Journal of Intelligent Information Systems
https://doi.org/10.1007/s10844-019-00584-7

A survey on question answering systems over linked data and documents

Eleftherios Dimitrakis[1,2] · Konstantinos Sgontzos[1,2] · Yannis Tzitzikas[1,2]

Received: 25 March 2019 / Revised: 29 September 2019 / Accepted: 4 October 2019 /
[redacted]

Abstract
Question Answering (QA) systems aim at supplying precise answers to questions, posed by users in a natural language form. They are used in a wide range of application areas, from bio-medicine to tourism. Their underlying knowledge source can be structured data (e.g. RDF graphs and SQL databases), unstructured data in the form of plain text (e.g. textual excerpts from Wikipedia), or combinations of the above. In this paper we survey the recent work that has been done in the area of stateless QA systems with emphasis on methods that have been applied in RDF and Linked Data, documents, and mixtures of these. We identify the main challenges, we categorize the existing approaches according to various aspects, we review 21 recent systems, and 23 evaluation and training datasets that are most commonly used in the literature categorized according to the type of the domain, the underlying knowledge source, the provided tasks, and the associated evaluation metrics.

Keywords Question answering · dialogue systems · RDF and linked data · evaluation collections

1 Introduction

Question Answering (QA) Systems can be considered as an extension of Search Engines in the sense, that they aim at automatically supplying users with precise answers to questions posed in natural language, instead of simply returning a ranked list of relevant sources based on a set of keywords. The history of QA systems starts from 1961 with the system Baseball

✉ Yannis Tzitzikas
tzitzik@ics.forth.gr

Eleftherios Dimitrakis
dimitrakis@ics.forth.gr

Konstantinos Sgontzos
sgontzos@ics.forth.gr

[1] FORTH, Institute of Computer Science, Heraklion, Greece

[2] Department of Computer Science, University of Crete, Heraklion, Greece

(Green et al. 1961), it includes the famous system LUNAR (Woods 1973) published in 1973, while a relatively recent (2010) and noteworthy system is IBM Watson (Ferrucci et al. 2010). Other commercial products in the area of personal assistants include Apple's Siri (in 2011), Amazon's Alexa (in 2014) and Microsoft's Cortana (in 2014). Nowadays QA systems are used in a wide range of application areas, such as in Bio-medicine, where doctors and nurses ask the QA system about medical care or diagnosis (e.g. IBM Watson (Ferrucci et al. 2010)), as well as in Tourism, where users ask questions about geographical entities and related information e.g. cities, locations etc. (e.g. LILOG (Herzog and Rollinger 1991), LD-SDS Papangelis et al. 2017).

The architecture of QA Systems highly depends on the type of the underlying knowledge sources. On the one hand, if the goal is to extract the answer from plain text, a common approach is to use traditional *Information Retrieval* techniques combined with *Machine Comprehension* (MC) methods, for the tasks of *Passage Retrieval* and *Answer Processing* respectively. On the other hand, if data is represented as a graph, e.g. using the Resource Description Framework[1] (RDF), methods relying on *Graph Processing* and *SPARQL Query Generation* are adopted in order to extract the desired information. Finally, if none of the above is effective, or when both kinds of sources are available (unstructured and structured), then mixed, or *hybrid*, methods are adopted, for creating the final answer.

Although there are various surveys of the area (described in Section 2), the current survey aims to complete the picture. Apart from covering more recent systems, up to 2018, this survey contains a detailed list of available *training/evaluation* datasets for QA, an aspect that is not covered by the past surveys. We believe that the list of evaluation datasets provided in this survey can be a valuable guide to researchers for choosing the datasets that best suit their needs and hypotheses. We have to note that this survey does not focus on datasets built for dialogue systems, since this part has already been covered in Serban et al. (2015), which presents a list of the available datasets for data-driven dialogue systems. Another distinctive characteristic of the current survey is that it discusses how different types of QAs and information sources can be *combined* to a unified pipeline, in order to help researchers to find combinatorial ways that can be more effective.

The rest of the paper is organized as follows: Section 2 synopsizes the current uses of QA systems, and discusses the current main *challenges* of QA. Section 3 introduces the main *aspects* for structuring the landscape of QA. Section 4 analyzes the *processes* of QA systems based on the type of the underlying knowledge source. Section 5 focuses on the *categorization* of the existing systems based on their type and the aspects presented in the previous section, and reports some quantitative observations. Section 6 discusses *evaluation metrics* and provides a comprehensive list of *evaluation datasets*. Section 7 discusses trends and sketches directions for further research. Finally, Section 8 summarizes and concludes.

2 Context, Past Surveys and Challenges

Application Areas After the extensive research of the past years on QA systems, they have achieved to be applicable in many areas such as Medical (Devi and Dua 2017; Abacha and Zweigenbaum 2015), Culture (Liu et al. 2017), Web Communities (Bonadiman et al. 2017), Tourism (Pathak and Mishra 2016), and others. An indicative example of a successful QA system, is IBM Watson (Ferrucci et al. 2010) which is not restricted to a certain

[1] https://www.w3.org/TR/rdf-concepts/

domain but instead aims at capturing a wide range of questions. At first, it was designed to answer natural logic questions as in the TV game show Jeopardy; the question-answering challenge represented by Jeopardy and the details of the technical approach of IBM Watson are described in the special issue (Ferrucci 2012). Watson competed in Jeopardy against former winners of the show, and won (Ferrucci et al. 2013). However even Watson which is considered to be one of the most successful QA systems, still suffers from weaknesses and ambiguities (evidenced by the wrong answers it provided). The last years we observe a wide adoption of QA-based *personal assistants* including Apple's Siri, Amazon's Alexa, Microsoft's Cortana, Samsung's Bixby, and Google Assistant, that are capable of answering a wide range of questions. Furthermore, applications of QA in *call centres* and *IoT applications* are rapidly increasing (Wang et al. 2019; Norouzi et al. 2019).

Past Surveys Here we describe past surveys (listed in chronological order) and the focus of each one of them.

Wang (2006) presented a comparison of the techniques used (at that time) for the task of *information extraction on factoid questions*. The comparison aspects included: (1) the information sources used, (2) the representation and the information extraction process and (3) the way of combining multiple data sources.

Allam and Haggag (2012) in 2012 surveyed the *architectures of QA* systems, and compared various approaches, their main contributions, experimental results, and limitations. Published in the same year, Gupta and Gupta (2012) surveyed the *QA system types* and presented a general pipeline for each one of them (that included steps like question classification and information retrieval) and made some suggestions regarding the architecture of QA systems.

Sasikumar and Sindhu (2014) in 2014 surveyed the methods applied in the *Natural Language Question Answering (NLQA)* field for general languages. It categorized QA systems to (1) linguistic, (2) statistical and (3) pattern matching, and presented, different types of QAs, such as Web-based, IR/IE based and Rule-based.

Höffner et al. (2016) in 2015 surveyed the challenges of the QA area based on *semantic data*. It described and analyzed 62 systems, published between 2010 and 2015, and discussed the common challenges for this particular research area and provided some recommendations for future systems.

Mishra and Jain (2016) in 2016 focused on the *classification of QA systems* based on a set of aspects of the underlying pipeline: (1) application domain (i.e. open or closed), (2) types of questions (e.g. factoid, list, etc.), (3) types of data sources (i.e. structured, semi-structured, unstructured).

The survey (Rodrigo 2017) in 2017 focused on *evaluation* of QAs over free text and it analyzed a set of QA collections in order to obtain better insights as regards the current challenges and future directions. Based on this analysis, it recommended some guidelines regarding evaluation. Published in the same year, Patra (2017) surveyed the area of *Community QA* with focus on: (1) question semantic matching, (2) answer ranking, and (3) answer retrieval, which are common tasks of a general QA pipeline.

Challenges Based on the analysis of the past surveys, and the current state-of-the art, we have identified the following main difficulties/challenges for effective QA.

Ambiguity: The problem of word sense ambiguity can be spotted both to structured and unstructured data. As regards the unstructured data, a word in a sentence may have a specific meaning, while the same word in another sentence may have a different sense (polysemy).

The task of resolving such issues is called Word Sense Disambiguation (WSD). Moreover there are entities that have multiple names (e.g. Apple, Apple Co., Macintosh). A user question can describe an entity, using one of its labels (e.g. Apple), while the underlying source(s) may use a different one (e.g. Macintosh), resulting to mismatches. Indexes that collect all equivalent names (like Mountantonakis and Tzitzikas 2018) from hundreds of KBs can be useful for this task. Apart from synonyms, we have the problem of homonyms, i.e. there are different entities that share the same name (e.g. Washington may refer either to the capital state or to the first President of the U.S.A.). In general, the task of resolving such problems is called Named Entity Disambiguation (NED) and ambiguity is still spotted as a challenge of the QA field, including linked data (Höffner et al. 2016), and various directions for better tackling this problem are under investigation including methods that attempt to disambiguate entities and relation jointly (Dubey et al. 2018), as well as the creation of resources for better managing slang words and neologisms (Dhuliawala et al. 2016).

Distributed Knowledge: It is unrealistic to assume that the available knowledge is stored in a single collection or dataset. Moreover, each different organization or user, represents and stores their data in different places and with different ways (different conceptualizations, models and formats). Therefore, approaches that can exploit the available data across different datasets, are of cornerstone importance for QA systems. There are two main challenges related to this topic. First, the different representations of the data used makes the task of aggregation, interlinking and integration very difficult (e.g. see Mountantonakis and Tzitzikas (2019) for a recent survey). Second, the selection of the appropriate dataset to exploit, between the wide range of the available sources, is not a trivial task, and *dataset search* is an open problem (Chapman et al. 2019).

Complex Questions: User questions are not always simple (i.e. require the extraction of a single fact to be answered), such as “When did Einstein died?”. There are many cases where the question, requires complex approaches for finding the associated answer. For example, consider the question “What is the prevailing opinion of the reviewers about the quality of the food, at restaurant X in Paris?”. An answer here is not trivial, since different reviewers may express different opinions. Furthermore, it is not trivial how to capture the opinion of a person for a specific subject, since it is not easy to figure out when a person talks with a “good” or a “bad” manner about that specific subject. Current state-of-the-art QAs are unable to solve such complex problems in the general case. Complex Questions were recognized as challenge of the QA field over linked data in Höffner et al. (2016) and over free text in Rodrigo (2017).

Multilinguality: Users wish to express their questions in their native language however this task is very challenging due to the large number of existing languages as well as the differences in their structure and idioms. Different languages may require to be studied separately and apply different techniques based on their characteristics. Moreover, the datasets that can be used for training depend on the natural language (there are plenty of datasets for some natural languages, and very few for others). Multilinguality is recognized as a challenge of the QA field also for the case where the underlying knowledge has the form of linked data in Höffner et al. (2016).

