive question-answering methods that are prevalent in current literature. It begins by explaining the foundational principles of question-answering systems, hence defining new notations and taxonomies to combine all identified works inside a unified framework. The reviewed published work on interactive question-answering systems is then presented and examined in terms of its proposed methodology, evaluation approaches, and dataset/application domain. We also describe trends surrounding specific tasks and issues raised by the community, so shedding light on the future interests of scholars. Our work is further supported by a GitHub page with a synthesis of all the major topics covered in this literature study. <https://sisinflab.github.io/interactive-question-answering-systems-survey/>

Additional Key Words and Phrases: Question Answering, Natural Language Processing, Interactive Systems, Human Computer Interaction, Artificial Intelligence

ACM Reference Format:

Giovanni Maria Biancofiore, Yashar Deldjoo, Tommaso Di Noia, Eugenio Di Sciascio, and Fedelucio Narducci. 2022. Interactive Question Answering Systems: Literature Review. In *Under Review, July 2022*. ACM, New York, NY, USA, 35 pages. <https://doi.org/xxxx.xxxx>

1 INTRODUCTION

Motivation. In the early literature, question-answering systems (QASs) were frequently contrasted to Search Engines (SEs). Their primary distinction was that the latter returns a ranked (often lengthy) list of documents, whereas the former produces an *answer* to the question posed by the user [42]. Today, the boundary between SEs and QASs is becoming increasingly blurred [29]. When we query Google seeking for “President of the United States of America”, we no longer receive a list of results in which we must search for the answer, but rather a concise snippet that contains the answer to our question (extracted from a web page).

Essentially, QASs share the characteristic to have the ability to provide a clear answer to the user inquiry, regardless of the type of question, i.e., factual (“Which kingdom does the animal Bird of Paradise belong to?”) [71], visual (“What color hair does the woman have?” combined with a picture of a woman) [112], or open-goal oriented (“How can I connect

*Corresponding author

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2022 Association for Computing Machinery.

Manuscript submitted to ACM

my fitbit sensors to the server?”) [39]. The answer can be extracted from a portion of a document (as in the preceding example) or generated by condensing/summarizing fragments of many words into a coherent whole [42]. *Extraction* and *generation* constitute the core building blocks of a QA algorithm.

The QA task progresses from the earliest systems characterized by one-shot requests, through *interactive question answering* (IQA), which allows users to continue interacting with the system after receiving an answer. In some recent conversational question answering, a multi-turn dialog is employed to determine the correct answer. In this direction, the great diffusion of virtual digital assistants (VDAs) such as Amazon Alexa, Google Assistant, Microsoft Cortana, or Apple Siri is playing a crucial role [76]. In fact, one of the principal tasks these VDAs are asked to perform is just to answer questions in several fields: health, weather, news, shopping [8], just to cite a few of them. Accordingly, the *interaction* between humans and QASs is becoming more and more natural. In this study, we use the term *interaction* to refer to any type of feedback or reaction/response from the user or system, not only the standard exchange of questions and answers seen in conversational question answering systems (CoQASs).

Interactive QA systems (IQASs) are a solution to several problems raised by QASs in satisfying the user requests. One of the core problem is related to the *disambiguation* of the user question. In fact, the user request is not always straightforwardly understandable by the system, and part of the interaction can be devoted to the disambiguation or, more generally, clarification of the user request. This scenario does not affect only conversational systems, but interactive ones in general. If we query Google with “When was Milan founded?”, the answer provided is related to the “Milan football club”. However, the related questions proposed by Google also contain “When the city of Milan was founded?”, since *Milan* is an ambiguous name. In case the user did not mean *Milan* as the football club, but as the city, she can click on the corresponding question and get the right answer.

Thus, the watershed between *non-interactive* and *interactive* QASs will be the capability of the latter to continue the interaction after the original query for *disambiguation* or *exploration* purposes. With regard to the exploration, in a system-driven scenario, the system proposes a set of other questions related to the topic (e.g., Milan, football clubs, etc.) and the user selects one of them. Conversely, in a user-driven scenario, the user can ask the system with further questions after receiving answers.

Undoubtedly, as highlighted by Guo et al. [37], between IQAS and CoQAS, the latter is the most challenging. In fact, in addition to the general problems of an IQA task, there are some peculiar issues related to the natural language processing. For example, these systems should be able to cope with complex problems such as the ellipsis phenomena (e.g., “Where was the President of the United States born?”, “Where did **he** graduated from?”) that characterizes dialogues between human beings. For this reason, a CoQAS has to keep track of the interaction *state*. This introduces an additional analysis dimension, exploited in this survey, related to the *stateful* or *stateless* characteristic of these systems.

Contributions. In this survey, we analyze QASs from the above mentioned perspectives by highlighting the different solutions proposed in the literature. We focused on work published over the last 10 years to provide a detailed picture of IQASs.

The principal contributions of this surveys are manifold:

- We provide formal definitions describing the purposes for which QASs are implemented in the literature and further state the primary characteristics that IQASs should possess (cf. Section 2);
- We present a unified and encompassing architecture highlighting the primary components of IQASs and their operation (cf. Section 3);

- We conduct an in-depth classification of state-of-the-art from the methodological and application perspectives (cf. Section 3, 4);
- We give detailed sights of the most frequently used datasets, as well as adopted evaluation protocols and metrics, organized by the paradigms and objectives of IQASs (cf. Section 4);

The rest of the survey is organized as follows: In Section 2 we outline several formal definitions about different kinds of QASs that establish the state of the art. For each of them we also provide detailed descriptions and useful examples, then resuming advances, challenges, and tasks. In Section 3 we go through methods and techniques of QASs, designing a general architecture that best represent all options discussed in this work. We describe different component operations via examples taken from state of the art, thus their interactions for solving the QA task. In Section 4, evaluation protocols and metrics are outlined according to tasks and challenges targeted by QASs. We provide a further classification of papers from metric and dataset perspectives. Finally, in Section 5 we illustrate the reached conclusions.

1.1 Strategy for Literature Search

To identify relevant publications for this survey, we adopted a multi-level search strategy. We started by finding relevant research works from top conferences in the fields of QA and *knowledge management*, i.e. EMNLP, SIGIR, CIKM, KDD, ESWC, and WSDM respectively, and collected a majority of the publications that address topics of interest. Having in mind that many other venues on information retrieval and natural language processing also publish relevant works, we also gathered a number of related publications by searching on Google Scholar and filtering for the name of the following venues: ACL, ACM, IEEE, AAAI, and Neurocomputing. Lastly, we performed some free search in Google Scholar and obtained a number of papers from other venues. We focused on conference proceedings and journals and to a much lesser extent on workshop publications. A common characteristic among these searches is the stemmed keywords set used to collect papers within the scope of this survey, which are *interact question answer*, *explain question answer*, *interact natural language interface*, *bert question answer*, *question answer conversat* and *question answer clarif*.

After having read the publications identified as explained above, we analyzed their bibliographies, and sometimes we used the "Related Work" option in Google Scholar. This way, we obtained a second list of publications. The venues of these publications were not limited to the ones mentioned above. While we are quite sure that we did not find *all* publications that address Question Answering problems by leveraging interactive processes, we are confident that we identified a relevant portion of related publications. For the sake of reproducibility, the comprehensive list of works focused on in this survey is provided on our github¹.

2 QUESTION ANSWERING APPROACHES

The focus of this survey is on IQASs, which can be seen as a multidisciplinary topic combining several fields such as information retrieval (IR), natural language processing (NLP), human computer interaction (HCI) and, more recently, artificial intelligence (AI), machine learning (ML) and knowledge management (KM). Thanks to the latest developments in NLP and ML areas, QASs have found a remarkable and growing interest from the AI community, especially for the role that they play for digital assistants (e.g., Google Assistant, Amazon Alexa, Apple Siri). QASs have usually been placed within the macro area of information retrieval systems (IRSs) because of their apparently similar function, enough to consider them as a sophisticated form of IRSs [32]. Nowadays, QASs have evolved to gain their own field of

¹<https://sisinlab.github.io/interactive-question-answering-systems-survey/>

study, whose goals continually increase by including new research topics like knowledge representation and semantic entailment.

In this section, we present the formal definition of IQA, including the QA problem (cf. Definition 2.1), two configurations of IQASs (cf. Definition 2.5, Definition 2.6) and CoQASs (cf. Definition 2.10). Then, we provide definitions of interactive session (cf. Definition 2.7), QA state (cf. Definition 2.8) and conversational history (cf. Definition 2.9), and different examples/pointers to state-of-the-art solutions that would help the reader to obtain a profound understanding of the topic.

Assumption. In information-access systems the user’s information need could be expressed through *keywords* (retrieval), a *question* (question-answering), or *user profile* (recommender system), while answers could be a piece of text, an image [? ?], or items of interest, most relevant to the information need. Systems such as task-oriented chatbots or dialog systems whose primary role is not providing information access remain outside the focus of this survey.

In order to highlight motivations behind this assumption, we consider the following definition.

Definition 2.1 (QA problem). Let $Q = \{q_1, \dots, q_N\}$ be the set of possible queries and $A = \{a_1, \dots, a_M\} \cup \{\text{NULL}\}$ the set of possible answers including the NULL symbol representing the situation where the system is not able to provide an answer. A QA system aims to find the most relevant answer to a given question in a single shot iteration. More formally:

$$\forall q \in Q, \hat{a} = \underset{a \in A}{\operatorname{argmax}} g(q, a)$$

with g being a utility function that considers how well a given answer a satisfies the proposed query. In probabilistic terms, we can define the problem as finding the most probable answer given an input question (and its context)

$$\forall q \in Q, \hat{a} = \underset{a \in A}{\operatorname{argmax}} p(a|q)$$

Users may interact with QASs for finding the information they need. A QAS will answer the question with a unique result well-formed in natural language, like: *"The Lord of the Rings' writer is J.R.R. Tolkien"*, saving the user to search for the needed information.

2.1 Interactive Query Answering as Exploration and Disambiguation

Finding the correct answer for a given question is the primary goal to be achieved by QASs. Although most of the QASs show high performance for this task, some intrinsic natural language issues cannot be solved by only analyzing the submitted question. The definition of a well-disambiguated request through a natural language question is not a trivial task. Indeed, it requires the usage of specific terms plus a cognitive effort that is not affordable for everyone. Hazrinaa et al. [40] examine the semantic QA task, which permits disambiguation of queries by leveraging context or by requesting further information from the user.

An extension of QASs, named **Interactive Question-Answering** systems, overcome this limitation with two specific goals:

Disambiguation. In the case there are *too many* or *too few* eligible answers for a given question, or there is ambiguity in the request, the system can ask a new question to the user [55]. For this reason, IQASs can be seen halfway between QASs and *dialog systems*. Let us consider the case when the original query q has a set $A' \subseteq A$ of candidate answers. The system can suggest a new query $q' \in Q$. The user answer a' to q' gives the information that leads to an unambiguous answer $\hat{a} \in A'$ to the previous question q . We can say that \hat{a} is an answer to the combination of q and q' .

Exploration. After the system returned the answer \hat{a} to the initial query q , a new set of queries Q' can be suggested by the system or posed by the user² in order to explore other relevant topics related to \hat{a} .

As noted, interactivity refers to the possibility of the system to pose/suggest new queries to the user. In both cases, when the system suggests a query, in principle, it might have already found one or more answers to the user request, thus resulting in starting up a disambiguation and exploratory step. Please note that while the disambiguation step is always **system driven**, exploration can also be **user driven**. In the latter case, the user decides the new aspects to explore related to \hat{a} .

Disambiguation and exploration may run for one step only or for a sequence of steps. Depending on the "memory" the QAS has about previous questions and answers, we may have a **stateless** QAS or a **stateful** one. This latter situation leads to what we call **Conversational Question-Answering** system. It is worth noting that during a conversation, in *system driven* interactions, we only have sequences of disambiguation or exploration steps. We do not have situations where the two steps are interleaved unless the systems allow a *user-driven* interaction.

In a conversation aiming to disambiguate the original query q , we may have situations where the answer a' to q' does not lead to \hat{a} . In these cases, the system computes a new query q'' based on a' and so on until \hat{a} is finally returned. It does not result useful to have exploratory steps while trying to disambiguate q in order to compute \hat{a} .

Example 2.2 (QA for disambiguation). Let us consider the case in which a user needs some information about music tracks, so she asks "Who sang the song Money". This question results ambiguous since the answer could refer to both the group *Pink Floyd* and the musician *David Gilmour*. In this case, the system replies with a question trying to disambiguate the user information need. It searches relevant information from the set of possible answers to reach this goal. As a consequence, it will pose disambiguation questions until it reaches what the user meant. In this example, the system asks "Do you mean the band who sang Money?" and the user agrees, receiving then the final answer "Pink Floyd".

It is worth noting that in the Example 2.2 the choice of which disambiguating question to pose the user merely depends on the implementation of the system. In fact, asking for "Do you mean the musician who sang Money?" leads to the same answer *Pink Floyd* once the user feedback is given (i.e. she disagrees). That is because Example 2.2 has a binary mutually exclusive ambiguity. In general, approaches for choosing disambiguating questions from a set that may be large depend on whether the system allows optimized interactions.

Differently from disambiguation, two scenarios are possible in an exploratory conversation. The queries may be computed by the QAS (system-driven) or the user (user-driven). In a system-driven scenario, new questions are proposed for which the answer is already known. Thus, disambiguation steps become useless.

Example 2.3 (QA for system driven exploration). Following the Example 2.2, once the final answer is reached, a QAS could start an exploratory session within the conversation. It could propose questions like "Do you know when Pink Floyd were founded?" or "Do you know when David Gilmour joined Pink Floyd?" and then provide the related data depending on the user answers. For instance, the user may know when the band was founded but she does not the answer to the second question, so the system will reply: "He joined the Pink Floyd in 1968 as a support to Syd Barrett".

On the other side, in a user driven scenario, the new queries are posed by the user, which may lead to ambiguous answers. In those cases, the system needs to start disambiguation steps to reach the relevant answer.

² Q' is also referred to as *follow-up questions* in the literature.

Example 2.4 (IQA for user driven exploration). Going back to the conversation in Example 2.3, the user may continue the dialogue by asking: "When was he born?". Here we can refer to both *David Gilmour* and *Syd Barret*, so the system will ask "Do you mean Syd Barret?". As a consequence, the user may agree with the the system or not. In the latter case, the QAS answers with "David Gilmour was born on 6 March 1946". The conversation will lasts until the user information need is totally satisfied.

With respect to *stateful* exploratory QAS it results useful to avoid interaction loops. In principle, the exploration of the knowledge space may run forever. In fact, in case we do not consider previous interactions and answers by users, the exploration may get stuck in a loop where the system computes questions already suggested in the previous steps.

Before we give a formal definition of the different interaction steps we have discussed so far, we introduce two unary operators to represent the possible actions an IQAS or a user may perform, e.g. generating a new query or an answer. We use $y \in \{u, s\}$ to state if the next query or the answer has been generated by the user (u) or by the system (s). Given $x \in A \cup Q$, $y \in \{u(ser), s(ystem)\}$, $a \in A$ and $q \in Q$ we will use:

- $x \xrightarrow{u} q$: the IQAS ($y = s$) or the user ($y = u$) generates a new query.
- $x \xrightarrow{y} a$: the IQAS ($y = s$) or the user ($y = u$) generates an answer.

Given a query q , we then represent the simple query-answering step as $q \xrightarrow{s} \hat{a}$. Analogously, for the exploratory IQA we have $q \xrightarrow{s} \hat{a} \xrightarrow{s} q'$ while the disambiguating Interactive QA steps are represented with $q \xrightarrow{s} q' \xrightarrow{u} a' \xrightarrow{s} \hat{a}$. In this case, a' is the answer (e.g. feedback) provided by the user to the query q' generated by the system.

