Situational data integration in Question Answering systems: a survey over two decades

Maria Helena Franciscatto, Luis Carlos Erpen de Bona, Celio Trois, Marcos Didonet del Fabro, Joao Carlos Damasceno Lima

Situational Data Integration in Question Answering Systems

A Survey Over Two Decades

Maria Helena Franciscatto[1*], Luis Carlos Erpen de Bona[1], Celio Trois[2], Marcos Didonet Del Fabro[3] and João Carlos Damasceno Lima[2]

[1*]Departamento de Informática, Federal University of Paraná, Curitiba, Brazil.
[2]Centro de Tecnologia, Federal University of Santa Maria, Santa Maria, Brazil.
[3]Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France.

*Corresponding author(s). E-mail(s): mhfranciscatto@inf.ufpr.br;
Contributing authors: bona@inf.ufpr.br; trois@inf.ufsm.br;
marcos.didonetdelfabro@cea.fr; caio@inf.ufsm.br;

Abstract

Question Answering (QA) systems provide accurate answers to questions, however, they lack the ability to consolidate data from multiple sources, making it difficult to manage complex questions that could be answered with additional data retrieved and integrated on-the-fly. This integration is inherent to Situational Data Integration (SDI) approaches that deal with dynamic requirements of ad-hoc queries that, neither traditional database management systems, nor search engines are effective in providing an answer. Thus, if QA systems include SDI characteristics, they could be able to return validated and immediate information for supporting users decisions. For this reason, we surveyed QA-based systems, assessing their capabilities to support SDI features, i.e., *Ad-hoc Data Retrieval, Data Management,* and *Timely Decision Support*. We also identified patterns concerning these features in the surveyed studies, highlighting them in a timeline that shows the SDI evolution in the QA domain. To the best of your knowledge,

this study is precursor in the joint analysis of SDI and QA, showing a combination that can favor the way systems support users. Our analyzes show that most of SDI features are rarely addressed in QA systems, and based on that, we discuss directions for further research.

Keywords: data integration, question answering, information retrieval, situational data, source discovery, decision support

1 Introduction

The last years have marked a paradigm shift in access to information, where Big Data raised the need to deal with large and heterogeneous data volumes on the Web [1]. Exploiting such large amounts of data makes finding information a complex and time-expensive task, encompassing communities such as information retrieval, natural language processing, and artificial intelligence to create solutions for retrieving the information [2, 3]. Information Retrieval (IR) approaches are a common way to capture information suitable for the user needs, but often the information provided is not relevant or cannot be easily processed [4, 5], which makes it difficult to provide the precise answers that users really want [6].

Question Answering (QA) systems have been considered capable of minimizing these problems, since they aim at providing a precise answer that satisfies a specific question [2, 5]. Consequently, this domain has grown steadily all the way to present-day, integrated in virtually all personal assistants (e.g., Siri, Cortana, Alexa Echo, and Google Home) which act as voice/question-oriented automatic answering, providing answers or suggestions to the submitted questions [3]. However, QA systems still present some challenges, for example, they cannot cope well with increasingly long input questions and complex information needs [7]. In addition, the majority of QA approaches are single information source systems (i.e., systems that use only internal information to generate an answer) and only a small part is able to deal with multiple information sources [8]. This disproportion not only demonstrates a prominent research need, but directly impacts the management of complex questions, since corroborating information from multiple sources helps to identify good answers across databases in the presence of low quality data containing erroneous, misleading, biased, or outdated information [9].

Another challenge in QA domain is the need to determine whether the data retrieved from the query contains the answer or not, and whether the information is complete [10]. Thus, in the cases where the information is insufficient, a multiple source QA system could be able to integrate other data sources such as documents, query logs, or even entire Knowledge Bases (KBs) with its own internal information to generate a complete answer [8]. Furthermore, the answer resulting from this integration could allow the end user to execute operational processes based on better informed decisions.

Historically, such operational processes have been supported by traditional data integration, where different ETL (Extract-Transform-Load) flows are executed for loading data repositories with external information of interest (e.g., an information that a business manager can use to obtain competitive advantage over other companies) [11]. In this case, the integration is periodic and the decision maker *owns* the structured data that are incorporated into the decision process [12]. In many application domains, and especially in a QA scenario, traditional data integration allows to obtain additional data for performing useful analysis; however, when there is a need for near *real-time information*, periodic updates are no longer beneficial. In other words, it may be impractical to wait for information only when ETL flows are executed [11]. In such cases, where immediate information is required, a **Situational Data Integration** can be performed.

A Situational Data Integration (SDI) is managed on-the-fly for dealing with dynamic requirements of ad-hoc queries that neither traditional database management systems nor search engines are effective in providing an answer [13–15]. Specifically, SDI integrates unstructured and real-time data with local data for inferring solutions that enable accurate decisions [16, 17]. This means that if an ad-hoc question cannot only be answered with local/stationary data, a SDI system might *discover* relevant sources that contain the data needed. These on-the-fly data are called *situational data*, i.e., data that are not owned by the user and have the role of providing a complete answer to a *specific* problem or need [15].

Considering this dynamic characteristic of SDI and the increased research interest respecting multiple source QA, we provoke the hypothesis that situational data discovered in the SDI process can be decisive for returning the right answer to the end user. Specifically, SDI presents features that, when integrated with a QA system, could make it smarter, thus potentially supporting the end user. To exemplify, suppose that a user queries a QA system, asking for specific information that can only be answered by querying situational data besides local/stationary data. When querying local data and receiving insufficient data[1], the QA system starts the search for situational data, i.e., relevant data sources that contain the information needed. The situational data source selected is then integrated with the stationary data, and the integration results are returned to the QA system and presented to the user. With this operation, there is a greater chance that he will receive, in a timely manner, the expected answer[2].

Although SDI is a concept originated in the Business Intelligence (BI) domain [13], its features can be applied in different areas where the end users need better support, such as Question Answering. The motivation for combining SDI and QA can be justified by the following:

[1] Stationary data could be considered insufficient if the input question requires information whose metadata is not present in the stationary data source schema, or if the information itself is not found in the stationary source.

[2] A complete scenario exemplifying SDI in QA domain is given in Subsection 2.1.

- `Multiple source and near real-time information is gaining focus:` More recent applications are no longer limiting their analysis to open-domain searches and structured databases, but they are demanding actionable information for attending specific needs and improving decision making [18]. QA systems are good candidates for these purposes, since they allow us to obtain concise and faster answers to questions stated in natural language from a collection of data sources [18, 19].
- `Changes in question formulation and systems dynamics:` With more complex questions being handled and QA systems moving from their static counterparts to more interactive ones (e.g., conversational settings, user feedback, and interpretability) [3], situational data sources are eligible for providing timely and complete answers, whereas SDI is required to handle these sources, which are highly dynamic and without guaranteed access [15].
- `Need to create more flexible and scalable systems:` QA systems are often disconnected from other systems, i.e., they need to flexibly integrate components to fulfill specific tasks [20]. In contrast, the SDI process could be applied to obtain and validate answers from multiple sources in cases where a single one is not enough to answer the question [8, 21]. In addition, SDI can be used to provide *contextual information*, specially when this information can significantly improve tasks such as retrieval and ranking [22, 23]. Hence, QA systems can be more intelligent when formulating answers.

In spite of the benefits of an SDI inclusion, situational or ad-hoc data integration is a big challenge in the literature [24], which may be an indication that SDI is still a gap in QA-based systems. Indeed, some SDI features already exist sparsely in Question Answering domain, but the literature lacks a systematic investigation of the relation between these concepts.

Following the aforementioned motivations, this paper surveys **Situational Data Integration in Question Answering systems**, aiming to establish the state-of-art of data retrieval, data management, and decision support in these systems. We surveyed more than 50 studies related to *multiple source QA* from the last two decades (i.e., between 1990 and 2020), for assessing the presence of situational data in the period. In this process, we analyzed the selected studies and realized different levels of coverage for SDI features in QA approaches. As results from the analysis, we constructed a timeline for highlighting the most prominent SDI features in each time range; also, we found that some features are moving towards consolidation in the QA domain, as they are often explored in the literature. Despite that, SDI is still challenging in the Question Answering domain, i.e., most SDI features are rarely present in the QA approaches.

It is important to clarify that the present work does not aim to detail specific characteristics of QA non-related to SDI, since several studies already address these aspects [2, 4, 21, 25, 26]. In contrast, we present characteristics of SDI that are covered by QA-based approaches in a complete or partial way, in order to evaluate the most challenging features in the area. To the best of our

knowledge, there is not yet a study that identifies SDI in Question Answering approaches, raising points that prove the usefulness and need of such integration. So, as main contributions, this survey: (1) analyzes QA approaches according to common SDI features found in literature; (2) presents techniques and patterns related to SDI in QA approaches, organizing these patterns in a timeline and highlighting the most outstanding features; and (3) discusses open research possibilities regarding SDI and QA, as a way to guide readers or future researchers interested in expanding knowledge and development in both domains.

The structure of the paper is presented as follows. In Section 2, we discuss the SDI concept and provide a practical example in QA domain. From Section 3 to Section 5 we review QA approaches regarding the main SDI features. In Section 6 we present analysis including SDI features supported, patterns, and management over time. Based on these analysis, we discuss research opportunities in Section 7. We conclude with our remarks in Section 8.

2 Situational Data Integration (SDI)

In Situational Data Integration, the term "*Situational*" comes from the concept *Situation Awareness* (SA), which is the perception of events related to an entity (i.e., the user) and the understanding of what is going on around, allowing to make accurate decisions [27]. By following this concept, a Situational Data Integration may be understood as an integration oriented by situations of interest to the user, providing prompt information that are at the basis of decision-making.

Although SA is a long-established concept (defined around the 1990s), situational integration only gained focus years later, in the *Business Intelligence* (BI) domain, where SDI was initially investigated due to its impact on operational decisions [13, 28]. The growing interest in obtaining and using real-time information motivated several studies to present the characteristics of SDI [11–16, 29, 30]. Based on the related literature, SDI is composed by three main features: *Ad-hoc Data Retrieval, Data Management*, and *Timely Decision Support*. Each feature, in turn, is associated with subfeatures, i.e., specific tasks that are involved in SDI, as demonstrated in Table 1. They are discussed next.

Table 1 Situational Data Integration Features

SDI Main Features	Subfeatures
Ad-hoc Data Retrieval	Ad-hoc Questions and Source Discovery
Data Management	Unstructured Data Preprocessing and Situational Source Inclusion
Timely Decision Support	User Guidance, Decision-making Support, and Response Time Improvement

Ad-hoc Data Retrieval refers to the abilities of a situational data integration approach to retrieve potential sources of information for dealing with

query requirements. It relies on *Ad-hoc Questions* and *Source Discovery*. Frequently, ad-hoc questions involve situational data, which have a narrow focus on a specific domain problem or a unique set of needs [12, 13]. Situational data are usually external to an organization control and hence without guaranteed access [11]. Thus, in Source Discovery, the goal is to find a provider for data not available internally, as well as systematically analyze the different potential sources of information at hand for attending specific and dynamic requirements [12, 29]. When data integration requests are changed or new requests are submitted on the fly, a new round of integration should be quickly performed to meet the new requirements [14]. Thus, we assume that a situational integration involves dynamic data sources.

Another main feature of SDI is **Data Management**. This feature refers to how an SDI system deals with unstructured and heterogeneous information from a discovered source, which is an essential step to ensure an effective situational integration. Two subfeatures are involved, named *Unstructured Data Preprocessing* and *Situational Source Inclusion*. Situational data are often scattered across heterogeneous and unstructured sources available on the Web [12, 15, 30], so, the integration has to tackle the fact that data sources usually contain erroneous, out-of-date, or conflicting data [31]. *Unstructured Data Preprocessing* deals with these challenges, making the extracted information feasible to be integrated. After the processing, situational data needs to be integrated with the current information in order to extend the information previously available and completely satisfy users' specific needs [12, 13, 15, 17, 32, 33]. This often requires a correlation analysis process for assessing whether the situational source is suitable for the task at hand [13].

The last main feature of SDI is **Timely Decision Support**. A situational picture is necessary to go beyond the simple perception of the elements in the environment, supporting the overall comprehension of the current situation and the user's decision making process [32, 33]. Thus, a situational data integration system provides *Decision-making Support* if it provides information for alerting the decision makers of situations that can potentially affect their activities. The support can also be achieved, for example, if the system leads the user through the sequence of steps he has to apply, subsequently providing explanation of how resolutions were made, or why a certain operator was selected [34].

Situational projects also require just-in-time delivery of good-enough solutions that are not addressed by traditional offerings [30], so *Response Time Improvement* is a highly desirable feature. Useful techniques involve, e.g., provide new compute nodes as needed, reduce data complexity for processing, or distribute computing tasks through parallel computing [13]. By providing timely insight into situations, an appropriate action can be taken in real time [16]. Concerning *User Guidance* subfeature, situational applications may involve active human participation when solving specific needs [13, 14]. With

human feedback, for instance, the system is able to optimize its processes and responses.

It is interesting to highlight that many of the subfeatures discussed in this section can be found in other integration approaches. E.g., Source Discovery is also present in link traversal approaches [35], User Guidance is a strong characteristic in pay-as-you-go integration [36], as well as Situational Source Inclusion can be observed in mashup-based approaches [37]. The present work, however, only focuses on situational integration and how it is applied to the QA domain.

2.1 Visualizing SDI: A Practical Example in QA Domain

We can reason about SDI features in QA taking as example a scenario involving speech therapy, since this domain has been explored in situational contexts [38]. Regarding Ad-hoc Data Retrieval, suppose that a Speech-Language Pathologist (SLP) wants to know whether the government investment in education affects children's speech abilities. The SLP owns a stationary database that contains information about the patients (including their age, schooling, region), but this database does not include any educational data of the target region, which would be necessary to effectively answer the question. So, he/she poses an **ad-hoc question** as input to a QA system (either by natural language text or voice), and the system triggers a **discovery process** in Web sources. Then, it finds several open databases in official government portals that provide data related to education, e.g., data on schools performance in national exams and educational indicators. By verifying patterns in the question that match each candidate source, the system selects the one that best fits the information need: a government open data source that describes investments in basic education, which could provide a complete answer to the SLP's question.

Next, regarding Data Management subfeatures, consider that the selected source contains raw situational data that are difficult to process and analyze. So, a **preprocessing** step is performed in order to obtain a smaller set of data that are relevant and applicable for the SLP's query. After this process, the next step is to **combine the situational data** with the stationary data (i.e., the patients data owned by the SLP), by using correlation techniques that semantically map both databases attributes. The result should be an augmented knowledge base that allows to identify all data patterns between educational and clinical data.

Concluding the speech therapy example with Timely Decision Support, the QA system now has a specialized data set where relations between educational and patients data can be extracted. Hence, the system can use ML and NLP techniques to rapidly infer these relations and present some suggestions to the SLP through a graphical interface. If incorrect inferences occur, the SLP could **guide the system** by informing feedbacks, which can positively impact future classifications. But most importantly, the system returns a relevant pattern, showing that "*most patients with lower pronunciation performances are from regions with lower educational investments*". This answer makes the

SLP **better informed while saving time**, and it can be used to guide the therapeutic planning. In other words, the SDI+QA system allowed the therapist to have a good view on the patient's condition and make appropriate decisions based on educational context.

2.2 Searching for SDI in QA Systems

As a summary of this section, we presented a set of features that compose the SDI concept. As the next step, based on this set of features, we aim to establish an overview of Situational Data Integration in recent QA-based approaches, thus enabling an understanding about the most urgent needs in this domain, as well as consolidated methods used in the researches.

Thus, for adequately investigating the state-of-art of SDI in QA systems, we selected *QA-based studies* that covered *multiple data sources* by means of a *data integration*. Based on these keywords, we covered studies from the last two decades, prioritizing those published between 2010 and 2020. Thus, we can track trends in time ranges, in terms of methods and data sources, as well as to raise hypotheses about future developments. Scopus search engine[3] and Google Scholar[4] were used as data sources, since they include other digital libraries such as ACM Digital Library[5], IEEE Xplore[6], and Science Direct[7].

For assessing the presence of SDI in the surveyed studies, we based on the features shown in Table 1. Specifically, given the particularities of the QA approaches, we present how each SDI feature is present in the studies, assessing whether its coverage occurs in a *complete or partial way*. In addition, as already mentioned, many SDI characteristics can be found in other integration variants. Thus, the analysis of each feature individually not only allows to obtain an overview of SDI in QA, but it can be also used to infer how the user support could be improved from other types of integration operating in QA systems.