3 Aspects of the landscape and types of systems

In this section we introduce a number of aspects for structuring the landscape of QA systems. Specifically, in Section 3.1 we identify cases based on the *domain* of the systems, in Section 3.2 we elaborate on *question types*, while in Section 3.3 we identify *types of knowledge sources*. Finally, in Section 3.4, we distinguish three *categories of QA-oriented systems*: Traditional QA Systems (QA), Dialogue Systems (DS), and Spoken Dialogue Systems (SDS). Figure 1 illustrates the above aspects, providing a kind of conceptual map.

3.1 Domain of knowledge

Open vs Closed Domain: *Closed-domain* QA applications are systems which focus on answering questions under a specific domain such as Medical and Culture (Devi and Dua 2017; Liu et al. 2017). Instead the term *Open-domain* refers to systems whose purpose is to answer questions about "anything" (Ferrucci et al. 2010; Watanabe et al. 2017; Dai et al. 2016). The techniques used for the implementation of these two types of systems differ in many aspects (Mollá and Vicedo 2007): At first, the size of the available data should be considered, that is, large amounts of underlying knowledge for *Open-domain* QAs while smaller in size knowledge sources for *Closed-domain* QAs. The size is critical, since the relative smaller size of close domain (in comparison to the open domain) makes it feasible to apply complex techniques e.g. deep NLP to the available resources, while in the *Open-domain* case we are unable to apply such techniques due to the large volumes of data. Second, the context of our available knowledge, plays an important role due to the possible senses of specific words or phrases. In a *Closed-domain* application, the WSD is much easier, since we can restrict to the possible senses of a specific word in a small (domain-specific) subset instead of considering all of them. Finally, attention should be paid to the kind of resources

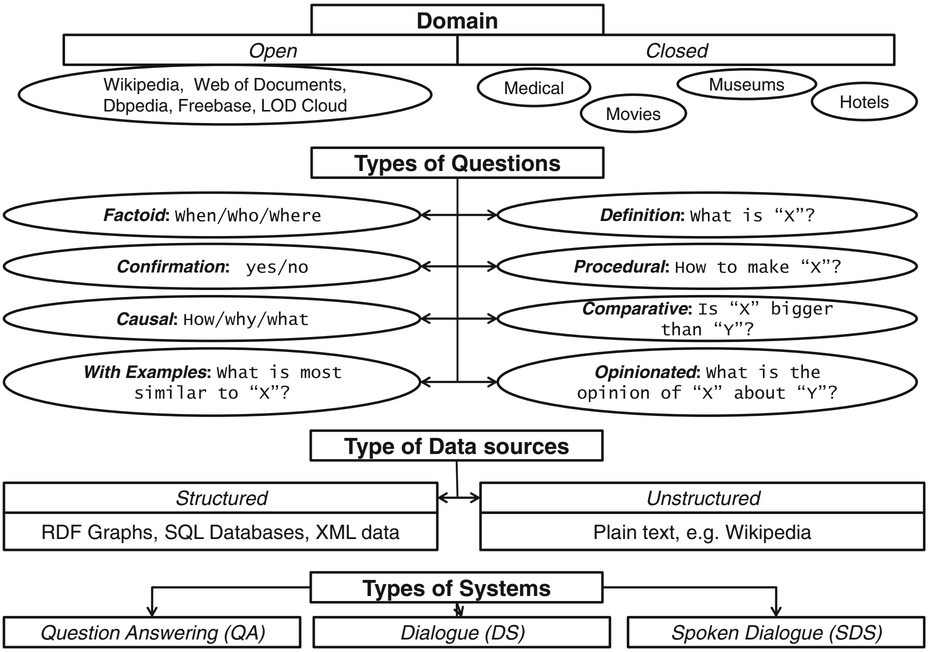
Domain
Open
Closed
Wikipedia, Web of Documents, Dbpedia, Freebase, LOD Cloud
Medical
Movies
Museums
Hotels
Types of Questions
Factoid: When/Who/Where
Definition: What is "X"?
Confirmation: yes/no
Procedural: How to make "X"?
Causal: How/why/what
Comparative: Is "X" bigger than "Y"?
With Examples: What is most similar to "X"?
Opinionated: What is the opinion of "X" about "Y"?
Type of Data sources
Structured
RDF Graphs, SQL Databases, XML data
Unstructured
Plain text, e.g. Wikipedia
Types of Systems
Question Answering (QA)
Dialogue (DS)
Spoken Dialogue (SDS)

Fig. 1 **Aspects of the Landscape:** (1) Domain of Knowledge (2) Type of Question (3) Type of Knowledge Source (4) Type of System

that could be used in each case. In the *Closed-domain* case, it is wise to rely on domain-specific resources, that capture the domain focused vocabulary as well as the terminology of the words it consists of, whereas *Open-domain* applications usually exploit general lexical resources and domain independent ontologies, such as WordNet (Miller 1995) and DBPedia (respectively), which try to capture all possible word senses and their associations with other words. In the case of *Closed-domain*, it is usually harmful to use general ontological resources, since, as it is stated by Mollá and Vicedo (2007), they are, in many cases, too coarse-grained for specific domains, whereas there are other cases where they are too fine-grained.

3.2 Question type

Questions can be categorized to various categories according to their complexity, response type, the techniques that should be used for answering them, and so on. Among the various possible categorizations, below we select the categorization mentioned in Mishra and Jain (2016):

1. **Factoid (when/who/where):** questions that essentially require a single fact or a small piece of text to be returned as answer.
   - e.g. "When did the WWI begin?"
2. **Confirmation (yes/no):** questions that require a yes/no answer.
   - e.g. "Is Athens the capital of Greece?"
3. **Definition:** questions that require answers that are definitions of terms.
   - e.g. "What is the Resource Description Framework?"
4. **Causal (how/why/what):** questions that require that the answer should be one or more consequences of a fact.
   - e.g. "What are the consequences of Iraq War?"
5. **Procedural:** questions that require a set of actions needed for accomplishing something.
   - e.g. "Which are the steps to get a master degree?"
6. **Comparative:** questions that require as an answer, a set of differences between two or more subjects.
   - e.g. "Which are the differences between SSD and HDD?"
7. **With Examples:** questions that target to find examples that best describe the reference point of the question.
   - e.g. "Which are the most similar disks to hard disk X?"
8. **Opinionated:** questions that require to find the opinion of someone about a subject or a fact.
   - e.g. "Which is the opinion of the Americans about the Iraq war?"

3.3 Knowledge source type

QAs can be classified based on the type of data source that they exploit in order to extract information. In this survey we focus on three distinct types: *(1) Documents*, *(2) Data* and *(3) Hybrid* (data and documents). The approaches used in each case for the analysis of the information, differ a lot. For example in the *Data* case, it is not always required to apply complex NLP techniques, since the explicit structure of the data is more convenient for capturing the semantic specifications of the information stored (e.g. Albert Einstein is a Person). In contrast, the *Documents* case, requires NLP and NLU (Natural Language Understanding) techniques e.g. POS (Part-of-Speech) tagging, NER (Named Entity Recognition), co-reference resolution etc., in order to extract such kind of information. As regards the *Hybrid* knowledge representation, it requires employing techniques from both structured and unstructured data, as well as methods for combining them in order to produce the final answer. Note that in Fig. 1, XML and RDF are considered as formats for representing structured data (not structured documents).

3.4 Types of systems

Question Answering Systems QA is a multidisciplinary topic that adopts methods and techniques coming from Information Retrieval (IR), Natural Language Processing (NLP), Information Extraction (IE) and more recently, other fields such as Artificial Intelligence (AI), Machine Learning (ML) and Linked Data (LD). Such systems are unable to interact with the user in a multi-step procedure. However, there is an extension of QA called Interactive Question Answering (IQA) which tackles this limitation. It is considered to be an intersection of QA and DS (Dialogue Systems), which enables seeking for the answer of a specified question instead of simply receiving the answer. As stated in Konstantinova and Orasan (2012), it employs dialogue for clarification of missing or ambiguous information.

Dialogue Systems A Dialogue System (DS) is a computer system designed to converse with a human. We can distinguish such systems to chatbots which are used mainly for fun (starting from the 1966 system ELIZA 1966) and dialogue agents which are goal/task oriented. In general, the input and output information is analyzed by the *dialogue manager*, that keeps the history and state of the dialogue as well as, manages the general flow of the conversation. Current state-of-the-art DSs, are mostly developed to answer questions in order to resolve a specific task, and these DSs are called "task-oriented" (Gustafson et al. 2000). They are restricted to a specific domain and the associated information, and due to this restriction, they use a specific and limited grammar, that is able to describe the context of the underlying data, including paraphrases, possible abbreviations etc. in order to resolve Natural Language Expression ambiguities. If we apply such architecture, to a different domain, a necessary step would be the adaptation of the new grammar, suitable for the target domain, from scratch. To tackle this limitation, current research focuses on automatic methods for learning an appropriate grammar for a given KB/domain (McCrae et al. 2011). However, resolving ambiguities of human utterances (disambiguation) automatically, is a very challenging task, since the phrases used for expressing their thoughts and preferences are reality-bound i.e. what they actually mean, strictly depends on the context of the discussion. Moreover, it has been observed from past works (Popescu et al. 2003) that in order to achieve high precision, limitations on language coverage is required and thus system "habitability" (how system supports the language that users use to interact with) (Ogden et al. 2008) reduces. Hence, to improve the performance of the disambiguation stage, when trying to learn a new grammar, without harming the "habitability", many researchers turned into *semi-automatic* methods (Moreo et al. 2017). Such kind of methods, require human effort, in order to make the mappings of the utterance phrases and words to the underlying knowledge. However, this human computer interaction should not assume familiarity of the users with the KB structure and NLP. To succeed that, the system should automatically resolve disambiguation and ask for human clarification only when it is unsure. In this way, we succeed to maintain both "habitability" and high precision, and additionally to unburden the user.

Spoken Dialogue Systems A Spoken Dialogue System (SDS) is a computer system that takes as input, and provides as output, voice, in order to converse with a human. It has two essential components that do not exist in a simple text DS: a speech recognizer and a text-to-speech module. It can be further distinguished from command and control speech systems that can respond to requests but do not attempt to maintain continuity over time. In such systems, the user interacts with the system through speech, making the interaction procedure free of constraints and limitations as regards the expressiveness. The last years, such systems find increasing applications in a wide range of application domains e.g. in personal assistants (Schmidt and Braunger 2018) or in the context of faceted search as in Dimitrakis et al. (2018).

