Survey on Challenges of Question Answering in the Semantic Web

Konrad Höffner, Sebastian Walter, Edgard Marx, Ricardo Usbeck, Jens Lehmann, Axel-Cyrille Ngonga Ngomo

a Leipzig University, Institute of Computer Science, AKSW Group
Augustusplatz 10, D-04109 Leipzig, Germany
E-mail: {hoeffner, marx, lehmann, ngonga, usbeck}@informatik.uni-leipzig.de
b CITEC, Bielefeld University
Inspiration 1, D - 33615 Bielefeld, Germany
E-mail: swalter@techfak.uni-bielefeld.de

Abstract. Semantic Question Answering (SQA) removes two major access requirements to the Semantic Web: the mastery of a formal query language like SPARQL and knowledge of a specific vocabulary. Because of the complexity of natural language, SQA presents difficult challenges and many research opportunities. Instead of a shared effort, however, many essential components are redeveloped, which is an inefficient use of researcher’s time and resources. This survey analyzes 62 different SQA systems, which are systematically and manually selected using predefined inclusion and exclusion criteria, leading to 72 selected publications out of 1960 candidates. We identify common challenges, structure solutions, and provide recommendations for future systems. This work is based on publications from the end of 2010 to July 2015 and is also compared to older but similar surveys.

Keywords: Question Answering, Semantic Web, Survey

1. Introduction

Semantic Question Answering (SQA) is defined by users (1) asking questions in natural language (NL) (2) using their own terminology to which they (3) receive a concise answer generated by querying an RDF knowledge base. Users are thus freed from two major access requirements to the Semantic Web: (1) the mastery of a formal query language like SPARQL and (2) knowledge about the specific vocabularies of the knowledge base they want to query. Since natural language is complex and ambiguous, reliable SQA systems require many different steps. For some of them, like part-of-speech tagging and parsing, mature high-precision solutions exist, but most of the others still present difficult challenges. While the massive research effort has led to major advances, as shown by the yearly Question Answering over Linked Data (QALD) evaluation campaign, it suffers from several problems: Instead of a shared effort, many essential components are redeveloped. While shared practices emerge over time, they are not systematically collected. Furthermore, most systems focus on a specific aspect while the others are quickly implemented, which leads to low benchmark scores and thus undervalues the contribution. This survey aims to alleviate these problems by systematically collecting and structuring methods of dealing with common challenges faced by these approaches. Our contributions are threefold: First, we complement existing work with 72 publications about 62 systems developed from 2010 to 2015. Second, we identify challenges faced by those approaches and collect solutions for them from the 72 publications. Finally, we draw conclusions and make recommendations on how to develop future SQA systems. The structure of the paper is as follows: Section 2 states the methodology used to find and filter surveyed publications. Section 3 compares this work to older, similar surveys as well as eval-
Challenges of Question Answering in the Semantic Web

Data evaluation campaigns and work outside the SQA field. Section 2 introduces the surveyed systems. Section 3 identifies challenges faced by SQA approaches and presents approaches that tackle them. Section 4 summarizes the efforts made to face challenges to SQA and their implication for further development in this area.

2. Methodology

This survey follows a strict discovery methodology; Objective inclusion and exclusion criteria are used to find and restrict publications on SQA.

Inclusion Criteria Candidate articles for inclusion in the survey need to be part of relevant conference proceedings or searchable via Google Scholar (see Table 1). The included papers from the publication search engine Google Scholar are the first 300 results in the chosen timespan (see exclusion criteria) that contain “question answering’ AND (‘Semantic Web’ OR ‘data web’)” in the article including title, abstract and text body. Conference candidates are all publications in our examined time frame in the proceedings of the major Semantic Web Conferences ISWC, ESWC, WWW, NDLB, and the proceedings which contain the annual QALD challenge participants.

Exclusion Criteria Works published before November 2010 or after July 2015 are excluded, as well as those that are not related to SQA, determined in a manual inspection in the following manner: First, proceeding tracks are excluded that clearly do not contain SQA related publications. Next, publications both from proceedings and from Google Scholar are excluded based on their title and finally on their content.