We surveyed more than 50 studies in QA domain, with respect to SDI features. All papers are detailed in Appendix A and discussed in the next sections.

3 SDI's Ad-hoc Data Retrieval in QA

Ad-hoc Data Retrieval refers to finding and accessing situational data, by discovering sources that are suitable for an information need. Its subfeatures, Ad-hoc Questions and Source Discovery, were evaluated in the surveyed QA studies as demonstrated in Table 2. An overview of results is shown in Figure 1.

[3] https://www.scopus.com/
[4] https://scholar.google.com.br/
[5] https://dl.acm.org/
[6] http://ieeexplore.ieee.org/Xplore/home.jsp/
[7] https://www.sciencedirect.com/

Table 2 SDI's Ad-hoc Data Retrieval Requirements

Ad-hoc Data Retrieval Features	**Requirement for "Complete Feature"**	**Requirement for "Partial Feature"**
Ad-hoc Questions	Situational data integration requirements are often immediate and cannot be totally defined in advance [29, 30]. We consider this feature *fully* supported if the questions or queries presented in the approach are not predefined nor recovered from databases or benchmarks.	We consider this feature *partially* supported if the question is built with the help of other techniques, e.g., auto-complete tools or parameters selection.
Source Discovery	We consider this feature *fully* supported if (i) the situational data source used in the approach is *not predefined*, but chosen at query time[8] [11, 13, 14, 29, 39] `AND` (ii) the approach includes some change in the data source(s) in order to meet query requirements [11, 13, 14, 29, 39].	The feature is *partially* considered when *predefined* sources are changed or adapted to meet query requirements.

[8]In some cases, the data source accessed is predefined as the whole Web, but the specific data sources (i.e., the websites accessed) are defined at query time. Exceptionally in this case, we consider the specific data sources when assessing Source Discovery.

3.1 Ad-hoc Questions

SDI is usually motivated by ad-hoc questions, which are defined at a moment of need. They closely relate to QA area, where users interact with a system looking for precise and fast answers. Everyday questions do not come from any repository, and usually do not follow query formats unfamiliar to the user, such as SQL or SPARQL. Users questions in QA are often informed in natural language, and mainly, they are informed *at the very moment of an information need*. Situational Data Integration aims to deal with this kind of questions (i.e., the ad-hoc questions), providing support by discovering data sources that fit in the question requirements.

The study in [40] answers ad-hoc queries posed in a global schema by exploring heterogeneous definitions of indicators formulas. Ad-hoc NL questions are answered in [41] by means of a multi-source QA system. NL questions are also present in [42], which proposes a QA system and an integration method based on ontologies, and in [43], through an interactive NL Interface for querying ontologies. [44] and [45] present MiPACQ (Multi-Source Integrated Platform for Answering Clinical Questions), an integrated framework for semantic-based QA that accepts free-text clinical questions.

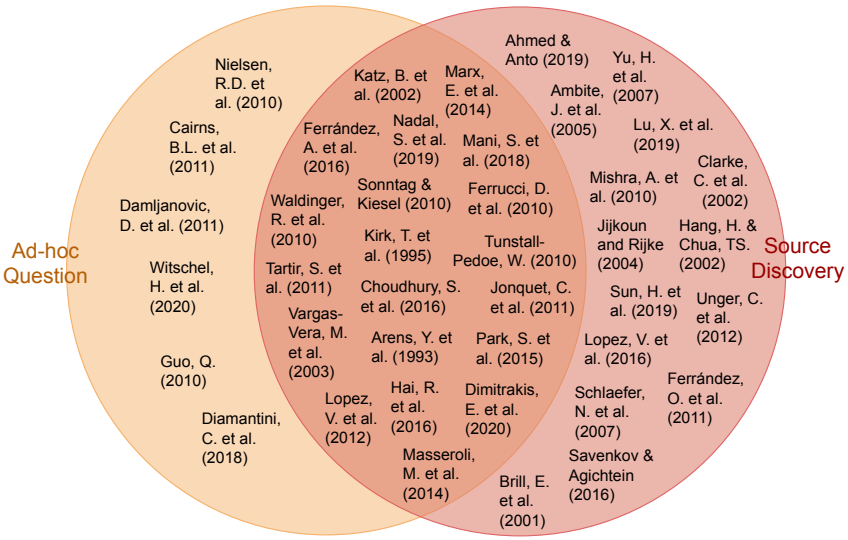

Fig. 1 Ad-hoc data retrieval in Question Answering systems.

Ad-hoc questions may also be covered in a *partial* way if they are not asked freely, but using selection and completion methods. E.g., the authors in [46] present a novel approach for context-aware Sequential Question Answering, based on a knowledge graph (KG) that integrates information from various sources. In their experiments, each participant receives a first predefined query, and all subsequent ones require participants to select a subset of the KG nodes that were displayed or to ask questions in natural language. Similarly, in [47] the query is built by parameters selection in an interface. In [48] and [49], there are selection-based and auto-complete approaches for query formulation, respectively.

3.2 Source Discovery

Besides ad-hoc questions, Ad-hoc Data Retrieval in SDI includes ***Source Discovery***. Many QA systems have to deal with complex questions that can only be answered using information from more than one data source. In some cases, real time information is *decisive* for the users tasks, but cannot be provided by stationary or predefined sources. SDI's Source Discovery could fill this gap, enabling to find the right sources that contain the information.

The study in [50] presents QUEST (QUEstion answering with Steiner Trees), a method for answering complex questions from textual sources on-the-fly, by joining evidence from different documents. QUEST constructs an ad-hoc, noisy knowledge graph by dynamically retrieving many question-relevant text documents from the Web. Non-predefined and dynamic pages from the Web are also accessed in the MedQA system proposed in [51], in the hybrid QA system described in [52], and in the Quartz system [53].

The QA system proposed in [54] also presents Source Discovery as it accesses non-predefined and dynamic websites that, in combination with WordNet [55], are able to find words lexically related to the original query terms. The same occurs in [56], which proposes a system that utilizes Google search engine to find answers on the Web. A set of potential answers is extracted from the summary text, then given a set of possible answers, an answer projection step is performed for searching supporting documents in the TREC QA document collection[9].

The authors in [63] present a novel approach that relies on a parse of the question to produce a SPARQL template representing the internal structure of the question. The query templates contain missing elements that have to be filled with URIs. For this task, a extraction is performed through a framework that uses the Linked Data Web as background knowledge and assumes a set of predicates for which equivalent NL expressions are to be detected from an arbitrary corpus (e.g., Wikipedia or the whole Web), so that labels of instances linked by these predicates determine the seed pages to be queried.

In [64] the authors present the QALL-ME framework, a reusable architecture for multi-lingual QA systems working on structured data sources. The architecture is based on Service Oriented Architecture (SOA), which uses web services for the framework components. The data sources used to retrieve the answers are non-predefined databases or XML documents, which are specified by the domain ontology. In addition, the web services accessed by the SOA-based architecture are interchangeable.

In [65], Text2KB is introduced, a system that enriches QA over a knowledge base using external text data. These data come from three sources: the whole Web, the Yahoo! Webscope L6 dataset[10] containing CQA (Community Question Answering) data, and text fragments with entity mentions. Similarly, [51, 52] and [66] also perform searches on the whole Web, by retrieving web pages according to information specified in the query.

Some studies *partially* handle Source Discovery, i.e., their data sources are dynamic, even though predefined. The authors in [67] use several predefined Linked Open Data sources including Wordnet and DBpedia[11], and these sources are dynamic in handling query requirements. The authors in [68] describe PullNet, an integrated neural framework that reasons with heterogeneous information to find the best answer for an input question. PullNet constructs a question-specific subgraph containing sentences from a corpus, and facts from a KB. Both data sources are predefined but dynamically changed: the model might use the KB when the required fact exists, and "back off" to text when the KB is missing information.

The QA system in [69] presents an "early-answering" strategy, in which a question can be answered directly from a structured collection (a database

[9]The Text REtrieval Conference (TREC) is a well-known reference for QA approaches due to Question-Answering tracks started in 1999. Several studies that search for answers on the Web mention the QA tracks in their methodologies, such as [57–62].

[10]https://webscope.sandbox.yahoo.com/

[11]Structured database of information extracted from Wikipedia. Available in https://wiki.dbpedia.org/

containing knowledge gathered from the Web), then a secondary corpus is used to support the answer found. In addition, if the question cannot be answered with the structured collection, publicly corpora are accessed for retrieving relevant passages. The sources are predefined, but not all of them are used for passage retrieval: passages are retrieved from at least three corpora, considering that the results in the early-answering process were insufficient. Therefore, there is source changing during the process.

Mishra A. et al. [70] present a context-aware geographical QA system that finds out the semantic relations between the question and candidate answers. Web information about various cities are collected and used to built an offline knowledge base used in the system. Although there is no discovery, source changing can be easily performed manually. The Ephyra QA system [71, 72] combines several techniques for question analysis and answer extraction, and incorporates multiple knowledge bases. The system uses predefined sources and a API for searching in dynamic sources. Similarly, the authors in [73] present a query answering system that integrates multiple knowledge sources of specific domain. The sources are predefined, including a Mediator reasoner that extracts useful information from about two dozens of websites. The access to all knowledge sources in the proposal is performed by a module called Query Manager, which determines which knowledge source are required to produce the answers. The Query Manager also considers each knowledge source as a reasoner, so that it can be recalled to provide additional answers when needed.

3.3 Ad-hoc Questions + Source Discovery

As we can see in Figure 1, most of the QA-based approaches that cover Ad-hoc Questions also deal with Source Discovery. The authors in [74] present a natural language QA system that understands users' ad-hoc queries and translates them into structured queries. Also, it provides an interface to multiple non-predefined knowledge sources from the Web. Similarly, the study in [49] ingests information from web crawls and articles to continuously stay abreast of information around drones. Web sources are also external, non-predefined, and dynamic in [48], [75], and [76]. These studies also allow the user to query data sources by means of natural language questions.

The study in [77] introduces LODQA, an approach that exploits simultaneously hundreds of linked datasets by means of a suite of services. The datasets are chosen at query time, according to their similarity with the question, and they are dynamic with respect to query requirements. Although LODQA was evaluated through a collection of predefined questions, the approach allows the user to type a query in natural language or RDF format. The QA system in [78] uses a novel hybrid answering model for providing IT support. The user can interact with a platform by asking a question or uploading a screenshot of the error he is facing. Predefined and heterogeneous data sources are used for answering the questions, with an *orchestrator* module that decides which source to call under what circumstances, in order to maximize the likelihood of getting a correct answer.

Graphical interfaces for users' ad-hoc queries are also present in [79] and [80]. In the former, the SIMS (Services and Information Management for decision Systems) is proposed: the system dynamically retrieves and integrates information from external and predefined databases, which can also be changed through the interface. In the latter, an ontology-based resource index called NCBO (National Center for Biomedical Ontology) provides unified access to more than twenty heterogeneous biomedical resources. Both ontologies and resources are changing often, so NCBO tables are regularly and automatically updated. Based on the user input, the index uses an auto-complete mechanism for suggesting terms and resources. *True Knowledge* [81] is another QA platform with an interface for submitting ad-hoc queries. For the answering process, the proposal combines a structured knowledge base, a NL translation system, and an inference system. The user may also provide additional knowledge sources, and a *System Assessment* module can switch off the conflicting ones.

Ad-hoc Data Retrieval in SDI is well represented in [82]: the authors argue that BI applications should consider external data sources for achieving crucial information that help taking the right decisions, thus they integrate the DW internal structured data with external unstructured data obtained with QA techniques to answer ad-hoc questions. A question is posed through a GUI element by the user, who also identifies the sources where to search the required information, and the sources are accessed by the QA system dynamically.

In [83], the authors present *openQA*, a modular and extensible open-source QA framework that integrates other QA systems [63, 84] for accessing DBpedia endpoint. The proposal covers an answer formulation process that receives an ad-hoc query, which can be interpreted in different formats such as SPARQL and SQL. New modules can be integrated within *openQA* via a plug-in architecture, and the user can enable and disable instances used for searching. The study in [85] presents a dialogue-based QA system that integrates several linked data sources at SPARQL endpoints. The user's ad-hoc question is analyzed and mapped to one or more suitable services found, which may answer parts of the query. With the information found in the result, additional services can be triggered for finding more information about the first result.

The work in [86] presents a Big Data Integration ontology and a query answering algorithm that converts ad-hoc queries posed over the ontology to queries over multiple sources. The approach builds upon two RDF named graphs (Global and Source graphs) and wrappers that accommodate different kinds of no-predefined data sources. The approach also deals with changes in the source schema via semi-automated transformations on the ontology. Ontologies are also present in [87] with *PowerAqua*, an ontology-based system that explores multiple and heterogeneous sources on the web. PowerAqua accesses the Semantic Web through the Watson SW Gateway, thus it only retrieves information if this has been crawled and indexed by Watson or in specified online repositories. PowerAqua provides plugins for accessing the repositories, which are loaded on demand. The system accepts users' queries

expressed in NL and retrieves precise answers by dynamically selecting and combining information from the resources.

The studies in [88, 89] describe *Information Manifold* (IM), a system for browsing and querying multiple web sources. The system determines exactly the set of information sources relevant to a query, adding their descriptions to the knowledge base by means of a representation language, allowing multiple sources to be accessed. Users can formulate queries either using templates that are available for the information categories, or by combining such templates into an arbitrary conjunctive query. The Data Lake proposed in [90], named *Constance*, manages QA based on metadata. In an unified interface, the users can define their queries by using a formal structured query language or simple keywords. Constance has an Ingestion Layer, which is responsible for importing data from heterogeneous sources into the DL system, and generic extractors adapt to data source formats. Through the interface, users can import local files, databases or remote data via web services, and new extractors can be easily added using a plug-in mechanism.

The QA system proposed by [41] merges a KB-based QA (KBQA), a IR-based QA (IRQA), and a keyword QA in its architecture. Multiple information sources are used, including curated KB, raw text, and auto-generated triples. The sources are predefined but they are alternated according to the input question, e.g., when the questions is a sentence, it is sent to the sources accessed by the KBQA/IRQA system; otherwise, it is sent to the keyword QA system. Multiple information sources are also observed in [91]: the authors develop a prototype called QUARK (Question Answering through Reasoning and Knowledge), which uses several knowledge resources. Ad-hoc NL questions are submitted to the automated deduction system SNARK, a general-purpose theorem equipped with an application-domain theory, which invokes the sources when appropriate. When a new resource is introduced to the system, it is provided one or more axioms in the theory that expresses what the resource can do.

Finally, the study in [47] presents an approach to support integrated search of distributed biomedical-molecular data, aimed at answering multi-topic complex biomedical questions. A Bioinformatics Search Computing application (Bio-SeCo) is modelled and published, in which bioinformatics services are registered. Bio-SeCo provides a platform which allows the user to express requests over the multiple services registered and find answers to his/her questions. Although the sources are predefined, the user can easily inspect the sources and obtained results, select the most appropriate, expand or refine them.

Retrieved data loses its value if not refined and integrated with current data to form a complete information. Thus, SDI's *Data Management* feature is addressed as follows.

4 SDI's Data Management in QA

Data Management in SDI includes the system's ability to deal with heterogeneous and noisy information from situational data source, and combine this source with the stationary data, in order to support decision-making. These abilities refer to two features, named *Unstructured Data Preprocessing* and *Situational Source Inclusion* requirements are detailed in Table 3. The feature results in QA are shown in Figure 2.

Table 3 SDI's Data Management Requirements

Data Management Features	**Requirement for "Complete Feature"**	**Requirement for "Partial Feature"**
Unstructured Data Preprocessing	We consider *full* coverage in approaches that apply preprocessing techniques in data sources, such as cleaning, normalization, noise removing, entity/link resolution, among others. We did not consider this feature when it occurs in the input question only.	Not applicable.
Situational Source Inclusion	Situational Source Inclusion is *fully* supported by papers that add a data source which (i) covers a targeted/specific information need `AND` (ii) is integrated with current data in order to provide complete insights or solutions[12] [12, 13, 15, 30, 39].	Situational Source Inclusion is *partially* considered when there is a situational source, but it does not complement the available data.

[12]Typically, a situational integration occurs when stationary data is combined with situational data (external or/and domain-specific data); however, the literature presents few cases containing local databases. In addition, there may be exceptions where external sources are not present, but integration with specific additional data occurs. Thus, with respect to this feature, we have considered a relaxed condition.