4 Processes for question answering

In this section we present the key processes for analyzing a question (in Section 4.1), a task that is needed independently of the underlying knowledge source, and then we refine according to the type of the underlying knowledge source, specifically we describe QA processes over a set of documents (in Section 4.2), a structured KB (in Section 4.3) or both (Section 4.4).

4.1 Processes for question analysis

Question Analysis is major step in a QA pipeline. It aims at analyzing the input utterance, in order to improve the informative quality of the question and to reduce the search space of the corresponding answer. In this task, challenges arise due to the small size of the user's utterance, missing information, as well as ambiguous information. There is a wide range of analysis types that could be applied, as shown in Fig. 2 (also aligned with Mishra and Jain 2016), each described in brief below.

- **Morphological analysis** aims at separating words into individual units called *morphemes*, which are the smallest grammatical units in a language, e.g. the word "Untouchable" comprises three morphemes: "un-", which is a bound morpheme signifying "not", "-touch-", which is the root and it can stand alone, and "-able", a free morpheme signifying that something "can be done", in our example it refers to "-touch-". This task is achieved by lemmatizing and stemming the input words. Morphological analysis can increase the recall of the system, however, it can also harm the precision, due to the stemming, which may change the semantic meaning of a word (e.g. starring → star). Such problems arise, when we deal with polysemus words like "star" which, based on WordNet, has eight senses as a noun, three as a verb and one as an adjective.

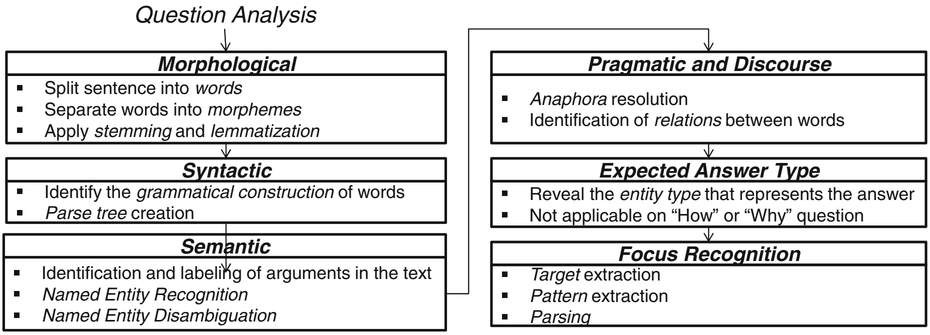
Question Analysis
Morphological
Split sentence into words
Separate words into morphemes
Apply stemming and lemmatization
Syntactic
Identify the grammatical construction of words
Parse tree creation
Semantic
Identification and labeling of arguments in the text
Named Entity Recognition
Named Entity Disambiguation
Pragmatic and Discourse
Anaphora resolution
Identification of relations between words
Expected Answer Type
Reveal the entity type that represents the answer
Not applicable on "How" or "Why" question
Focus Recognition
Target extraction
Pattern extraction
Parsing

Fig. 2 Types of *Question Analysis*

- **Syntactical analysis** is the task of recognizing a sentence and assigning a syntactic structure to it. It consists of the creation of parse trees and consequently the identification of the grammatical construction of words (e.g. noun, verb, adjective or adverb). It is commonly used to define syntactic-based similarity measures and transform the input query into a declarative sentence. As a consequence, the candidate statement (sentence or triple) selection step becomes more accurate.
- **Semantic analysis** aims at identifying and labeling the arguments in the text. The arguments for a given verb consists of all the constituents in the parse tree that do not contain that predicate; commonly the arguments of a predicate are contained within the minimal clause that contains the predicate (e.g. see Abend et al. 2009). Semantic analysis is achieved by analyzing the parse tree or by applying statistical and Machine Learning classification techniques. It can be very useful, especially in hybrid systems, in order to align the textual part of the information with the ontology of the structured part. As mentioned in Mishra and Jain (2016), Semantic analysis is currently investigated only in lexical or sentence level, but not on document level.
- **Pragmatic and Discourse analysis** is an NLU process which targets on revealing the relations between sentences. It includes *Anaphora Resolution* which aims at replacing words like pronouns (which have no semantic context) with proper nouns in the text (e.g. in "This t-shirt looks nice, I will buy it." we need to identify that "it" refers to "t-shirt" and then replace it). As concerns the Discourse analysis, the goal is to capture the kind of relation between *two or more sentences*, i.e. elaboration, explanation or contrast. This type of analysis helps at answering complex questions like opinionated, causal, hypothetical or confirmation.
- **Expected Answer Type Analysis** aims at identifying the *answer type*, which reveals the *entity type* we are searching (e.g. Person, Thing). This analysis, strictly depends on the question types that a system aims to answer, since some types, e.g. *why* or *how*, do not have a specific answer type. As stated in Li and Roth (2006), in order to identify the question answer type, two kinds of features should be extracted: syntactic and semantic. The first refers to POS tags, chunks (non-overlapping phrases in a sentence Abney 1991) and head chunks (e.g. the first noun chunk and the first verb chunk after the question word in a sentence). As regards the semantic features, they usually include: (i) Named Entities, (ii) Word Senses (e.g. from WordNet Miller 1995), (iii) Class-Specific Related Words (i.e. a set of words that frequently occur in the specific answer type)

and (iv) Distributional Similarity Based Categories (i.e. distributions that capture the likelihood of words occurring in identical syntactic structures in sentences).

- **Focus Recognition** aims at identifying the *question topic* i.e. the user's interests, as well as the question focus which represents the information need of the user, i.e. the certain aspects or descriptive features of the question topic. It can be applied both on question analysis and text analysis if the underlying knowledge source is a set of documents, but only on question analysis step, if the knowledge is represented as a structured graph. It consists of the identification of (a) the *head of the focus*, (b) a *list of modifiers* (Ferret et al. 2001). The head of the focus, is a noun, while the list of modifiers is determined by the noun phrase in which the head is enclosed. For example, in the sentence "Who was the first man on the moon?", the focus is "first man on the moon", having as head the noun "man" followed by the modifiers "first", "moon".

4.2 Processes for QA over documents

The general pipeline of a QA System over Documents is illustrated in Fig. 3(left). It can include several tasks, here we organize them into three modules: 1) *Question Analysis*, 2) *Passage Retrieval* and 3) *Answer Processing*. Each of these is briefly discussed below.

1. Question Processing/Analysis module: Common tasks here include:

- *Question Type Classification* aims at identifying/classifying the question type (e.g. factoid, list, etc.) for applying the most suitable techniques and exploiting the expected

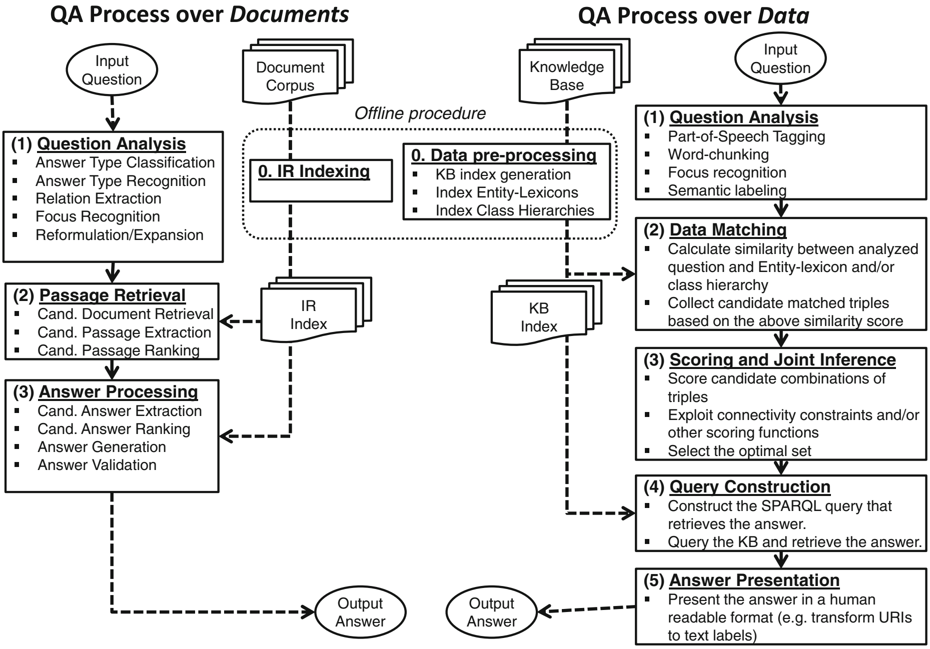

Fig. 3 Left: QA pipeline over *Unstructured Data*. Right: QA pipeline over *Structured Data*

form of the answer. Approaches here include: hand-written rules based on regular expressions, for identifying a usually pre-defined hierarchy of question types, as well as ML classifiers, or combination of the above.

- *Question Focus Recognition* aims at recognizing the question topic as well as the question focus. For example, Watson (Lally et al. 2012) uses Prolog with a set of rules to achieve this task.
- *Expected Answer Type (EAT)* aims at identifying the expected Named Entity type of the answer (e.g. Person, Location) appropriate for generating answers mainly for factoid and list type questions. Several approaches have been proposed including: 1) hand-written rules, e.g. regular expression-based, as well as, approaches with rules using question headword, 2) ML classifiers, and 3) hybrid, i.e. combination of the above.
- *Relation extraction* aims at detecting relations between Named Entities in the input question. Such relations are extracted using NLP tools, such as Stanford CoreNLP Toolkit (Manning et al. 2014) and DBpediaSpotlight (Mendes et al. 2011).
- *Question Reformulation* and/or *Question Expansion* focuses on enhancing question phrasing and on transforming it into a semantically equivalent one, consisting of additional or more descriptive information in order to improve the accuracy and recall of the overall system. Here resources, such as lexicons (e.g. WordNet) are exploited in order to retrieve information, like sets of synonyms for particular words, as well as POS Taggers, NE Taggers etc. Other approaches use hand-written reformulation rules like those presented in Lin (2007).