Definition 2.5 (Interactive Question Answering for Disambiguation). An interactive question answering for disambiguation **IQAD** is a system that takes a user query q as input and computes a new query q' to reach the answer \hat{a} to q . More formally, we have

$$\hat{a} = \operatorname{argmax}_{a \in A, q' \in Q} p(q'|q), p(a|q \xrightarrow{s} q' \xrightarrow{u} a')$$

The idea here is to find the best next question q' given the initial query q in order to compute the best answer to $q \xrightarrow{s} q' \xrightarrow{u} a$. As for $p(q'|q)$ and $p(a|q \xrightarrow{s} q' \xrightarrow{u} a')$, in principle, we do not make any assumption on their (in)dependency.

Figure 1 depicts an example of IQAS. The question asked to the interactive system is: "Who is the president of America?". As already highlighted in the example, the user obtains an answer that satisfies her query plus some more suggestions useful to explore the related exploration space. Here the picture outlines a collection of information and interactive area that allow users to continue her exploratory interaction.

The first information shown by Google in the example is the answer "Joe Biden", which includes the entities and relations of a knowledge graph recognized in the question, respectively "United States" and "President". The second one proposes a set of follow-up queries that could come after the initial question. For instance, other people searched for "Who was the previous president of America?" that was "Donald Trump", or "Who is the vice-president of America?" that is "Kamala Harris". In this way, the user may reach all this data by simply interacting with the suggested images, enriching the solution to her information need. The same area also groups different clickable questions asked by people to achieve the same above correlated answers. In a way, it gives information about the context understood by the system. The user who obtains an unexpected answer could rephrase her question to disambiguate its sense intention/meaning and then receive the appropriate response from the system.

Furthermore, the system provides a suite of possible multimodal interactions, like images, audio and text, which aim at improving both the user experience and the correctness of the results. A feedback slot allows people to further

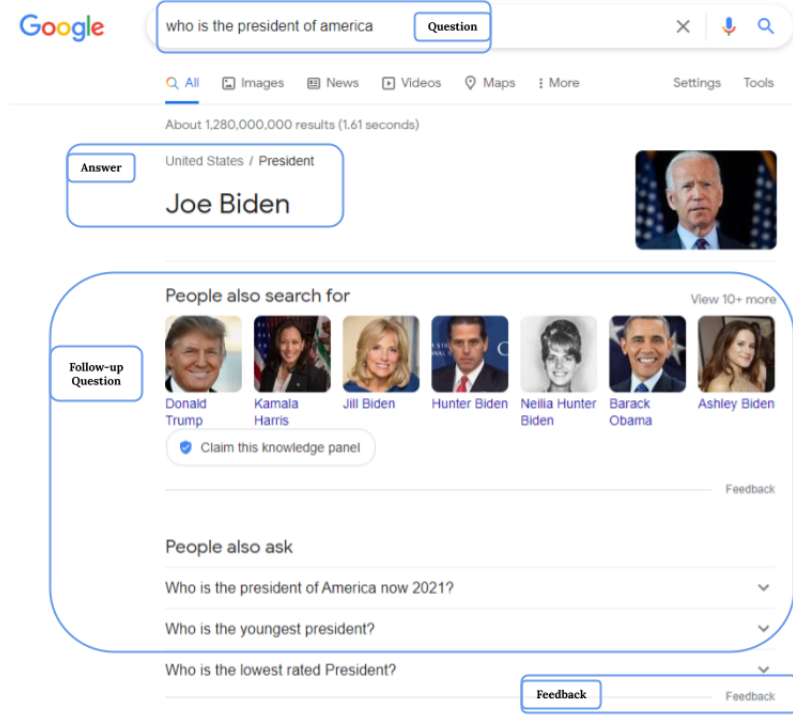


Fig. 1. Example of an interactive QAs answer.

interact with the system in offering comments about the answer. Whether the solution is wrong, the user can specify the motivations and send them to the system, which will improve its behaviour about that type of question.

Definition 2.6 (Interactive Query Answering for Exploration). An interactive question answering system for exploration IQA^e aims at guiding the user through the exploration of a knowledge space. Given a question q , it provides both an answer and a set of follow-up questions closed to q . More formally, we have:

$$\langle \hat{a}, \hat{q}' \rangle = \operatorname{argmax}_{q' \in Q, a \in A} p(a|q), p(q'|a)$$

The aim of an IQA^e is that of finding an answer to q and, at the same time, suggesting a new query q' that is related to the computed answer. Without loss of generality, in the previous definition we consider only one follow-up question q' . The definition can be easily extended to a set Q' of follow-up questions. Also in this case, we do not make any assumption on the (in)dependency between $p(q'|a)$ and $p(a|q)$. This is something that, in case, is considered in the implementation of the IQA^e .

Therefore, we may surely have also the hybrid situation where the exploration is supported by a disambiguation step:

$$q \xrightarrow{s} q' \xrightarrow{u} a' \xrightarrow{s} \hat{a} \xrightarrow{s} q''$$

where the initial query is disambiguated before the answer \hat{a} and the next question q'' are computed. In a nutshell, an IQAS allows the user to further interact with the system, once its reply is given, via **interactive sessions**.

Definition 2.7 (Interactive Session). A set of user enabled actions following the system reply to a previous user question defines an *interactive session*:

$$I = \{a', q' : a \xrightarrow{u} q', q \xrightarrow{u} a'\}$$

The *interactive session* is always supported by the IQAS to users. The cardinality of this set determines which actor leads the interaction, while its elements define the interaction type (i.e. exploratory or disambiguating). An *interactive session* with infinite cardinality remarks the interaction leads by the user. This is the case of some IQASs for exploration where users are free to pose any question to the system response. In contrast, a finite *interactive session* cardinality outlines interactions conducted by the system. Therefore, a bounded number of actions (i.e. answers to disambiguating questions or enabled exploratory questions) are proposed by the IQAS to the user. Here, the interaction takes place when the user gives her feedback to the *interactive session* (e.g. answering with a disambiguating information).

2.2 Conversational Query Answering

The interactive systems shown in Figure 1 does not keep any memory of previous interactions with the user. When the user explores the knowledge space, every single exploration step does not consider previous ones. However, the system needs a set of information to understand the user *intents* behind each question. With **context** we refer to that data helping the system in understanding the meaning intended by the user behind her questions. Thus, the context is the set of data C which supports the QAS in finding the answer a to the user question q . Nevertheless, the overall process can be seen as a *stateless* application of a sequence of IQA^e . Then, the *state* can be formalized as follows:

Definition 2.8 (QA State). Let q be the user question, C a *context* supporting q , a the system answer and I the following *interactive session* enabled to the user. The *QA state* is the tuple that collects all the information of a QA interaction:

$$S_{QA} = \langle q, C, a, I \rangle$$

In other words, the *QA state* host all the data that are exchanged through a user-system interaction. Thus, it is totally outlined at the end of each interaction.

The QA state context C determines the *statefulness* property of an IQAS. In fact, *stateless* IQASs host supporting information in their *QA state context* C that does not belong to other *QA states*. This can also lead to loops in the exploratory task. On the other hand, in case the system had memory of previous QA states through the *context*, it could avoid to propose questions whose answer has been already visited in the past by the user. Analogously, an IQA^d can only consider the original question q and the answer given by the user to q' . In general, IQA^d and IQA^e are *stateless* and do not take into account the **history** referring to previous interactions between the user and the system.

In case the IQA model had access to the search history of the user, it could grant new rounds of interaction helpful, e.g., in handling some linguistic issues like the *Coreference Resolution* problem. The *coreference resolution* is the task of finding all expressions that refer to the same entity in a text [19], which frequently appears by exchanging follow-up questions with the system. For example, assuming that the first user query is "Who wrote the Lord of Rings?", a follow-up question could be "When **he** wrote **it**?", where "he" refers to the answer "J.R.R. Tolkien" and "it" to the entity "Lord of the Rings". In these situations, the system must know what the user asked in the past and the given answers to solve this issue, thus moving from a *stateless* configuration to a *stateful* one. The *stateful* configuration of an IQAS is also known in the literature as *conversational QA system* due to its ability to treat conversations having dialogues with the user.

In CoQASs we can have both user driven and system driven exploratory interactions as well as their combination. We will see later that, in these systems, disambiguation interactions occur as intermediate steps in a user-driven exploration.

The previous example related to *"The Lord of the Rings"* is a clear user driven exploratory interaction. Indeed, thanks to the *coreference resolution* and the corresponding *conversational context* (c.f. Section 3) computed by the CoQAS, the user is allowed to explore the knowledge space related to their original query q . Analogously, a CoQAS can be used in a system driven disambiguation process in case the answer to q' is not satisfactory to provide an answer to q . It is worth noticing that in a *system driven* CoQAS for *exploration* we will never need any disambiguation step. In fact, the new queries are suggested by the system which already knows the answers (cf. Example 2.3). This cannot be the case for *user driven exploration* via a CoQAS. Here, a new query posed by the user after some exploration steps may require a disambiguation process by the system, (cf. Example 2.4).

In a **System Driven Conversational Question Answering System for Exploration**, we have a sequence of exploratory steps in the form:

$$q \xrightarrow{s} a \rightsquigarrow q' \xrightarrow{s} a' \rightsquigarrow \dots \xrightarrow{s} a^n$$

where the transitions between the *exploratory* questions posed by the system and the related answers are interleaved by user feedback, in case she is knowledgeable or not on those topics.

Dually, in a **User Driven Conversational Question Answering System for Exploration** we obtain:

$$q \xrightarrow{s} a \rightsquigarrow q' \xrightarrow{s} a' \rightsquigarrow \dots \xrightarrow{s} a^n$$

Differently from IQA^e , in the *exploratory* case the system-generated query q^i at the i -th step also considers the two sets Q^i and A^i containing the previously generated queries $Q^i = \{q', \dots, q^{i-1}\}$ as well as their corresponding answers $A^i = \{a, a', \dots, a^{i-1}\}$. The same happens in the disambiguation case for the system-generated answer a^i .

Definition 2.9 (Conversation History and Conversation Span). Given a conversational sequence of exploratory or disambiguation steps we define **Conversation History** at step i as $h^i = Q^i \cup A^i$ where $Q^i = \{q', \dots, q^{i-1}\}$ and $A^i = \{a, a', \dots, a^{i-1}\}$. For a step l and a step i , with $l < i$ we define **Conversation Span** as $s^{l,i} = h^i - h^l$.

A conversation history contains all the user-driven and system-driven interactions up to a certain point of the conversation. They can be interpreted as the context for the next question to answer. A conversation span represents the conversation history from a certain step l up to a step i . We can see that $h^i = s^{0,i}$. So, what we will say for $s^{l,i}$ always holds also for h^i . Once again, we see that in a system-driven conversational exploration, at each step the system always generates the answer a^i to the query q^i and the next question to ask q^{i+1} . Also in this case, since the next query is generated by the system, it is always unambiguous. Hence, no disambiguation interaction is needed. This could not be the case for a user-driven conversational exploration. In fact, the user-generated query q^i may result ambiguous to the system and then require a further interaction for disambiguation. Here, we can generalize with respect to the definition of a IQA^d and assume multiple steps of interaction to disambiguate q^i in a situation like in the following:

$$q \xrightarrow{s} a \rightsquigarrow q' \xrightarrow{s} a' \rightsquigarrow \dots \rightsquigarrow q^i \rightsquigarrow \tilde{q}' \xrightarrow{s} \tilde{a}' \rightsquigarrow \tilde{q}'' \xrightarrow{s} \dots \rightsquigarrow \tilde{q}^k \xrightarrow{s} \tilde{a}^k \xrightarrow{s} a^i \rightsquigarrow q^{i+1} \xrightarrow{s} \dots \xrightarrow{s} a^n$$

$\underbrace{\hspace{10em}}_{\text{exploration}}$
 $\underbrace{\hspace{10em}}_{\text{disambiguation}}$
 $\underbrace{\hspace{10em}}_{\text{exploration}}$

where the conversational system is not able to disambiguate q^i given a conversational span $s^{l,i}$, and then it starts k multiple rounds of disambiguation steps until it gets enough information to compute the answer a^i . Then, the user driven interaction may continue to explore the addressed information space.

For the sake of presentation, we assume the systems does not have a bounded number m of queries to pose during the disambiguation phase. In case it can ask at most m queries to disambiguate, and after the m -th answer it is not able to compute an unambiguous answer, the system returns NULL and the overall conversation ends.

Definition 2.10 (Conversational Question Answering). A Conversational Question Answering system aims at exploring an information space alternatively via user-driven or system-driven interactions.

A **system-driven Conversational Question Answering system** $CoQA^s$ computes answers to the current query and the next question to ask, given a Conversation Span. More formally, we have:

$$\langle \hat{a}, \hat{q}^{i+1} \rangle = \operatorname{argmax}_{q^i \in Q, a^i \in A} p(q^{i+1} | a^i, s^{l,i+i}), p(a^i | q^i, s^{l,i})$$

In a **user driven Conversational Question Answering system** $CoQA^u$ we formally distinguish between the exploration and the disambiguation as:

$$\hat{a} = \operatorname{argmax}_{a^i \in A} p(a^i | q^i, s^{l,i}) \quad \text{and} \quad \hat{q} = \operatorname{argmax}_{q^i \in Q} p(q^i | s^{l,i})$$

Please note that in $CoQA^s$ the aim is to maximize both the probability that a certain answer satisfies the current query and the probability of the next question to pose. Conversely, in $CoQA^u$ we are only interested in computing the answer maximizing the probability that it satisfies the current query during exploration. On the other side, in a disambiguation step we are only interested in computing the next query to pose to the user in order to provide them an answer. In the following, we summarize the principal characteristics of all types of IQAS based on five dimensions that are usually found in the literature: *interactivity*, *statefulness*, *robustness*, *naturalness*, and *initiative*.

Statefulness is related to the capability of the system to keep track of the state of the interaction. Conversations are the most common method to obtain information and knowledge between two or more agents [69]. In a real-world dialogue, during the conversation there are high informative contextual relationships which bring new knowledge about a given topic to the interacting agents. As shown in the following example related to a $CoQA^u$, each conversation defines its dialogue context. A CoQAS stores the set of data derivable from the conversation until the end of the interaction with the user.

Table 1. An example of stateful Question-Answering Interaction

q	Who is the founder of Apple?
a	Apple has three founders: Steve Jobs, Ronald Wayne and Steve Wozniak.
q'	When was he born?
\tilde{q}'	Who do you refer to?
\tilde{a}'	Steve Jobs
a'	24 February 1955
q''	When was it founded?
a''	1 April 1976, Los Altos, California, United States
q'''	Which was the first launched product?
a'''	The company's first product is the Apple I, a computer designed and hand-built entirely by Wozniak.

The first question and the related answer provide a context about Apple and its founders, enabling users ask for data related to both the industry and the people. In the example, the system links the "it" pronoun of the second question with Apple, understanding that the user wants to know when Apple was founded. Moreover, thanks to the dialogue context, the CoQAS can get the implied subject of the third question, which is still Apple. In the previous example we consider a Conversation Span $s^{0,i} = h^i - h^0$ since the information needed to answer the

first question refers to the original question q . In a *stateless* IQAS, every information exchanged instead is not stored by the system, so the users need to ask detailed questions each time they seek new information, resulting in less natural interactions with respect to the ones offered by *stateful* IQAS.

Interactivity According to Definitions 2.5, 2.6 and 2.10, IQASs and CoQASs are the only ones capable of interacting with the user. Here the *interactivity* is intended as the capability of the systems to grant users the possibility to perform varied interactions as a response to an action by the system. We call the set of user reactions made available by the system as *interactive session*. In the example reported in Figure 1 we can observe that after the user question, the system gives the answer, but also provide an *interactive session* as a set of next possible questions the user can interact with. In the case the user clicks on one of the follow-up questions listed by the system, it will react by providing the related answer and starting a new *interactive session*. The interactivity is more straightly perceived in a CoQAS. In that case, a system message (reaction) corresponds to each user action.