4.1 Unstructured Data Processing

The QA of next generation will have to take into consideration presently heterogeneous and unstructured data [21]. Although data preprocessing is already a common task in QA systems [10], handling unstructured data, in particular, is essential for successfully integrating discovered data sources. Consequently, a well-performed integration may determine the value of the QA system to the user.

With respect to Data Preprocessing, different techniques are used by QA-based systems. The novel framework named HSIN (Heterogeneous Social Influential Network) proposed in [92] integrates and simultaneously learns

the questions textual contents, their related categories information and user's social interaction. The experiments were based on a large dataset from Quora service and Twitter social network, in which duplicated data were removed before composing training and testing sets. Duplicates and ambiguous data are also handled in [93], which presents a model based on Integer Linear Programming (ILP) for exploring multiple KBs in QA. The approach unites the construction of alignments among KBs, and query construction for translating a NL query into a SPARQL query. These tasks can involve ambiguous data, so the approach combines potential alignments to obtain a *Disambiguation Graph*.

Many other studies also perform disambiguation and summarization tasks for preprocessing data: The IR-based system in [41] uses a multi-source tagged text database that includes disambiguation results, types of named entities and NLP results. The system in [94] is embedded in a hybrid QA system architecture called *QUETAL*, which selects one or more information data sources to retrieve answer candidates and prepare a final answer. These processes include disambiguation of concepts and database tables, based on recognized entities. In the Ephyra system [71, 72], answer candidates (extracted from external sources) that contain frequent keywords are favored, and additional filtering techniques are used to drop redundant and non-informative answers. Similarly, the QA model in [95] prunes candidate answers that are incompatible with the question, using a convergence mechanism and an arc-search to reach the most relevant answers. Irrelevant sources are also pruned in the Information Manifold system [88, 89], which helps to solve completeness and redundancy issues.

The Constance system [90] discovers, extracts and summarizes structural metadata from the data sources, also annotating data with semantic information to resolve ambiguities. The MedQA approach [51] includes a Text Summarization step that removes redundant sentences by clustering similarities into the same group. In the OpenQA system [83], answers may be extracted from different sources, so it can be ambiguous and redundant. To alleviate this issue, results that appear multiple times are fused by means of clustering and ranking. The FREyA system [43] also covers a desambiguation step, and in case of the system failing to automatically interpret the question, the disambiguation may occur manually, i.e., with the user interacting with a dialog box.

The study in [96] focuses on capturing the full views of user topical expertise in CQA services, by considering information from other social media websites in which they participate, e.g. the GitHub. For data preprocessing, the text from social media is tokenized, and all code snippets, stopwords and HTML tags are discarded. In fact, tokenization and stopwords removal are very frequent preprocessing methods in QA. E.g., in [44, 45], both the question and candidate answer are processed through tokenization, stemming, part of speech detection, named entity recognition and dependency parsing. In [97], the authors present a QA approach that allows precise browsing from

web news data by using text and multimedia information simultaneously. The crawled news documents were parsed by using a language processing toolkit, and filtered with a stopword list.

In [56], stopwords are not permitted to appear in any potential n-gram answers when searching for candidate answers. Also, an answer tiling algorithm is applied for solving overlap in shorter n-grams. For example, "A B C" and "B C D" are tiled to "A B C D". Answer tiling is also used in the Quartz system [53]. The system analyzes an incoming question, sends it to six streams in parallel, and each stream produces a ranked list of relevant answer candidates. The answer tiling is a subsequent process, where similar answer candidates from the six streams are identified and merged to obtain the highest confidence answer.

Noise removal is another technique that frequently appears in QA proposals. The PowerAqua's architecture [87], e.g., deals with noisy and incomplete data throughout the retrieval process. Noise removal from audio files occurs in [78]. The clinical QA system *AskHERMES* [19] handles complex questions through the use of a structured domain-specific ontology that integrates five types of external sources. The system removes noise from texts, and merges relevant passages as parts of a potential answer, so it can retain semantic content of the data sources. In [70], the extracted documents are preprocessed through noise removal, tokenization, sentence splitting and tagging. The system architecture proposed in [63] includes a cleaner module for removing noise from crawled text. In this study, an indexer is also present.

Indexes are frequent in the analyzed proposals. The study in [98] presents PIQUANT, a modular architecture that allows for multiple answering agents to address the same question in parallel. Among the answering agents in PIQUANT, a knowledge-based agent performs predictive annotation for indexing the information with semantic classes. The QA system in [78] comprises many preprocessing techniques by running the *Ingestor Component*, responsible for converting raw data into structured knowledge. In the document ingestion phase, structural heuristics and Latent Semantic Indexing [99] are used to induce formatting into the documents. In the NCBO Resource Index [80], after retrieving the resource elements, there are steps of concept recognition, resolution of references, data annotations, as well as semantic expansion for making the data elements more informative.

In the QA system proposed in [42], the Pervasive Agent Ontology is constructed by means of concept extraction methods and analysis methods, used for discovering internal relations between concepts and making the ontology more complete and consistent. The authors in [100] propose to extract evidence from heterogeneous knowledge sources, and use it for question answering. Preprocessing was performed on the information from the two sources, by means of identification of entities, Semantic Role Labeling (SRL) and stopwords removal.

In the Bio-Seco application proposed by [47], attributes of the connected services were normalized in order to join their values semantically. In the

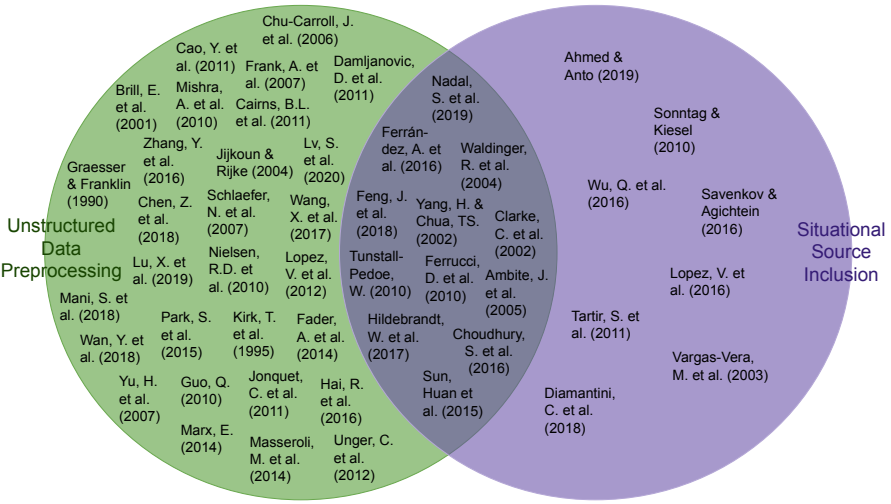

Fig. 2 Data Management in Question Answering systems.

QUEST system presented in [50], a knowledge graph was constructed by retrieving many question-relevant text documents from the Web. These documents were preprocessed through POS tagging, named entity recognition and lightweight coreference resolution for linking pronouns to entities. The resulting graph is treated as the knowledge source for answering questions. Finally, the authors in [101] present an Open Question Answering (OQA) approach that leverages both curated and extracted knowledge bases. The union of these KBs forms a single resource containing a billion noisy, redundant, and inconsistent assertions. The approach includes lemmatization and stopwords removal before computing cosine similarity.

4.2 Situational Source Inclusion

Other feature in SDI's Data Management is ***Situational Source Inclusion***. This step is what makes Source Discovery effective, since it includes situational data into the answering process. The capability of incorporating situational data could make the QA system aware of current situations, thus being able to respond complex and specific questions more precisely.

In this sense, the dialogue-based QA system presented in [85] implements an interface to linked data sources, which relies on mapping between a given data source to corresponding Web resources. The dialogue platform is based on ontology structures for processing dialogue grammars and a external service connector. When a query is informed, the appropriate Web resources are retrieved dynamically through semantic search. These resources cover specific information needs, which are combined with the stationary data (the ontology) for providing complete answers.

In [48], the questions are answered using semantic knowledge stored in ontology schemas, by means of query triples mapped to ontology elements.

The approach covers Situational Source Inclusion since external information are retrieved when there is a need for complementing the current data: if the question cannot be answered solely from the ontology, SemanticQA detects the failing parts and send them to a document web search engine, for extracting answers from snippets of web documents. After the extraction, they are matched against ontology instances.

The hybrid QA system in [52] aims to combine textual and structured knowledge base data for question answering. The system consists of three main modules: a Knowledge base, an online module, and a Text-To-KB transformer. The NL question is taken as input, and the knowledge base (e.g., DBpedia) is used to retrieve the answer. As there is no context information for KB-based QA modules, a Web search is performed in order to fill this gap. In its turn, the online module searches text to find answers, and the candidate answers from the Web are merged with those in the KB. This process is concluded by the Text-To-KB transformer, which detects KB triples in both snippets and documents and then store them in the KB.

The study in [65] also presents a hybrid QA system, named Text2KB, that extends the information in an existing knowledge base. The KB used as baseline corpus is *Aqqu* KBQA system [102], which accesses Freebase data. However, some tasks such as question interpretation, candidate answer generation and candidate ranking are challenging for a KBQA system due to lack of information for solving redundancies or composing a complete answer. Thus, external web-based sources are employed, providing additional edges in the KB and allowing to produce a final answer.

The approach in [40] is based on a query reformulation procedure that correlates and adapts the global query to the schema of the local sources by using the materialized views of the local data mart. The materialized views are accessed by exploiting a state space that integrates a dimensional lattice with a formula graph. The formula graph is an additional source generated automatically, containing specific data that complement the dimensional lattice in order to answer specific queries.

In the QuerioDALI QA system proposed by [67], Linked Open Data and Knowledge Graphs (KGs) are exploited to answer complex NL queries. Given the query, Predicate Argument Structures (PAS) are extracted for finding entities and links that matter. Relevant graphs are selected based on their coverage for a given PAS and the candidate URIs are found based on this correlation. From these candidates, PAS triples are translated into Graph Patterns (GPs), which convey into a formal query executed against the KGs. If the query can only be answered by merging across graphs, the GPs from the same or different sources are merged, i.e., the GP search is expanded only if required and the situational sources are included when the current data are insufficient.

In [103] is proposed a LSTM framework for VQA (Visual Question Answering) that combines an internal representation of an image's content with text-based information extracted from a general KB to answer a broad range of image-based questions. Given a image-question pair, a set of image attributes

are predicted and used to extract relevant information from the external KB (DBpedia). Then, the paragraphs extracted from the KB are encoded and used to train an LSTM, by maximizing the probability of the correct answer. DBpedia is therefore used as situational source for providing targeted data which, integrated with current data, is able to answer questions referred to information not contained in the image.

The AQUA system [76] *partially* covers Situational Source Inclusion, i.e., it retrieves external information when is needed, but the retrieved data does not complement the existing Knowledge Base. AQUA's algorithm executes the query against the knowledge base, and if it does no succeed, the query is reformulated and used for launching a search engine that retrieve documents which satisfy the new query. So, these data are situational, since they are retrieved motivated by the insufficiency of the KB.

4.3 Unstructured Data Preprocessing + Situational Source Inclusion

A set of QA approaches fully cover Data Management by handling ***Unstructured Data Preprocessing and Situational Source Inclusion*** simultaneously. E.g., the work in [86] applies Local-As-View mappings to characterize elements of the sources schemata in an ontology. The ontology is semantically annotated, enabling to resolve problems of ambiguity. Also, the ontology accommodates situational data and is able to deal with their evolution, by exploiting the impact of feedback and monitored data for improving the user's quality of experience.

The study in [49] presents a framework to build domain specialized knowledge graphs, by fusing curated KBs with extracted knowledge, in a way that the resulting graph is used to answer a set of queries. In the use case, given the existing information about use of drones in the KB (YAGO2 ontology [104]), texts are retrieved from the Web and preprocessed for entity and relation extraction. The extracted triples are mapped to entities present in YAGO2, generating a drone graph that allows to continuously stay abreast of information around drones.

In [73], the knowledge sources are accessed by a Query Manager which is based on a shared ontology called CALO. The approach also has a Mediator reasoner, responsible for extracting information from the Web, so that extracted data are mapped into CALO ontology by means of translation axioms. In addition to this correlation, the Mediator provides suitable semantics to the data returned by Web extraction. The Situational Source Inclusion is motivated by insufficiency of current data: first, the answer is searched locally, and when the answering requires facts residing in other sources, the system breaks down for searching partial proofs in the the appropriate sources.

The authors in [105] present an approach to answering definition questions, in which the goal is to return as many relevant "nuggets" of information about a target concept as possible, by using a relational database and a Web dictionary as corpus. The retrieved documents are tokenized into individual

sentences, discarding candidate sentences that do not contain the target term. If no answers are found by using database and dictionary lookup, the system employs traditional document retrieval to extract relevant nuggets, and then the results from all sources are integrated to produce a final answer.

In [106] is presented QuASE (QUestion Answering via Semantic Enrichment), a QA system that mines answers directly from the Web, and meanwhile employs Freebase as a significant auxiliary to further boost the QA performance. Answer candidates are detected from sentences extracted from the Web and linked to entities in Freebase, by means of entity linking tools. By linking answer candidates to the KB, similar answer candidates can be automatically merged, significantly reducing redundancy and noise. Freebase is used as situational source when integrated into a ML model, since it provides the information required for completely answering questions.

One problem faced in the QUARK system proposed by [91] is the lack of uniform conventions in notation by the knowledge resources. For example, for resources that deal with latitudes and longitudes, some may adopt a decimal notation, while others employ degrees, minutes, and seconds. Due to this, QUARK includes important agents that preprocess notations, converting one to another. Each knowledge resource acts as a situational source, invoked when is needed, and the axiomatic theory is responsible for their correlation and integration. In True Knowledge [81], knowledge extraction includes a four-stage process of sentence extraction, simplification, translation, and bootstrapping that allows to extract high-quality facts for the knowledge base. The proposal has a general inference system that dynamically generates fact as needed. These facts can be seen as situational data, since they are generated based on a need, and complement the current knowledge for presenting the user with a concise explanation.

As previously mentioned, the QA system in [69] uses an early-answering strategy for answering a question from structured data and justifying the answer with a secondary corpus. However, if no acceptable justification can be found, or if the question cannot be answered with the structured collection, situational sources are invoked through a basic question strategy, and merged in order to produce a final result. Moreover, the approach includes an Entity Extractor component to eliminate unacceptable or unlike answer candidates (i.e., n-grams) from a retrieved passage, which includes, e.g., removing stopwords and n-grams that only appear in a single passage.

The study in [75] describes the implementation of *Watson*, the IBM's well-known QA system based on *DeepQA*, an architecture that performs at human expert levels of precision, confidence and speed. The sources for Watson include a wide range of encyclopedias, dictionaries, articles, literary works, including sources of specific information. Given a baseline corpus, DeepQA identifies related Web documents, extracts text nuggets from them, score the nuggets by relevance and merge the most informative ones into the expanded corpus. Preprocessing techniques are used to get more detailed analysis of the search

results, such as named entity detection and soft filtering (to prune candidate answers).

The proposal in [82] integrates an internal DW with an external and unstructured data source. The external data are obtained through QA techniques, thus given a question, it is sent to a set of specialized nodes (DW and QA nodes, composed by ontologies) that process it. An Information Retrieval tool in the QA node retrieves the set of documents that is more likely to contain the answer and, once the running of each specialized node is finished, a semi-automatic mapping process is carried out for detecting connections between the QA and DW ontologies. Preprocessing is performed through a normalization process for obtaining the lemma of ontologies' classes and properties.

The following studies *fully* cover Preprocessing and *partially* cover Situational Source Inclusion. The medical QA system (MQAS) proposed in [107] establishes a mapping process between the question and answers on the basis of the datasets. Situational and specific-domain sources are employed for expanding the terms in the question, allowing to use these terms for discovering relationships in the knowledge sources. These sources, although situational, are not used for complementing the existing datasets. In a later preprocessing step, all textual descriptions of the data fields from the datasets are normalized and combined.

Similarly, the proposal in [54] investigates the integration of lexical and external knowledge (Wordnet and Web, respectively) to bridge the gap between query space and document space in QA. The Web can only provide words that occur frequently with the original query terms, but it lacks information on lexical relationships between these terms. To overcome this need, Wordnet is used as situational source to expand the query that are used for searching answer candidates. NL analysis is performed on the candidate answers to extract POS, base noun phrases, and named entities, thus minimizing the noise introduced by the external resources.