2. Passage Retrieval module: After the analysis of the input question and the extraction of the clues, as regards the user interests, search in the underlying knowledge source is performed for finding relevant information. Common tasks here include:

- *Document Retrieval* aims at searching in the available information space for returning a ranked small subset of the most relevant documents. Typically, Information Retrieval techniques and systems are used for this task, e.g. see Chu-Carroll et al. (2012) for the related tasks in IBM Watson. The output of the previous module is exploited in order to increase the efficiency and the accuracy, e.g. EAT for restricting the search space and search for certain type of entities.
- *Passage Extraction* aims at reducing both the size of the text to analyze as well as the amount of noise it contains, since the output of the previous step could be large in size. In this task the relevant documents are segmented into shorter units, in the size of paragraphs, for maintaining only a reasonable amount of information to process and therefore improve the efficiency of the system. This is analogous to the web search engines that for each hit they return a snippet, i.e. a small excerpt of the document, that has all (or most) of the words in the query. Returning to QA, this task is also related to the features that are used for Passage Ranking (described next). A recent paper that evaluates passage retrieval approaches is Yang et al. (2016).
- *Passage Ranking* aims at ranking the candidate passages, based on a set of features, which reflect the plausibility degree of containing the correct answer. The features include the rank of the document that contains the passage, the number of question words in passage and their proximity, the number of named entities having the right type (Answer type) in passage, and others. It outputs a ranked list of candidate passages to be processed by the final module. Approaches for this module, usually contain hand-written rules or ML classifiers which are trained to produce a ranking.

3. Answer Processing module: After the extraction of the relevant information with respect to the user's question, the final step is the extraction and validation of the final answer. Common tasks in this module include:

- *Candidate Answers Extraction* aims at identifying the answer candidates from the ranked list of passages. An approach for this module is to parse the passages with part-of-speech and Named Entity taggers and return the pieces of text which match the EAT (Expected Answer Type).
- *Candidate Answers Ranking* aims at ranking the candidate answers based on a set of features which reflect the plausibility degree of being the correct answer. It outputs a ranked list of candidate answers created by hand-written rules or ML classifiers trained to produce a ranking.
- *Answer Generation* aims at generating the final answer from the set of candidates. This task is useful, when we face the problem of partial answers which range between paragraphs, documents or even different sources.
- *Answer Validation* aims at calculating a confidence score that reflects how sure we are for the correctness of the generated answer.

4.3 Processes for QA over data

Over time, Knowledge Base Question Answering (KBQA) systems have converged to two major approaches (Yao et al. 2014a): (I) *Semantic Parsing*, and (II) *Information Extraction*. The Semantic Parsing approach focuses on question understanding and therefore attempts to convert sentences into their semantic representation, such as *logical forms* (Berant et al. 2013, 2014, 2015; Yih et al. 2015). On the other hand, the Information Extraction (IE) approach (Bast and Haussmann 2015; Yao and Van Durme 2014b) aims at identifying topic/focus entities in the input question and then, via pre-defined templates, map the question to the KB predicates, and finally, explore the KG (Knowledge Graph) neighborhood of the matched entities. For instance, according to the first approach, for a question of the form *"Did Aristotle influence Ioannes Georgius Gadamer?"* the objective is to translate it to a SPARQL query of DBpedia, while according to the second approach, the idea is to fetch all information about the related entities and their associations, and then to try to answer the question using the derived graph, without necessarily having to translate the question to a SPARQL query. As also stated in Savenkov and Agichtein (2016), such systems, rely on a lexicon, learned from labeled training data, or even supported by additional resources, such as question paraphrases (Berant and Liang 2014) and weakly labelled sentences from a large text collection (Yao and Van Durme 2014b). However, the size of such training data tends to be very small compared to the number of distinct predicates and literals in the KB, and consequently the produced lexicons have limited coverage.

In this section we analyze the case, where the underlying knowledge is represented as a Knowledge Graph (KG) or Knowledge Base (KB), based on RDF (e.g. DBpedia (Auer et al. 2007), Freebase (Bollacker et al. 2008) etc). An RDF KG is a graph that describes real world entities and facts as well as the relations between them. Entities are represented as nodes, while the relations between them are represented as edges. Each entity can be classified using one or more classes. Classes can be organized hierarchically, the same is true for relations. Nodes also stand for literals, i.e. values of entity properties (e.g. dates, phones, colors etc.). Moreover, RDF is equipped with reasoning services enabling the derivation of inferred triples based on the semantics of RDF Schema, OWL as well from custom inference rules. Finally, relative to the integration of the available data, there are many tools/techniques that make the task easier, by interlinking different KBs based on ontology mapping methodologies (Song et al. 2017; Anam et al. 2016) or by transforming and properly integrate other data source types (e.g. CSV, SQL tables) into RDF ontologies using semantic labeling techniques (Pham et al. 2016; Ramnandan et al. 2015). As a result, the growth of this type of data representation increases over time, a fact that make KGs able to support QA systems both on *Open* and *Closed-domains*. Here, we present and briefly discuss, general steps in order to build a QA system, based on Linked Data alongside with some possible approaches.

A general pipeline of a QA system over Data is illustrated in Figure 3(Right). The overall process includes a preprocessing step (marked below with the number 0) and five basic steps, all described next.

- **0. Data preprocessing:** It aims at increasing the performance both in terms of execution time and predictive accuracy. It consists of the construction of indexes of the KG for efficient search. Moreover, it can be used for vocabulary expansion by exploiting synonyms, acronyms etc. of the KG resources. Thus, it increases the recall of the *Data Matching* step, since it enables the system to capture different expressions of specific entities. Attention should be paid, to avoid harming the precision due to homonyms (i.e. the same word can refer to different concepts) and polysemy (i.e. same string refers to different but related concepts).
- **1. Question analysis:** It aims at analyzing the input question. The process, usually relies on a linguistic analysis of the question, which involves part-of-speech taggers and parsers to capture the syntactic structure of the question, plus some ways of capturing the semantics of it, e.g. Named Entity Recognition (NER). These can be carried out using techniques which are implemented in various widely used tools like Stanford CoreNLP (Manning et al. 2014) or DBpedia Spotlight (Mendes et al. 2011). This step mainly focuses on revealing: (a) the question type, (b) the EAT and (c) the focus of the question (as discussed in more detail in Section 4.1).
- **2. Data matching:** This step is a challenging task due to the lexical gap and the ambiguity between user's utterance and the underlying KG vocabulary. It targets on matching question words or phrases to their counterparts in the KG. More specifically, the purpose of this step is to identify specific word chunks of the input question that refer to specific entities, classes or properties in the KG. This procedure can be naively implemented by simple string matching (Zhang et al. 2016), or by more sophisticated techniques such as approaches that take advantage of word embeddings (Lukovnikov et al. 2017). It is worth mentioning, that in the computation of such similarities, synonymy, hypernymy and hyponymy should be taken into account, since they adjust conceptual meaning to the words. On the one hand, such information already exists in the KG, in the form of class hierarchies and the expanded vocabulary sets mentioned in the *Data preprocessing* step. On the other hand, the input question lacks of such structured knowledge due to the plain text format. Thus, external (to the query) resources such as WordNet (Miller 1995), PATTY (Nakashole et al. 2012), ReVerb (Fader et al. 2011) and Word2Vec (Mikolov et al. 2013) (as well as more recent techniques for word embedding such as Glove (Pennington et al. 2014), ELMO (Peters et al. 2018) and BERT Devlin et al. 2018) are essential, since they provide/capture such information.
- **3. Scoring and Joint Inference:** After the Data Matching step, we have a set of candidate components. Having this set does not imply having the final answer, since there are many combinations of the components resulting in different answers or even invalid ones. This is due to the lack of connectivity between them in the KG especially in cases that we deal with multiple knowledge sources (for connectivity results

over the datasets published as linked data see Mountantonakis and Tzitzikas 2016). The scoring of the candidates is required to find the optimal solution. Another problem is the conflicting components. For example in the question: *"Who won the European football Championship in 2017?"*, the candidate resources, for the surface form *Championship* that emerged from the previous step, may be both $ChampionsLeague2017$ and $EuropaLeague2017$, since European football championship may refer to both *"Champions League"* and *"Europa League"*. In such cases, the scoring step should be able to select only one between the candidates, in order to construct a single SPARQL[2] query in the next steps, from the following:

- `select ?team where  ?team prefix:winner prefix:ChampionsLeague2017.`
- `select ?team where  ?team prefix:winner prefix:EuropaLeague2017.`

To disambiguate that, approaches involve string matching, combined with Linear Programming as in Zhang et al. (2016) or matching based on the semantic similarity (e.g. word embedding scores like WMD on Word2Vec's word vectors) between the candidate combinations of the answer components and the question components (again combined with Linear Programming to find the optimum candidate resource combination).

- **4. Query construction:** Having identified the features of the input question, matched them in the KG, scored them and detected both the question and the answer types, the next step is the question translation to a SPARQL query in order to retrieve the answer. For this task, as mentioned in Höffner et al. (2016), current approaches are along two paths: *(1) template-based approaches* (He et al. 2013), which map input questions to either manually or automatically created SPARQL query templates or *(2) template-free approaches* (Xu et al. 2014; Zou et al. 2014) that try to build SPARQL queries based on the given syntactic structure of the input question.
- **5. Answer presentation:** For returning the final answer to the user's question a kind of processing is required for making it intelligible to users. Since most people are not able to read structured representation of knowledge, and are not familiar with URIs, the translation of the RDF graph that corresponds to the answer to a natural language form is required. For instance, for factoid questions (as in Bast and Haussmann 2015), only the last part of the URI can be presented, or the value of the property `rdfs:label`. However, more complex processing is required in cases of complex questions, as in the case of Cao et al. (2011) where clustering and summarization is employed.

4.4 Processes for QA over data and documents (hybrid approaches)

A limitation of the previous approaches is that a single type of source can be insufficient to provide the ideal answer. The latter could be the synthesis of information extracted from various resources with different types of information representation. In such cases, the best approach is the adoption of hybrid approaches which combine different types of knowledge representation and exploit all the available information regardless their type. Hybrid approaches, are essentially systems that exploit data of different forms (e.g. plain text, RDF

[2] https://www.w3.org/TR/rdf-sparql-query/

datasets, SQL tables etc.) in order to understand the input question and extract the corresponding answer. In this work, we refer to Hybrid QA systems as the ones that exploit specifically, plain text and RDF linked data. Such systems, may exploit text to enrich the vocabulary of the structured knowledge (Feng et al. 2016; Savenkov and Agichtein 2016; Xu et al. 2016), a procedure that uses the ontology background knowledge to add semantic meaning to the available text and then interlink the mentioned extracted information to the original KB. Other approaches (Das et al. 2017), transform the data that hail from different types of sources in a unique global representation such as Universal schema (Riedel et al. 2013) and then use this schema as the underlying KB. The latter approach is helpful to train end-to-end NNs (Neural Networks) for the QA task, since they require a single representation of the data to be trained on. Furthermore, the approach of the Universal schema tackles the difficulties of interlinking unstructured text to RDF KBs, which is a challenging task, especially for large KBs (e.g. DBpedia, Freebase etc.) that contain many ambiguous entities, relations and types.