Notable exclusions We exclude the following approaches since they do not fit our definition of SQA (see Section 2): Swoogle [45] is independent of any specific knowledge base but instead builds its own index and knowledge base using RDF documents found by multiple web crawlers. Discovered ontologies are ranked based on their usage intensity and RDF documents are ranked using authority scoring. Swoogle can only find single terms and cannot answer natural language queries and is thus not a SQA system. WolframAlpha is a natural language interface based on the computational platform Mathematica [138] and aggregates a large number of structured sources and a large number of natural language queries and is thus not a SQA system. WolframAlpha is a natural language interface based on the computational platform Mathematica [138] and aggregates a large number of structured sources and a large number of natural language queries and is thus not a SQA system. However, it does not support Semantic Web knowledge bases and the source code and the algorithm is not published. Thus, we cannot identify whether it corresponds to our definition of a SQA system.

Result The inspection of the titles of the Google Scholar results by two authors of this survey led to 153 publications, 39 of which remained after inspecting the full text (see Table 1). The selected proceedings contain 1660 publications, which were narrowed down to 980 by excluding tracks that have no relation to SQA. Based on their titles, 62 of them were selected and inspected, resulting in 33 publications that were categorized and listed in this survey. Table 1 shows the number of publications in each step for each source. In total, 1960 candidates were found using the inclusion criteria in Google Scholar and conference proceedings and then reduced using track names (conference proceedings only, 1280 remaining), then titles (214) and finally the full text, resulting in 72 publications describing 62 distinct SQA systems.

3. Related Work

This section gives an overview of recent QA and SQA surveys and differences to this work, as well as QA and SQA evaluation campaigns, which quantitatively compare systems.

3.1. Other Surveys

QA Surveys Cimiano and Minock [32] present a data-driven problem analysis of QA on the Geobase dataset. The authors identify eleven challenges that QA has to solve and which inspired the problem categories of this survey: question types, language “light” [3], lexical ambiguities, syntactic ambiguities, scope ambiguities, spatial prepositions, adjective modifiers and superlatives, aggregation, comparison and negation operators, non-compositionality, and out of scope [4]. In contrast to our work, they identify challenges by manually inspecting user provided questions instead of existing systems. Mishra and Jain [94] propose eight classification criteria, such as application domain, types of questions and type of data. For each criterion, the different classifications are given along with their advantages, disadvantages and exemplary systems.

---

3. The time before is already covered in Cimiano and Minock [32].

4. Cannot be answered as the information required is not contained in the knowledge base.
Table 1
Sources of publication candidates along with the number of publications in total, after excluding based on conference tracks (I), based on the title (II), and finally based on the full text (selected). Works that are found both in a conference's proceedings and in Google Scholar are only counted once, as selected for that conference. The QALD 2 proceedings are included in ILD 2012, QALD 3 [24] and QALD 4 [133] in the CLEF 2013 and 2014 working notes.

<table>
<thead>
<tr>
<th>Venue</th>
<th>Google Scholar Top 300</th>
<th>All</th>
<th>I</th>
<th>II</th>
<th>Selected</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISWC 2010 [105]</td>
<td>70</td>
<td>70</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>ISWC 2012 [53]</td>
<td>66</td>
<td>66</td>
<td>4</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>ISWC 2013 [4]</td>
<td>72</td>
<td>72</td>
<td>4</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>ISWC 2014 [61]</td>
<td>31</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>WWW 2011 [139]</td>
<td>81</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>WWW 2012 [140]</td>
<td>108</td>
<td>6</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>WWW 2013 [131]</td>
<td>137</td>
<td>137</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>WWW 2014 [142]</td>
<td>84</td>
<td>33</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>WWW 2015 [133]</td>
<td>131</td>
<td>131</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>ESWC 2011 [5]</td>
<td>67</td>
<td>58</td>
<td>3</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>ESWC 2012 [121]</td>
<td>53</td>
<td>43</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>ESWC 2013 [55]</td>
<td>42</td>
<td>34</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>ESWC 2014 [108]</td>
<td>51</td>
<td>31</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>ESWC 2015 [59]</td>
<td>42</td>
<td>42</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>NLDB 2011 [90]</td>
<td>21</td>
<td>21</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>NLDB 2012 [31]</td>
<td>36</td>
<td>36</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>NLDB 2013 [89]</td>
<td>36</td>
<td>36</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>NLDB 2014 [90]</td>
<td>39</td>
<td>30</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>NLDB 2015 [10]</td>
<td>45</td>
<td>10</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>QALD 1 [130]</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>ILD 2012 [122]</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>CLEF 2013 [52]</td>
<td>208</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>CLEF 2014 [27]</td>
<td>160</td>
<td>24</td>
<td>8</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>Σ(conference)</td>
<td>1660</td>
<td>980</td>
<td>61</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>Σ(all)</td>
<td>1960</td>
<td>1280</td>
<td>214</td>
<td>72</td>
<td></td>
</tr>
</tbody>
</table>