Discovering and including a situational source in an approach usually has a definite motivation: supporting human actions and decisions by offering a complete information. In this context, the next subsection presents the Timely Decision Support and discuss how QA systems handle this feature.

5 SDI's Timely Decision Support in QA

The last SDI feature (see Table 1) is **Timely Decision Support**. As stated by [16], if users were aware of events that impact their operations and relationships that are affected by such events, they would have the opportunity to take immediate action. Thus, a situational integration system must provide means to assist users in their decision processes. This involves three subfeatures, which we call *User Guidance, Decision-making Support*, and *Response Time Improvement*. Their requirements are detailed in Table 4, and results are shown in Figure 3.

Table 4 SDI's Timely Decision Support Requirements

Timely Decision Support Features	Requirement for "Complete Feature"	Requirement for "Partial Feature"
Decision-making Support	Approaches present *full* coverage of this feature if (i) they provide valuable responses (such as alerts, recommendations, reports or predictions) that (ii) provide Situation Awareness to the user `AND` support his actions or decisions [17, 32, 33].	This feature is *partially* supported in approaches that do not provide explicit action support, but do provide opportune Situation Awareness to the user [15–17].
User Guidance	We have considered User Guidance in QA systems that presented user participation both in the system operation `OR` evaluation, i.e., by deciding which information are relevant or giving feedback on the responses returned [12, 14].	Not applicable.
Response Time Improvement	We consider this feature *fully* supported in studies that present any technique of time optimization for retrieving answers, e.g., search indexing or parallel processing.	Not applicable.

5.1 Decision-Making Support

Question answering and decision support systems have been independently developed for decades. However, the QA scenario is a friendly environment for decision processes to occur: with the development of high-performance QA systems, which combine natural language processing and information retrieval, users can directly interact with an information system to evaluate evidence gathered automatically [108]. With SDI, this evidence may contain situational data that are at the basis of decision-making, and most importantly, can be used to recommend an action to the user or trigger an alert about the situation identified.

Analyzing studies that only covered ***Decision-Making Support***, the proposal in [40] presents a query answering mechanism for answering ad-hoc queries, and by exploring heterogeneous definitions of indicators formulas, it supports evaluation of cross-organizations performances and produces meaningful comparisons. In [96], the topical interest and expertise from the cross CQA sources are integrated with the Bayesian model in an unified probabilistic, called MultiTEM. MultiTEM is applied to a specific task of expert user recommendation for a given a question, since an expert user tends to provide a good answer.

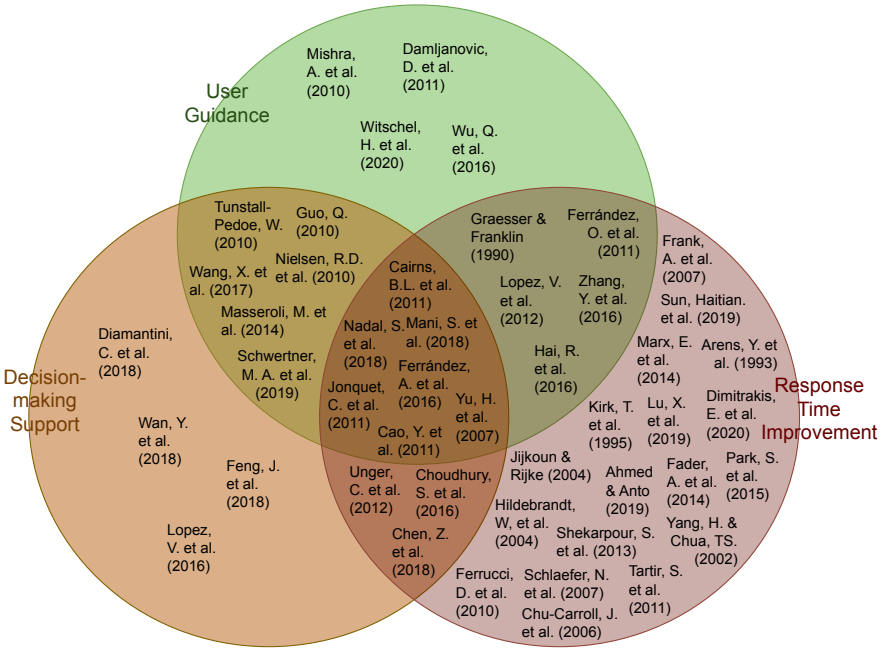

Fig. 3 Timely Decision Support in Question Answering systems.

The algorithm of the medical QA system proposed in [107] creates answers that bridges the question and the corresponding datasets. Optimal answers for decision support are returned to users, including the question, the topic searched, and the knowledge extracted from the datasets. For example, an output of the medical question *Why is my son having a continuous chest congestion?* generates a topic "symptom" and a set of the selected data fields from the datasets "procedure, diagnosis".

The QuerioDALI QA system in [67] was evaluated in two scenarios: an open-domain scenario (using DBpedia and Freebase datasets as QA KGs) and a Smarter Care scenario (containing enterprise data about patient conditions and biomedical ontologies). In the Smarter Care scenario, specifically, the situational integration satisfies the information need of healthcare professionals, by answering questions e.g. related to what are the side-effects for all the medications of a patient.

5.2 User Guidance

Once situational data are retrieved and returned, the user may decide that they are not suitable for the task at hand [12]. This feedback is useful, for example, for the QA system to redirect its search based on learning (i.e., ML techniques). Besides, user guidance can reduce system mistakes if the user informs it, before the Source Discovery process, which information he *is not* looking for.

A few approaches cover ***User Guidance*** individually. In the experiments of the study described in [46], each participant receives a first predefined query, and all subsequent ones require participants to write a question or select a subset of the KG nodes that were displayed. These experiments were based on a specific scenario (domain of books) and had a group of 5 participants that performed exploratory searches on the KG.

In [103], the model was evaluated on two publicly available VQA datasets[13] and involved ground truth answers generated by 10 human subjects. Similarly, in [70], human participants have elaborated a database of questions containing 152 questions related to city domain, in order to evaluate the geographical system proposed.

In the FREyA system [43], the user has an important role to improve the performance of NLIs to ontologies. When the system fails to automatically generate the answer, it will prompt the user with a dialog comprising two tasks: *disambiguation*, in which the user resolves identified ambiguities; and *mapping*, in which the user maps query terms to ontology concepts suggested. Thus, the system learns from the user's selections, and improves its performance over time.

5.3 User Guidance + Decision-Making Support

Some studies have covered both ***Decision Support and User Guidance***: [111] presents an approach for natural language QA over a knowledge base containing medical texts information. For conducting the work, an EHR (Electronic Health Record) system dedicated to Oncology was used, which provided EHR records containing medical appointment data. From the involvement of professionals, it was possible to define sentences of interest, e.g. "what symptoms does the patient report?" and "is the symptom associated with which diagnoses?". The QA process returned a set of answers from the KB, with indication of the correct tuples that support the professional's needs.

In [97], Decision-Making Support is covered since relevant news' documents and representative images are retrieved and presented to the user, who can be aware of the current news. The experiments involved fifteen participants; each one of them selected three query questions from a list of questions provided, and gave evaluation scores to the results returned from the system. The BioSeCo application presented in [47] uses a highly efficient algorithm for rank aggregation, and consensus ranking methods to get a global ranking of results. The results are shown in the interface, so that users can find answers to their biomedical questions, and provide feedback about the relevance of the system and its ranking strategy.

In [42], historical data in medicine were used to structure an automatic diagnosis decision-making table, i.e., a kind of knowledge expression system where conditions are associated with decision rules. Potential diagnosis rules were extracted from the decision-making table, which can offer effective diagnosis service. The PAO ontology (see Subsection 3.1) was provided to users

[13] Toronto COCO-QA [109] and VQA [110].

as an access interface, where they entered the system and raised NL questions. Also, PAO was tested on the QA system in medicine and the generated answers were compared with manual answers of authoritative experts.

In [81], User Guidance is present by means of a process called User Assessment, in which users are able to contradict or endorse existing facts, optionally providing additional sources for the knowledge in platform. The system identifies whether there is a missing knowledge and provides the user with a link to add it, so the user may guide the process. Also, a query-processing engine is capable of tracing the path it followed to generate answers, in order to create a detailed explanation of how those answers were generated. The facts used as part of that proof can be extracted and presented to the user as concise explanation.

5.4 Response Time Improvement

Considering a QA system running on a server, the time required to compute answers and the overhead of the network requests must be taken into account, mainly because the user expects response times of around 1 second [112]. Since agility is crucial for SDI, it could favor the response delivery in QA. Moreover, the situation-aware aspect of SDI favors the acquisition of immediate information, and consequently, a holistic view of various activities, which will be an important factor for making the "best" decision, especially when allied to the interaction feature of QA [113].

This subfeature is the most present one in the reviewed approaches. In the work proposed by [114] (an extension work of [84]), it was implemented parallelization over some components of the system, in order to speedup runtime. Parallelization is also present in [98] (multiple answering agents address the same question in parallel, with results combined) and in [53] (the system sends the question to six streams in parallel).

The QUEST system [50] computes answers in interactive time, with median run-time of 1.5 seconds. The PullNet framework [68] applies ADAM optimizer [115] in the learning process, as well as indexing for fast retrieval. In the Ephyra system [71, 72], duplicated text snippets retrieved from the Web are removed, in order to reduce the processing time. *Response Time Improvement* also appears in the OpenQA framework [83], where a cache service stores the results of processes, so that future requests to the same process can be executed faster.

In the IM system proposed by [88, 89], the query planning algorithms used provides a query interface to distributed structured information sources. The plan executor tries to access the sources in parallel, minimizing the time taken to provide answers. Also, a query processor allows to prune information sources in order to solve completeness and redundancy issues. Dealing with redundancy also improves time, since a minimal set of information sources is determined for answering the query.

Databases and ontologies are queried in the QA system proposed by [94]. Thus, answers can be returned by both MySQL and *Sesame* [116], which

includes the SeRQL query language for handling RDF data. In the search process, path constraints are specified and added as further constraints to the `WHERE` clause of the SeRQL query, simplifying it and speeding up the query performance.

Given a query, the SIMS system [79] generates and executes a *query plan* for accessing the appropriate information sources. Before executing a query, the system performs a query reformulation to minimize the cost and the amount of data processed, so that generated subqueries are executed in parallel. Besides the reformulation and parallelism, the approach presents a cache mechanism for data that are required frequently or are very expensive to retrieve.

In IBM Watson [75] fast runtime indices are created using the Hadoop map-reduce framework[14], so that the system is able to answer more than 85% of the questions in 5 seconds or less. The systems in [77] and [41] include search indexes for offering faster question responses. Similarly, the OQA system in [101] uses a simple KB abstraction where ground facts are represented as string triples (argument1, relation, argument2). Triples are used from curated and extracted KBs, and they are stored in an inverted index that allows for efficient keyword search. Indexes are also used in the QA system proposed in [54], in the SemanticQA system [48], in [105], and in the hybrid QA system described in [52].

5.5 Decision-Making Support + Response Time Improvement

Some studies cover both features, although they do not cover User Guidance. The NOUS framework presented in [49] builds domain specialized KGs by fusing curated KBs with extracted knowledge. The user can visualize the resultant graph and a summarization of quality-related statistics, which provides an overview of the current situation (Situation Awareness). Also, the streaming graph mining algorithm developed in the proposal increases the execution speed of the proposal when discovering trends in streaming data.

The HSIN framework developed in [92] for CQA platforms integrates and simultaneously learns the questions textual contents, their related categories information and user's social interaction. When a user poses a new question, HSIN uses the integrated information for ranking similar historical questions proposed by other users, along with corresponding answers. This recommendation provides Decision-Making Support through Situation Awareness. In addition, the authors used an optimization method to speed up for training time in their HSIN framework.

The proposal in [63] implements a prototype as a freely accessible web application, which allows users to enter their questions and receive answers. The answers are show in a tabular view if appropriate, and the view allows the user to enrich the generated answers by displaying further information for the

[14] https://hadoop.apache.org/

returned resources. The web application thus contributes to Situation Awareness. With respect to Response Time Improvement, a BOA index [117] was created based on Lucene indexer[15], which allows for time-efficient search, e.g., it improves the mapping of properties in natural language queries compared to using a text index.

In all cases where *Decision-Making Support* was covered in addition to *Response Time Improvement*, the support was *partial* through Situation Awareness, which means that there was not an explicit assistance, but the approach was able to make the user aware of the current situation, thus contributing to *Decision-Making Support*.

5.6 User Guidance + Response Time Improvement

The study in [87] covers ***User Guidance and Response Time Improvement***, since the user participates in the evaluation of the system, and the Triple Mapping Component improves time by handling indexing and computational expensive queries. In [90], users can define quality metrics for data quality management in the Constance system. Also, the approach identifies potential foreign keys from the user query, and builds indexes on the corresponding attributes to improve the performance.

In the evaluation of QALL-ME framework [64], users received a brief description of the system (just to make them aware of its capabilities) and elaborated a set of questions containing 304 cinema questions referred to Italy region. The QALL-ME workflow is managed by the QA Planner, which orchestrates the web service components by receiving input parameters, including spatial-temporal context and pattern mappings. As the system response time is directly related to the size of the question pattern set, this set is reduced as much as possible to contain only the essential patterns, thus improving the response time.

In the model proposed by [95], the information sources were constructed with User Guidance: the authors simply approached 10 adults, asked them questions, and tape-recorded their answers. Also, as already mentioned, the model includes arc-search procedures and a constraint propagation component that provide a satisfactory solution to the convergence problem, since they reduce the node space to a minimal set of good answers. Pruning down non-informative answers enables Response Time Improvement.

The joint model proposed in [93] employs *User Guidance* as five questioners were asked to pose questions independently, in order to enrich the evaluation dataset. The model is based on Integer Linear Programming, which was implemented by using Gurobi[16], a fast and powerful mathematical optimization solver that has shown efficiency w.r.t. the average speed in processing questions.

[15] http://lucene.apache.org/
[16] https://www.gurobi.com/

5.7 User Guidance + Decision-Making Support + Response Time Improvement

Some approaches cover all features of SDI's Timely Decision Support together. One of them is [19], which presents the AskHERMES system for answering complex clinical questions. In the Summarization module of the system, the answers are grouped by content terms, thus helping physicians quickly and effectively browsing answer clusters (Response Time Improvement). In the evaluation phase, AskHERMES was compared with state-of-the-art systems, and manual evaluation of the output was performed by three physicians, aiming to examine how well each of the three systems answer the questions. The evaluation found that AskHERMES is competitive with the state-of-the-art systems and its answer presentation interface helps physicians easily obtain information from different points of view (Decision Support).

The work in [86] exploits end-user feedback and runtime data, with the overall goal of improving the quality of experience. Also, wrappers are modeled to accommodate different sources, which deal with query complexity, so that runtime is improved. The results of the study have shown that a great number of changes performed in real-world APIs could be semi-automatically handled by the proposed wrappers and the ontology, thus providing Decision Support when ingesting and analyzing the data.

The MiPACQ system proposed by [44, 45] was evaluated through a clinical dataset and human-annotated answers (User Guidance). The approach includes an Answer Summarization process that performs answer ranking and decides whether to present statistical information about subsets of the results or other potentially interesting aspects of the set of patient records returned (Decision Support). Lucene was used as indexing tool in the Information Retrieval module as time improvement method.

The QA system for IT support in [78] monitors the user's feedback after every dialog turn, and uses this feedback to improve its knowledge. W.r.t. Decision Support, the system provides IT support to users' questions by exploring multimedia data, and furthermore, it comprises a Resolution Automation process, i.e., the platform supports an automation to be attached to an answer. Response Time Improvement is achieved by means of Apache Lucene indexing that enables fast retrieval.

In [80], wrappers are developed for accessing the biomedical resources. This process is performed with a subject matter expert to determine which metadata fields must be processed (User Guidance). The approach provides Situation Awareness by means of the graphical interface that suggests relevant resources to be explored. Also, the Resource Index improves runtime information retrieval.