5 Categorizing the existing systems

In this section we briefly present and categorize 21 *Open-domain* QA systems that were published from 2013 until 2017, according to the following aspects: (1) the type of their underlying knowledge, (2) the methods used, (3) the evaluation metrics, and (4) the evaluation datasets used (evaluation is further discussed in Section 6). The categorization is shown in Table 1. Subsequently we briefly discuss the systems of each category.

5.1 Documents Oriented QA Systems

DrQA (Chen et al. 2017) is an *Open-domain* factoid QA system which exploits Wikipedia as knowledge source. The process comprises two steps: (1) Document Retrieval and (2) Machine Comprehension. In the former it exploits tf-idf vectors and local word order features to retrieve relevant articles, while in the latter it uses a Multi-Layer RNN (Recurrent Neural Network) in order to detect answer text spans.

Watanabe et al. (2017) present an *Open-domain* factoid QA system which exploits Wikipedia as knowledge source. It follows the two steps: (1) Document Retrieval and (2) Machine Comprehension. In the former it employs hand crafted heuristics and an RNN which exploits Word Level Attention to retrieve relevant articles. In the latter, it uses an RNN with Attention Sum architecture in order to detect answer text spans either from retrieved documents or a fixed vocabulary.

Wang et al. (2017) present an *Open-domain* factoid QA system with an end-to-end NN architecture which also supports causal questions. It uses Gated self-matching networks to approach the overall process as a reading comprehension task. It follows the steps: (1) matching the input question with a given passage to obtain a question-aware representation of the passage (using gated attention-based recurrent networks), (2) refine the produced representation by applying the proposed self-matching attention mechanism, and finally, (3) locate the answer boundaries by employing the pointer networks.

Xiong et al. (2016) present an *Open-domain* factoid QA system with an end-to-end NN architecture which also supports causal questions. It uses Dynamic co-attention networks to tackle the limitation of other proposed deep learning models, which are unable to recover from local maxima that correspond to incorrect answers, “due to their single-pass nature”. The proposed model consists of: (1) a co-attention encoder, which produces co-dependent

Table 1 Categorization of 21 *Open-domain* QA systems

System	Year	Method	Evaluation dataset	Data type
DrQA (Chen et al. 2017)	2017	Document Retrieval, Machine Comprehension	SQuAD, CuratedTREC, WebQuestions, Wikimovies	Documents
Watanabe et al. (2017)	2017	Document Retrieval, Machine Comprehension	Wikimovies	Documents
Wang et al. (2017)	2017	end-to-end NN	SQuAD	Documents
Xiong et al. (2016)	2017	end-to-end NN	SQuAD	Documents
CFO (Dai et al. 2016)	2016	Conditional Focused NN	SimpleQuestions	Data
Lukovnikov et al. (2017)	2017	end-to-end NN	SimpleQuestions	Data
Bast and Haussmann (2015)	2015	Entity Matching, Template Matching, Relation Matching, Query Ranking	Free917, WebQuestions	Data
He et al. (2013)	2013	question analysis, resource mapping, SPARQL generation	QALD-3	Data
gAnswer (Zou et al. 2014)	2014	Semantic Parsing, Semantic Query Graph Generation, Graph Matching	QALD-3	Data
Berant and Liang (2015)	2015	Imitation Learning, Agenda-based Semantic Parsing, Resource Mapping, SPARQL construction	WebQuestions	Data
Xser (Xu et al. 2014)	2014	Phrasal Semantic Parsing, Resource mapping, SPARQL generation	QALD-4	Data
SEMPRE (Berant et al. 2013)	2013	Semantic Parsing, Resource mapping	Free917, WebQuestions	Data
PARASEMPRE (Berant and Liang 2014)	2014	Semantic Parsing, Paraphrasing	Free917, WebQuestions	Data
STAGG (Yih et al. 2015)	2015	Semantic Parsing, Graph Generation, CNN	WebQuestions	Data
jacana-freebase (Yao and Van Durme 2014b)	2014	Information Extraction, Information Retrieval, Classification	ReverbMapping, CluewebMapping, WebQuestions	Data
Zhang et al. (2016)	2016	ILP, Alignment Construction, Query Construction	QALD-4	Data
ISOFT (Park et al. 2015)	2015	Text Information Extraction, Text Indexing, SPARQL Template Generation, Sequential Query Generation	QALD-5	Hybrid
(Das et al. 2017)	2017	Universal schema, Memory Networks	Spades	Hybrid

Table 1 (continued)

System	Year	Method	Evaluation dataset	Data type
Text2KB (Savenkov and Agichtein 2016)	2016	Entity Matching, Template Matching, Relation Matching, Query Ranking	WebQuestions	Hybrid
(Feng et al. 2016)	2016	Entity Linking, Relation Extraction, ILP	WebQuestions	Hybrid
(Xu et al. 2016)	2016	Entity Linking, Relation Extraction, Answer Refinement	WebQuestions	Hybrid

System: Name and/or citation of the system. Year: Publishing Year. Method: Methods used by the system namely. Evaluation Dataset: Dataset used for system evaluation. Data Type: Format of underlying data used by the system

representations of the input question and the corresponding document and (2) a dynamic decoder, which iteratively estimates the answer boundaries.

5.2 Data Oriented QA Systems

CFO (Dai et al. 2016) is an *Open-domain* factoid KBQA (Knowledge Base Question Answering) system which approaches the overall process as a binary classification task, i.e. each subject-predicate pair is a candidate answer to an input question which is classified as answer or not. It follows the steps: (1) compute the probability of each pair given the question, using a Conditional-focused NN approach, (2) factorize the probabilities by first inferring the target relation, and then the subject that is associated with the candidate relations.

Lukovnikov et al. (2017) present an *Open-domain* factoid KBQA system with an end-to-end NN architecture which approaches the overall process as a binary classification task. The proposed approach, includes an RNN which is trained on character and word level, which is then used as an encoder for the question as well as the subjects and the predicates. The input features used, are character encodings (using Gated Recurrent Unit - GRU) for the entity labels, and word embeddings (using GloVe (Pennington et al. 2014)) for predicates URIs and subject types.

Aqqu (Bast and Haussmann 2015) is an *Open-domain* factoid KBQA system. It follows the steps: (1) Entity Identification, to retrieve candidate entities, (2) Template Matching, to construct candidate queries based on three handcrafted templates, (3) Relation Matching, to find candidate predicates of the KB that match the input question's phrases and (4) Rank Queries, to find the most suitable query based on a set of features.

Casia (He et al. 2013) is an *Open-domain* factoid KBQA system that follows the steps: (1) generate graph pattern templates, based on the question type, the Named Entities and the POS tags, (2) map the KB resources and the patterns (using WordNet, PATTY and similarity measures), and (3) build SPARQL queries, to retrieve the answer.

gAnswer (Zou et al. 2014) is an *Open-domain* factoid KBQA system. It follows the steps: (1) convert the input question into a semantic query graph, (2) retrieve candidate matches with confidence scores, (3) retrieve top-k matches over RDF subgraphs.

Berant and Liang (2015) present an *Open-domain* factoid KBQA system which (1) exploits a semantic parser, trained through imitation learning techniques over WebQuestions, (2) proposes an agenda-based parsing that scales to the size of a KB such as Freebase, (3) exploits tools such as Freebase Search API, Stanford CoreNLP, λ-DCS (Lambda Dependency-Based Compositional Semantics) logical forms and lexicons in order to extend the parser to complete QA system.

Xser (Xu et al. 2014) is an *Open-domain* KBQA system. It follows the steps: (1) assign semantic labels to question phrases via phrasal semantic parsing, (2) use a structured perceptron model to jointly solve the mappings between semantic phrases and KB items.

SEMPRE (Berant et al. 2013) is an *Open-domain* KBQA system that (1) constructs a lexicon, in order to map NL phrases to logical predicates (via alignment of large text with Free base), and (2) generates logical predicates which are compatible with neighboring predicates. The proposed approach, tackles the limitations of other approaches as regards the need of annotated logical forms for training and the limited domains they operate.

PARASEMPRE (Berant and Liang 2014) is an *Open-domain* KBQA system. It applies semantic parsing based on paraphrasing, which exploits large amount of external text. It follows the steps: (1) use a deterministic procedure to generate a set of candidate logical forms, along with a canonical realization, and (2) use a paraphrase model to choose the realization that best paraphrases the question and outputs the logical form.

STAGG (Yih et al. 2015) is an *Open-domain* KBQA system. It follows the steps: (1) represent the meaning of the input question as a query-graph (that can be mapped into a logical form), (2) apply entity linking as well as a Convolutional Neural Network in order to match questions and predicate sequences.

jacana-freebase (Yao and Van Durme 2014b) is an *Open-domain* KBQA system that treats the KB as an interlinked collection of “topics”. It follows the steps: (1) retrieve the corresponding topic-node for each entity in the question (using the Freebase Search API), (2) retrieve topic-graph, based on these topics (using the Freebase Topic API), (3) extract features for the question and the topic-graph, finally (4) use a classifier to determine the answer node.

Zhang et al. (2016) present an *Open-domain* KBQA system using multiple KBs.

The proposed approach, consists of a joint method based on Integer Linear Programming (ILP), which jointly considers the two interactive tasks of the Alignment Construction and the Query Construction.

5.3 Hybrid QA systems

Park et al. (2015) present an *Open-domain* Hybrid QA system that exploits free text for enriching the knowledge and the vocabulary of the KB and aims at answering factoid, comparative and confirmation questions. It follows the steps: (1) analyze input question to extract sub-questions, POS tags, EAT etc. (2) search in the multi-information tagged text database (built from Wikipedia articles using NLP and Disambiguation tools) to extract candidate answers, and (3) generate SPARQL queries based on specific kind of words i.e. that reveal the type of the question, e.g. for comparative words like ‘deeper’ etc.

Das et al. (2017) present an *Open-domain* Hybrid QA system which exploits and extends the Universal schema (Riedel et al. 2013). It employs memory networks for the construction of a KB with mixed data using a common representation, as well as a question encoder in order to create its representation in a common space. Therefore, since all the available information is aligned in a common embedded space, the system can support reasoning over the data despite their type and is able to serve as a QA system.

Text2KB (Savenkov and Agichtein 2016) is an *Open-domain* Hybrid QA system that extends the system Aqqu (Bast and Haussmann 2015). It exploits external text information, to improve the tasks: a) Entity Linking, b) Predicate Matching and c) Candidate Ranking. Particularly, it exploits: (1) Web search results to improve the task (a), (2) Community QA data in order to improve the task (b) and finally (3) Documents collection with detected KB entity mentions to improve the task (c).