Robustness QASs may allow a more or less complex interaction with the user depending on a third dimension that emerges from the literature, that is *robustness*. When the user question is not properly formulated or leads to ambiguous answers, a classical QAS reveals a point of failure returning a wrong answer or NULL. This also applies to IQA^e systems, designed for exploration purposes (cf. Definition 2.6). Conversely, IQA^d and CoQA^{s/u} systems allow the user to fix errors and ambiguities producing follow-up queries. IQA^d systems need different *interactive sessions* to solve the user errors or ambiguities due to their *stateless* configuration. Instead, a *stateful* setting, i.e. CoQA^{s/u} systems, stores all the *interactive sessions* in a context having relevant information to solve these issues from the first moment they appear. It follows that CoQASs have an higher *robustness* than IQA^d systems.

Naturalness is strictly dependant to the *interactivity* of a QAS. In our analysis, a QAS is *Natural* when it grants users spontaneous and immediate interactions using natural language whenever possible. It is worth noticing that naturalness is not a synonym of *user friendliness* because we can have an interface very friendly and effective (as that shown in Figure 1) that is not *natural* in our sense. In fact, the *Naturalness* can be achieved when users cannot sense the changing of interactive sessions during the usage of a QAS. That is the case of CoQASs, where users feel a single flow of interaction instead of a sequence of separated interactive sessions. Conversely, IQASs may break this continuity, e.g., in a *stateless* configuration, by posing exploratory/disambiguating questions regarding information that are already encountered by the user in the previous interactive sessions.

Initiative Basically, the *initiative* is intended as the possibility for an actor (either the user or the system) to choose which information composes the *interactive session*. In details, whether only the QAS forms the *interactive session* with a limited number of admissible data the user may interact with (i.e. a set of pre-computed follow-up question or feedback to a disambiguating question), then its *initiative* is *system-based*. Instead, the QAS initiative is *user-based* when she is free to form their own *interactive session*, e.g. asking any follow-up questions to the system answer. Otherwise, we have a *mixed initiative* QAS when both the previous mechanisms are enabled.

2.3 Task and Challenge based classification of Question Answering Systems

QASs can be built in several way, covering features like robustness or naturalness and ensuring a more o less sophisticated interactivity. Nevertheless, all these systems are implemented to fulfill some specific tasks, which decline the QA problem (cf. Definition 2.1) in multiple variations. It emerges from literature the following *task-based* taxonomy of QASs.

Open-Goal QA. Here the QASs exploit unstructured text to solve the QA problem. Forum messages or bounded sets of answers related to a specific domain, commonly known as Frequently Asked Questions (FAQ), fed the knowledge source of *open-goal* QASs. More in depth, depending on the QA model and its knowledge source, the open-goal QA is further classified into *community* QA (cQA) [132] and *classifier-based* QA (CB QA) [74]. Selecting the best answer from a pool of candidate ones, usually built on top of forum thread, is referred to as a *community* Question-Answering. In detail, cQA models foresee ranking mechanisms to achieve their goal. Conversely, a *classifier-based* QAS chooses appropriate answers by categorizing questions into default classes provided by the knowledge source (e.g. FAQ). In fact, each group is mapped to a specific answer which best satisfies the related information need.

Factoid QA. It answers questions that refer to a specific fact, e.g. "*Who is Leonardo Di Caprio?*" or "*What is Interstellar?*" and "*Where Christopher Nolan is born?*". The fact answering a given natural language question has to be retrieved from the QA knowledge source, which takes the shape of a *knowledge base*. The latter can occur as a set of unstructured documents hosting facts (i.e. Wikipedia, business documents, ect.) or as a collection of structured rules expressed in several forms (e.g. logic programming rules with Prolog and graph triples for knowledge graphs). In case, we distinguish *machine reading comprehension* QA (MRC QA) [144] from *knowledge-based* QA (KB QA) [150] tasks respectively. Teaching machines to read and understand texts on which to infer answers to user questions defines the *machine reading comprehension* (MRC) Question-Answering task. MRC QASs reply to questions either by pointing words span in documents or by generating a new text string, both enclosing facts satisfying the user information need. Differently, KB QAS implements a model to translate the user questions into queries allowed by the KB for seeking answers. In other words, it provides a universally accessible natural language interface to factual knowledge [117].

Visual QA. The goal of this task is to generate answers that encapsulates a truthful description of a picture on which questions are asked. The aim of Visual QA (VQA) is to find out a correct answer for a given question which is consistent with visual content of a given image [27].

All the collected publications are distributed among the previously described task-oriented classes as in Table 2. It is noteworthy that, although a great number of works aims to realize an IQAS able to read and understand huge number of textual documents for providing answers to users questions, all analyzed research efforts are still evenly distributed among the highlighted categories, showing a high interest from the QA community to all these tasks.

In addition, QA approaches vary depending on the challenges they target. We found different features in QA methods aiming for different goals, e.g. optimize system performances (i.e. answers accuracy, response time, etc.) or the overall user experience. Therefore, QASs can be further classified based on objectives they cover.

System Performances (SP). Starting from other state-of-the-art systems, new approaches are continually proposed in order to design QASs with improved effectiveness. To this purpose, extensive offline experiments are usually carried out on several public datasets.

User Experience, Trust, and Transparency (UTTP). QASs aiming to solve this goal are designed to engage users in a more compliant way. Three main aspects are usually evaluated regarding users relationship with the system, that is *user experience*, *trustworthiness*, and *transparency*. Whether the first one deals with *satisfaction* and *usability*, the *trustworthiness* is more oriented to measure user expectations about results returned by the system, that is how users trust system functionalities [79]. The *transparency* instead, evaluates the interpretability and comprehensibility of system processes [90].

Usability and Interaction Analysis (UIA). This challenge does not need any kind of new QASs implementation. In fact, its scope is to analyze the usage of already existing QASs or users behaviours towards interacting with them. Thus, statistical analysis are always provided and discussed for this goal [9].

Specific Domain-Dependent Task (SDDT). Most of papers challenging this task design a complete QA environment aiming to solve a specific issue. The proposed system usually drives users to execute domain dependent processes or tasks in a easier way. As a consequence, these solutions strictly fit the problem for which they are designed and tested. Thus, they are not thought to be generalized. This kind of publications often lacks comparisons with other state-of-the-art models [72].

Table 2 also depicts the distribution of publications among the aforementioned challenges, where emerges a not balanced QA community interest. Research efforts are almost totally focused on improving QASs performances, while a very few works emphasize statistical analysis about usability and interaction. Implementation modalities and algorithms still have shortcomings in reaching strong performances about not specialized QASs, that is the IQA field is far to be solved. To underline this evidence we see a lower interest in User Experience, Trust, and Transparency challenges.

Table 2. Work distribution over the dimensions identified.

	Open-Goal QA		Factoid QA		Visual QA
	cQA	CB QA	MRC QA	KB QA	VQA
SP	Wu et al. [132] Hu et al. [44] Wu et al. [131] Xiong et al. [135]	Waltinger et al. [124] Nie et al. [89] Su et al. [119]	Han et al. [39], Yang et al. [139] Yuan et al. [144], Chada [13] Das et al. [24], Mass et al. [80] Li et al. [65], McCarley [81] Xie [134], Li et al. [69] Bhattacharjee et al. [10] Kuo et al. [60], Kundu et al. [59] Osama et al. [91], Qu et al. [99] Wang et al. [126], Su et al. [118] Qi et al. [98], Li et al. [67] Yang et al. [138], Qu et al. [101] Qu et al. [100], Ju et al. [50] Zhu et al. [153]	Zheng et al. [150], Zhang et al. [148] Zhang et al. [145] Petukhova et al. [96] Perera et al. [95] Moon et al. [85] Damjanovic et al. [23] Liu et al. [71] Christmann et al. [18] Müller et al. [86] Shen et al. [113] Guo et al. [37]	Shi et al. [114] Do et al. [27] Gao et al. [34] Shao et al. [112] Pradhan et al. [97] Gordon et al. [35] Yang et al. [141]
UTTP	Rücklé et al. [107] Zhang et al. [149] Zhou et al. [152] Xiong et al. [135] Wong et al. [129]	Liu et al. [73] Latcinnik et al. [61] Sugiyama et al. [121]	Chiang et al. [16] Baheti et al. [5], Mandya et al. [78] Mandya et al. [79] Vakulenko et al. [122] Reddy et al. [104], Basu [6]	Sorokin et al. [117] Wu et al. [133] Habibi et al. [38] Xu et al. [137]	Alipour et al. [1] Jin et al. [48] Shin et al. [115] Li et al. [68] Li et al. [66]
UIA	Sen et al. [111] Kulkarni et al. [57]	Siblini et al. [116]	Hulburd [46], Otegi et al. [92] Aken et al. [123]	Le Berre et al. [9] Aken et al. [123]	-
SDDT	Zhang et al. [147] Kulkarni et al. [57]	Maitra et al. [77], Lockett et al. [74] Lee et al. [62], He et al. [41] Alloatti et al. [2] Sakata et al. [109]	Schwarzer et al. [110] Siblini et al. [116] Kumar et al. [58]	Li et al. [64] Habibi et al. [38]	Riley et al. [106]

2.4 Interaction Modalities

IQAS allow different interaction modalities to engage users in the process of seeking answers. Here, the interaction modality is intended as the physical channel available for users to exchange information with the system, e.g. clicking on dedicated sections of a web page, typing text messages or talking with an agent. Thus, QASs can be differentiated based on interaction modalities they allow.

Clicks & Touches. Here users interact with the system by means of specific visual regions, which are designed to host only supported information. This is the case of the IQAS depicted in Figure 1, where the user can only react to an interactive session by clicking on exploratory entities and questions.

Text. The majority of QASs provide an interaction modality that foresees natural language texts as input. Users express their questions through texts and receive answers in the same format. For example, Zhang et al. [147] propose a QAS which exploits a multi-scale deep neural network to extract deep semantic information from medical text to provide textual answers at different levels of granularity. That is, this framework allows only textual questions from the interaction with the user.

Speech. This interaction modality permits users to pose their questions through speech. QASs that takes advantage of users voice are often implemented within VDAs like Siri, Google Assistant, Amazon Alexa, etc. Other relevant works are proposed by Lockett et al. [74] and Mészáros et al. [83]. They both implement IQASs which allow users to talk with them. The first solution addresses the operator routing task within a customer care service, where the answer is the most suitable operator to a raised issue. The latter builds a smart cart that leads users, during their shopping, to the place where an asked product is located.

Visual. QASs allowing users to exploit images for the QA task belong to *visual* QASs. This interaction modality allows visual input like images, photos or even videos. Alipour et al. [1] implement an IQAS that supports images paired with natural language questions generally asking for what is represented in the picture (e.g., a photo of a football player and a question like "What sport is this?"). Then, it computes the answer and its motivations by selecting informative *region of interest* maps on the input images and exploiting *deep learning* models. More straightforwardly, Shin et al. [115] grant the user to send pictures to their system on which it automatically answer to the question "What is in this picture?". In addition, their system asks the user several disambiguating questions to generate an answer (i.e. description) which best fit her subjective view.

Depending on the allowed interaction modalities, an IQAS can be further distinguished into *uni-modal* (i.e., only a single interaction modality is allowed) or *multi-modal* (i.e., more interaction modalities are supported) QAS. See the work by Deldjoo et al. [?] for more information about the multi-modality in information seeking systems.

3 INTERACTIVE QA SYSTEMS: ARCHITECTURE AND TECHNIQUES

In this section we propose a general architecture of IQAS representing the prominent methodologies that exist in the literature. We also describe the implementations of each architectural element according to the state-of-the-art solutions.

3.1 Architecture

Figure 2 outlines a high-level architecture with the components that an IQAS should implement. Four major components are proposed: the *interaction engine*, the *state tracker*, the *QA module*, and the *knowledge source*.

The **Interaction Engine** (IE) is the module that manages the interaction with the user, receiving her requests and providing the response to her. This module enables the interaction between user and system by leveraging different interfaces according to the system capabilities. For instance, in case the user-system interaction is based on click (e.g. automaTA by Lee et al. [62]), the IE enables the system to capture the user request and prepare it in a suitable form for the *state tracker* (cf. Section 3.3). In a similar way, when the interaction is based on natural language messages, then the IE supports the reception of textual messages [131], as well as images [106] or even videos [48] for visual

user-systems interactions. In a nutshell, the IE manages different *interaction modalities* (cf. Section 2.4) according to the system implementation.

The complexity of the IE depends on several factors: (i) the *interactivity* level, (ii) the *operation modality*. The *interactivity* level refers to the distinction between standard and interactive QASs. The IE enables users to engage with the system in both *single-step* (e.g., QAS that returns the most relevant answer as a direct result to the user question [109]) or *multi-turn* interaction (e.g., systems allow further steps to refine or expand the given system response [115, 119]). The higher the interactivity level is, the more complex the IE-module implementation will be. The *operation modality* is defined as the capability of the IE to deal with one or more *interaction modes*, making a IQAS *uni-modal* [98] and/or *multi-modal* [34]. Finally, the *initiative* defines whether the user [59], the system [137] or both [128] can trigger a disambiguation/exploration step, i.e. with disambiguation/follow-up questions.

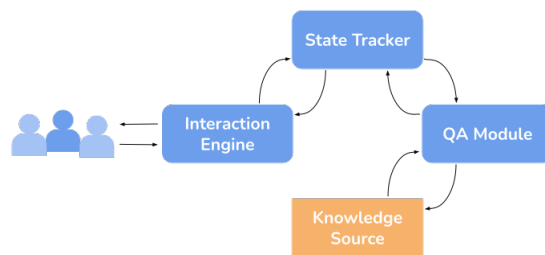


Fig. 2. General Architecture of Interactive Question Answering Systems

The **State Tracker** (ST) manages all the information the *interaction engine* and *QA module* need in order to handle the QA request. In greater detail, it aims to fill up the *QA state* (cf. Definition 2.8) and to collect all the information exchanged between user and system up to that time. In addition, referring to the Definitions 2.5, 2.6 and 2.10, an IQA task can also be seen as a *chain of interactive sessions* (cf. Definition 2.7) grouped by the user's information need. Accordingly, the ST keeps track of the exact point of the *chain* in which the system is at any given time: this is the *QA state* that represents a sort of snapshot of the system at time t .

The ST operation mainly depends on two elements: (i) the *context* and (ii) the *tracking methods*. For CoQASs the same action could have different reactions and meanings given the *dialogue context*, while for *stateless* IQASs, the ST has no information about past interactions, dealing with *session context* as exploitable information (e.g., images or textual snippets) that completes the user input to the system.

With regard to *tracking methods*, in the literature they are classified as *implicit* or *explicit*. An *explicit tracking method* keeps information about the previous *QA states* in a structured form. For example, natural language tokens or knowledge graph entities can be collected [133]. Conversely, *implicit tracking methods* store previous *QA states* in a latent shape like embeddings or trained parameters of a model. That is the case of systems that rely on deep learning models, which learn patterns on real dialogue sets and perform conversations without needing explicit state tracking techniques [99]. In that case, the state is embedded in the learned model.

The **Question Answering Module** (QAM) aims to provide the most suitable answer to the user question. This component is not specific of IQASs, but is implemented in every QASs. It retrieves or generates the answer to the user by exploiting the context, if any, managed by the ST. The answer depends on the *data representation* adopted by the QAS (e.g., text, images, etc.) as well as the interaction mode.

In order to deal with them, the QA models can implement *data-driven* and *instruction-based* approaches. The former belongs to the ML field and encodes all the collected documents through *numerical representations* such as TF-IDF, one-hot encoding, Word2Vec, etc. [68]. The latter operates with both *categorical information*, such as pure text or KG entities [87], and *numerical representations* outlined before [71].

Finally, the **Knowledge Source** (KS) stores all the knowledge exploited by QAS to answer users questions. The KS data can be *structured* (e.g Knowledge Graphs, Relational Database, ect.) or *raw* (i.e. images, videos, texts, ect.). The data type can depend on the task that the QAS aims to solve. For instance, a *factoid* QASs can rely on both *structured* information as a KG, a set of triples in the form of subject-predicate-object [113], and *raw format* data as a collection of textual documents [138].