The MedQA system [51] presents a user with both Web definitions and MEDLINE sentences suitable for his question. These results are shown in an interface, which displays a summary of potential clusters of sentences, along with other relevant sentences that might be additionally important to the biologists or the physicians. The approach uses the indexing tool Lucene in the

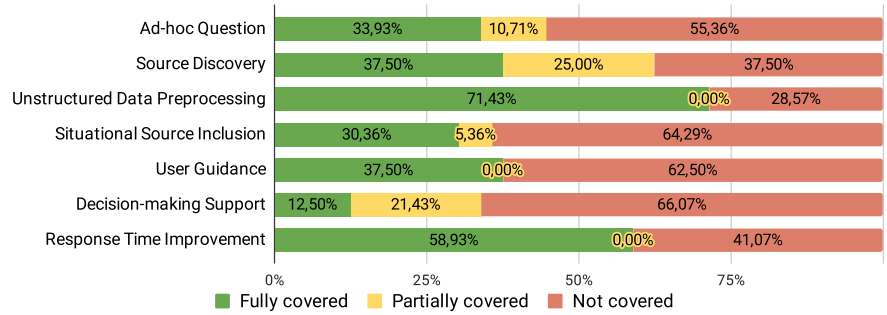

Fig. 4 SDI general amplitude in the surveyed studies.

Document Retrieval step, a technique for Response Time Improvement. User Guidance was also covered in MedQA, since a group of physicians volunteered to participate in the study, evaluating the performance of the system and comparing it with other online information systems. Lastly, in the framework proposed by [82], the question is sent to specialized nodes and the fused output information is sent back to a GUI element, where a dashboard integrates external and internal data as result. The result provides Decision Support and allows the user to correct the information if needed. In the experiments, the QA system consisted of a indexation phase that aimed to prepare all the information required for the search phase, thus optimizing the time response.

Along this section, we reasoned about characteristics of several Question Answering systems that support the presence of SDI features and subfeatures. In addition to this preliminary categorization, in the next section we present some analysis and discussions that provide an overview of SDI in the QA area.

6 Trends and Patterns of Situational Data Integration

This section presents analysis regarding SDI in QA domain. In a first moment, we present an overview of SDI features in the area and outstanding methodological aspects. Next, we show a timeline of SDI in QA, which organizes approaches and features overs two decades. Based on the timeline, we discuss the evolution of SDI and trends in each period of time.

6.1 SDI Coverage in QA Studies

This subsection summarizes how SDI is covered in QA papers, showing the most prominent features. Patterns related to SDI+QA are also discussed, i.e., tasks, sources, and methods that are more frequent in the surveyed studies. Our results are summarized in Figure 4, and a detail description of all SDI features covered in each QA approach is given in Appendix A.

The features coverage in Figure 4 shows that *Unstructured Data Preprocessing* and *Response Time Improvement* were addressed by most of the studies

(71,43% and 58,93%, respectively). Preprocessing is desirable in a SDI-based system, since situational data are often spread over heterogeneous and unstructured sources and need to be accessible for efficiently supporting the user. A summary of Preprocessing techniques in the surveyed papers is presented in Table 5. Among the studies that presented *Unstructured Data Preprocessing*, most deal with duplicates and ambiguities in the data. Resolution of entities - as well as relations between them - are also handled by a considerable number of studies. Concerning *Response Time Improvement*, almost 60% of the studies cover this feature to speed up the processing and delivery of responses to the user. As shown in Table 6, indexing methods are the most used ones for data retrieval and, in this category, it is interesting to highlight that *Lucene* tool [118] is applied in more than half of QA approaches using indexing.

Table 5 Preprocessing Techniques in QA-based Approaches

Preprocessing Techniques	**Application**
Data conversions	[78, 81, 82, 91]
Filtering	[58, 69, 75, 98, 119]
Normalization	[47, 82, 107, 119]
Stopwords removal	[56, 60, 96, 97, 100, 101]
Noise removal/cleaning	[19, 63, 70, 78, 87, 106]
Resolution/exclusion of duplicates and ambiguities	[19, 41, 72, 86–88, 90, 92, 105] [43, 51, 53, 56, 59, 83, 93, 94, 106]
POS tagging	[41, 50, 54, 70, 119]
Entity and Relation Resolution	[41, 44, 49, 50, 54, 58, 75, 100]
Tokenization	[44, 60, 70, 96, 105, 119]
Exclusion/pruning of irrelevant data	[53, 60, 95, 96]
Completeness verification	[42, 87]
Semantic annotations	[60, 73, 80, 86, 90, 98, 119]
Stemming and/or lemmatization	[44, 101, 119]

Another prominent feature is *Source Discovery*, which represents the ability of a system to find a provider for needed data on-the-fly, and adapt the data source according to new requests. We can see in Figure 4 that more than 35% of the studies *fully* cover this feature, and 25% *partially* includes it.

Table 6 Response Time Techniques in QA-based Approaches

Response Time Techniques	**Application**
Search indexing	[41, 48, 54, 75, 77, 82, 90, 101] [44, 51, 52, 63, 68, 78, 80, 87, 105]
Constrained data retrieval	[64, 82, 88, 94, 95]
Parallel processing	[53, 79, 88, 98, 114]
Query rewriting	[79]
Graph-based optimizers	[49, 68]
Cache-based optimizers	[79, 83]
Complexity reduction	[19, 86]
Training acceleration (ML-based approaches)	[68, 92, 119]
Redundancy removal	[72, 88]
Mathematical optimizer	[93]

This means that the majority of QA-based approaches (i.e., more than 60% of them) performs Source Discovery in some way. The use of a non-predefined source is an important requirement for Source Discovery, thus in the papers where the *partial* feature was covered (see subsections 3.2 and 3.3), the sources were not predefined but they had a dynamic nature (e.g. they were updated, exchanged, or "switched off" based on the posed query). Almost all discovered sources are Web-based (including Websites, publicly available KBs, ontologies, and linked datasets) and many studies employ well-known KBs, such as DBpedia, Wikipedia, and Freebase. The considerable amount of studies that partially covers *Source Discovery* indicates that this SDI feature is still missing in QA systems, but is being increasingly explored in a way that could become consolidated in future research.

All other SDI features shown in Figure 4, i.e., *Ad-hoc Question, User Guidance, Situational Source Inclusion*, and *Decision Support*, are integrated less often in the studies. The main challenges lie in the last two mentioned, which means that QA systems do not usually include a situational source for augmenting the data available internally, and are not able to support the user in his actions or decisions. These characteristics allow us to establish some research opportunities with respect to SDI, which will be addressed in Section 7.

Considering the main features of SDI and how they are distributed in the reviewed papers, some characteristics stand out. When considering *Ad-hoc Data Retrieval*, the fact that most studies that cover *Ad-hoc Questions* also cover *Source Discovery* (Figure 1) reinforces the idea that searching for a situational source is often triggered by a specific and momentary need, even though it might also occur in order to answer a conventional query (e.g., a benchmark query). With respect to *Data Management* (Figure 2), besides the evidence of high coverage of *Unstructured Data Preprocessing*, we can observe that this subfeature is also covered by more than half of the studies that presented *Situational Source Inclusion*, demonstrating a concern with the treatment of the data before their effective use. Furthermore, in what concerns *Timely Decision Support* (Figure 3), there is a strong correlation between *Decision-making Support* and *User Guidance*. This is because these subfeatures are more applied together than in isolation, indicating that authors concerned with developing decision support approaches will often take into account the orientation of the end user, e.g., which interventions he finds interesting or how the system can improve its assistance. In addition, *Decision-making Support* is mostly covered in the partial way, indicating that it is still explored at an early level, e.g., without explicit and accurate support.

Only two studies covered Situational Data Integration in a complete way (see Appendix A): the framework for enriching DWs with QA data by [82] and the integration-oriented ontology in [86]. Analyzing these works, we can acknowledge some similarities, such as the use of web-based external sources. The framework described in [82] uses QA for retrieving data from blogs and social networks, whereas [86] retrieves information from Web sources in the form of REST APIs. All external sources contain unstructured data that

complement the information of a structured source, i.e., a dataset [82] or an ontology [86]. Both approaches performed ontological mappings as correlation method when including the situational source. Regarding the query language, natural language queries are considered in [82], whereas [86] manipulates SPARQL queries.

6.2 Timeline Evolution

For investigating the evolution of SDI throughout the years in QA approaches, we have constructed a timeline in which we expose features that have been intensely researched in each time range. The timeline is shown in Figure 5. For constructing it, we have prioritized the most relevant papers, i.e., those with more SDI features covered, besides their overall influence (based on number of citations). For space reasons, we have chosen 40 studies from the total of surveyed papers for composing the timeline, inspired by the feature weighting approach in [120]. For each paper, a relevance score was calculated based on its coverage of SDI features and overall influence, as demonstrated in equations 1, 2 and 3.

$$MFScore(p, m) = \sum_{i=1}^{m_n} (\frac{p_{c_i}}{n}) \tag{1}$$

$$CScore(p) = \frac{p_{tc}}{(y+1) - p_{pub}} \tag{2}$$

$$TScore(p) = w_c \cdot CScore(p) + w_M \sum_{m=1}^{M} MFScore(p, m) \tag{3}$$

The Equation 1 calculates the main feature score $MFScore$ for a paper p, considering a given main feature m. m_n is the total number of subfeatures $\in m$ and p_{c_i} is the paper coverage of each subfeature, which may range from 0 to 10 where 10 means "fully covered", 5 is "partially covered", and 0 stands for "not covered". We do not only consider SDI main features for papers ranking, but also their relevance by means of citations per paper age. So, in Equation 2, we calculated the citation score $CScore$ for a paper p, where p_{tc} is the total citations of the paper[17], y is the current year (i.e. 2020), and p_{pub} is the paper publication year. Finally, Equation 3 determines the total score ($TScore$) of a paper p; it considers the $CScore$ weighted by w_c, added with the summing $MFScore$ for all main features M, considering w_M as its weight. The weight assigned for w_M was 0.3 and the assigned weight for citations w_c received 0.1. The top 40 papers with highest $TScore$ were chosen to compose the SDI timeline.

The timeline shows the most relevant QA studies along with *highlights*, i.e., SDI features that were frequent in each time range. From 2002 to 2005, the majority of the studies incorporated *Data Management*. This means that

[17] We have considered citations number exhibited in August 2020, in Google Scholar, for each paper.

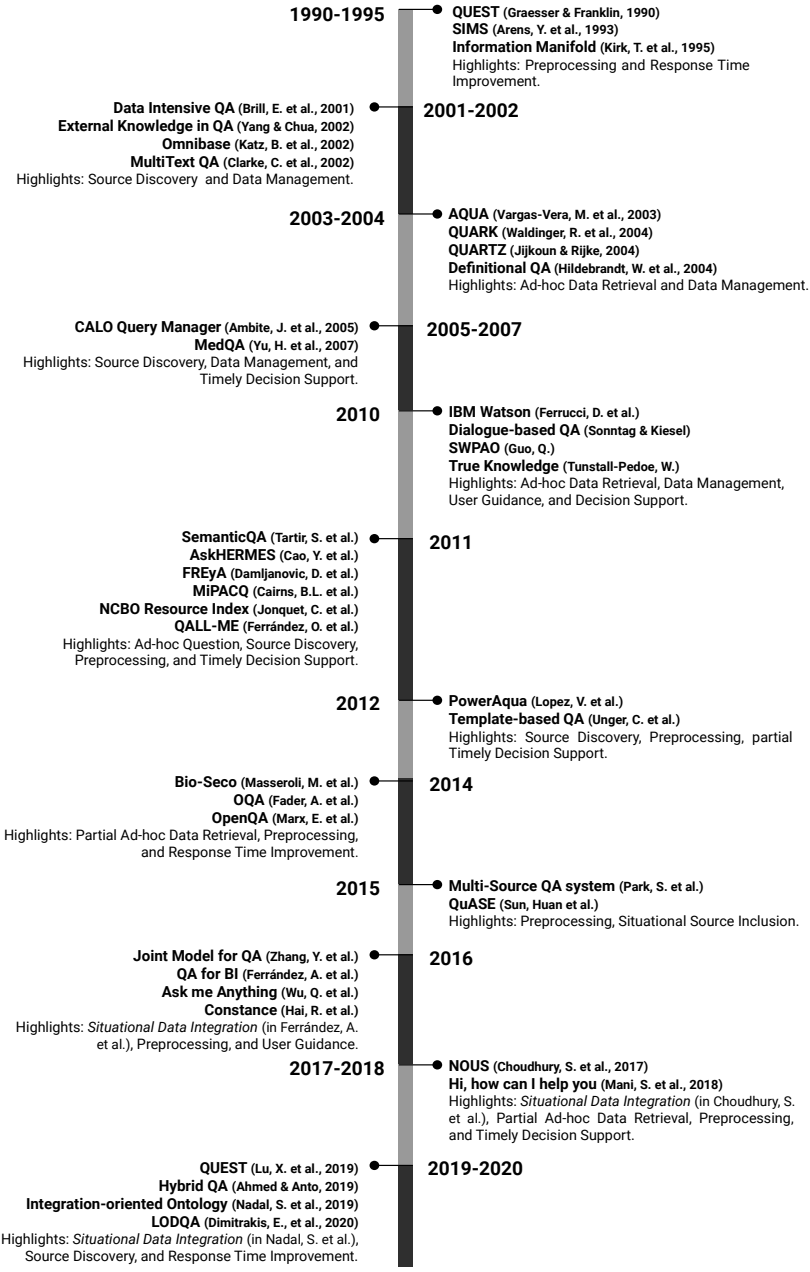

Fig. 5 Timeline of QA approaches. The studies were ranked based on a weighted combination of SDI features and citations, and the top 40 were included in the timeline.

both preprocessing and situational sources were present in the proposals. Situational sources were included by merging components performing syntactic and semantic matchings [73, 74, 105], axiom-based theories [73, 91], contextual and lexical correlations [54], and pattern matching [69]. Situational sources are exploited again in 2010, and included using ensembles of matching algorithms and evidence-gathering techniques [75], as well as semantic mappings [81].

The first occurrence of *Timely Decision Support* was in 2007, with MedQA system [51]. This feature had the highest occurrence in 2011, and interestingly, all papers that covered it in that year and in 2007 were restricted-domain ones, specifically medicine [19, 44, 51] and biomedicine [80]. This fact demonstrates that decision support is being explored not only in areas that involves human action, but also where such human action is decisive (i.e., with low tolerance for errors), in a way that an assistive system can greatly impact the efficiency of the offered services.

In 2010 and 2011 there were several new approaches exploring SDI features, and also in these years, *Ad-hoc Data Retrieval* was present (at least partially) in almost all approaches. In this period, knowledge bases and ontologies stood out either as internal sources [19, 48, 64, 70, 75, 81, 85] or external sources [19, 42–44, 70, 75, 80, 81], and correlation methods were mostly based on semantic and ontological mappings in these studies. In addition, natural language and SPARQL queries were predominant in all studies excepting [80], where the search is performed using keywords.

Unstructured Data Preprocessing was predominant in all time ranges until 2019 and the most consolidated SDI feature, as in every time range there was a concern about handling noisy and raw data to obtain better system performance. The methods are diverse (consider Table 5): in the previous years, they were less present and more focused on filtering [69, 98], pruning resources [53, 95] and removing data duplicates [53, 56, 88, 105]; as of 2007, NLP and semantic-based methods (such as POS tagging, tokenization, entity resolution, stopwords and noise removal) have gained focus.

Finally, Situational Data Integration was completely covered for the first time in 2016, in the framework proposed by [82]. It also occurred in 2017 [49] and 2019 [86]. However, there is a trend starting in 2010, since other studies which covered most part of SDI features were published after 2009 [44, 75, 78, 80, 81, 87, 90]. Of these studies, the greater part (i.e., 5/7) was published between 2010 and 2012. In this time range, *Situational Source Inclusion* was still a challenging feature, whereas *Timely Decision Support* starts to become more frequent in the proposals. In this context, and considering the beginning of SDI in the timeline, we can infer that existing needs since 2010 were gradually being addressed as strategic decisions in several areas became more urgent.

7 Research Opportunities

Based on the analysis of Section 6, in this section we draw some research opportunities concerning Situational Data Integration, based on the most challenging features encountered in QA approaches. As stated in Section 6, some SDI features are moving towards consolidation in QA systems (such as *Unstructured Data Preprocessing, Response Time Improvement*, and *Source Discovery*), even though they still involve many aspects to be explored. Despite this fact, all other features are rarely covered (see Figure 4), e.g., *Situational Source Inclusion, User Guidance, Ad-hoc Question*, and *Decision Support*, which are discussed in the following.