Feng et al. (2016) present an *Open-domain* Hybrid QA system that exploits two KBs (DBPedia and Freebase) as well as the English Wikipedia as text resource. It follows the steps: (1) perform entity linking using DBpedia Lookup[3] and S-MART (Yi and Chang 2016), (2) simultaneously perform relation extraction over KB and free text, using a Multi-Channel Convolutional Neural Network and a paraphrase model respectively, and (3) apply an Integer Linear Program (ILP) model to infer the candidates and provide an optimal solution.

Xu et al. (2016) present an *Open-domain* Hybrid QA system that exploits the Freebase KB and the English Wikipedia as text resource. It follows the steps: (1) perform joint entity linking using S-MART (Yi and Chang 2016) and relation extraction using a Multi-Channel Convolutional Neural Network, (2) apply the answer refinement process which is treated as a binary classification task using Support Vector Machines. The features used, were extracted from the Wikipedia pages of the identified Freebase entities.

6 Evaluation methods

TREC[4] has had a question answering track since 1999. We should note that creating the equivalent of a standard Information Retrieval test collection is a difficult problem. In an IR test collection, the unit that is judged, the document, has an unique identifier, and it is easy to decide whether a document retrieved is the same document that has been judged. For QA, the unit that is judged is the entire string that is returned by the system and quite often different QA runs return different answer strings, hence it is difficult to determine automatically whether the difference between a new string and a judged string is significant with respect to the correctness of the answer. One method to tackle this problem is to use so-called answer patterns and accept a string as correct if it matches an answer pattern for the question, answer patterns are described in Voorhees and Tice (2000). Having tackled such issues, below follow a list of the metrics that are more commonly used.

Accuracy: If Q denotes the set of questions that are used in the evaluation, and AQC those that were answered correctly, then the Accuracy is the fraction of the questions that were answered correctly i.e. $Accuracy = \frac{|AQC|}{|Q|}$ (e.g. this is the way it is used in Bordes et al. (2015) and Pasupat and Liang (2015) and others). The number of wrongly answered questions divided by Q is called *error* (obviously $error = 1 - Accuracy$). Sometimes the word accuracy refers to other kinds of accuracy, e.g. to the ability of the system to compute correct confidence scores.

For instance if “X” is a correct answer and a system (say A) predicts “X as the answer”, with confidence score equals to 0.7 and another system (say B) predicts the same, but with confidence score equals to 0.9, then we can say that system B is more accurate than system

[3] http://wiki.dbpedia.org/projects/dbpedia-lookup

[4] https://trec.nist.gov/data/qa.html

A, despite the fact that they achieved the same precision, since they both predicted the correct answer (as in Iyyer et al. 2014).

Precision, Recall and F-measure: When evaluating a list question q, i.e. a question whose responses is a set of elements, or if several correct answers to a question are possible,] then evaluation metrics from IR (Information Retrieval) are commonly employed. If $Correct(q)$ is the set of elements that forms the perfect answer for q, and $Found(q)$ are those that the QA system returned, then the *Precision* (of the system as regards q) is the fraction of responses that are correct, i.e. $Precision(q) = \frac{|Found(q) \cap Correct(q)|}{|Found(q)|}$, while Recall is the fraction of the correct elements that the system found, i.e. $Recall(q) = \frac{|Found(q) \cap Correct(q)}{|Correct(q)|}$. Now the *F-measure* (also called F-score or F1-score), represents the harmonic mean of Precision and Recall i.e. $F1 = 2\frac{Precision * Recall}{Precision + Recall}$. It is used in order to capture both precision and recall capabilities of a system in a single score and it is used in several well known QA benchmarks like QALD.

Average Precision (AveP): It is the average value of the precision as a function of the recall. It requires computing the precision and recall at every position in the ranked list (to take into account the order of the results). It is mainly used to evaluate ranked-result questions, like the metric MRR that will be described next.

Mean Average Precision (MAP): It is the mean of the AveP scores for a set of queries, e.g. for set Q of questions that are used in the evaluation, and it is calculated by summing all AveP scores for each query, and then dividing by the total number of queries.

Mean Reciprocal Rank (MRR): For a system returning a ranked list of elements the Mean Reciprocal Rank (MRR), over a set of questions Q, is defined as $MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i}$ where $rank_i$ refers to the rank position of the first relevant hit for the i-th question. This metric can be used not only in QA systems but also in dialogue systems, e.g. for measuring the improvement because of dialogue act information (as in Kumar et al. 2018).

Table 2 lists 23 commonly used *evaluation/training datasets* in QAs and DS. They are categorized according to the following aspects: domain (open or closed), underlying knowledge source, available tasks, and finally, the associated evaluation metrics.

7 Current trends and challenges

General Observations A general observation is that most of the works focus on answering factoid questions. A few quantitative observations, about the systems shown in Table 1, follow: (a) *all* of the document-based systems employ *Neural Networks* and half of them exploit *Wikipedia*, (b) *half* of the data-based systems employ a kind of *ML technique*, and (c) *most* of the hybrid systems employ a kind of *ML technique*.

Tools By analyzing the bibliography the list of tools that are mentioned very frequently in the literature are: *Word2Vec* (Mikolov et al. 2013)[5], *FRED* (Presutti et al. 2012; Gangemi et al. 2017), *Qanary* (Diefenbach et al. 2017b; Singh et al. 2016; Diefenbach et al. 2017a),

[5] https://deeplearning4j.org/word2vec.html

Table 2 A guide to QA/DS datasets

Dataset	Domain	Tasks	Evaluation metrics	Knowledge source
SQuAD (Rajpurkar et al. 2016)	open	Machine Comprehension	F1, EM	Documents
CNN/Daily Mail (Hermann et al. 2015)	open	Machine Comprehension	Accuracy	Documents
CBT (Hill et al. 2015)	open	Exploit Linguistic Context, Predict missing word	Accuracy	Documents
QuizBowl (Iyyer et al. 2014)	open	Answer Triggering	Accuracy	Documents
WikiQA (Yi et al. 2015)	open	Answer Sentence Selection, Answer Triggering	Precision, Recall, F1, MAP, MRR	Documents
Question Classification Corpus (Li and Roth 2006)	open	EAT Identification	Precision	Documents
CuratedTREC (Baudiš and Šedivỳ 2015)	open	QA	Accuracy@k, F1, Recall, MRR	Documents
WebQuestions (Berant et al. 2013)	open	Semantic Parsing	F1, Accuracy	Data
SimpleQuestions (Bordes et al. 2015)	open	QA	F1, Accuracy	Data
Frames (El Asri et al. 2017)	closed	Frame Tracking	Frame Creation, Frame Identification	Data
QALD1	both	QA over RDF	F1, Precision, Recall	Data
QALD2	both	QA over RDF	F1, Precision, Recall	Data
QALD3 (Cimiano et al. 2013)	both	Multilingual QA, Lexicalization	F1, Precision, Recall, Lexical Precision, Lexical Recall, Lexical Accuracy	Data
LC-QuAD (Trivedi et al. 2017)	open	Large Scale QA over RDF	F1, Precision, Recall	Data
WikiTable (Pasupat and Liang 2015)	open	Semantic Parsing	Accuracy	Data
SPADES (Bisk et al. 2016)	open	Slot Filling	Simplified Labeled F1, Undirected Unlabeled F1, Slot Filling Performance,	Hybrid
WikiMovies (Miller et al. 2016)	closed	QA	hits@k	Hybrid
WikiReading (Hewlett et al. 2016)	open	Natural Language Understanding	F1	Hybrid
QALD4 (Penas et al. 2014)	both	Multilingual QA, QA over Interlinked Data, Hybrid QA	F1, Precision, Recall	Hybrid
QALD5 (Unger et al. 2015)	open	Multilingual QA, Hybrid QA	F1, Precision, Recall, F1-Global	Hybrid

Table 2 (continued)

Dataset	Domain	Tasks	Evaluation metrics	Knowledge source
QALD6 (Unger et al. 2016)	both	Multilingual QA, Hybrid QA, Statistical QA over RDF datacubes	F1, Precision, Recall, F1-Global	Hybrid
QALD7	open	Multilingual QA, Hybrid QA, Large Scale QA over RDF, English question answering over Wikidata	F1, Precision, Recall	Hybrid
MovieDialog (Dodge et al. 2015)	closed	QA, Recommendation, Dialogue	hits@k	Hybrid

Dataset: Name and citation of the data set. Domain: Type of the domain the dataset focuses on (i.e. *Open* or *Close Domain*). Tasks: The tasks for which the dataset is provided for. Evaluation Metrics: Evaluation metrics that the dataset proposes. Knowledge Source: The underlying format of knowledge (i.e. *Data*, *Documents* or *Hybrid*)

PATTY (Nakashole et al. 2012)[6], *WordNet* (Miller 1995)[7], *Stanford CoreNLP* (Manning et al. 2014)[8], *DBpedia Spotlight* (Mendes et al. 2011),[9], and *FOX (Federated knOwledge eXtraction Framework)* (Speck and Ngomo 2014)[10].

Challenges and Directions Although the current paper covered many different types of QA systems, i.e. document-based QA (in Section 4.2), KB-based QA (in Section 4.3), and Hybrid QA (in Section 4.4), according to our opinion the most interesting is the latter. Since, different knowledge sources come along with their pros and cons, as we analyzed in this survey, a reasonable direction is the one that aims at exploiting different types of underlying knowledge in order to minimize the limitations of each individual one. It is undeniable that the biggest amount of factual information on the web, currently is covered by plain text (e.g. documents, sites, blogs etc.). Based on the above facts, a direction that is worth further research is the task of *exploiting external plain-text knowledge* in order to: enrich the vocabulary and the contents of the KB (for improving tasks like disambiguation, recall the *ambiguity* challenge). The opposite direction is also interesting, that is the pursuit for more flexible and effective methods for structuring and interlinking plain text information with structured representations of knowledge such as RDF graphs, in an attempt to approach a solution to the challenge of *distributed knowledge*. Finally, evaluation datasets and benchmarks are crucial for obtaining comparable results (and thus for comparing and advancing the effectiveness of QA methods), therefore the *evaluation process* per se is also an interesting research direction.