SQA Surveys For each participant, problems and their solution strategies are given: Athenikos and Han [8] give an overview of domain specific QA systems for biomedicine. After summarising the state of the art for biomedical QA systems in 2009, the authors describe different approaches from the point of view of medical and biological QA. In contrast to our survey, the authors do not sort the presented approaches by challenges, but by more broader terms such as “Non-semantic knowledge base medical QA systems and approaches” or “Inference-based biological QA systems and approaches”. Lopez et al. [85] present an overview similar to Athenikos and Han [8] but with a wider scope. After defining the goals and dimensions of QA and presenting some related and historic work, the authors summarize the achievements of SQA so far and the challenges that are still open. Another related survey from 2012, Freitas et al. [56], gives a broad overview of the challenges involved in constructing effective query mechanisms for Web-scale data. The authors analyze different approaches, such as Treo [55], for five different challenges: usability, query expressivity, vocabulary-level semantic matching, entity recognition and improvement of semantic tractability. The same is done for architectural elements such as user interaction and interfaces and the impact on these challenges is reported. Lopez et al. [86] analyze the SQA systems of the participants of the QALD 1 and 2 evaluation challenge, see Section 3.2. While there is an overlap in the surveyed approaches between Lopez et al. [86] and our paper, our survey has a broader scope as it also analyzes approaches that do not take part in the QALD challenges.

In contrast to the surveys mentioned above, we do not focus on the overall performance or domain of a system, but on analyzing and categorizing methods that tackle specific problems. Additionally, we build upon the existing surveys and describe the new state of the art systems, which were published after the before mentioned surveys in order to keep track of new research ideas.

3.2. Evaluation Campaigns

Contrary to QA surveys, which qualitatively compare systems, there are also evaluation campaigns, which quantitatively compare them using benchmarks. Those campaigns show how different open-domain QA systems perform on realistic questions on real-world
knowledge bases. This accelerates the evolution of QA in four different ways: First, new systems do not have to include their own benchmark, shortening system development. Second, standardized evaluation allows for better research resource allocation as it is easier to determine, which approaches are worthwhile to develop further. Third, the addition of new challenges to the questions of each new benchmark iteration motivates addressing those challenges. And finally, the competitive pressure to keep pace with the top scoring systems compells emergence and integration of shared best practises. On the other hand, evaluation campaign proceedings do not describe single components of those systems in great detail. By focussing on complete systems, research effort gets spread around multiple components, possibly duplicating existing efforts, instead of being focussed on a single one.

**Question Answering on Linked Data (QALD)** is the most well-known all-purpose evaluation campaign with its core task of open domain SQA on lexicographic facts of DBpedia [62]. Since its inception in 2011, the yearly benchmark has been made progressively more difficult. Additionally, the general core task has been joined by special tasks providing challenges like multilinguality, hybrid (textual and Linked Data) and its newest addition, SQA on statistical data in the form of RDF Data Cubes [73].

**BioASQ** [127,103,9,10] is a benchmark challenge which ran until September 2015 and consists of semantic indexing as well as an SQA part on biomedical data. In the SQA part, systems are expected to be hybrids, returning matching triples as well as text snippets but partial evaluation (text or triples only) is possible as well. The introductory task separates the process into annotation which is equivalent to named entity recognition (NER) and disambiguation (NED) as well as the answering itself. The second task combines these two steps.

**TREC LiveQA**, starting in 2015 [4], gives systems unanswered Yahoo Answers questions intended for other humans. As such, the campaign contains the most realistic questions with the least restrictions, in contrast to the solely factual questions of QALD, BioASQ and TREC’s old QA track [59].

### 3.3. System Frameworks

System frameworks provide an abstraction in which an generic functionality can be selectively changed by additional third-party code. In document retrieval, there are many existing frameworks, such as Lucene[5], Solr[6] and Elastic Search[7]. For SQA systems, however, there is still a lack of tools to facilitate implementation and evaluation process of SQA systems.