Almost 65% of the studies did not cover *Situational Source Inclusion*. In the context of SDI, it is strongly connected to *Source Discovery* since the situational source is not a pre-existing source, but usually the result of a search for specific data. In other words, the core of SDI is a *discovered and dynamic situational source*, meaning that situational data loses its value if it is not retrieved when needed. Traversal-based query approaches might be investigated to deal with this task, since they traverse data links on the Web of Linked Data, allowing to discover data from initially unknown sources [121].

Another opportunity regarding discovery and inclusion of situational data may be the exploration of semi-automatic methods for source matching, since purely automated methods are often uncertain when discovering data [122]. Semi-automatic methods could involve, e.g., the integration of *human knowledge* for solving ambiguities and generating rules for the system operation [123]. So, when the system is searching for the "best" data source to be integrated, the user feedback can help refine the results and fine-tune data models. Indeed, QA systems are propitious environments for this knowledge to be captured: A QA system empowered with language models, for example, could incrementally retrieve and integrate data over time by asking users to confirm matches on-the-fly, as occurs in pay-as-you-go data integration [124]. Thus, when the feedback is positive, the source recommendation is reinforced, otherwise, a more suitable option is retrieved and presented next time.

When analyzing *Ad-hoc Question* in Figure 4, we see that it is covered by more than 30% of the studies, but still this represents a low inclusion rate. In fact, a considerable part of the approaches includes benchmarks of complex questions, which are quite valuable for evaluation purposes, but since the questions are stored and reused, they do not reflect specific and momentary needs the same way as ad-hoc questions. It is essential that this momentary need is captured and managed in order to deliver dynamic and timely responses, as opposed to results that may suffer depreciation (i.e., lose meaning or value) over time. *Chatbots* are conversational interfaces that have this ability: they simulate an intelligent conversation with the user, capturing immediate needs and bridging the gap between him and an information source [125, 126], thus they can favor the capture of ad-hoc questions. Also, we identified that, among the question benchmarks used by the reviewed proposals, there was no specific collection to evaluate SDI in QA. However, with recent advances in NLP

and deep learning, it is necessary to introduce new datasets for testing the ability of language models to make inferences that rely on Situation Awareness [127]. Thus, comparative results involving SDI features in QA domain could be promoted.

With respect to *User Guidance* and *Decision Support*, although they are independent features, the former has great impact on the latter's efficiency. Consider, for example, the application of SDI in a clinical QA system (see Subsection 2.1): after integrating situational data with data owned by the therapist, the system can identify patterns and infer in which therapeutic activities these patterns impact. Knowing this information, the therapist can make appropriate decisions, considering the patient context. Although in this scenario *Decision Support* is beneficial, this support can be optimized with *User Guidance*, e.g., if the physician define portions of data that are of his interest, by including constraints or assigning weights to sentences involved in the query. Following the example of user feedback applied to source discovery and integration, here the guidance may also be a feedback from the professional: the manual correction of mistaken correlations made by the system, or a manifestation about the usefulness of the results. In both cases (instructions or feedbacks), *Decision Support* can be leveraged, seeing that past experiences can be used to train the system to give better responses.

Another way to leverage *Decision Support* could be investing more on user experience, through interfaces where the user can interact with results and recommendations. QA systems also favor this interactivity. Some reviewed studies [47, 80] expressed future goals which we have taken as clues on how to strengthen the adoption of this feature: the former aims to improve the user interface, whereas the latter aims to guide the exploration of available data resources towards the one that are more appropriate for his needs. Both tasks can potentially improve decision support. Similarly, [97] wants to explore a daily report for generating informative sentences in a short period, which favors decision support in a timely manner.

In a general way, the research opportunities considering SDI in Question Answering include, primarily: (1) Decision Support; (2) Situational Source Inclusion; (3) User Guidance; (4) Ad-hoc Question. Features that move towards consolidation, but still need attention in researches are: (1) Unstructured Data Preprocessing; (2) Response Time Improvement; (3) Source Discovery. It is important to recall that the optimization of one feature can positively impact other features. E.g., if Source Discovery is improved in the sense that relevant knowledge is mined and summarized, the amount of time to provide answer could decrease, thus bettering the coverage of Response Time Improvement. Likewise, promoting the inclusion of situational data sources can leverage decision support, and especially when combined with QA systems, many areas such as healthcare, education, e-Tourism, and BI can be benefited. On the other hand, trade-offs among the features are also a possibility, e.g., focusing on achieving the finest Source Discovery method, with great accuracy, might slow down the response time. Thus, investigating positive and negative

effects when working on SDI features, as well as assessing an ideal fine-tune, are really interesting research directions.

Finally, the features presented in this paper and analyzed in QA-based studies are not exclusive to SDI, since some of them can be found in other data integration variants [36, 128–130]. This means that the research opportunities mentioned in this section are not limited to SDI and QA, but can be extended to other types of data integration that run in QA and have tasks in common with situational integration.

7.1 The Years Ahead

This survey covered studies published over two decades (2000-2020), as a way to track the SDI inclusion in these years. Considering this time range assumed as requirement for our analyzes, it is important to recall that papers published after 2020 were not considered, however, they represent opportunities for future investigation.

Visual Question Answering is a hot topic, specially in the medical area, where the systems provide support (for both doctors and patients) during the treatment. The authors in [131] point out that most of the existing medical VQA methods rely on external data for transfer learning. In fact, there is a new trend to solve VQA by introducing external knowledge, which has already achieved promising results [132], so that we can infer the SDI usefulness for providing targeted external data to overcome eventual data limitation issues (either for training a model or complementing a knowledge base). Recent VQA approaches seem to present SDI features by, e.g., performing information fusion with different sources [133] or including human collaboration [134].

Knowledge graphs are also on focus in recent QA studies, supported by the continuous growth of Linked Open Data. The authors in [135] present a very interesting approach that, instead of relying on data retrieval from *static* knowledge graphs, generates contextually relevant knowledge, which is needed for the integration but not often available in current KBs. Graph completion for QA is also being widely studied in recent years, as a way to better represent the knowledge [136, 137]. Situational sources could be investigated in this context, in order to complete missing facts in KGs.

A recent survey [138] has proposed a taxonomy for classifying QA skills along five dimensions: inference, retrieval, input interpretation/manipulation, world modeling, and multi-step. We believe that SDI could be particularly interesting for the "world modeling" dimension, where knowledge sources often need to be combined with spatial, temporal, causal and motivational elements. When considering these elements for situational integration, features such as *Source Discovery* and *Decision Support* could be potentially improved.

Besides the topics mentioned above, we call the attention to the extensive advances involving deep learning models [139–142]. In fact, we have been witnessing the "explosion" of Large Language Models (LLMs) such as GPTs (Generative Pre-trained Transformers), which can understand and generate content according to the current context and specific situations [126]. Since

LLMs are trained on vast amounts of data (including situational data), they have been applied in several QA-based solutions that leverage context and memory to generate a specialized conversation flow [143–146]. An outstanding example is the ChatGPT[18], a powerful chatbot based on the GPT architecture that answers complex questions with human-level performance.

A system like ChatGPT differs from a QA system designed for situational integration since it does not have direct access to databases, but operates based on the data it has been trained on [147]. Also, it aims at simulating a human-like conversation through text generation, rather than facilitating information retrieval or data manipulation (as occurs in a QA system focused on situational tasks). Still, these models can be used to assist SDI in many innovative ways. E.g., they can be an effective source of situational data due to API integration capabilities, retrieving real-time information from heterogeneous sources, specially when data augmentation is needed [148]. Also, LLMs can be customized for operating in specific scenarios, and their ability of context interpretation represents a notable opportunity for enhancing *Decision Support* for users. In clinical domain, for example, the use of ChatGPT as QA tool has proven valuable for assisting the professional in diagnosis and treatment planning [149, 150]. Besides medicine, there is a wide range of applications involving large language models that should be leveraged by future studies for building QA interfaces with situational capabilities [151].

As previously mentioned, an important research opportunity concerns the investigation of *User Guidance* as strategy for incremental learning in QA, particularly for *Source Discovery* processes. Considering LLMs as a huge trending topic, they could be intensely explored to retrieve a source not only based on similarity with the question (or the users perception about the source suitability), but on organized evidence and temporal evolution of the facts, which are also relevant for building a recommendation [152].

Although the integration of human knowledge can compensate a possible lack of quality or completeness when retrieving situational data, it also requires attention in the way this knowledge is used. As discussed in [138], the intensive use of social media brings us several challenges with a highly informal speech that contains unique terms and misspellings, influencing the way in which data is discovered and retrieved. Most importantly, handling human knowledge in future QA tasks should consider security, ethical and privacy issues, that become more critical with the advent of generative models [153].

While the rapid evolution of NLP undoubtedly warrants attention, it is important to emphasize that our focus on Situational Data Integration serves a distinct purpose that complements, rather than competes with, the advancements in question answering facilitated by LLMs. SDI addresses the complex challenge of integrating and synthesizing heterogeneous data sources to provide contextually relevant responses, which can be assisted by a natural language interface such as a QA system. While LLMs play a crucial role in enhancing question answering capabilities, our research contributes to the broader NLP

[18] https://openai.com/chatgpt

landscape by offering insights into the practical application of SDI features within conversational systems.

We believe the opportunities mentioned in this subsection can contribute to the evolving scenario of situational data management in the QA domain. Future researchers can take advantage of the challenging topics, towards the development of more dynamic QA systems that can make room for situational data inclusion in several applications areas, with user support as ultimate goal.

8 Conclusions

In this paper we have surveyed Situational Data Integration (SDI) in the Question Answering domain, by identifying common characteristics of SDI (i.e., *Ad-hoc Data Retrieval, Data Management*, and *Timely Decision Support*) presented by QA papers published in the last two decades. In our selection process, we have chosen more than 50 studies related to multiple source QA, as this characteristic favors the investigation of situational data. We analyzed the selected studies according to different levels of coverage for SDI features in the approaches and verifying which of them were more prominent. In this regard, we identified that some features such as *Unstructured Data Preprocessing, Response Time Improvement*, and *Source Discovery* are researched to a greater extent, as they are covered by more than half of the studies.

Although few studies covered all the main features of SDI, we explored these studies with more details to detect patterns. We have discovered some similarities shared by them, such as the presence of web-based external sources and ontological mappings when including situational data. When analyzing SDI over time through a Timeline of QA approaches starting in 1990 and ending in 2020, we realized that several innovative approaches in 2010 and 2011 (most of them presenting *Ad-hoc Data Retrieval*), the predominance of *Data Management* from 2002 to 2005 and *Unstructured Data Preprocessing* in all time ranges, as well as the first occurrence of full SDI in 2016.

Considering the aspects discussed, we presented some research opportunities encompassing SDI features that are still challenging in the Question Answering domain. Sorting by the most demanding needs, we can mention: *Decision-making Support, Situational Source Inclusion, User Guidance*, and *Ad-hoc Question*. These issues demonstrate that QA systems of the last decades are often unable to include human knowledge in their actions, comprehend prompt and specific needs, augment the current data with relevant discovered information, and use this integration for supporting human action.

Also, considering the rapid evolution of NLP techniques in the current decade, we discussed some opportunities identified in a set of recent publications, highlighting the role of large language models (LLMs) for supporting situational data management. By elucidating the specific contributions of our work in bridging the gap between Situational Data Integration and QA systems, we aimed to provide valuable insights for researchers and practitioners

seeking to leverage both traditional NLP techniques and emerging LLM technologies, which can address the unique challenges posed by situational data integration in conversational AI applications.

Finally, based on the research needs accentuated in this survey, we expect the results shared to be useful as a guide for incorporating Situational Data Integration in several applications. As future work, we aim to apply SDI features in a QA system for querying open databases, specially those involving speech therapy data, since we believe that including SDI in Question Answering methodologies can leverage smart and adaptive systems. We are also interested in applying language models and user feedback for improving Source Discovery tasks in such situational QA system. Finally, acknowledging the need for a timely answer as an indispensable aspect to provide user support, we aim to investigate trade-offs involving response time in our future studies.

Acknowledgments. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brazil (CAPES) - Finance Code 001.

Appendix A SDI Features in QA-based Approaches

This appendix provides information on all QA approaches surveyed and SDI features covered by them. These information are shown in Table A1.

References

[1] Diefenbach, D., Lopez, V., Singh, K., Maret, P.: Core techniques of question answering systems over knowledge bases: a survey. Knowledge and Information systems **55**(3), 529–569 (2018)

[2] Bouziane, A., Bouchiha, D., Doumi, N., Malki, M.: Question answering systems: survey and trends. Procedia Computer Science **73**, 366–375 (2015)

[3] Roy, R.S., Anand, A.: Question answering over curated and open web sources. arXiv preprint arXiv:2004.11980 (2020)

[4] Utomo, F.S., Suryana, N., Azmi, M.S.: Question answering system: a review on question analysis, document processing, and answer extraction techniques. Journal of Theoretical and Applied Information Technology **95**(14), 3158–3174 (2017)

[5] Ferrández, A., Peral, J.: The benefits of the interaction between data warehouses and question answering. In: Proceedings of the 2010 EDBT/ICDT Workshops, pp. 1–8 (2010)

Table A1 SDI features fully covered (✓), partially covered (❊), and not covered (✗) in QA-based approaches

	Ad-hoc Data Retrieval		Data Management		Timely Decision Support		
Approach	*Ad-hoc Question*	*Source Discovery*	*Unstructured Data Preprocessing*	*Situational Source Inclusion*	*User Guidance*	*Decision Support*	*Response Time Improvement*
Ferrández, A. et al. [82]	✓	✓	✓	✓	✓	✓	✓
Wang, X. et al. [97]	✗	✗	✓	✗	✓	❊	✗
Shekarpour, S. et al. [84][114]	✗	✗	✗	✗	✗	✗	✓
Schwertner, M. A. et al. [111]	✗	✗	✗	✗	✓	✓	✗
Diamantini, C. et al. [40]	✓	✗	✗	✓	✗	❊	✗
Masseroli, M. et al. [47]	❊	❊	✓	✗	✓	❊	✗
Sonntag & Kiesel [85]	✓	✓	✗	✓	✗	✗	✗
Lu, X. et al. [50]	✗	✓	✓	✗	✗	✗	✓
Wan, Y. et al. [96]	✗	✗	✓	✗	✗	❊	✗
Choudhury, S. et al. [49]	❊	✓	✓	✓	✗	❊	✓
Guo, Q. [42]	✓	✗	✓	✗	✓	✓	✗
Nadal, S. et al. [86]	✓	✓	✓	✓	✓	❊	✓
Tartir, S. et al. [48]	❊	✓	✗	✓	✗	✗	✓
Lopez, V. et al. [67]	✗	❊	✗	✓	✗	❊	✗
Chen, Z. et al. [92]	✗	✗	✓	✗	✗	❊	✓
Wu, Q. et al. [103]	✗	✗	✗	✓	✓	✗	✗
Witschel, H. et al. [46]	❊	✗	✗	✗	✓	✗	✗
Cao, Y. et al. [19]	✗	✗	✓	✗	✓	✓	✓
Katz, B. et al. [74]	✓	✓	✗	✗	✗	✗	✗
Kirk, T. et al. [88][89]	❊	✓	✓	✗	✗	✗	✓
Vargas-Vera, M. et al. [76]	✓	✓	✗	❊	✗	✗	✗
Hildebrandt, W. et al. [105]	✗	✗	✓	✓	✗	✗	✓
Ferrucci, D. et al. [75]	✓	✓	✓	✓	✗	✗	✓
Schlaefer, N. et al. [72][71]	✗	❊	✓	✗	✗	✗	✓
Lopez, V. et al. [87]	✓	✓	✓	✗	✓	✗	✓
Cairns & Nielsen et al. [44][45]	✓	✗	✓	✗	✓	✓	✓
Dimitrakis, E. et al. [77]	✓	✓	✗	✗	✗	✗	✓
Park, S. et al. [41]	✓	❊	✓	✗	✗	✗	✓
Ambite, J. et al. [73]	✗	❊	✓	✓	✗	✗	✗
Fader, A. et al. [101]	✗	✗	✓	✗	✗	✗	✓
Mishra, A. et al. [70]	✗	❊	✓	✗	✓	✗	✗
Waldinger, R. et al. [91]	✓	❊	✓	✓	✗	✗	✗
Hai, R. et al. [90]	✓	✓	✓	✗	✓	✗	✓
Marx, E. et al. [83]	✓	❊	✓	✗	✗	✗	✓
Lv, S. et al. [100]	✗	✗	✓	✗	✗	✗	✗
Feng, J. et al. [107]	✗	✗	✓	❊	✗	✓	✗
Tunstall-Pedoe, W. [81]	✓	❊	✓	✓	✓	❊	✗
Yang & Chua [54]	✗	✓	✓	❊	✗	✗	✓
Frank, A. et al. [94]	✗	✗	✓	✗	✗	✗	✓
Unger, C. et al. [63]	✗	✓	✓	✗	✗	❊	✓
Damljanovic, D. et al. [43]	✓	✗	✓	✗	✓	✗	✗
Arens, Y. et al. [79]	✓	❊	✗	✗	✗	✗	✓
Ferrández, Ó. et al. [64]	✗	✓	✗	✗	✓	✗	✓
Brill, E. et al. [56]	✗	✓	✓	✗	✗	✗	✗
Graesser & Franklin [95]	✗	✗	✓	✗	✓	✗	✓
Jonquet, C. et al. [80]	❊	❊	✓	✗	✓	❊	✓
Yu, H. et al. [51]	✗	✓	✓	✗	✓	❊	✓
Chu-Carroll, J. et al. [98]	✗	✗	✓	✗	✗	✗	✓
Ahmed & Anto [52]	✗	✓	✗	✓	✗	✗	✓
Savenkov & Agichtein [65]	✗	✓	✗	✓	✗	✗	✗
Sun, Huan et al. [106]	✗	✗	✓	✓	✗	✗	✗
Mani, S. et al. [78]	✓	❊	✓	✗	✓	✓	✓
Sun, Haitian et al. [68]	✗	❊	✗	✗	✗	✗	✓
Zhang, Y. et al. [93]	✗	✗	✓	✗	✓	✗	✓
Jijkoun & Rijke [53]	✗	✓	✓	✗	✗	✗	✓
Clarke, C. et al. [69]	✗	❊	✓	✓	✗	✗	✗