[6] https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/patty/

[7] http://wordnet.princeton.edu/

[8] https://stanfordnlp.github.io/CoreNLP/

[9] https://github.com/dbpedia-spotlight/dbpedia-spotlight/wiki/Installation

[10] http://aksw.org/Projects/FOX.html

8 Concluding remarks

In this paper we surveyed Question Answering and Dialogue systems with emphasis on the first type of systems. We discussed the main challenges, we structured the landscape according to various aspects, and we described the steps of the QA process that is followed by three main categories of systems, as well as the methods and techniques that are commonly used for realizing each of these steps. We then categorized 21 recent and successful systems according to various aspects (knowledge source, types of questions, and domain type). Since evaluation is a topic of utmost importance, we collected a set of 23 evaluation and training datasets that are most commonly used in the literature and we categorized them according to the type of the domain (open or closed), the underlying knowledge source, the provided available tasks, and the associated evaluation metrics used. Finally, we listed practical tools that are widely used for carrying out certain tasks of the QA pipeline, and we discussed trends, directions and challenges.

Acknowledgments We would like to thank Michalis Mountantonakis, Katerina Papantoniou and Yannis Marketakis for making suggestions that improved the paper.

References

Abacha, A.B., & Zweigenbaum, P. (2015). Means: A medical question-answering system combining nlp techniques and semantic web technologies. *Information Processing and Management*, *51*(5), 570–594.

Abend, O., Reichart, R., Rappoport, A. (2009). Unsupervised argument identification for semantic role labeling. In *Procs of the Joint Conf. of the 47th Annual Meeting of the ACL and the 4th Intern. Joint Conf. on Natural Language Processing of the AFNLP* (pp. 28–36): Association for Computational Linguistics.

Abney, S.P. (1991). Parsing by chunks. In *Principle-based parsing* (pp 257–278): Springer.

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

Anam, S., Kim, Y.S., Kang, B.H., Liu, Q. (2016). Adapting a knowledge-based schema matching system for ontology mapping. In *Proceedings of the Australasian Computer Science Week Multiconference, ACSW '16* (pp. 27:1–27:10). New York: ACM.

El Asri, L., Schulz, H., Sharma, S., Zumer, J., Harris, J., Fine, E., Mehrotra, R., Suleman, K. (2017). Frames: A corpus for adding memory to goal-oriented dialogue systems. CoRR, arXiv:1704.00057.

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z. (2007). Dbpedia: A nucleus for a web of open data. *The semantic web*, 722–735.

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

Baudiš, P., & Šedivỳ, J. (2015). Modeling of the question answering task in the yodaqa system. In *International Conference of the Cross-Language Evaluation Forum for European Languages* (pp. 222–228): Springer.

Berant, J., Chou, A., Frostig, R., Liang, P. (2013). Semantic parsing on freebase from question-answer pairs. *EMNLP*, *2*, 6.

Berant, J., & Liang, P. (2014). Semantic parsing via paraphrasing. In *ACL (1)* (pp. 1415–1425).

Berant, J., & Liang, P. (2015). Imitation learning of agenda-based semantic parsers. *Transactions of the Association for Computational Linguistics*, *3*, 545–558.

Bisk, Y., Reddy, S., Blitzer, J., Hockenmaier, J., Steedman, M. (2016). *Evaluating induced ccg parsers on grounded semantic parsing*. arXiv:1609.09405.

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J. (2008). Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data* (pp. 1247–1250): ACM.

Bonadiman, D., Uva, A.E., Moschitti, A. (2017). *Multitask learning with deep neural networks for community question answering*. CoRR, arXiv:1702.03706.

Bordes, A., Usunier, N., Chopra, S., Weston, J. (2015). *Large-scale simple question answering with memory networks*. CoRR, arXiv:1506.02075.

Cao, Y., Liu, F., Simpson, P., Antieau, L., Bennett, A., Cimino, J.J., Ely, J., Yu, H. (2011). Askhermes: An online question answering system for complex clinical questions. *Journal of Biomedical Informatics*, *44*(2), 277–288.

Chapman, A., Simperl, E., Koesten, L.aura., Konstantinidis, G., Ibáñez-Gonzalez, L.-D., Kacprzak, E., Groth, P.aul. (2019). *Dataset search: a survey*. arXiv:1901.00735.

Chen, D., Fisch, A., Weston, J., Bordes, A. (2017). *Reading wikipedia to answer open-domain questions*. CoRR, arXiv:1704.00051.

Chu-Carroll, J., Fan, J., Schlaefer, N., Zadrozny, W. (2012). Textual resource acquisition and engineering. *IBM Journal of Research and Development*, *56*(3.4), 4–1.

Cimiano, P., Lopez, V., Unger, C., Cabrio, E., Ngonga Ngomo, A., Walter, S. (2013). *Data: Multilingual Question Answering over Linked (QALD-3): Lab Overview*, (pp. 321–332). Berlin: Springer.

Dai, Z., Li, L., Xu, W. (2016). *Cfo: Conditional focused neural question answering with large-scale knowledge bases*. arXiv:1606.01994.

Das, R., Zaheer, M., Reddy, S., McCallum, A. (2017). *Question answering on knowledge bases and text using universal schema and memory networks*. arXiv:1704.08384.

Devi, M., & Dua, M. (2017). Adans: An agriculture domain question answering system using ontologies. In *2017 International Conference on Computing, Communication and Automation (ICCCA)* (pp. 122–127). IEEE.

Devlin, J., Chang, M.-W., Lee, K., Toutanova, K. (2018). *Bert: Pre-training of deep bidirectional transformers for language understanding*. CORR arXiv:1810.04805.

Dhuliawala, S., Kanojia, D., Bhattacharyya, P. (2016). Slangnet: A wordnet like resource for english slang. In *LREC*.

Diefenbach, D., Hormozi, N., Amjad, S., Both, A. (2017a). Introducing feedback in qanary: How users can interact with QA systems. In *European Semantic Web Conference* (pp. 81–86): Springer.

Diefenbach, D., Singh, K., Both, A., Cherix, D., Lange, C., Auer, S. (2017b). The qanary ecosystem: getting new insights by composing question answering pipelines. In *International Conference on Web Engineering* (pp. 171–189): Springer.

Dimitrakis, E., Sgontzos, K., Papadakos, P., Marketakis, Y., Papangelis, A., Stylianou, Y., Tzitzikas, Y. (2018). On finding the relevant user reviews for advancing conversational faceted search. EMSASW@ ESWC.

Dodge, J., Gane, A., Zhang, X., Bordes, A., Chopra, S., Miller, A.H., Szlam, A., Weston, J. (2015). Evaluating prerequisite qualities for learning end-to-end dialog systems. CoRR, arXiv:1511.06931.

Dubey, M., Banerjee, D., Chaudhuri, D., Lehmann, J. (2018). Earl: Joint entity and relation linking for question answering over knowledge graphs. In *International Semantic Web Conference* (pp. 108–126): Springer.

Fader, A., Soderland, S., Etzioni, O. (2011). Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (pp. 1535–1545): Association for Computational Linguistics.

Feng, Y., Huang, S., Zhao, D., et al. (2016). Hybrid question answering over knowledge base and free text. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers* (pp. 2397–2407).

Ferret, O., Grau, B., Hurault-Plantet, M., Illouz, G., Monceaux, L., Robba, I., Vilnat, A. (2001). Finding an answer based on the recognition of the question focus. In *TREC*.

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. (2010). Building watson: an overview of the deepqa project. *AI Magazine*, *31*(3), 59–79.

Ferrucci, D.A. (2012). Introduction to ”this is watson”. *IBM Journal of Research and Development*, *56*(3.4), 1–1.

Ferrucci, D., Levas, A., Bagchi, S., Gondek, D., Mueller, E.T. (2013). Watson: beyond jeopardy!. *Artificial Intelligence*, *199*, 93–105.

Gangemi, A., Presutti, V., Recupero, D.R., Nuzzolese, A.G., Draicchio, F., Mongiovì, M. (2017). Semantic web machine reading with fred. *Semantic Web*, *8*(6), 873–893.

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

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

Gustafson, J., Bell, L., Beskow, J., Boye, J., Carlson, R., Edlund, J., Granström, B., House, D., Wirén, M. (2000). Adapt—a multimodal conversational dialogue system in an apartment domain. In *The Sixth International Conference on Spoken Language Processing (ICSLP)* (pp. 134–137). Beijing.

He, S., Liu, S., Chen, Y., Zhou, G., Liu, K., Zhao, J. (2013). Casia@ qald-3: A question answering system over linked data. In *CLEF (Working Notes)*.

Hermann, K.M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., Blunsom, P. (2015). Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems* (pp. 1693–1701).

Herzog, O., & Rollinger, C.-R. (1991). *Text understanding in LILOG: integrating computational linguistics and artificial intelligence: final report on the IBM Germany LILOG-project*. Berlin: Springer.

Hewlett, D., Lacoste, A., Jones, L., Polosukhin, I., Fandrianto, A., Han, J., Kelcey, M., Berthelot, D. (2016). Wikireading: A novel large-scale language understanding task over wikipedia. CoRR, arXiv:1608.03542.

Hill, F., Bordes, A., Chopra, S., Weston, J. (2015). The goldilocks principle: Reading children's books with explicit memory representations. arXiv:1511.02301.

Höffner, K., Walter, S., Marx, E., Usbeck, R., Lehmann, J., Ngonga Ngomo, A.-C. (2016). Survey on challenges of Question Answering in the Semantic Web. *Semantic Web Journal*.

Iyyer, M., Boyd-Graber, J.L., Claudino, L.M.B., Socher, R., Daumé, H. III (2014). A neural network for factoid question answering over paragraphs. In *EMNLP* (pp. 633–644).

Konstantinova, N., & Orasan, C. (2012). Interactive question answering. *Emerging Applications of Natural Language Processing: Concepts and New Research* (pp. 149–169).

Kumar, H., Agarwal, A., Joshi, S. (2018). Dialogue-act-driven conversation model: an experimental study. In *Proc. of the 27th International Conference on Computational Linguistics* (pp. 1246–1256).

Lally, A., Prager, J.M., McCord, M.C., Boguraev, B.K., Patwardhan, S., Fan, J., Fodor, P., Chu-Carroll, J. (2012). Question analysis: How watson reads a clue. *IBM Journal of Research and Development*, *56*(3.4), 2–1.

Li, X., & Roth, D. (2006). Learning question classifiers: The role of semantic information. *Natural Language Engineering*, *12*(3), 229–249.

Lin, J. (2007). An exploration of the principles underlying redundancy-based factoid question answering. *ACM Transactions on Information Systems (TOIS)*, *25*(2), 6.

Liu, H., Hu, Q., Zhang, Y., Xing, C., Sheng, M. (2017). A knowledge-based health question answering system. In *Smart Health* (pp. 286–291). Cham: Springer International Publishing.