In the following, we aim to give an in depth snapshot of the state-of-the-art about IQASs with respect to the components outlined in Figure 2. On this line, Table 3 groups some relevant examples provided by the literature according to the IQAS modules, their aspects, and technical specifications for their implementation.

Table 3. A summary of components in the IQAs representation pipeline, as sketched in Fig.2

Module	Aspect	Approach and Example work
Interaction Engine	Interactivity	single-step [9, 35], multi-round [52, 150].
	Operation Modality <ul style="list-style-type: none"> • Single • Multiple 	text [69], audio [74]. images & text [1, 114], audio & text [60], video & text [141].
	Initiative	system-driven [43, 148], user-driven [27, 100], mixed [23].
State Tracker	Tracking Methods	implicit [118], explicit [18, 72].
	Context	session [85], dialogue [99].
Question Answering	Image Representation	ResNet [44], Faster-RCNN [112], VGG Net [97], CNN [35].
	Text Representation	inverted index [22], pure text [33], term frequency [66], Word2Vec [44], Glove [112], LSTM [35], WordPiece [109].
	Model <ul style="list-style-type: none"> • Instruction-based 	keyword matching [22, 130], rules execution [18, 150].
	<ul style="list-style-type: none"> • Data driven 	supervised [65, 132], reinforced [24, 144].
Knowledge Source	Structured	knowledge graph [117], relational database [87].
	Unstructured	texts [123], images [68], video [48].

3.2 Interaction Engine

The *interaction engine* is the module that manages the user-system interactions flow. Its main objective is to allow users to communicate with the system in a natural manner. As a result, the IE makes use of interfaces that are tailored to the system's characteristics, while also properly formatting the information intended for the user. For example, CoQASs require an IE that supports dialogues via chat-box interfaces. Rather than that, IQASs that solve the VQA task require this component to provide capability for attaching photos as an additional feature. The IE can incorporate numerous system design choices, such as allowing users to ask their own follow-up questions (i.e. exploratory question) or choosing a query from a predefined set. In a nutshell, the IE is characterized by three primary characteristics that include: *interactivity*, *operation modality*, and the *authorized initiative*.

The **interactivity** of a QAS determines how users engage in the process of seeking answers. In fact, this aspect denotes the distinction between conventional QASs and IQASs. Traditionally, QASs deliver immediate responses to user queries, excluding her from further discussions. In comparison, IQASs accumulate several interaction sessions that allow both users and the system to refine/enrich the initial result. In a nutshell, interactivity defines the QAS required to support interactive user sessions.

In the literature, systems that do not anticipate the use of any interactive sessions to support achieved replies are referred to as *single-step* interactive systems. Such systems receive inquiries from the user and eventually supporting data (i.e. images, document extracts, audio track, etc.) in order to produce a unique and appropriate response. Thus, the system-user interaction concludes when the user receives the system response.

For example, Aken et al. [123] analyze the ways BERT-based QASs answer to user questions. Their system addresses the MRC QA task, with a focus on understanding BERT operations in retrieving the correct answer on a given question. Therefore, authors omit the system to enable *interactive sessions* for the computed answer, granting only *single-step* interactions with the user.

Conversely, QASs that enable *interactive sessions* to the answers given to the user are named as *multi-round* interactive systems. This includes the systems described by Definitions 2.6, 2.5 and 2.10. The *interactive sessions* allow both the user and system to refine their outcomes through *multi-round* communications (e.g., disambiguating the ambiguous user questions or further exploring the system responses).

For instance, CROWN, a CoQAS implemented by Kaiser et al. [52], exploits a supervised approach to allow the context propagation through multi-round interactions. As a *stateful* IQAS, CROWN is able to understand the context left implicit by users during their conversations. The *interactive sessions* are composed of the user exploratory questions to system answers. Given the question "When did Nolan make his Batman movies?", the CoQAs retrieves the appropriate document snippet from its *knowledge source* as the answer. Then, it enables an *interactive session* in which the user can ask "Who played the role of Alfred?". In this case, the hidden context "Batman movies directed by Nolan" is known since the system has memory of the previous QA states.

It is worth noticing that the IE *interactivity* has no direct relations with the *statefulness* property, which belongs to the *State Tracker*. Thus, the *interactivity* does not distinguish *stateless* IQASs from *stateful* ones (i.e. CoQASs). As a further proof, the example in Figure 1 shows an IQAS that support multi-round interactions in a *stateless* configuration.

The QAS's **operation modality** specifies which kind of interaction mode the IE has to deal with (cf. Section 2.4). QASs that manages more than one physical channel as a means of interaction (i.e. text and visual or speech and clicks) exploit a *multi-modal* IE. Instead, systems that allow unique *interaction modalities* in communicating with users (e.g. only text [122] or speech [60]) have their IE being *uni-modal*.

Qi et al. [98] implement a QAS based on *siamese networks* to learn user preferences and answer her subjective questions. They concatenate user questions representation, i.e. Bi-LSTM embeddings, with her latent factor profile. Then a classifier is trained on the resulting representations to select personal answers from a multi-choice QA dataset. To this goal, the QAS IE implements an *uni-modal operation modality*, which requires only textual questions.

Conversely, *multi-modal* IE deals with different *operation modalities* at the same time, i.e. speech and text [74], visual and texts [66] or texts and clicks [62].

Gordon et al. [35] address the VQA task through an autonomous agent that dynamically interacts with a visual environment to reach the answer. They design an IQAS that requires in input both a textual question and a visual context. Thus, the agent may seek other visual views for disambiguating the user question or compute the answer by means of several modules, i.e. a Scanner to capture images, a Navigator to change the system view and a Manipulator.

However, the actor mainly involved to configure *interactive sessions* is designated by the IE **initiative**, which is three-folded. We refer to *user-driven initiative* when users can freely choose their questions as a follow-up to the system response (i.e. *exploratory question*). In contrast, the *system-driven initiative* specifies the *interactive session* being a finite set of data on which users may interact (e.g. picking a follow-up question from a collection computed by the system or answering *disambiguating questions*). The *mixed initiative* allows both the previous two cases.

Referring to Section 2, systems enabling the *disambiguation* have at most a *system-driven initiative*, while exploratory interactions can also include a *user-driven initiative*. For example, the work of Su et al. [119] implements an IQAS which allows users to ask about the composition of API calls. In addition, it supports the user to refine the system answer by adding/removing parameters to the returned API call, exploring new related ones. These parameters are provided by the system as the *interactive session* related to the reached answer. Thus, the proposed system *initiative* is *system-driven*.

Zhang et al. [148] designed instead an equipping tool for existing KB QASs called IMPROVE-QA with a *mixed initiative* IE. It provides the capability of continuously adding new concepts/entities to the equipped QA *knowledge source*, improving its accuracy. To achieve this goal, the authors implement a disambiguation step where users are asked to mark wrong and correct answers from a whole posed by the system. Moreover, they are also allowed to add further correct missing entities in the *interactive session* according to the *why-not* problem [14]. The latter corresponds to an *exploratory question* that may be answered both by inferences on the *knowledge source* or by the new added information.

Finally, Google Assistant and the system implemented by Qu et al. [100] are CoQASs that allow multiple user defined interactions, which make *user-driven* their *initiative*. In both cases, the user is free to ask any exploratory question, moving the system to seek manifold information of a topic.

3.3 State Tracker

The *state tracker* is the module that deals with *QA states* as described in Definition 2.8. As a middle component, it supports communications between the IE and QA modules to solve the QA task. This means that the ST takes data from both of them to update the *QA state* at each interaction step. It enables the QA module to exploit information (i.e. context data) in addition to the *knowledge source* for computing answers to user questions. At the same time, it supports the IE to configure the authorized interactions based on data currently stored in the *QA state*.

In literature, the ST strictly depends on the *statefulness* property of a QAS. It assumes different behaviours based on both *QA state tracking methods* and the type of *context* (cf. Definition 2.8). In particular, the *context* may differ in two types, i.e. *session context* and *dialogue context*.

The *session context* is a set of supporting data contained in the *QA state* that can be obtained with just a single interaction with the user. Thus, it is peculiar of *stateless* IQASs. The data hosted in this type of context depends on the QAS *operation modality* and the way with which that information is represented. For instance, Shi et al. [114] design a Quaternion Block Network to solve the VQA task, which admits images in a numeric representation in its *session context*. Similarly, Das et al. [24] expect a set of textual paragraphs in a numerical form to support the user questions.

In contrast, Pradhan et al. [97] implement a QAS enhanced by Social Networks to help visually impaired people in using these platforms. The solution they propose takes advantage of a *session context* shaped on multiple information sources. In fact, the ST gets contextual data like images and audio for each user question from the IE. Given the question "How many dogs are in the photo?", the context will have a picture as a supporting visual information to find the answer besides an audio source of the question, if any, to double check any typos.

The *dialogue context* is instead typical of CoQASs whose data is continuously gained after each interaction with the user. All the QA states feed the context of the next one, forming a *conversation* with the user. For example, IHAF is a

CoQAS implemented by Perera et al. [95] that hosts textual data in form of a *conceptual graph* within the context of each *QA state*. During the conversation, it receives from users further information that expands the *contextual graph* at each interaction step. This allow the system to return more accurate answers the longer the conversation is.

A similar intuition is adopted by Fukumoto et al. [33] in implementing an expansion mechanism of the *dialogue context* to address the problem of ambiguous questions. Once the system receives the user query, it will ask for confirmation it is understood before returning an answer. The system solves a MRC QA task allowing users to ask *factual questions* like "Who is a gold medalist at Olympic game?". In that case, the user may intend the Olympic games held in *London*, *Tokyo*, or any other city. In addition, it is not known which sport the gold medalist refers to, e.g. *Swimming*, *Judo* or *Tennis*. In fact, when the CoQAS gets entities that need a *disambiguation*, it will query the user with questions like "Do you mean London Olympic game?" or "Do you refer to the gold medalist of swimming?". Then, all the clue words selected by the user will feed the *dialogue context* of the ST until no more disambiguation is needed. At this point, the system will be able to compute the most relevant answer to the user question.

To keep in memory all the *QA states*, the ST relies on **tracking methods**. In literature, we distinguish *implicit* from *explicit tracking methods* regardless the type of context hosted by *QA states* (i.e. session or dialogue context).

On the one hand, the *implicit tracking methods* deals with the storage of *QA states* in numeric forms (e.g. latent representation). For instance, Su et al. [118] model a CoQAS as an adaptive framework based on the sequence to sequence model. It manages numerical representations of all the memorized *QA states* in a *conversation history*, which can be combined with current questions (also numerically represented) to retrieve refined answers.

Conversely, Chiang et al. [16] implement three CoQASs with different *implicit tracking methods* to test their ability to comprehend textual contents regardless their performances on the MRC QA task. They show how ML systems like FlowQA [45], BERT [25], and the proposed SDNet rely more on previous *QA state* answers instead of any contents host in the KS. In a nutshell, answers are computed by exclusion from the previous ones instead of being logically inferred from both the implicit tracked *dialogue context* and the KS.

On the other hand, the *explicit tracking methods* grant a direct access to explicit data stored by the system. This information type is understandable by humans, enabling the QAS to be explainable and interpretable. Here, *QA states* are not recorded in a latent form, thus preserving the structure outlined in Definition 2.8. The system designed by Wong et al. [130] represents a first attempt to realize a CoQAS with an *explicit tracking method* for storing *QA states*. This solution expands the *dialogue context* with all the keywords arose during the conversation with the user. Therefore, the *QA Model* infers appropriate answers relying on both its KS and dialogue context, which hosts some weighted elements computed according to a decay function over conversation steps.

Instead, Zheng et al. [150] exploit conversations to improve the performances on the *factoid QA* task. The main objective is to answer user questions through a knowledge graph, enabling cost-optimized user interactions to solve ambiguities. Thus, the ST is driven by an interaction graph which schedules the best disambiguation questions to ask requiring the lowest interaction effort. The *dialogue context* is built with fragments of *Basic Graph Patterns*, a set of triples usually exploited in query languages like SPARQL, that holds the facts collected during the conversation with the user. Once the final node is reached on the interaction graph, the *QA module* returns the final answer.

3.4 QA module

The core component of a QAS is the *QA Module*, whose goal is to find appropriate answers to user questions given a *knowledge source* and possible *contexts*. Thus, a *QA module* has to be designed to understand user queries and infer the requested information from an often large collection of data.

The QAM differs on the type of data the QAS has to deal with and the modeling strategies that fit the QA task resolution. Therefore, it is described by two characteristics, i.e. the *data representation* and the *modeling approach*.

The **data representation** depends on both the supported *interaction modality* and its *modeling approach*, which in turn counts on the information types hold by the *Knowledge Source* (cf. Section 3.5). In fact, we distinguish *categorical* data (e.g., a sequence of textual strings) from *numerical* one (e.g., vectors as well as matrices). It is worth noting that the QAM *data representation* is intended as the data model that best fits the requirements of a QAS, which may differ from the ones of the *knowledge source*. In a nutshell, the QAM supports a *numerical data representation* when its model requires data being expressed with numeric elements to compute an answer (e.g. dealing with images/audio or deep learning architectures). For example, the work of Hu et al. [44] enriches the semantic information of possible answers by adding images in their knowledge source. Thus, the QAM requires a *numeric data representation* to evaluate semantic similarities between queries and possible answers.

Mandya et al. [78] focus instead on the Co-reference resolution task in a conversational setting through ML models. In particular, they highlight a set of co-reference question-answer chains in the *dialogue context* searching for terms that may refer to the same entities of the user question. Their implemented model automatically achieves the goal by means of attention mechanisms, which requires data to be represented in a *numerical* form. In detail, authors embed the user question in a numeric vector by concatenating its character and word embeddings, which are obtained through a trainable numeric vector and the GloVe [94] pre-trained model respectively.

On the other hand, Fukumoto et al. [33] leverage the conversational issues by means of matching keywords methods and Named Entity Recognizer. Thus, their QAM relies on data expressed with their *categorical* values.

Schwarzer et al. [110] implement a *stateless system-driven IQA^d* in the e-government domain for the public administration of Berlin. They indexed all the *knowledge source* data (i.e. services) in an Elasticsearch³ inverted index, enriching it with meta information like services popularity rankings and additional keywords (i.e. synonyms and stems from services description words). The implemented QAM enables users to find their answers operating on the inverted indexes with three different methods (i.e. keyword scoring, Elasticsearch full-text search and a custom scoring), which all require a *categorical representation* of data.

The **Modeling Approach** identifies the strategies adopted by the QAS to make practical and effective reaching an answer given a question. In other words, it refers to the framework that existing algorithms or new ones exploit to solve a QA task. The choice of a model relies on three factors, which are the QA task to be solved (cf. Section 2.3), the supported functionalities (e.g. targeted challenges or the allowed interactivity), and the available data (i.e. the *knowledge source*). Nevertheless, each implementation can be classified as an *instructions-based* or *data-driven* model. The former takes advantage of well-designed instructions sets to its goal, while the latter tries to take out and learn patterns from a huge set of data (i.e. query-response pairs) to reply to the user questions.

Hence, the *instructions-based* methods relies on a set of rules designed a priori to accomplish its task. The resulting QAM is totally unaware of data hold by the *knowledge source*. Thus, its implementation covers a finite number of scenarios that may emerge during the process of answering questions. The state-of-the-art QASs can be further divided into three categories: (i) *keyword matching* models, (ii) *pipeline execution* algorithms an (iii) *translation* models.

The *keyword matching* strategy aims to find the answer to a question based on the number of matches between their key terms. For instance, the previous work of Schwarzer et al. [110] implements several scoring functions that retrieve

³<https://www.elastic.co/elasticsearch/>

the answer among the most relevant document based on how many question keywords are found in them. Each scoring function is designed to consider a specific keyword feature by means of different weight functions.

Conversely, Wong et al. [130] realize an first attempt of CoQAS by concatenating keywords extracted from user questions with the ones stored in the ST *dialogue context* with a gradually-decaying weighting function. Thus, keywords that are far in the dialogue context have a less relevance than the most recent or repeated ones. The system searches for answers giving high priority to documents with relevant matching keywords at specific moment.