[6] Allam, A.M.N., Haggag, M.H.: The question answering systems: A survey. International Journal of Research and Reviews in Information Sciences (IJRRIS) **2**(3) (2012)

[7] Tong, P., Zhang, Q., Yao, J.: Leveraging domain context for question answering over knowledge graph. Data Science and Engineering **4**(4), 323–335 (2019)

[8] Kodra, L., Meçe, E.K.: Question answering systems: A review on present developments, challenges and trends. International Journal of Advanced Computer Science and Applications **8**(9), 217–224 (2017)

[9] Wu, M., Marian, A.: Corroborating answers from multiple web sources. In: WebDB, pp. 1–6 (2007)

[10] Rasiq, G.R.I., Al Sefat, A., Hossain, T., Munna, M.I.-E.-H., Jisha, J.J., Hoque, M.M.: Question answering system over linked data: A detailed survey. ABC Research Alert **8**(1), 32–47 (2020)

[11] Jovanovic, P., Romero, O., Abelló, A.: A unified view of data-intensive flows in business intelligence systems: a survey. Transactions on Large-Scale Data-and Knowledge-Centered Systems XXIX, 66–107 (2016)

[12] Abelló, A., Darmont, J., Etcheverry, L., Golfarelli, M., Mazón, J.-N., Naumann, F., Pedersen, T., Rizzi, S.B., Trujillo, J., Vassiliadis, P., *et al.*: Fusion cubes: Towards self-service business intelligence. International Journal of Data Warehousing and Mining (IJDWM) **9**(2), 66–88 (2013)

[13] Löser, A., Hueske, F., Markl, V.: Situational business intelligence. In: International Workshop on Business Intelligence for the Real-Time Enterprise, pp. 1–11 (2008). Springer

[14] Han, Y., Wang, G., Ji, G., Zhang, P.: Situational data integration with data services and nested table. Service Oriented Computing and Applications **7**(2), 129–150 (2013)

[15] Vo, Q.D., Thomas, J., Cho, S., De, P., Choi, B.J.: Next generation business intelligence and analytics. In: Proceedings of the 2nd International Conference on Business and Information Management, pp. 163–168 (2018)

[16] Castellanos, M., Gupta, C., Wang, S., Dayal, U., Durazo, M.: A platform for situational awareness in operational bi. Decision Support Systems **52**(4), 869–883 (2012)

[17] Nadj, M., Morana, S., Maedche, A.: Towards a situation-awareness-driven design of operational business intelligence & analytics systems. In: At the Vanguard of Design Science: First Impressions and Early Findings from Ongoing Research Research-in-Progress Papers and Poster Presentations from the 10th International Conference, DESRIST 2015. Dublin, Ireland, 20-22 May., pp. 33–40 (2015). DESRIST 2015

[18] Vila, K., Ferrández, A.: Model-driven restricted-domain adaptation of question answering systems for business intelligence. In: Proceedings of the 2nd International Workshop on Business Intelligence and the WEB, pp. 36–43 (2011)

[19] Cao, Y., Liu, F., Simpson, P., Antieau, L., Bennett, A., Cimino, J.J., Ely, J., Yu, H.: Askhermes: An online question answering system for

complex clinical questions. Journal of biomedical informatics **44**(2), 277–288 (2011)

[20] Singh, K., Radhakrishna, A.S., Both, A., Shekarpour, S., Lytra, I., Usbeck, R., Vyas, A., Khikmatullaev, A., Punjani, D., Lange, C., *et al.*: Why reinvent the wheel: Let's build question answering systems together. In: Proceedings of the 2018 World Wide Web Conference, pp. 1247–1256 (2018)

[21] Gupta, P., Gupta, V.: A survey of text question answering techniques. International Journal of Computer Applications **53**(4) (2012)

[22] Figueroa, A., Neumann, G.: Learning to rank effective paraphrases from query logs for community question answering. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 27, pp. 1099–1105 (2013)

[23] Figueroa, A., Neumann, G.: Category-specific models for ranking effective paraphrases in community question answering. Expert Systems with Applications **41**(10), 4730–4742 (2014)

[24] Nargesian, F., Zhu, E., Miller, R.J., Pu, K.Q., Arocena, P.C.: Data lake management: challenges and opportunities. Proceedings of the VLDB Endowment **12**(12), 1986–1989 (2019)

[25] Patil, C., Patwardhan, M.: Visual question generation: The state of the art. ACM Computing Surveys (CSUR) **53**(3), 1–22 (2020)

[26] Bavaresco, R., Silveira, D., Reis, E., Barbosa, J., Righi, R., Costa, C., Antunes, R., Gomes, M., Gatti, C., Vanzin, M., *et al.*: Conversational agents in business: A systematic literature review and future research directions. Computer Science Review **36**, 100239 (2020)

[27] Kantorovitch, J., Niskanen, I., Kalaoja, J., Staykova, T.: Designing situation awareness - addressing the needs of medical emergency response. In: Proceedings of the 12th International Conference on Software Technologies (ICSOFT 2017), pp. 467–472 (2017)

[28] Endsley, M.R.: Toward a theory of situation awareness in dynamic systems. Human factors **37**(1), 32–64 (1995)

[29] Wang, G., Fang, J., Han, Y.: Interactive recommendation of composition operators for situational data integration. In: 2013 International Conference on Cloud and Service Computing, pp. 120–127 (2013). IEEE

[30] Ahmed, S., Ruhi, U.: Towards a functional taxonomy of enterprise business intelligence mashups. In: 2013 Second International Conference on Informatics & Applications (ICIA), pp. 98–103 (2013). IEEE

[31] Mountantonakis, M., Tzitzikas, Y.: Large-scale semantic integration of linked data: A survey. ACM Computing Surveys (CSUR) **52**(5), 1–40 (2019)

[32] Bonura, S., Cammarata, G., Finazzo, R., Francaviglia, G., Morreale, V.: A novel webgis-based situational awareness platform for trustworthy big data integration and analytics in mobility context. In: OTM Confederated International Conferences" On the Move to Meaningful Internet Systems", pp. 86–98 (2017). Springer

[33] Dragos, V., Gatepaille, S.: On-the-fly integration of soft and sensor data for enhanced situation assessment. Procedia computer science **112**, 1263–1272 (2017)

[34] Serban, F., Vanschoren, J., Kietz, J.-U., Bernstein, A.: A survey of intelligent assistants for data analysis. ACM Computing Surveys (CSUR) **45**(3), 1–35 (2013)

[35] Hartig, O., Özsu, M.T.: Walking without a map: Ranking-based traversal for querying linked data. In: International Semantic Web Conference, pp. 305–324 (2016). Springer

[36] Curry, E., Derguech, W., Hasan, S., Kouroupetroglou, C., ul Hassan, U.: A real-time linked dataspace for the internet of things: enabling "pay-as-you-go" data management in smart environments. Future Generation Computer Systems **90**, 405–422 (2019)

[37] Sehar, U., Ghazal, I., Mansoor, H., Saba, S.: A comprehensive literature review on approaches, techniques & challenges of mashup development. International Journal of Scientific & Engineering Research **13** (2022)

[38] Franciscatto, M.H., Augustin, I., Lima, J.C.D., Maran, V.: Situation awareness in the speech therapy domain: A systematic mapping study. Computer Speech & Language **53**, 92–120 (2019)

[39] Zirpins, C.: Situational data-analytics for the web-of-things. In: Proceedings of the 1st International Workshop on Mashups of Things and APIs, pp. 1–4 (2016)

[40] Diamantini, C., Potena, D., Storti, E.: Multidimensional query reformulation with measure decomposition. Information Systems **78**, 23–39 (2018)

[41] Park, S., Kwon, S., Kim, B., Han, S., Shim, H., Lee, G.G.: Question answering system using multiple information source and open type answer merge. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics:

Demonstrations, pp. 111–115 (2015)

[42] Guo, Q.-l.: A novel approach for question answering and automatic diagnosis based on pervasive agent ontology in medicine. International Journal of Intelligent Systems **25**(7), 655–682 (2010)

[43] Damljanovic, D., Agatonovic, M., Cunningham, H.: Freya: An interactive way of querying linked data using natural language. In: Extended Semantic Web Conference, pp. 125–138 (2011). Springer

[44] Cairns, B.L., Nielsen, R.D., Masanz, J.J., Martin, J.H., Palmer, M.S., Ward, W.H., Savova, G.K.: The mipacq clinical question answering system. In: AMIA Annual Symposium Proceedings, vol. 2011, p. 171 (2011). American Medical Informatics Association

[45] Nielsen, R.D., Masanz, J., Ogren, P., Ward, W., Martin, J.H., Savova, G., Palmer, M.: An architecture for complex clinical question answering. In: Proceedings of the 1st ACM International Health Informatics Symposium, pp. 395–399 (2010)

[46] Witschel, H.F., Riesen, K., Grether, L.: Kvgr: A graph-based interface for explorative sequential question answering on heterogeneous information sources. In: European Conference on Information Retrieval, pp. 760–773 (2020). Springer

[47] Masseroli, M., Picozzi, M., Ghisalberti, G., Ceri, S.: Explorative search of distributed bio-data to answer complex biomedical questions. BMC bioinformatics **15**(S1), 3 (2014)

[48] Tartir, S., Arpinar, I.B., McKnight, B.: Semanticqa: exploiting semantic associations for cross-document question answering. In: International Symposium on Innovations in Information and Communications Technology, pp. 1–6 (2011). IEEE

[49] Choudhury, S., Agarwal, K., Purohit, S., Zhang, B., Pirrung, M., Smith, W., Thomas, M.: Nous: Construction and querying of dynamic knowledge graphs. In: 2017 IEEE 33rd International Conference on Data Engineering (ICDE), pp. 1563–1565 (2017). IEEE

[50] Lu, X., Pramanik, S., Saha Roy, R., Abujabal, A., Wang, Y., Weikum, G.: Answering complex questions by joining multi-document evidence with quasi knowledge graphs. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 105–114 (2019)

[51] Yu, H., Lee, M., Kaufman, D., Ely, J., Osheroff, J.A., Hripcsak, G., Cimino, J.: Development, implementation, and a cognitive evaluation

of a definitional question answering system for physicians. Journal of biomedical informatics **40**(3), 236–251 (2007)

[52] Ahmed, W., Anto, P.B.: A hybrid question answering system. Current Journal of Applied Science and Technology, 1–7 (2019)

[53] Jijkoun, V., De Rijke, M.: Answer selection in a multi-stream open domain question answering system. In: European Conference on Information Retrieval, pp. 99–111 (2004). Springer

[54] Yang, H., Chua, T.-S.: The integration of lexical knowledge and external resources for question answering. In: In the Proceedings of the Eleventh Text REtrieval Conference (TREC 2002), pp. 1–6 (2002)

[55] Miller, G.A.: Wordnet: An electronic lexical database. MIT press (1998)

[56] Brill, E., Lin, J., Banko, M., Dumais, S., Ng, A., *et al.*: Data-intensive question answering. In: TREC, vol. 56, p. 90 (2001)

[57] Abacha, A.B., Agichtein, E., Pinter, Y., Demner-Fushman, D.: Overview of the medical question answering task at trec 2017 liveqa. In: TREC Conference, pp. 1–12 (2017)

[58] Katz, B., Marton, G., Borchardt, G.C., Brownell, A., Felshin, S., Loreto, D., Louis-Rosenberg, J., Lu, B., Mora, F., Stiller, S., *et al.*: External knowledge sources for question answering. In: TREC (2005)

[59] Figueroa, A., Neumann, G., Atkinson, J.: Searching for definitional answers on the web using surface patterns. Computer **42**(4), 68–76 (2009)

[60] Androutsopoulos, I., Galanis, D.: A practically unsupervised learning method to identify single-snippet answers to definition questions on the web. In: Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, pp. 323–330 (2005)

[61] Katz, B., Lin, J., Loreto, D., Hildebrandt, W., Bilotti, M.W., Felshin, S., Fernandes, A., Marton, G., Mora, F.: Integrating web-based and corpus-based techniques for question answering. In: TREC, pp. 426–435 (2003)

[62] Yang, Y., Yih, W.-t., Meek, C.: Wikiqa: A challenge dataset for open-domain question answering. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 2013–2018 (2015)

[63] Unger, C., Bühmann, L., Lehmann, J., Ngonga Ngomo, A.-C., Gerber,

D., Cimiano, P.: Template-based question answering over rdf data. In: Proceedings of the 21st International Conference on World Wide Web, pp. 639–648 (2012)

[64] Ferrández, Ó., Spurk, C., Kouylekov, M., Dornescu, I., Ferrández, S., Negri, M., Izquierdo, R., Tomás, D., Orasan, C., Neumann, G., *et al.*: The qall-me framework: A specifiable-domain multilingual question answering architecture. Journal of Web Semantics **9**(2), 137–145 (2011)

[65] Savenkov, D., Agichtein, E.: When a knowledge base is not enough: Question answering over knowledge bases with external text data. In: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 235–244 (2016)

[66] Yang, H., Chua, T.-S.: Web-based list question answering. In: COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics, pp. 1277–1283 (2004)

[67] Lopez, V., Tommasi, P., Kotoulas, S., Wu, J.: Queriodali: Question answering over dynamic and linked knowledge graphs. In: International Semantic Web Conference, pp. 363–382 (2016). Springer

[68] Sun, H., Bedrax-Weiss, T., Cohen, W.W.: Pullnet: Open domain question answering with iterative retrieval on knowledge bases and text. arXiv preprint arXiv:1904.09537 (2019)

[69] Clarke, C.L., Cormack, G.V., Kemkes, G., Laszlo, M., Lynam, T.R., Terra, E.L., Tilker, P.L.: Statistical selection of exact answers (multitext experiments for trec 2002). In: TREC 2002 Conference Notebook, pp. 1–9 (2002)

[70] Mishra, A., Mishra, N., Agrawal, A.: Context-aware restricted geographical domain question answering system. In: 2010 International Conference on Computational Intelligence and Communication Networks, pp. 548–553 (2010). IEEE

[71] Schlaefer, N., Gieselman, P., Sautter, G.: The ephyra qa system at trec 2006. In: TREC 2006, pp. 1–10 (2006)

[72] Schlaefer, N., Ko, J., Betteridge, J., Pathak, M.A., Nyberg, E., Sautter, G.: Semantic extensions of the ephyra qa system for trec 2007. In: TREC, vol. 1, p. 2 (2007)

[73] Ambite, J.L., Chaudhri, V.K., Fikes, R., Jenkins, J., Mishra, S., Muslea, M., Uribe, T., Yang, G.: Integration of heterogeneous knowledge sources in the calo query manager. Lecture notes in computer science **3762**, 30 (2005)