Lukovnikov, D., Fischer, A., Lehmann, J., Auer, S. (2017). Neural network-based question answering over knowledge graphs on word and character level. In *Procs of the 26th International Conference on World Wide Web* (pp. 1211–1220).

Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J.R., Bethard, S., McClosky, D. (2014). The stanford corenlp natural language processing toolkit. In *ACL (System Demonstrations)* (pp 55–60).

McCrae, J., Spohr, D., Cimiano, P. (2011). Linking lexical resources and ontologies on the semantic web with lemon. In *Extended Semantic Web Conference* (pp. 245–259). Springer.

Mendes, P.N., Jakob, M., García-Silva, A., Bizer, C. (2011). Dbpedia spotlight: Shedding light on the web of documents. In *Procs of the 7th International Conference on Semantic Systems, I-Semantics '11*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems* (vol. 26, pp. 3111–3119).

Miller, G.A. (1995). Wordnet: A lexical database for english. *Communications of the ACM*, *38*(11), 39–41.

Miller, A.H., Fisch, A., Dodge, J., Karimi, A., Bordes, A., Weston, J. (2016). Key-value memory networks for directly reading documents. CoRR, arXiv:1606.03126.

Mishra, A., & Jain, S.K. (2016). A survey on question answering systems with classification. *Journal of King Saud University-Computer and Information Sciences*, *28*(3), 345–361.

Mollá, D., & Vicedo, J.L. (2007). Question answering in restricted domains An overview. *Computer Linguistics*, *33*(1), 41–61.

Moreo, A., Castro, J. L., Zurita, J. M. (2017). Towards portable natural language interfaces based on case-based reasoning. *Journal of Intelligent Information Systems*, 1–34.

Mountantonakis, M., & Tzitzikas, Y. (2016). On measuring the lattice of commonalities among several linked datasets. *Proceedings of the VLDB Endowment*, *9*(12), 1101–1112.

Mountantonakis, M., & Tzitzikas, Y. (2018). Scalable methods for measuring the connectivity and quality of large numbers of linked datasets. *ACM Journal of Data and Information Quality (JDIQ)*, *9*(3), 15.

Mountantonakis, M., & Tzitzikas, Y. (2019). Large scale semantic integration of linked data: A survey. *ACM Computing Surveys, 52*(5).

Nakashole, N., Weikum, G., Suchanek, F. (2012). Patty: A taxonomy of relational patterns with semantic types. In *Procs of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (pp. 1135–1145).

Norouzi, N., Bruder, G., Belna, B., Mutter, S., Turgut, D., Welch, G. (2019). A systematic review of the convergence of augmented reality, intelligent virtual agents, and the internet of things. In *Artificial Intelligence in IoT* (pp. 1–24): Springer.

Ogden, W., Mcdonald, J., Bernick, P., Chadwick, R. (2008). Habitability in question-answering systems. In *Advances in Open Domain Question Answering* (pp. 457–473). Springer.

Papangelis, A., Papadakos, P., Kotti, M., Stylianou, Y., Tzitzikas, Y., Plexousakis, D. (2017). Ld-sds: Towards an expressive spoken dialogue system based on linked-data. In *Search Oriented Conversational AI, SCAI 17 Workshop (co-located with ICTIR)* (pp. 17).

Park, S., Kwon, S., Kim, B., Lee, G.G. (2015). Isoft at qald-5: Hybrid question answering system over linked data and text data. In *CLEF (Working Notes)*.

Pasupat, P., & Liang, P. (2015). Compositional semantic parsing on semi-structured tables. arXiv:1508.00305.

Pathak, S., & Mishra, N. (2016). Context aware restricted tourism domain question answering system. In *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)* (pp. 534–539): IEEE.

Patra, B. (2017). A survey of community question answering. CoRR, arXiv:1705.04009.

Penas, A., Unger, C., Ngomo, A.-C.N. (2014). *Overview of CLEF Question Answering Track 2014*, (pp. 300–306). Cham: Springer International Publishing.

Pennington, J., Socher, R., Manning, C. (2014). Glove: Global vectors for word representation. In *Procs of the 2014 conf. on empirical methods in natural language processing (EMNLP)* (pp. 1532–1543).

Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L. (2018). Deep contextualized word representations. CORR arXiv:1802.05365.

Pham, M., Alse, S., Knoblock, C.A., Szekely, P. (2016). *Semantic Labeling: A Domain-Independent Approach*, (pp. 446–462). Cham: Springer International Publishing.

Popescu, A.-M., Etzioni, O., Kautz, H. (2003). Towards a theory of natural language interfaces to databases. In *Proceedings of the 8th international conference on Intelligent user interfaces* (pp. 149–157): ACM.

Presutti, V., Draicchio, F., Gangemi, A. (2012). Knowledge extraction based on discourse representation theory and linguistic frames. In *International Conference on Knowledge Engineering and Knowledge Management* (pp. 114–129): Springer.

Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P. (2016). Squad: 100,000+ questions for machine comprehension of text. arXiv:1606.05250.

Ramnandan, S.K., Mittal, A., Knoblock, C.A., Szekely, P. (2015). *Assigning Semantic Labels to Data Sources*, (pp. 403–417). Cham: Springer International Publishing.

Riedel, S., Yao, L., McCallum, A., Marlin, B.M. (2013). Relation extraction with matrix factorization and universal schemas. In *Procs of the 2013 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Rodrigo, A. (2017). Anselmo penas A study about the future evaluation of question-answering systems. *Knowledge-Based Systems*, *137*, 83–93.

Sasikumar, U., & Sindhu, L. (2014). A survey of natural language question answering system. *International Journal of Computer Applications, 108*(15).

Savenkov, D., & Agichtein, E. (2016). When a knowledge base is not enough: Question answering over knowledge bases with external text data. In *Procs of the 39th Intern. ACM SIGIR Conference*.

Schmidt, M., & Braunger, P. (2018). A survey on different means of personalized dialog output for an adaptive personal assistant. In *Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization* (pp. 75–81): ACM.

Serban, I.V., Lowe, R., Henderson, P., Charlin, L., Pineau, J. (2015). A survey of available corpora for building data-driven dialogue systems. CoRR, arXiv:1512.05742.

Singh, K., Both, A., Diefenbach, D., Shekarpour, S., Cherix, D., Lange, C. (2016). Qanary–the fast track to creating a question answering system with linked data technology. In *International Semantic Web Conference* (pp. 183–188): Springer.

Song, S., Zhang, X., Qin, G. (2017). *Multi-domain ontology mapping based on semantics*. Cluster Computing.

Speck, R., & Ngomo, A.-C.N. (2014). Ensemble learning for named entity recognition. In *International semantic web conference* (pp. 519–534): Springer.

Trivedi, P., Maheshwari, G., Dubey, M., Lehmann, J. (2017). Lc-quad: A corpus for complex question answering over knowledge graphs. In *International Semantic Web Conference* (pp. 210–218): Springer.

Unger, C., Forascu, C., López, V., Ngomo, A.-C.N., Cabrio, E., Cimiano, P., Walter, S. (2015). Question answering over linked data (QALD-5). In *Working Notes of CLEF 2015 - Conference and Labs of the Evaluation forum*: Toulouse, France.

Unger, C., Ngomo, A.-C.N., Cabrio, E. (2016). *6th Open Challenge on Question Answering over Linked Data (QALD-6)*, (pp. 171–177). Cham: Springer.

Voorhees, E.M., & Tice, D.M. (2000). Building a question answering test collection. In *Procs of the 23rd annual inter. ACM SIGIR conf. on Research and development in information retrieval* (pp. 200–207): ACM.

Wang, M. (2006). A survey of answer extraction techniques in factoid question answering. *Computational Linguistics, 1*(1).

Wang, W., Yang, N., Wei, F., Chang, B., Zhou, M. (2017). Gated self-matching networks for reading comprehension and question answering. In Procs of the 55th Annual Meeting of the Association for Computational Linguistics (vol. 1, pp. 189–198).

Wang, Z., Wang, Z., Long, Y., Wang, J., Xu, Z., Wang, B. (2019). Enhancing generative conversational service agents with dialog history and external knowledge. *Computer Speech & Language*, *54*, 71–85.

Watanabe, Y., Dhingra, B., Salakhutdinov, R. (2017). Question answering from unstructured text by retrieval and comprehension. CoRR, arXiv:1703.08885.

Woods, W.A. (1973). Progress in natural language understanding: an application to lunar geology. In *Procs of the June 4-8, 1973, national computer conference and exposition* (pp. 441–450): ACM.

Xiong, C., Xhong, V., Socher, R. (2016). Dynamic coattention networks for question answering. arXiv:1611.01604.

Xu, K., Zhang, S., Feng, Y., Zhao, D. (2014). Answering natural language questions via phrasal semantic parsing. In *Natural Language Processing and Chinese Computing* (pp. 333–344): Springer.

Xu, K., Reddy, S., Feng, Y., Huang, S., Zhao, D. (2016). Question answering on freebase via relation extraction and textual evidence. arXiv:1603.00957.

Yang, L., Ai, Q., Spina, D., Chen, R.-C., Pang, L., Croft, W.B., Guo, J., Scholer, F. (2016). Beyond factoid qa: effective methods for non-factoid answer sentence retrieval. In *European Conference on Information Retrieval* (pp. 115–128): Springer.

Yi, Y., Yih, W.-T., Meek, C. (2015). Wikiqa: A challenge dataset for open-domain question answering. In *EMNLP* (pp. 2013–2018).

Yi, Y., & Chang, M.-W. (2016). S-mart: Novel tree-based structured learning algorithms applied to tweet entity linking. arXiv:1609.08075.

Yao, X., Berant, J., Van Durme, B., Qa, F. (2014a). Information extraction or semantic parsing. In *Proceedings of ACL.*

Yao, X., & Van Durme, B. (2014b). Information extraction over structured data: Question answering with freebase. In *ACL (1)* (pp. 956–966).

Yih, W.-T., Chang, M.-W., He, X., Gao, J. (2015). Semantic parsing via staged query graph generation: Question answering with knowledge base. In *Procs of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th Intern, Joint Conference on Natural Language Processing* (vol. 1).

Zhang, Y., He, S., Liu, K., Zhao, J. (2016). A joint model for question answering over multiple knowledge bases. In *AAAI* (pp. 3094–3100).

Zou, L., Huang, R., Wang, H., Yu, J.X., He, W., Zhao, D. (2014). Natural language question answering over rdf: a graph data driven approach. In *Procs of the SIGMOD international conference on Management of data* (pp 313–324): ACM.