The *pipeline execution* models foresee a sequences of functions performed in cascade to solve the QA task. Christmann et al. [18] take advantage of this modeling strategy to implement Convex, a *factoid* CoQAS based on the Wikidata⁴. Authors design a set of processing steps to be executed in pipeline for solving the KB QA task. Starting from the user question, a (i) named entity recognition and disambiguation (NERD) system identifies Wikidata entities contained in the text. Then, a (ii) context sub-graph is built with Wikidata entities in the neighborhood of the one recovered by (i). Each element of this sub-graph and its frontier nodes represent a potential answer to the user question. Thus, (iii) key entities are retrieved based on three relevance evaluation (i.e. relevance to the question words, to the sub-graph context, and to the knowledge source priors). These three signals are then aggregated through (iv) the Fagin's Threshold Algorithm [30] and the answer is obtained from (v) a scored list of entities, which assumes the solution be in the near proximity of the key entities and the ones stored in the *dialogue context*.

Furthermore, *translation* approaches perform a translation of questions from natural language to a formal one, which depends on the QAS *knowledge source*. It allows users to have a direct access to complex structured data collections (e.g. knowledge graphs or relational databases) without be expert in the related query languages (i.e. SPARQL and SQL). Naeem et al. [87] implement a QAS that translates the natural language user questions into OLAP queries, while FREyA, by Damjanovic et al. [23], is one of the first example of KB QASs that translates the user questions into SPARQL queries. It identifies a set of ontology concepts from the natural language question through the combination of syntactic parsing and ontology reasoning techniques or by means of string similarities evaluation, synonyms detection and user engaging. The SPARQL query is built based on the retrieved ontology concepts.

Differently, *data-driven* strategies refers to models trained on often huge data collections to learn patterns for answering user questions. These models build their own knowledge looking at the examples provided during a training phase. Then, they take advantage of their training to solve a specific task (i.e. answering questions). The QAM *data-driven* strategies found in the literature are classified in *supervised* and *reinforced*.

The *supervised data-driven* model relies on a labelled dataset to learn which answers are associated to a given question example. They further be divided into *detecting* and *generative* implementation, which depends on the method with which the answer is computed. With the **detecting** approach the QAM learns how to detect the desired response from an information set. Thus, the answer can be either found as a limited sequence of words within a document or picked from a set of answers hold by the knowledge source. For example, the work of Alloatti et al. [2] proposes a CB QAS on the e-invoicing domain and its regulation. The authors opt for a BERT-based model with an added layer to classify user questions into specific groups, which are in turn labeled with the related answer. The training procedure matches with the fine-tuning of BERT in learning a one-hot encoded vector whose indexes refer to the different group of answers.

Li et al. [69], instead, design a conversational MRC QAS with a directional attention weaving (DAW) mechanism to extract answers from documents hold in the *knowledge source*. They implement a model that estimates the start and the end of the answer within a document for each question. To achieve this goal in a conversational setting, they train the

⁴<https://www.wikidata.org/>

model on the CoQA dataset [104] with a combination of different attention types to leverage both data coming from the knowledge source and the dialogue context. In contrast, *generative* approaches learn how to build the answer by composing appropriate words to the given question. In other words, the output is generated from scratch instead of being selected among the existing ones.

Li et al. [65] train a CoQAS to answer the user question about information that are previously given to the system. It expects as input a list of sentences defining a queryable "story". Then the user ask for data about that story in a conversational setting. In fact, the system also learn to pose disambiguation questions to the user when its current knowledge lacks of crucial information for computing a reply. The *generative data-driven* model proposed by authors is based on encoding both sentences and questions via Gated Recurrent Unit (GRU). The two results are then combined and decoded by another GRU to compose the answer or a disambiguation question. This framework is trained on a large corpus of conversations to learn when a disambiguation question is needed and how a sentence (i.e. question and answer) is built by means of words.

Finally, the *reinforced data-driven* models adopt the paradigm of *reinforced learning* to comprehend the QA task. Hence, the QAM performs some *actions* based on *observations* and *states*, which may results in *transitions* to other *states* eventually *rewarded*. In a nutshell, the training procedure states the system in attempting to solve the QA problem through an intrinsic logic. When the QAS answers correctly, it receives a reward that usually minimizes its training cost function. Otherwise, no incentives, or penalties, are given to the system. A clear example is provided by Gordon et al. [35], where the proposed visual QAS is modeled as an agent able to explore the environment of a given picture through several actions (i.e. navigating, manipulating, scanning and detecting as well as answering). Here, authors enable the hierarchical reinforced learning paradigm to train an high-level controller in selecting and invoking the right sub-task (i.e. the previous actions) to reach the correct answer in an efficient way. The reward is received when the system planning produces a positive outcome (i.e. the right answer).

3.5 Knowledge source

The information needed by the system to compute answers to user questions is hosted by the **knowledge source** (KS). Data about facts, domain specific instructions, and community opinions are here stored in different forms and modalities. This component represents the bases on which the QAS knowledge is built, defining what it is aware of and what it does not know. The hold information is made available both at running time and at an eventual training time for the QAS. This distinction determine the type of the adopted QAM modeling approach (cf. Section 3.4), which also depends on the kind of data contained in the KS.

Although the KS module usually mirrors the main features of datasets exploited by the system to accomplish its task (cf. Section 4), in this section we provide a view oriented to the QA processes instead of its data. In fact, we focus on the organization of information collections at a high-level of abstraction to label the state-of-the-art QASs approaching methods with data. Hence, data modalities and formats are not concerned to be discussed here. The QAS *knowledge source* can be classified into *structured*, *unstructured*, and *mixed*.

The **structured** KS manages the information composing the system knowledge by means of well-designed structures. In other words, it accepts only data that presents a standard structured organization universally recognized (e.g. knowledge graphs, relational databases). This is the case of Zheng et al. [150] and Zhang et al. [148], where the implemented systems are designed to rely on knowledge graphs (i.e. Dbpedia [82] and Freebase [11]) to retrieve the answer. Instead, the collection of question-answer strings pairs (e.g. QALD-6 and WebQuestion datasets) are used only to evaluate the systems performances. Li et al. [64] instead foresee a relational database as their CoQAS *structured* KS.

In contrast, **unstructured** KSs allow data sources to lack of an intrinsic structure. They cover the majority of QASs populating today literature, which rely on paragraphs lists or images collections to reply the user queries. Wu et al. [131] implement a cQA that depends on question-answer pairs (i.e. string sentences) that can be retrieved from any forum websites. Instead, Gao et al. [34] use a lists of question-images pairs labelled with answers as a KS for their VQAS.

Finally, the **mixed** KS enables the system to rely on both structured and unstructured source of data. An example is given by Zhang et al. [149], where the implemented QASs relies both on a multi-modal knowledge graphs (with images) and collections of qualified doctor advice to consultants, in the form of question-answer strings, to build its knowledge. Shen et al. [113] also exploit a combination of structured and unstructured information to train their CoQAS. In this case, authors take advantage of the Wikidata KG for the semantic knowledge and the Complex Sequential Question Answering (CSQA) [108] dataset to deal with dialog.

4 EVALUATION AND DATASET

Even while all IQASs in the literature can be characterized using the unified architecture seen in Figure 2, the datasets used to train and evaluate their models do not share a common set of properties. Depending on the QAS purpose, the modality type, and the interactivity setting (i.e. QA, IQA^e , IQA^d or CoQA), the nature and characteristics of a dataset might be vastly different. Consequently, evaluation methods also end up being rather distinct.

In this section we provide guidelines about evaluation protocols and measures usually exploited in literature to test QASs. We also explain which datasets constitute the state-of-the-art regarding QA tasks (cf. Section 2.3), QAS *modeling approaches* (cf. Section 3.4) and evaluation goals deriving from the QA challenges.

4.1 Evaluation protocols

The evaluation phase of IQASs determines the efficacy of the system in relation to the intended task and challenge. It is essential to permit comparisons between current solutions, therefore identifying the most effective for a given objective. It also allows researchers to determine whether there are any problems with the development of their methods. The evaluation step is essential to advancing the state of the art in literature.

An evaluation protocol refers to a set of actions that must be followed in order to gather data about a system's performance. In each of case studies explored, a specific set of qualities is tested in relation to a given goal (e.g., questions with grammatical errors to test the linguistic robustness).

There are two types of protocols: *offline* and *online* which are used for different types of evaluations.

Similarly to ML/RS models, *offline* evaluations for IQASs use pre-compiled offline datasets, and evaluation is carried out through cross-validation. Systems are hence analyzed on their ability to predict the missing data. Conversely, *online* tests require the system to work with real users to evaluate its functioning and returned results.

A few prominent examples of research works focused on offline evaluation is as followed: Shin et al. [115] evaluate their VQAS on the UTTP challenge, which require the data to include a set of high quality human written answers for each sample (i.e. images). They focus on automatically testing how natural the replies generated by the system is. Thus, no interactions with the user is enabled for this testing scope, making use of bilingual evaluation understudy (BLEU) algorithm [93] to evaluate the quality of the generate text. Riley et al. [106] test instead the ability of their VQA solution to cover the SDDT challenge by computing the accuracy on three datasets they generated. The ground truth here is the correct result that the system must return for each domain-specific scenario (i.e. classifying road signals or structures stability). It is important to keep in mind that the *offline* setting does not preclude the possibility of analyzing QASs using *data-driven* models. Because of this, a set of metrics must be developed to quantify the system's performance based on the test goals, and these measures must also allow for comparisons between different solutions.

In contrast, *online* evaluations engage users in working sessions to study the system’s functionality. Each session is intended to simulate certain use scenarios by emphasizing the most pertinent system aspects in order to study system behaviors.

Then, performances can be evaluated by collecting users opinions by means of surveys (i.e. user studies) or by monitoring certain operational measures. For example, Wong et al. [137] involve 11 participants in order to assess their CoQAS on the UTTP challenge. The authors design the user session as a conversation of 10-minute duration. At the end of each session, participants are asked to rate the system utterances for naturalness and coherence using two 5-points Likert scales. Instead, Li et al. [64] take use of measures taken during user engagement. They evaluate the proposed SDDT KB QAS’s usability by calculating the average time required to successfully complete the inquired tasks.

However, both the offline and online configurations can be exploited concurrently to realize a further detailed system evaluation. For instance, Zhang et al. [148] and Shin et al. [115] implement this mixed strategy to evaluate their systems in a wider range of features. The former engage 300 college students to analyze the system performance changes in increasing the number of hints given by human participants, while the latter ask up to 3000 workers to evaluate response features like diversity, attractiveness and expressiveness. In fact, these test goals are hard to be simulated and automatically evaluated.

In what follows, we will classify state-of-the-art evaluation techniques based on specific characteristics shared in relation to the objectives of the tests. In particular, these aspects are recognized as the result produced by the system (for example, a single answer, a list of ranked replies, or a dialogue), which is the primary focus of the evaluation.

Single Answer. Systems are assessed based on their ability to respond to user queries, with the returned responses serving as the primary metric to be evaluated. Depending on the problems, tasks, and modeling methodologies covered by the QAS, solutions can be assessed at several levels, which in turn establish assessment objectives such as *correctness*, *reliability*, and *sensitivity*, as well the *naturalness* and the *expressiveness*.

Ranked List This group refers to the system capacity to retrieve pertinent resources to a given queries. Although the QA task allows a single result to return to the user, some state-of-the-art works also permit other replies returned by the system after the initial response. This list reveals the *reasoning capabilities* of the QAS, as well as which features/information are deemed important for locating appropriate responses. The examined test objectives are comparable to those of the first group, although being able to be evaluated on a deeper level (e.g., answers positions in the returned ranked list).

Interaction Rather than the solution itself, the attention is on the interaction enabled by the system in order to arrive at it. The interaction influences the quality of the system’s responses. Consequently, interactions are assessed based on their *cost* and their required *user efforts*, as well as their *effectiveness* and *efficacy*. The majority of studies focusing on this element employs online evaluation techniques. Nonetheless, offline metrics are also utilized for stateful IQASs, which may be further evaluated based on coherence, context, and naturalness.

Table 4 summarizes the research findings. Each entry is characterized by (i) the reference to the specific paper, (ii) the year it was published, (iii) the primary problem addressed by the proposed solution, (iv) the type of evaluation setting used to test the system, (v) the element on which test objectives are evaluated, (vi) the dataset exploited to evaluate the implemented system (which are detailed in Section 4.2), and (vii) the QA task implemented. Although the targeted elements are similar both offline and online settings, we chose to emphasize trends over the years of the two evaluation protocols by noting them individually.

Table 4. Common datasets and evaluation metrics used in the IQAS literature

Authors	Year	Main chall.	Evaluation									Dataset	Task	
			type	metrics										
				offline metrics					online metrics					
				Efficiency	Error	Dialogue	Answer	Rank	User Imp.	Rank	Efficacy			Answer
Wu et al. [132]	2020	SP	offline					x				TREC, YH, SE, WQA	cQA	
Alipour et al. [1]	2020	UTTP	online							x			VG, VQA	VQA
Shi et al. [114]	2020	SP	offline				x						VG, VQA 2.0	VQA
Maitra et al. [77]	2020	SDDT	offline				x						FB, DL, InsD, IndD	CBQA
Zheng et al. [150]	2019	SP	both				x				x		QALD-5, WQ, GQ	FQA
Zhang et al. [148]	2019	SP	both				x					x	QALD-6, WQSP	FQA
Han et al. [39]	2019	SP	offline				x	x					InsQA, WPQA	MRCQA
Yuan et al. [144]	2019	SP	offline				x						QAit	MRCQA
Lockett et al. [74]	2019	SDDT	offline				x						SG	CBQA
Do et al. [27]	2019	SP	offline				x						V7W, VQA 2.0, TDIUC	VQA
Gao et al. [34]	2019	SP	offline				x						VQA 2.0, TDIUC	VQA
Das et al. [24]	2019	SP	offline				x						TQA, SeQA, Q-T, SQuAD	MRCQA
Shao et al. [112]	2019	SP	offline				x						VQA 2.0	VQA
Lee et al. [62]	2019	SDDT	online								x		SG	CBQA
Jin et al. [48]	2019	UTTP	offline		x		x						TGIF-QA, MSVD-QA, MSVRTT-QA	VQA
Zhang et al. [147]	2018	SDDT	offline					x					cMedQA 2.0	cQA
Pradhan et al. [97]	2018	SP	offline		x								COCO-QA	VQA
Gordon et al. [35]	2018	SP	offline				x	x					IQuAD v1	VQA
Shin et al. [115]	2018	UTTP	both				x				x	x	COCO-QA	VQA
Sorokin et al. [117]	2018	UTTP	online									x	-	FQA
Hu et al. [44]	2018	SP	offline					x					SG	cQA
Zhang et al. [145]	2018	SP	offline				x						SQ, WQ	FQA
Wu et al. [131]	2017	SP	offline				x	x					SemEval 2015, BZ	cQA
Rücklé et al. [107]	2017	UTTP	offline				x						InsQA, SE	cQA
Wu et al. [65]	2017	SP	offline		x		x						bAbI, IQA	MRCQA
Xie [134]	2017	SP	offline					x					DBQA	MRCQA
Schwarzer et al. [110]	2016	SDDT	offline					x					LeiKa	MRCQA
Petukhova et al. [96]	2015	SP	offline			x	x						EAT	FQA
Perera et al. [95]	2014	SP	offline				x						TREC 8	FQA
Liu et al. [73]	2013	UTTP	both					x				x	BZ	CBQA
Waltinger et al. [124]	2012	SP	both				x		x				CLEF-2007, SW	CBQA
Latcinnik et al. [61]	2020	UTTP	both				x			x			ComSQA, QASC	CBQA
Riley et al. [106]	2019	SDDT	offline	x			x						SS, TS, RA	VQA
Moon et al. [85]	2019	SP	offline					x					MemQA	FQA
Zhang et al. [149]	2019	UTTP	offline				x	x					S-C, SG	cQA
Nie et al. [89]	2019	SP	both	x						x	x		NCD, BC, LM-1B, NC, WSC-G, COPA	CBQA
Li et al. [68]	2018	UTTP	offline			x		x					VQA-E	VQA
Li et al. [66]	2018	UTTP	offline				x						VQA-Real	VQA
Su et al. [119]	2018	SP	both				x		x		x		NL2API	CBQA
Li et al. [64]	2014	SDDT	online						x			x	MASD	FQA
Damljanovic et al. [23]	2010	SP	offline	x			x	x					MGD	FQA
Wu et al. [133]	2020	UTTP	offline	x		x	x						SQ	FQA

4.2 Dataset

There are a large number of data resources in the literature that are focused on the implementation and evaluation of QAS. Depending on the QA modeling technique and the particular QA task covered by the IQAS, certain datasets produce findings that are more appropriate.