[74] Katz, B., Felshin, S., Yuret, D., Ibrahim, A., Lin, J., Marton, G., McFarland, A.J., Temelkuran, B.: Omnibase: Uniform access to heterogeneous data for question answering. In: International Conference on Application of Natural Language to Information Systems, pp. 230–234 (2002). Springer

[75] Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A.A., Lally, A., Murdock, J.W., Nyberg, E., Prager, J., *et al.*: Building watson: An overview of the deepqa project. AI magazine **31**(3), 59–79 (2010)

[76] Vargas-Vera, M., Motta, E., Domingue, J.: Aqua: An ontology-driven question answering system. New Directions in Question Answering **8** (2003)

[77] Dimitrakis, E., Sgontzos, K., Mountantonakis, M., Tzitzikas, Y.: Enabling efficient question answering over hundreds of linked datasets. In: International Workshop on Information Search, Integration, and Personalization, pp. 3–17 (2019). Springer

[78] Mani, S., Gantayat, N., Aralikatte, R., Gupta, M., Dechu, S., Sankaran, A., Khare, S., Mitchell, B., Subramanian, H., Venkatarangan, H.: Hi, how can i help you?: Automating enterprise it support help desks. In: Thirty-Second AAAI Conference on Artificial Intelligence, pp. 1–8 (2018)

[79] Arens, Y., Chee, C.Y., Hsu, C.-N., Knoblock, C.A.: Retrieving and integrating data from multiple information sources. International Journal of Intelligent and Cooperative Information Systems **2**(02), 127–158 (1993)

[80] Jonquet, C., LePendu, P., Falconer, S., Coulet, A., Noy, N.F., Musen, M.A., Shah, N.H.: Ncbo resource index: Ontology-based search and mining of biomedical resources. Journal of Web Semantics **9**(3), 316–324 (2011)

[81] Tunstall-Pedoe, W.: True knowledge: Open-domain question answering using structured knowledge and inference. AI Magazine **31**(3), 80–92 (2010)

[82] Ferrández, A., Maté, A., Peral, J., Trujillo, J., De Gregorio, E., Aufaure, M.-A.: A framework for enriching data warehouse analysis with question answering systems. Journal of Intelligent Information Systems **46**(1), 61–82 (2016)

[83] Marx, E., Usbeck, R., Ngomo, A.-C.N., Höffner, K., Lehmann, J., Auer, S.: Towards an open question answering architecture. In: Proceedings of the 10th International Conference on Semantic Systems, pp. 57–60 (2014)

[84] Shekarpour, S., Ngonga Ngomo, A.-C., Auer, S.: Question answering on interlinked data. In: Proceedings of the 22nd International Conference on World Wide Web, pp. 1145–1156 (2013)

[85] Sonntag, D., Kiesel, M.: Linked data integration for semantic dialogue and backend access. In: 2010 AAAI Spring Symposium Series: Linked Data Meets Artificial Intelligence (2010)

[86] Nadal, S., Romero, O., Abelló, A., Vassiliadis, P., Vansummeren, S.: An integration-oriented ontology to govern evolution in big data ecosystems. Information systems **79**, 3–19 (2019)

[87] Lopez, V., Fernández, M., Motta, E., Stieler, N.: Poweraqua: Supporting users in querying and exploring the semantic web. Semantic web **3**(3), 249–265 (2012)

[88] Kirk, T., Levy, A.Y., Sagiv, Y., Srivastava, D., *et al.*: The information manifold. In: Proceedings of the AAAI 1995 Spring Symp. on Information Gathering from Heterogeneous, Distributed Enviroments, vol. 7, pp. 85–91 (1995)

[89] Levy, A., Rajaraman, A., Ordille, J.: Querying heterogeneous information sources using source descriptions. Technical report, Stanford InfoLab (1996)

[90] Hai, R., Geisler, S., Quix, C.: Constance: An intelligent data lake system. In: Proceedings of the 2016 International Conference on Management of Data, pp. 2097–2100 (2016)

[91] Waldinger, R.J., Appelt, D.E., Dungan, J.L., Fry, J., Hobbs, J.R., Israel, D.J., Jarvis, P., Martin, D.L., Riehemann, S., Stickel, M.E., *et al.*: Deductive question answering from multiple resources. New Directions in Question Answering **2004**, 253–262 (2004)

[92] Chen, Z., Zhang, C., Zhao, Z., Yao, C., Cai, D.: Question retrieval for community-based question answering via heterogeneous social influential network. Neurocomputing **285**, 117–124 (2018)

[93] Zhang, Y., He, S., Liu, K., Zhao, J.: A joint model for question answering over multiple knowledge bases. In: Thirtieth AAAI Conference on Artificial Intelligence, pp. 1–7 (2016)

[94] Frank, A., Krieger, H.-U., Xu, F., Uszkoreit, H., Crysmann, B., Jörg, B., Schäfer, U.: Question answering from structured knowledge sources. Journal of Applied Logic **5**(1), 20–48 (2007)

[95] Graesser, A.C., Franklin, S.P.: Quest: A cognitive model of question

answering. Discourse processes **13**(3), 279–303 (1990)

[96] Wan, Y., Xu, G., Chen, L., Zhao, Z., Wu, J.: Exploiting cross-source knowledge for warming up community question answering services. Neurocomputing **320**, 25–34 (2018)

[97] Wang, X., Li, Z., Tang, J.: Multimedia news qa: Extraction and visualization integration with multiple-source information. Image and Vision Computing **60**, 162–170 (2017)

[98] Chu-Carroll, J., Prager, J., Welty, C., Czuba, K., Ferrucci, D.: A multi-strategy and multi-source approach to question answering. Technical report, IBM THOMAS J WATSON RESEARCH CENTER YORKTOWN HEIGHTS NY (2006)

[99] Hofmann, T.: Probabilistic latent semantic indexing. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 50–57 (1999)

[100] Lv, S., Guo, D., Xu, J., Tang, D., Duan, N., Gong, M., Shou, L., Jiang, D., Cao, G., Hu, S.: Graph-based reasoning over heterogeneous external knowledge for commonsense question answering. In: AAAI, pp. 8449–8456 (2020)

[101] Fader, A., Zettlemoyer, L., Etzioni, O.: Open question answering over curated and extracted knowledge bases. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1156–1165 (2014)

[102] Bast, H., Haussmann, E.: More accurate question answering on freebase. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, pp. 1431–1440 (2015)

[103] Wu, Q., Wang, P., Shen, C., Dick, A., Van Den Hengel, A.: Ask me anything: Free-form visual question answering based on knowledge from external sources. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4622–4630 (2016)

[104] Hoffart, J., Suchanek, F.M., Berberich, K., Weikum, G.: Yago2: A spatially and temporally enhanced knowledge base from wikipedia. Artificial Intelligence **194**, 28–61 (2013)

[105] Hildebrandt, W., Katz, B., Lin, J.: Answering definition questions with multiple knowledge sources. In: Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004, pp. 49–56 (2004)

[106] Sun, H., Ma, H., Yih, W.-t., Tsai, C.-T., Liu, J., Chang, M.-W.: Open domain question answering via semantic enrichment. In: Proceedings of the 24th International Conference on World Wide Web, pp. 1045–1055 (2015)

[107] Feng, J., Zhang, R., Chen, D., Zhang, W.: Extracting meaningful correlations among heterogeneous datasets for medical question answering with domain knowledge. In: 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), pp. 297–301 (2018). IEEE

[108] Yang, Z., Li, Y., Cai, J., Nyberg, E.: Quads: question answering for decision support. In: Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval, pp. 375–384 (2014)

[109] Ren, M., Kiros, R., Zemel, R.: Image question answering: A visual semantic embedding model and a new dataset. Proc. Advances in Neural Inf. Process. Syst **1**(2), 5 (2015)

[110] Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Lawrence Zitnick, C., Parikh, D.: Vqa: Visual question answering. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2425–2433 (2015)

[111] Schwertner, M.A., Rigo, S.J., Araújo, D.A., Silva, A.B., Eskofier, B.: Fostering natural language question answering over knowledge bases in oncology ehr. In: 2019 IEEE 32nd International Symposium on Computer-Based Medical Systems (CBMS), pp. 501–506 (2019). IEEE

[112] Diefenbach, D., Singh, K., Maret, P.: On the scalability of the qa system wdaqua-core1. In: Semantic Web Evaluation Challenge, pp. 76–81 (2018). Springer

[113] Ghimire, S., Luis-Ferreira, F., Nodehi, T., Jardim-Goncalves, R.: Iot based situational awareness framework for real-time project management. International Journal of Computer Integrated Manufacturing **30**(1), 74–83 (2017)

[114] Shekarpour, S., Marx, E., Ngomo, A.-C.N., Auer, S.: Sina: Semantic interpretation of user queries for question answering on interlinked data. Journal of Web Semantics **30**, 39–51 (2015)

[115] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

[116] Broekstra, J., Kampman, A., Van Harmelen, F.: Sesame: A generic architecture for storing and querying rdf and rdf schema. In: International Semantic Web Conference, pp. 54–68 (2002). Springer

[117] Gerber, D., Ngomo, A.-C.N.: Bootstrapping the linked data web. In: 1st Workshop on Web Scale Knowledge Extraction@ ISWC, vol. 2011, pp. 1–16 (2011)

[118] McCandless, M., Hatcher, E., Gospodnetić, O., Gospodnetić, O.: Lucene in action. Manning Greenwich **2** (2010)

[119] Surdeanu, M., Ciaramita, M., Zaragoza, H.: Learning to rank answers to non-factoid questions from web collections. Computational linguistics **37**(2), 351–383 (2011)

[120] Tafreshi, A.S., Tafreshi, A.S., Ralescu, A.L.: Ranking based on collaborative feature weighting applied to the recommendation of research papers. en), International Journal of Artificial Intelligence & Applications **9**, 53 (2018)

[121] Hartig, O., Freytag, J.-C.: Foundations of traversal based query execution over linked data. In: Proceedings of the 23rd ACM Conference on Hypertext and Social Media, pp. 43–52 (2012)

[122] Liu, C., Wang, J., Han, Y., *et al.*: Discovery of service hyperlinks with user feedbacks for situational data mashup. International Journal of Database Theory and Application **8**(4), 71–80 (2015)

[123] Li, G.: Human-in-the-loop data integration. Proceedings of the VLDB Endowment **10**(12), 2006–2017 (2017)

[124] Jeffery, S.R., Franklin, M.J., Halevy, A.Y.: Pay-as-you-go user feedback for dataspace systems. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, pp. 847–860 (2008)

[125] Daniel, G., Cabot, J., Deruelle, L., Derras, M.: Xatkit: a multimodal low-code chatbot development framework. IEEE Access **8**, 15332–15346 (2020)

[126] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., *et al.*: Training language models to follow instructions with human feedback. Advances in Neural Information Processing Systems **35**, 27730–27744 (2022)

[127] Laine, R., Meinke, A., Evans, O.: Towards a situational awareness benchmark for llms. In: Socially Responsible Language Modelling Research (2023)

[128] Chen, X., Han, Y., Wen, Y., Zhang, F., Liu, W.: A keyword-driven data service mashup plan generation approach for ad-hoc data query. In: 2017 IEEE International Conference on Services Computing (SCC), pp. 394–401 (2017). IEEE

[129] Jarke, M., Quix, C.: Federated data integration in data spaces. Designing Data Spaces, 181 (2022)

[130] Azuan, N.A.A.: Exploring manual correction as a source of user feedback in pay-as-you-go integration. PhD thesis, The University of Manchester (United Kingdom) (2021)

[131] Do, T., Nguyen, B.X., Tjiputra, E., Tran, M., Tran, Q.D., Nguyen, A.: Multiple meta-model quantifying for medical visual question answering. In: Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part V 24, pp. 64–74 (2021). Springer

[132] Wu, Y., Ma, Y., Wan, S.: Multi-scale relation reasoning for multi-modal visual question answering. Signal Processing: Image Communication **96**, 116319 (2021)

[133] Zheng, W., Yin, L., Chen, X., Ma, Z., Liu, S., Yang, B.: Knowledge base graph embedding module design for visual question answering model. Pattern recognition **120**, 108153 (2021)

[134] Wang, T., Li, J., Kong, Z., Liu, X., Snoussi, H., Lv, H.: Digital twin improved via visual question answering for vision-language interactive mode in human–machine collaboration. Journal of Manufacturing Systems **58**, 261–269 (2021)

[135] Bosselut, A., Le Bras, R., Choi, Y.: Dynamic neuro-symbolic knowledge graph construction for zero-shot commonsense question answering. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 4923–4931 (2021)

[136] Chen, X., Zhang, N., Li, L., Deng, S., Tan, C., Xu, C., Huang, F., Si, L., Chen, H.: Hybrid transformer with multi-level fusion for multimodal knowledge graph completion. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 904–915 (2022)

[137] Liu, L., Du, B., Xu, J., Xia, Y., Tong, H.: Joint knowledge graph completion and question answering. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 1098–1108 (2022)

[138] Rogers, A., Gardner, M., Augenstein, I.: Qa dataset explosion: A taxonomy of nlp resources for question answering and reading comprehension. ACM Computing Surveys **55**(10), 1–45 (2023)

[139] Esteva, A., Kale, A., Paulus, R., Hashimoto, K., Yin, W., Radev, D., Socher, R.: Covid-19 information retrieval with deep-learning based semantic search, question answering, and abstractive summarization. NPJ digital medicine **4**(1), 68 (2021)

[140] Abbasiantaeb, Z., Momtazi, S.: Text-based question answering from information retrieval and deep neural network perspectives: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **11**(6), 1412 (2021)

[141] Sharma, H., Jalal, A.S.: Visual question answering model based on graph neural network and contextual attention. Image and Vision Computing **110**, 104165 (2021)

[142] Limna, P., Kraiwanit, T., Jangjarat, K., Klayklung, P., Chocksathaporn, P.: The use of chatgpt in the digital era: Perspectives on chatbot implementation. Journal of Applied Learning and Teaching **6**(1) (2023)

[143] Saito, K., Sohn, K., Lee, C.-Y., Ushiku, Y.: Unsupervised llm adaptation for question answering. arXiv preprint arXiv:2402.12170 (2024)

[144] Shen, Y., Song, K., Tan, X., Li, D., Lu, W., Zhuang, Y.: Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. Advances in Neural Information Processing Systems **36** (2024)

[145] Bratić, D., Šapina, M., Jurečić, D., Žiljak Gršić, J.: Centralized database access: Transformer framework and llm/chatbot integration-based hybrid model. Applied System Innovation **7**(1), 17 (2024)

[146] Guo, J., Li, J., Li, D., Tiong, A.M.H., Li, B., Tao, D., Hoi, S.: From images to textual prompts: Zero-shot visual question answering with frozen large language models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10867–10877 (2023)

[147] Totlis, T., Natsis, K., Filos, D., Ediaroglou, V., Mantzou, N., Duparc, F., Piagkou, M.: The potential role of chatgpt and artificial intelligence in anatomy education: a conversation with chatgpt. Surgical and Radiologic Anatomy **45**(10), 1321–1329 (2023)

[148] Huang, Y., Huang, J.X.: Exploring chatgpt for next-generation information retrieval: Opportunities and challenges. In: Web Intelligence, pp. 1–14 (2024). IOS Press

[149] Ferdush, J., Begum, M., Hossain, S.T.: Chatgpt and clinical decision support: Scope, application, and limitations. Annals of Biomedical Engineering, 1–6 (2023)

[150] Liu, S., Wright, A.P., Patterson, B.L., Wanderer, J.P., Turer, R.W., Nelson, S.D., McCoy, A.B., Sittig, D.F., Wright, A.: Using ai-generated suggestions from chatgpt to optimize clinical decision support. Journal of the American Medical Informatics Association **30**(7), 1237–1245 (2023)

[151] Hadi, M.U., Qureshi, R., Shah, A., Irfan, M., Zafar, A., Shaikh, M.B., Akhtar, N., Wu, J., Mirjalili, S., et al.: A survey on large language models: Applications, challenges, limitations, and practical usage. Authorea Preprints (2023)

[152] Hu, X., Chen, J., Li, X., Guo, Y., Wen, L., Yu, P.S., Guo, Z.: Do large language models know about facts? arXiv preprint arXiv:2310.05177 (2023)

[153] Wu, X., Duan, R., Ni, J.: Unveiling security, privacy, and ethical concerns of chatgpt. Journal of Information and Intelligence (2023)