According to the literature, the most frequent used datasets for building, testing, and evaluating an IQAS are shown in the table 5. For the interest of completeness, we present a comprehensive list of datasets found in the literature. Each is complemented with a concise explanation of its content and structure.

Table 5. Main relevant datasets exploited in IQASs literature grouped by tasks and QAM model types.

Task	Modeling Approach	Dataset
cQA	Data Driven Instruction Based	Yahoo!, StackExchange
CB QA	Data Driven Instruction Based	Domain Dependant Datasets
MRC QA	Data Driven Instruction Based	CoQA, SQUAD, TriviaQA, SearchQA CLEF, QuAC, SQUAD
KB QA	Data Driven Instruction Based	SQUAD, CSQA WebQesitons, QALD-N, SimpleQuestions
Visual QA	Data Driven	VQA, VQA2.0, TDIUC, COCO-QA

TREC a collection of datasets published at the Text REtrieval Conference. With TREC dataset we refer to all data collections belonging to the homonym QA challenges available on this [link](#). These datasets contains questions and response patterns, as well as a pool of documents returned by competing teams [125].

Yahoo! (YH) identifies a set of information obtained from the website Yahoo! Answers hosted by [Yahoo!](#). The dataset contains 10 million question pages of Yahoo! Answers and 100.000 user queries about them [151].

Stack Exchange (StEx) is a [web platform](#) of exchanging information among users through the question-answer modality. The dataset collects over 7 thousand questions posted on the website equipped with related answers ordered by their relevance [3].

WikiQA is realized by Yang et al. [140] to give researchers a valuable source of data for the open-domain QA task. The main feature of this collection is the proposal of the answer triggering task, which identifies the possible absence of the correct answer among the documents associated with each question. The structure is similar to previous datasets (questions with a set of sentences ordered by their correctness), and it is available at this [link](#).

Visual Genome (VG) hosts data enabling to answer questions about objects depicted in images [56]. The data collection is proposed to solve the VQA task, collecting dense annotations of objects, attributes and relationships for each image. It is available on the [author's website](#).

Visual Question Answering (VQA) is a dataset firstly proposed by Antol et al. [4] to define the VQA task. Their dataset holds for each question a related image and a set of equally appropriate answers expressed in different form. A second version (VQA 2.0) is realized by Goyal et al. [36] to overcome the language biases issue that characterizes the first version. In fact, authors demonstrate that systems trained on the previous dataset version tend to ignore the visual features due to language priors. However, both datasets are available at this [link](#).

Question Answering over Linked Data (QALD) is a collection of challenges designed to test the QA task over the Linked Data. The first appearance of this type of dataset dates back to the work of Lopez et al. [75]. All the resources are available on the QALD [web platform](#), which hosts at this time 9 QALD challenges.

WebQuestions (WQ) is a set of question-answers pairs built by Berant et al. [7]. Starting from a single factoid question, they exploited the Google Suggest API to retrieve further questions related to the first one and link a set of possible answering entities to each question. This dataset is available at this [link](#). It also available at this [link](#) the semantic parsed (SP) version of WQ (i.e. WebQuestionsSP), which contains testing values of gathering SPARQL queries and answers for each question of the original dataset [143].

GraphQesitons (GQ) holds more than 5000 logical form-question pairs associated with answers from knowledge base to enable fine-grained analysis of QASs. These queries have been automatically generated by Su et al. [120]

and then refined by human operators to reduce redundancy and commonness. The whole dataset is made available at the following [GitHub page](#).

WikiPassageQA (WPQA) is a Wikipedia based collection specific for non factoid answer passage retrieval produced by Cohen et al. [21]. It contains thousands of questions with annotated answers queried by human workers based on Wikipedia document. The whole collection is available at this [platform](#).

InsuranceQA represents a data set collecting question and answer pairs from the web in the insurance domain. It was released on the following [GitHub page](#) by the work of Feng et al. [31].

QAit is a dataset designed by Yuan et al. [144] focused on procedural knowledge to answer the user questions. In fact, the answering procedure is here treated as a text game, where agents can interact with the environment (i.e. textual descriptions) to reach the answer (i.e. objects). Replies are generated as actions sequences, which depend on the environment where the question is posed. The data resource is made available on this [GitHub page](#).

Task Driven Image Understanding Challenge (TDIUC) identifies a data collection hosting more than 1.6 million questions organized in 12 different categories. It was built by Kafle et al. [51] to overcome the unfair evaluation of different algorithm abilities (e.g., object detection, object and attribute classification, positional reasoning, counting) by grouping samples in dedicated categories. The dataset is available at this [web page](#).

Visual7W is proposed by Zhu et al. [154] by establishing a semantic link between textual descriptions and image regions by object-level grounding. They enable VQAs to return visual answers besides textual ones, organizing the dataset in a set of questions-images-multi-choice answers triples, where each answer refers to a specific image region. Authors hold their data resource at this [page](#).

TriviaQA is produced by Joshi et al. [49] as a MRC QA dataset with more than 650 thousand question-answer-evidence documents triples. Their goal is to generate a resource that test systems on their potential bias about question style or content, requiring them to select the best documents that support answers they compute. The dataset is available at this [link](#).

SearchQA is a large-scale dataset for the MRC QA task published by Dunn et al. [28]. The collection is generated starting from existing question-answer pairs and augmenting it with text snippets retrieved by Google. Their resource hosts more than 140 thousand question-answer pairs, each of them associated on average with 49.6 snippets, and it is available on the following [GitHub page](#).

Quasar-T is built by Dhingra et al. [26] from the software entity tags on the Stack Overflow ⁵ website. Each dataset record includes a question, a relevant context document, a set of candidate solutions and the correct one. It is specifically designed for the MRC QA task and it is publicly available at this [GitHub page](#).

Stanford Question Answering Dataset (SQuAD) is a MRC dataset, consisting of questions about the Wikipedia articles. Answers are extracted as a segment of text (i.e. span) from the passage related to each question. The first version was published by Rajpurkar et al. [103] to address the need of large and high-quality resource for the MRC QA task. The dataset was updated to the 2.0 version (SQuAD 2.0) by Rajpurkar et al. [102] to include unanswerable questions that look similar to answerable ones. Both the dataset are available on this [link](#).

TGIF-QA is a collection proposed by Jang et al. [47] to address the video QA task, which require spatial-temporal reasoning from videos to answer questions correctly. They extended the Tumblr GIF dataset [70] with more than 100 thousand query-reply pairs. The dataset is accessible on this [GitHub page](#).

⁵<https://stackoverflow.com/>

Microsoft Research Video Description Corous (MSVD-QA) is a dataset introduced by Chen et al. [15] that consists of 120 thousand sentences summarizing actions in short video snippets, available on this [link](#).

MSRVTT-QA is a large-scale video benchmark for video understanding by Xu et al. [136]. The collection cover the video to text task, with 10 thousand web video clips and 200 thousand clip-sentence pairs. Each clip is annotated with about 20 natural language sentences thanks to human operator. This dataset is available at this [link](#).

Chinese Medical Question Answer dataset (cMedQA) is built by Zhang et al. [146] to cover a medical question answering task in Chinese. The resource gets their questions and answers from online Chinese health community and it is hosted on this [GitHub page](#). A second version (cMedQA 2.0) is produced by Zhang et al. [147] by expanding and cleaning the question-answer pairs of the previous one and is available at this [GitHub page](#).

COCO-QA holds question-answer pairs for the VQA task obtained through a question generation algorithm proposed by Ren et al. [105]. It converts image descriptions into QA pairs, then rejecting those ones that appear too rarely or too often. The overall collection is hosted on the following [website](#).

Interactive Question Answering Dataset (IQUAD v1) identifies a set of simulated environments designed by Gordon et al. [35] to evaluate agents in answering questions through interactions with surrounding objects. The dataset contains more than 75 thousand multiple choice questions, each one equipped with an unique scene configuration. Authors published their resource on this [GitHub page](#).

SimpleQuestions (SQ) is a large-scale dataset based on Freebase built by Bordes et al. [12]. It holds more than 100 thousand questions written by human annotators and associated to Freebase facts. The goal is to evaluate systems, trained on complex reasoning, solving easy to answer questions. It is available on this [GitHub page](#).

SemEval-2015 is a challenge designed by Nakov et al. [88] to leverage the answer selection for cQASs. Authors select Arabic and English question-answers collections from two relatively communities, then labelled by human operators to be available on the SemEval Task 3 challenge, available on this [link](#).

bAbI dataset is realized by Weston et al. [127] to test systems to answer questions via chaining facts, simple induction or deduction. The collection has different independent test cases, that can be eventually merged to evaluate specific scenario. The whole work is publicly available on this [website](#).

IQA dataset (ibAbI) is designed by Li et al. [65] to extend the bAbI dataset by adding interactive scenario to the QA task. They simulate three different representative scenarios of incomplete or ambiguous information, called as ambiguous actor problem, ambiguous object problem and the missing information problem. It adopts the same standard of bAbI dataset and it is available at this [link](#).

Question Answering via Sentence Composition dataset (QASC) produced by Khot et al. [53] is a multi-hop reasoning dataset that requires retrieving and composing facts to answer a multiple-choice question. Each entry is composed by a question, a set of answers enclosing a single correct one and a set of supporting facts on which the system must rely to response the question. The dataset is available at this [link](#).

Commonsense QA is designed by Talmor et al. [55] to investigate question answering with prior knowledge. It collects a set of commonsense questions relying on concept and complex semantic relations to be answered. The entire dataset is hosted on this [website](#).

OpenBookQA (OBQA) is a dataset about elementary level science facts built by Mihaylov et al. [84]. This work focuses on evaluating the ability of QASs to leverage specific languages and understanding related to the science domain. The collection is available at this [link](#).

A12 Reasoning Challenge dataset (ARC) is designed to require systems powerful knowledge and reasoning capabilities to answer questions. Sabharwal et al. [20] divided the collection in a challenge set and an easy one.

The former contains only questions answered incorrectly by the majority of QASs to stimulate the community in designing improved systems. The dataset is available on this [website](#).

Pororo QA is a dataset containing more than 16 thousand scene-dialogue pairs, each one with fine-grained sentences for scene description and story related QA pairs. The resource was built by Kim et al. [54] to perform video story QA by learning from a large amount of cartoon videos. It is available on this [GitHub page](#).

TVQA is a large-scale video QA dataset base on 6 popular TV shows. Questions are designed by Lei et al. [63] to be compositional in nature, requiring systems to recognize relevant moments within clips, comprehend subtitles and visual concepts to adequately respond. The resource is available at this [link](#).

HotpotQA by Yang et al. [142] is a Wikipedia-based QA dataset whose questions require searching and reasoning over supporting documents to be answered. To this, sentence-level supporting facts required for reasoning are given, which enables strong supervision and explanations. The resource is available on this [website](#).

Question Answering in Context (QuAC) is a collection of 14 thousand information-seeking QA dialogues designed by Choi et al. [17] to deal with real world questions that are usually open-ended, unanswerable or meaningful within a dialogue context. The dataset is available on this [website](#).

Conversational Question Answering dataset (CoQA) by Reddy et al. [104] contains 127 thousand QA pairs obtained from 8 thousand conversations about text passages of specific domains. Questions are presented in a conversational form while answers are free-form text with corresponding evidence highlighted in the passage. This frequently used collection to test CoQASs is available on the following [GitHub page](#).

Complex Sequential QA dataset (CSQA) is designed by Saha et al. [108] to combine the task of answering factual questions through complex inference over realistic-sized KG and learning to converse through a series of coherently linked QA pairs. It contains around 200 thousand of dialogues with a total of 1.5 million turns. The dataset is available on this [web page](#).

5 CONCLUSION

In conclusion, we have reviewed a substantial collection of interactive question answering systems (IQASs)-related literature published during the past decade. We discovered the literature to be diverse, beginning with adopted methodologies for addressing multiple QA tasks and concluding with a vast array of diverse resources (i.e. knowledge source, and datasets) that are typically utilized to create and evaluate question answering systems (QASs). Despite the fact that the state-of-the-art is defined by several types of QA solutions, we were able to determine the characteristics shared by the suggested systems that constitute a shared framework. To the best of our knowledge, we are the first to present a unified and comprehensive design that emphasizes the fundamental components and functions of IQASs. For each component, we have performed an in-depth categorization of the literature from the methodological and application perspectives, categorizing the works by tasks, difficulties, and interaction modes. In addition, in order to address the demands of the IQAS community, we give explicit definitions of the implementation goals and features of IQASs. To achieve this goal, we have categorized QASs based on their behaviors and enabled interaction so that the whole IQASs landscape may be characterized using these definitions and features. Then, we detailed trends regarding specific tasks and problems, demonstrating the community's keen interest in enhancing the system's performance and openness. Lastly, we have included a basic classification of evaluation approaches often used in the literature to assess QASs, together with a full list of datasets used in the evaluation process, general descriptions, and links to the authors' hosting platforms.

Despite the fact that the state-of-the-art examined in this survey comprises a large number of solutions encompassing a wide range of IQAS elements and characteristics, we believe there is still considerable potential for advancement as future challenges. In particular, based on the developing patterns from the literature analysis, the QA community is headed toward four significant problems. A first step toward the unification of conversational search and question answering is the development of new dialog-based search and answer algorithms. This condition is a direct result of the blurring of the lines between question-answering systems and information retrieval that is becoming increasingly apparent. The second difficulty follows the never-ending efforts to enhance the AI systems' comprehension of human semantics and ontology. This line focuses mostly on the development of techniques to extract relevant data from disparate data sources i.e., *multi-modality*. It pursues a third future trend that seeks more natural and multi-modal interfaces/interaction strategies to enhance the user's enjoyment of the IQAS. The final objective of future research will be to compile a comprehensive dataset that can be used to evaluate all IQAS kinds explored thus far under the constraints and tasks outlined in this literature review.

REFERENCES

- [1] Kamran Alipour, Jürgen P. Schulze, Yi Yao, Avi Ziskind, and Giedrius Burachas. 2020. A Study on Multimodal and Interactive Explanations for Visual Question Answering. In *SafeAI@AAAI (CEUR Workshop Proceedings, Vol. 2560)*. CEUR-WS.org, 54–62.
- [2] Francesca Alloatti, Luigi Di Caro, and Gianpiero Sportelli. 2019. Real Life Application of a Question Answering System Using BERT Language Model. In *SIGdial*. Association for Computational Linguistics, 250–253.
- [3] Ashton Anderson, Daniel Huttenlocher, Jon Kleinberg, and Jure Leskovec. 2012. Discovering value from community activity on focused question answering sites: a case study of stack overflow. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. 850–858.
- [4] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*. 2425–2433.
- [5] Ashutosh Baheti, Alan Ritter, and Kevin Small. 2020. Fluent Response Generation for Conversational Question Answering. In *ACL*. Association for Computational Linguistics, 191–207.
- [6] Kinjal Basu. 2019. Conversational AI : Open Domain Question Answering and Commonsense Reasoning. In *ICLP Technical Communications (EPTCS, Vol. 306)*. 396–402.
- [7] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*. 1533–1544.
- [8] Ana Berdasco, Gustavo López, Ignacio Díaz-Oreiro, Luis Quesada, and Luis A. Guerrero. 2019. User Experience Comparison of Intelligent Personal Assistants: Alexa, Google Assistant, Siri and Cortana. In *UCAmI (MDPI Proceedings, Vol. 31)*. MDPI, 51.
- [9] Guillaume Le Berre and Philippe Langlais. 2020. Attending Knowledge Facts with BERT-like Models in Question-Answering: Disappointing Results and Some Explanations. In *Canadian Conference on AI (Lecture Notes in Computer Science, Vol. 12109)*. Springer, 356–367.
- [10] Santanu Bhattacharjee, Rejwanul Haque, Gideon Maillette de Buy Wenniger, and Andy Way. 2020. Investigating Query Expansion and Coreference Resolution in Question Answering on BERT. In *NLDB (Lecture Notes in Computer Science, Vol. 12089)*. Springer, 47–59.
- [11] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. 1247–1250.
- [12] Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. 2015. Large-scale Simple Question Answering with Memory Networks. *CoRR* abs/1506.02075 (2015).
- [13] Rakesh Chada. 2019. Gendered Pronoun Resolution using BERT and an extractive question answering formulation. *CoRR* abs/1906.03695 (2019).
- [14] Adriane Chapman and H. V. Jagadish. 2009. Why not?. In *SIGMOD Conference*. ACM, 523–534.
- [15] David L. Chen and William B. Dolan. 2011. Collecting Highly Parallel Data for Paraphrase Evaluation. In *ACL*. The Association for Computer Linguistics, 190–200.
- [16] Ting-Rui Chiang, Hao-Tong Ye, and Yun-Nung Chen. 2020. An Empirical Study of Content Understanding in Conversational Question Answering. In *AAAI*. AAAI Press, 7578–7585.
- [17] Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. QuAC: Question answering in context. *arXiv preprint arXiv:1808.07036* (2018).
- [18] Philipp Christmann, Rishiraj Saha Roy, Abdalghani Abujabal, Jyotsna Singh, and Gerhard Weikum. 2019. Look before you Hop: Conversational Question Answering over Knowledge Graphs Using Judicious Context Expansion. In *CIKM*. ACM, 729–738.
- [19] Kevin Clark and Christopher D. Manning. 2016. Improving Coreference Resolution by Learning Entity-Level Distributed Representations. In *Association for Computational Linguistics (ACL)*.

- [20] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457* (2018).
- [21] Daniel Cohen, Liu Yang, and W Bruce Croft. 2018. WikiPassageQA: A benchmark collection for research on non-factoid answer passage retrieval. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 1165–1168.
- [22] Sérgio Curto, Ana Cristina Mendes, Pedro Curto, Luisa Coheur, and Ângela Costa. 2014. JUST.ASK, a QA system that learns to answer new questions from previous interactions. In *LREC*. European Language Resources Association (ELRA), 2603–2607.
- [23] Danica Damjanovic, Milan Agatonovic, and Hamish Cunningham. 2010. Natural Language Interfaces to Ontologies: Combining Syntactic Analysis and Ontology-Based Lookup through the User Interaction. In *ESWC (1) (Lecture Notes in Computer Science, Vol. 6088)*. Springer, 106–120.
- [24] Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, and Andrew McCallum. 2019. Multi-step Retriever-Reader Interaction for Scalable Open-domain Question Answering. In *ICLR (Poster)*. OpenReview.net.
- [25] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
- [26] Bhuwan Dhingra, Kathryn Mazaitis, and William W Cohen. 2017. Quasar: Datasets for question answering by search and reading. *arXiv preprint arXiv:1707.03904* (2017).
- [27] Tuong Do, Huy Tran, Thanh-Toan Do, Erman Tjiputra, and Quang D. Tran. 2019. Compact Trilinear Interaction for Visual Question Answering. In *ICCV*. IEEE, 392–401.
- [28] Matthew Dunn, Levent Sagun, Mike Higgins, V Ugur Guney, Volkan Cirik, and Kyunghyun Cho. 2017. Searchqa: A new q&a dataset augmented with context from a search engine. *arXiv preprint arXiv:1704.05179* (2017).
- [29] Oren Etzioni. 2011. Search needs a shake-up. *Nat.* 476, 7358 (2011), 25–26.
- [30] Ronald Fagin, Amnon Lotem, and Moni Naor. 2003. Optimal aggregation algorithms for middleware. *Journal of computer and system sciences* 66, 4 (2003), 614–656.
- [31] Minwei Feng, Bing Xiang, Michael R Glass, Lidan Wang, and Bowen Zhou. 2015. Applying deep learning to answer selection: A study and an open task. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. 813–820.
- [32] Antonio Ferrández and Jesús Peral. 2010. The benefits of the interaction between data warehouses and question answering. In *EDBT/ICDT Workshops (ACM International Conference Proceeding Series)*. ACM.
- [33] Jun-ichi Fukumoto, Noriaki Aburai, and Ryosuke Yamanishi. 2013. Interactive Document Expansion for Answer Extraction of Question Answering System. In *KES (Procedia Computer Science, Vol. 22)*. Elsevier, 991–1000.
- [34] Peng Gao, Haoxuan You, Zhanpeng Zhang, Xiaogang Wang, and Hongsheng Li. 2019. Multi-Modality Latent Interaction Network for Visual Question Answering. In *ICCV*. IEEE, 5824–5834.
- [35] Daniel Gordon, Aniruddha Kembhavi, Mohammad Rastegari, Joseph Redmon, Dieter Fox, and Ali Farhadi. 2018. IQA: Visual Question Answering in Interactive Environments. In *CVPR*. Computer Vision Foundation / IEEE Computer Society, 4089–4098.
- [36] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 6904–6913.
- [37] Daya Guo, Duyu Tang, Nan Duan, Ming Zhou, and Jian Yin. 2018. Dialog-to-Action: Conversational Question Answering Over a Large-Scale Knowledge Base. In *NeurIPS 2018*. 2946–2955.
- [38] Maryam Habibi, Parvaz Mahdabi, and Andrei Popescu-Belis. 2016. Question answering in conversations: Query refinement using contextual and semantic information. *Data Knowl. Eng.* 106 (2016), 38–51.
- [39] Hojae Han, Seungtaek Choi, Haeju Park, and Seung-won Hwang. 2019. MICRON: Multigranular Interaction for Contextualizing RepresentatiON in Non-factoid Question Answering. In *EMNLP/IJCNLP (1)*. Association for Computational Linguistics, 5889–5894.
- [40] Sofian Hazrina, Nurfadhilina Mohd Sharef, Hamidah Ibrahim, Masrah Azrifah Azmi Murad, and Shahrul Azman Mohd. Noah. 2017. Review on the advancements of disambiguation in semantic question answering system. *Inf. Process. Manag.* 53, 1 (2017), 52–69.
- [41] Shizhu He, Kang Liu, and Weiting An. 2019. Learning to Align Question and Answer Utterances in Customer Service Conversation with Recurrent Pointer Networks. In *AAAI*. AAAI Press, 134–141.
- [42] Lynette Hirschman and Robert J. Gaizauskas. 2001. Natural language question answering: the view from here. *Nat. Lang. Eng.* 7, 4 (2001), 275–300.
- [43] Yining Hong, Jialu Wang, Yuting Jia, Weinan Zhang, and Xinbing Wang. 2019. Academic Reader: An Interactive Question Answering System on Academic Literatures. In *AAAI*. AAAI Press, 9855–9856.
- [44] Jun Hu, Shengsheng Qian, Quan Fang, and Changsheng Xu. 2018. Attentive Interactive Convolutional Matching for Community Question Answering in Social Multimedia. In *ACM Multimedia*. ACM, 456–464.
- [45] Hsin-Yuan Huang, Eunsol Choi, and Wen-tau Yih. 2018. Flowqa: Grasping flow in history for conversational machine comprehension. *arXiv preprint arXiv:1810.06683* (2018).
- [46] Eric Hultburd. 2020. Exploring BERT Parameter Efficiency on the Stanford Question Answering Dataset v2.0. *CoRR* abs/2002.10670 (2020).
- [47] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. 2017. TGIF-QA: Toward Spatio-Temporal Reasoning in Visual Question Answering. In *CVPR*. IEEE Computer Society, 1359–1367.
- [48] Weike Jin, Zhou Zhao, Mao Gu, Jun Yu, Jun Xiao, and Yueting Zhuang. 2019. Multi-interaction Network with Object Relation for Video Question Answering. In *ACM Multimedia*. ACM, 1193–1201.

- [49] Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551* (2017).
- [50] Ying Ju, Fubang Zhao, Shijie Chen, Bowen Zheng, Xuefeng Yang, and Yunfeng Liu. 2019. Technical report on Conversational Question Answering. *CoRR abs/1909.10772* (2019).
- [51] Kushal Kafle and Christopher Kanan. 2017. An analysis of visual question answering algorithms. In *Proceedings of the IEEE International Conference on Computer Vision*. 1965–1973.
- [52] Magdalena Kaiser, Rishiraj Saha Roy, and Gerhard Weikum. 2020. Conversational Question Answering over Passages by Leveraging Word Proximity Networks. In *SIGIR*. ACM, 2129–2132.
- [53] Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. Qasc: A dataset for question answering via sentence composition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 8082–8090.
- [54] Kyung-Min Kim, Min-Oh Heo, Seong-Ho Choi, and Byoung-Tak Zhang. 2017. Deepstory: Video story qa by deep embedded memory networks. *arXiv preprint arXiv:1707.00836* (2017).
- [55] Natalia Konstantinova and Constantin Orasan. 2013. Interactive question answering. In *Emerging applications of natural language processing: concepts and new research*. IGI Global, 149–169.
- [56] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision* 123, 1 (2017), 32–73.
- [57] Mayank Kulkarni and Kristy Elizabeth Boyer. 2018. Toward Data-Driven Tutorial Question Answering with Deep Learning Conversational Models. In *BEA@NAACL-HLT*. Association for Computational Linguistics, 273–283.
- [58] Girish Kumar, Matthew Henderson, Shannon Chan, Hoang Nguyen, and Lucas Ngoo. 2018. Question-Answer Selection in User to User Marketplace Conversations. In *IWSDS (Lecture Notes in Electrical Engineering, Vol. 579)*. Springer, 397–403.
- [59] Souvik Kundu, Qian Lin, and Hwee Tou Ng. 2020. Learning to Identify Follow-Up Questions in Conversational Question Answering. In *ACL*. Association for Computational Linguistics, 959–968.
- [60] Chia-Chih Kuo, Shang-Bao Luo, and Kuan-Yu Chen. 2020. An Audio-Enriched BERT-Based Framework for Spoken Multiple-Choice Question Answering. In *INTERSPEECH*. ISCA, 4173–4177.
- [61] Veronica Latcinnik and Jonathan Berant. 2020. Explaining Question Answering Models through Text Generation. *CoRR abs/2004.05569* (2020).
- [62] Changyoon Lee, Donghoon Han, Hyoungho Jin, and Alice Oh. 2019. automaTA: Human-Machine Interaction for Answering Context-Specific Questions. In *L@S*. ACM, 44:1–44:4.
- [63] Jie Lei, Licheng Yu, Mohit Bansal, and Tamara L Berg. 2018. Tvqa: Localized, compositional video question answering. *arXiv preprint arXiv:1809.01696* (2018).
- [64] Fei Li and H. V. Jagadish. 2014. Constructing an Interactive Natural Language Interface for Relational Databases. *Proc. VLDB Endow.* 8, 1 (2014), 73–84.
- [65] Huayun Li, Martin Renqiang Min, Yong Ge, and Asim Kadav. 2017. A Context-aware Attention Network for Interactive Question Answering. In *KDD*. ACM, 927–935.
- [66] Qing Li, Jianlong Fu, Dongfei Yu, Tao Mei, and Jiebo Luo. 2018. Tell-and-Answer: Towards Explainable Visual Question Answering using Attributes and Captions. In *EMNLP*. Association for Computational Linguistics, 1338–1346.
- [67] Qian Li, Hui Su, Cheng Niu, Daling Wang, Zekang Li, Shi Feng, and Yifei Zhang. 2019. Answer-Supervised Question Reformulation for Enhancing Conversational Machine Comprehension. In *MRQA@EMNLP*. Association for Computational Linguistics, 38–47.
- [68] Qing Li, Qingyi Tao, Shafiq R. Joty, Jianfei Cai, and Jiebo Luo. 2018. VQA-E: Explaining, Elaborating, and Enhancing Your Answers for Visual Questions. In *ECCV (7) (Lecture Notes in Computer Science, Vol. 11211)*. Springer, 570–586.
- [69] Ronghan Li, Zejun Jiang, Lifang Wang, Xinyu Lu, and Meng Zhao. 2020. Directional attention weaving for text-grounded conversational question answering. *Neurocomputing* 391 (2020), 13–24.
- [70] Yuncheng Li, Yale Song, Liangliang Cao, Joel R. Tetreault, Larry Goldberg, Alejandro Jaimes, and Jiebo Luo. 2016. TGIF: A New Dataset and Benchmark on Animated GIF Description. In *CVPR*. IEEE Computer Society, 4641–4650.
- [71] Aiting Liu, Ziqi Huang, Hengtong Lu, Xiaojie Wang, and Caixia Yuan. 2019. BB-KBQA: BERT-Based Knowledge Base Question Answering. In *CCL (Lecture Notes in Computer Science, Vol. 11856)*. Springer, 81–92.
- [72] Siyuan Liu, Sourav S. Bhowmick, Wanlu Zhang, Shu Wang, and Wanyi Huang. 2018. NEURON: An Interactive Natural Language Interface for Understanding Query Execution Plans in RDBMS. *CoRR abs/1805.05670* (2018).
- [73] Song Liu, Yixin Zhong, and Fuji Ren. 2013. Interactive Question Answering Based on FAQ. In *CCL (Lecture Notes in Computer Science, Vol. 8202)*. Springer, 73–84.
- [74] James Lockett, Sanith Wijesinghe, Jasper Phillips, Ian Gross, Michael Schoenfeld, Walter T. Hiranpat, Phillip J. Marlow, Matt Coarr, and Qian Hu. 2019. Intelligent Voice Agent and Service (iVAS) for Interactive and Multimodal Question and Answers. In *FQAS (Lecture Notes in Computer Science, Vol. 11529)*. Springer, 396–402.
- [75] Vanessa Lopez, Christina Unger, Philipp Cimiano, and Enrico Motta. 2013. Evaluating question answering over linked data. *Journal of Web Semantics* 21 (2013), 3–13.
- [76] Yoelle Maarek. 2018. Alexa and Her Shopping Journey. In *CIKM*. ACM, 1.

- [77] Anutosh Maitra, Shivam Garg, and Shubhashis Sengupta. 2020. Enabling Interactive Answering of Procedural Questions. In *NLDB (Lecture Notes in Computer Science, Vol. 12089)*. Springer, 73–81.
- [78] Angrosh Mandya, Danushka Bollegala, and Frans Coenen. 2019. Evaluating Co-reference Chains Based Conversation History in Conversational Question Answering. In *PACLING (Communications in Computer and Information Science, Vol. 1215)*. Springer, 280–292.
- [79] Angrosh Mandya, James O'Neill, Danushka Bollegala, and Frans Coenen. 2020. Do not let the history haunt you: Mitigating Compounding Errors in Conversational Question Answering. In *LREC*. European Language Resources Association, 2017–2025.
- [80] Yosi Mass, Haggai Roitman, Shai Erera, Or Rivlin, Bar Weiner, and David Konopnicki. 2019. A Study of BERT for Non-Factoid Question-Answering under Passage Length Constraints. *CoRR abs/1908.06780* (2019).
- [81] J. S. McCarley. 2019. Pruning a BERT-based Question Answering Model. *CoRR abs/1910.06360* (2019).
- [82] Pablo N Mendes, Max Jakob, and Christian Bizer. 2012. *DBpedia: A multilingual cross-domain knowledge base*. European Language Resources Association (ELRA).
- [83] Tamás Mészáros and Tadeusz P. Dobrowiecki. 2017. Agent-based Reconfigurable Natural Language Interface to Robots - Human-Agent Interaction using Task-specific Controlled Natural Languages. In *ICAART (2)*. SciTePress, 632–639.
- [84] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789* (2018).
- [85] Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. 2019. Memory Graph Networks for Explainable Memory-grounded Question Answering. In *CoNLL*. Association for Computational Linguistics, 728–736.
- [86] Thomas Müller, Francesco Piccinno, Peter Shaw, Massimo Nicosia, and Yasemin Altun. 2019. Answering Conversational Questions on Structured Data without Logical Forms. In *EMNLP/IJCNLP (1)*. Association for Computational Linguistics, 5901–5909.
- [87] M. Asif Naeem, Saif Ullah, and Imran Sarwar Bajwa. 2012. Interacting with Data Warehouse by Using a Natural Language Interface. In *NLDB (Lecture Notes in Computer Science, Vol. 7337)*. Springer, 372–377.
- [88] Preslav Nakov, Lluís Màrquez, Walid Magdy, Alessandro Moschitti, James Glass, and Bilal Randeree. 2019. Semeval-2015 task 3: Answer selection in community question answering. *arXiv preprint arXiv:1911.11403* (2019).
- [89] Allen Nie, Erin D. Bennett, and Noah D. Goodman. 2019. Learning to Explain: Answering Why-Questions via Rephrasing. *CoRR abs/1906.01243* (2019).
- [90] Florian Nothdurft and Wolfgang Minker. 2016. Justification and transparency explanations in dialogue systems to maintain human-computer trust. In *Situated Dialog in Speech-Based Human-Computer Interaction*. Springer, 41–50.
- [91] Reham A. Osama, Nagwa M. El-Makky, and Marwan Torki. 2019. Question Answering Using Hierarchical Attention on Top of BERT Features. In *MRQA@EMNLP*. Association for Computational Linguistics, 191–195.
- [92] Arantxa Otegi, Aitor Gonzalez-Agirre, Jon Ander Campos, Aitor Soroa, and Eneko Agirre. 2020. Conversational Question Answering in Low Resource Scenarios: A Dataset and Case Study for Basque. In *LREC*. European Language Resources Association, 436–442.
- [93] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *ACL*. ACL, 311–318.
- [94] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.
- [95] Rivindu Perera and Parma Nand. 2014. Interaction History Based Answer Formulation for Question Answering. In *KESW (Communications in Computer and Information Science, Vol. 468)*. Springer, 128–139.
- [96] Volha Petukhova, Desmond Darma Putra, Alexandr Chernov, and Dietrich Klakow. 2015. Understanding Questions and Extracting Answers: Interactive Quiz Game Application Design. In *LTC (Lecture Notes in Computer Science, Vol. 10930)*. Springer, 246–261.
- [97] Akshit Pradhan, Pragya Shukla, Pallavi Patra, Rohit Pathak, and Ajay Kumar Jena. 2018. Enhancing Interaction with Social Networking Sites for Visually Impaired People by Using Textual and Visual Question Answering. In *CICBA (2) (Communications in Computer and Information Science, Vol. 1031)*. Springer, 3–14.
- [98] Zihao Qi, Dario Bertero, Ian D. Wood, and Pascale Fung. 2019. Incorporate User Representation for Personal Question Answer Selection Using Siamese Network. In *ICASSP*. IEEE, 7540–7544.
- [99] Chen Qu, Liu Yang, Cen Chen, Minghui Qiu, W. Bruce Croft, and Mohit Iyyer. 2020. Open-Retrieval Conversational Question Answering. In *SIGIR*. ACM, 539–548.
- [100] Chen Qu, Liu Yang, Minghui Qiu, W. Bruce Croft, Yongfeng Zhang, and Mohit Iyyer. 2019. BERT with History Answer Embedding for Conversational Question Answering. In *SIGIR*. ACM, 1133–1136.
- [101] Chen Qu, Liu Yang, Minghui Qiu, Yongfeng Zhang, Cen Chen, W. Bruce Croft, and Mohit Iyyer. 2019. Attentive History Selection for Conversational Question Answering. In *CIKM*. ACM, 1391–1400.
- [102] Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. In *ACL (2)*. Association for Computational Linguistics, 784–789.
- [103] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100, 000+ Questions for Machine Comprehension of Text. In *EMNLP*. The Association for Computational Linguistics, 2383–2392.
- [104] Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A Conversational Question Answering Challenge. *Trans. Assoc. Comput. Linguistics* 7 (2019), 249–266.

- [105] Mengye Ren, Ryan Kiros, and Richard S. Zemel. 2015. Exploring Models and Data for Image Question Answering. In *NIPS*. 2953–2961.
- [106] Heather Riley and Mohan Sridharan. 2019. Integrating Non-monotonic Logical Reasoning and Inductive Learning With Deep Learning for Explainable Visual Question Answering. *Frontiers Robotics AI* 6 (2019), 125.
- [107] Andreas Rücklé and Iryna Gurevych. 2017. End-to-End Non-Factoid Question Answering with an Interactive Visualization of Neural Attention Weights. In *ACL (System Demonstrations)*. Association for Computational Linguistics, 19–24.
- [108] Amrita Saha, Vardaan Pahuja, Mitesh M. Khapra, Karthik Sankaranarayanan, and Sarath Chandar. 2018. Complex Sequential Question Answering: Towards Learning to Converse Over Linked Question Answer Pairs with a Knowledge Graph. *arXiv:1801.10314* (2018).
- [109] Wataru Sakata, Tomohide Shibata, Ribeka Tanaka, and Sadao Kurohashi. 2019. FAQ Retrieval using Query-Question Similarity and BERT-Based Query-Answer Relevance. In *SIGIR*. ACM, 1113–1116.
- [110] Malte Schwarzer, Jonas Düver, Danuta Ploch, and Andreas Lommatzsch. 2016. An Interactive e-Government Question Answering System. In *LWDA (CEUR Workshop Proceedings, Vol. 1670)*. CEUR-WS.org, 74–82.
- [111] Bhaskar Sen, Nikhil Gopal, and Xinwei Xue. 2020. Support-BERT: Predicting Quality of Question-Answer Pairs in MSDN using Deep Bidirectional Transformer. *CoRR abs/2005.08294* (2020).
- [112] Huan Shao, Yunlong Xu, Yi Ji, Jianyu Yang, and Chunping Liu. 2019. Intra-Modality Feature Interaction Using Self-attention for Visual Question Answering. In *ICONIP (5) (Communications in Computer and Information Science, Vol. 1143)*. Springer, 215–222.
- [113] Tao Shen, Xiubo Geng, Tao Qin, Daya Guo, Duyu Tang, Nan Duan, Guodong Long, and Daxin Jiang. 2019. Multi-Task Learning for Conversational Question Answering over a Large-Scale Knowledge Base. In *EMNLP-IJCNLP (1)*. Association for Computational Linguistics, 2442–2451.
- [114] Lei Shi, Shijie Geng, Kai Shuang, Chiori Hori, Songxiang Liu, Peng Gao, and Sen Su. 2020. Multi-Layer Content Interaction Through Quaternion Product for Visual Question Answering. In *ICASSP*. IEEE, 4412–4416.
- [115] Andrew Shin, Yoshitaka Ushiku, and Tatsuya Harada. 2018. Customized Image Narrative Generation via Interactive Visual Question Generation and Answering. In *CVPR*. Computer Vision Foundation / IEEE Computer Society, 8925–8933.
- [116] Wissam Siblini, Charlotte Pasqual, Axel Lavielle, and Cyril Cauchois. 2019. Multilingual Question Answering from Formatted Text applied to Conversational Agents. *CoRR abs/1910.04659* (2019).
- [117] Daniil Sorokin and Iryna Gurevych. 2018. Interactive Instance-based Evaluation of Knowledge Base Question Answering. In *EMNLP (Demonstration)*. Association for Computational Linguistics, 114–119.
- [118] Lixin Su, Jiafeng Guo, Yixing Fan, Yanyan Lan, Ruqing Zhang, and Xueqi Cheng. 2019. An Adaptive Framework for Conversational Question Answering. In *AAAI*. AAAI Press, 10041–10042.
- [119] Yu Su, Ahmed Hassan Awadallah, Miaosen Wang, and Ryan W. White. 2018. Natural Language Interfaces with Fine-Grained User Interaction: A Case Study on Web APIs. In *SIGIR*. ACM, 855–864.
- [120] Yu Su, Huan Sun, Brian Sadler, Mudhakar Srivatsa, Izzeddin Gür, Zenghui Yan, and Xifeng Yan. 2016. On generating characteristic-rich question sets for qa evaluation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 562–572.
- [121] Hiroaki Sugiyama, Toyomi Meguro, and Ryuichiro Higashinaka. 2016. Evaluation of Question-Answering System About Conversational Agent's Personality. In *IWSDS (Lecture Notes in Electrical Engineering, Vol. 427)*. Springer, 183–194.
- [122] Svitlana Vakulenko, Shayne Longpre, Zhucheng Tu, and Raviteja Anantha. 2021. Question Rewriting for Conversational Question Answering. In *WSDM*. ACM, 355–363.
- [123] Betty van Aken, Benjamin Winter, Alexander Löser, and Felix A. Gers. 2019. How Does BERT Answer Questions?: A Layer-Wise Analysis of Transformer Representations. In *CIKM*. ACM, 1823–1832.
- [124] Ulli Waltinger, Alexa Breuing, and Ipke Wachsmuth. 2012. Connecting Question Answering and Conversational Agents - Contextualizing German Questions for Interactive Question Answering Systems. *Künstliche Intell.* 26, 4 (2012), 381–390.
- [125] Mengqiu Wang, Noah A Smith, and Teruko Mitamura. 2007. What is the Jeopardy model? A quasi-synchronous grammar for QA. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*. 22–32.
- [126] Zhiguo Wang, Patrick Ng, Xiaofei Ma, Ramesh Nallapati, and Bing Xiang. 2019. Multi-passage BERT: A Globally Normalized BERT Model for Open-domain Question Answering. In *EMNLP/IJCNLP (1)*. Association for Computational Linguistics, 5877–5881.
- [127] Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart Van Merriënboer, Armand Joulin, and Tomas Mikolov. 2015. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698* (2015).
- [128] Wilson Wong, Lawrence Cavedon, John Thangarajah, and Lin Padgham. 2012. Mixed-initiative conversational system using question-answer pairs mined from the web. In *CIKM*. ACM, 2707–2709.
- [129] Wilson Wong, Lawrence Cavedon, John Thangarajah, and Lin Padgham. 2012. Strategies for Mixed-Initiative Conversation Management using Question-Answer Pairs. In *COLING*. Indian Institute of Technology Bombay, 2821–2834.
- [130] Wilson Wong, John Thangarajah, and Lin Padgham. 2011. Health conversational system based on contextual matching of community-driven question-answer pairs. In *CIKM*. ACM, 2577–2580.
- [131] Fei Wu, Xinyu Duan, Jun Xiao, Zhou Zhao, Siliang Tang, Yin Zhang, and Yueting Zhuang. 2017. Temporal Interaction and Causal Influence in Community-Based Question Answering. *IEEE Trans. Knowl. Data Eng.* 29, 10 (2017), 2304–2317.
- [132] Jinmeng Wu, Tingting Mu, Jeyarajan Thiyagalingam, and John Yannis Goulermas. 2020. Building interactive sentence-aware representation based on generative language model for community question answering. *Neurocomputing* 389 (2020), 93–107.

- [133] Zhiyong Wu, Ben Kao, Tien-Hsuan Wu, Pengcheng Yin, and Qun Liu. 2020. PERQ: Predicting, Explaining, and Rectifying Failed Questions in KB-QA Systems. In *WSDM*. ACM, 663–671.
- [134] Zhipeng Xie. 2017. Enhancing Document-Based Question Answering via Interaction Between Question Words and POS Tags. In *NLPCC (Lecture Notes in Computer Science, Vol. 10619)*. Springer, 136–147.
- [135] Kun Xiong, Anqi Cui, Zefeng Zhang, and Ming Li. 2016. Neural Contextual Conversation Learning with Labeled Question-Answering Pairs. *CoRR* abs/1607.05809 (2016).
- [136] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. MSR-VTT: A Large Video Description Dataset for Bridging Video and Language. In *CVPR*. IEEE Computer Society, 5288–5296.
- [137] Jingjing Xu, Yuechen Wang, Duyu Tang, Nan Duan, Pengcheng Yang, Qi Zeng, Ming Zhou, and Xu Sun. 2019. Asking Clarification Questions in Knowledge-Based Question Answering. In *EMNLP-IJCNLP (1)*. Association for Computational Linguistics, 1618–1629.
- [138] Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. End-to-End Open-Domain Question Answering with BERTserini. In *NAACL-HLT (Demonstrations)*. Association for Computational Linguistics, 72–77.
- [139] Wei Yang, Yuqing Xie, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. Data Augmentation for BERT Fine-Tuning in Open-Domain Question Answering. *CoRR* abs/1904.06652 (2019).
- [140] Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. Wikiqa: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 conference on empirical methods in natural language processing*. 2013–2018.
- [141] Zekun Yang, Noa Garcia, Chenhui Chu, Mayu Otani, Yuta Nakashima, and Haruo Takemura. 2020. BERT Representations for Video Question Answering. In *WACV*. IEEE, 1545–1554.
- [142] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600* (2018).
- [143] Wen-tau Yih, Matthew Richardson, Christopher Meek, Ming-Wei Chang, and Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 201–206.
- [144] Xingdi Yuan, Marc-Alexandre Côté, Jie Fu, Zhouhan Lin, Chris Pal, Yoshua Bengio, and Adam Trischler. 2019. Interactive Language Learning by Question Answering. In *EMNLP/IJCNLP (1)*. Association for Computational Linguistics, 2796–2813.
- [145] Hongzhi Zhang, Guandong Xu, Xiao Liang, Tinglei Huang, and Kun Fu. 2018. An Attention-Based Word-Level Interaction Model: Relation Detection for Knowledge Base Question Answering. *CoRR* abs/1801.09893 (2018).
- [146] Sheng Zhang, Xin Zhang, Hui Wang, Jiajun Cheng, Pei Li, and Zhaoyun Ding. 2017. Chinese Medical Question Answer Matching Using End-to-End Character-Level Multi-Scale CNNs. *Applied Sciences* 7, 8 (2017), 767.
- [147] Sheng Zhang, Xin Zhang, Hui Wang, Lixiang Guo, and Shanshan Liu. 2018. Multi-Scale Attentive Interaction Networks for Chinese Medical Question Answer Selection. *IEEE Access* 6 (2018), 74061–74071.
- [148] Xinbo Zhang, Lei Zou, and Sen Hu. 2019. An Interactive Mechanism to Improve Question Answering Systems via Feedback. In *CIKM*. ACM, 1381–1390.
- [149] Yingying Zhang, Shengsheng Qian, Quan Fang, and Changsheng Xu. 2019. Multi-modal Knowledge-aware Hierarchical Attention Network for Explainable Medical Question Answering. In *ACM Multimedia*. ACM, 1089–1097.
- [150] Weiguo Zheng, Hong Cheng, Jeffrey Xu Yu, Lei Zou, and Kangfei Zhao. 2019. Interactive natural language question answering over knowledge graphs. *Inf. Sci.* 481 (2019), 141–159.
- [151] Guangyou Zhou, Yin Zhou, Tingting He, and Wensheng Wu. 2016. Learning semantic representation with neural networks for community question answering retrieval. *Knowledge-Based Systems* 93 (2016), 75–83.
- [152] Zhiheng Zhou, Man Lan, Yuanbin Wu, and Jun Lang. 2017. Single turn Chinese emotional conversation generation based on information retrieval and question answering. In *IJALP*. IEEE, 103–106.
- [153] Chenguang Zhu, Michael Zeng, and Xuedong Huang. 2018. SDNet: Contextualized Attention-based Deep Network for Conversational Question Answering. *CoRR* abs/1812.03593 (2018).
- [154] Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. 2016. Visual7w: Grounded question answering in images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4995–5004.