Document retrieval frameworks usually split the retrieval process in tree steps (1) query processing, (2) retrieval and (3) ranking. In the (1) query processing step, query analyzers identify documents in the data store. Thereafter, the query is used to (2) retrieve documents that match the query terms resulting from the query processing. Later, the retrieved documents are (3) ranked according to some ranking function, commonly tf-idf [122]. Developing an SQA framework is a hard task because many systems work with a mixture of NL techniques on top of traditional IR systems. Some systems make use of the syntactic graph behind the question [131] to deduce the query intention whereas others, the knowledge graph [117]. There are hybrid systems that to work both on structured and unstructured data [134] or on a combination of systems [63]. Therefore, they contain very peculiar steps. This has led to a new research sub field that focuses on QA frameworks, that is, the design and development of common features for SQA systems.

**openQA** [87] is a modular open-source framework for implementing and instantiating SQA approaches. The framework’s main work-flow consists of four stages (interpretation, retrieval, synthesis, rendering) and adjacent modules (context and service). The adjacent modules are intended to be accessed by any of the components of the main work-flow to share common features to the different modules e.g. cache. The framework proposes the *answer formulation* process in a very likely traditional document retrieval fashion where the query processing and ranking steps are replaced by the more general *Interpretation* and *Synthesis*. The interpretation step comprises all the pre-processing and matching techniques required to deduce the question whereas the synthesis is the process of ranking, merging and confidence estimation required to produce the answer. The authors claims that openQA enables a unification of different architectures and methods.

---

https://lucene.apache.org  
https://solr.apache.org  
https://www.elastic.co  
http://openqa.aksw.org
4. Systems

The 72 surveyed publications describe 62 distinct systems or approaches. The implementation of a SQA system can be very complex and depending on, thus reusing, several known techniques. SQA systems are typically composed of two stages: (1) the query analyzer and (2) retrieval. The query analyzer generates or formats the query that will be used to recover the answer at the retrieval stage. There is a wide variety of techniques that can be applied at the analyzer stage, such as tokenisation, disambiguation, internationalization, logical forms, semantic role labels, question reformulation, coreference resolution, relation extraction and named entity recognition amongst others. For some of those techniques, such as natural language (NL) parsing and part-of-speech (POS) tagging, mature all-purpose methods are available and commonly reused. Other techniques, such as the disambiguating between multiple possible answers candidates, are not available at hand in a domain independent fashion. Thus, high quality solutions can only be obtained by the development of new components. This section exemplifies some of the reviewed systems and their novelties to highlight current research questions, while the next section presents the contributions of all analyzed papers to specific challenges.

Hakimov et al. \[64\] proposes a SQA system using syntactic dependency trees of input questions. The method consists of three main steps: (1) Triple patterns are extracted using the dependency tree and POS tags of the questions. (2) Entities, properties and classes are extracted and mapped to the underlying knowledge base. Recognized entities are disambiguated using page links between all spotted named entities as well as string similarity. Properties are disambiguated by using relational linguistic patterns from PATTY \[97\], which allows a more flexible mapping, such as “die” to \[\text{dbo:deathPlace}\]. Finally, (3) question words are matched to the respective answer type, such as “who” to person, organization or company and “while” to place.

The results are then ranked and the best ranked result is returned as the answer.

PARALEX \[48\] only answers questions for subjects or objects of property-object or subject-property pairs, respectively. It contains phrase to concept mappings in a lexicon that is trained from a corpus of paraphrases, which is constructed from the question-answer site WikiAnswer.\[10\] If one of the paraphrases can be mapped to a query, this query is the correct answer for the paraphrases as well. By mapping phrases between those paraphrases, the linguistic patterns are extended. For example, “what is the \(r\) of \(e\)” leads to “how \(r\) is \(e\)”, so that “What is the population of New York” can be mapped to “How big is NYC”. There is a variety of other systems, such as Bordes et al. \[18\], that make use of paraphrase learning methods and integrate linguistic generalization with knowledge graph biases. They are however not included here as they do query RDF knowledge bases and thus do not fit the inclusion criteria.

Xser \[144\] is based on the observation that SQA contains two independent steps. First, Xser determines the question structure solely based on a phrase level dependency graph and second uses the target knowledge base to instantiate the generated template. For instance, moving to another domain based on a different knowledge base thus only affects parts of the approach so that the conversion effort is lessened.

QuASE \[124\] is a three stage open domain approach based on web search and the Freebase knowledge base.\[11\] First, QuASE uses entity linking, semantic feature construction and candidate ranking on the input question. Then, it selects the documents and according sentences from a web search with a high probability to match the question and presents them as answers to the user.

DEV-NLQ \[123\] is based on lambda calculus and an event-based triple store\[12\] using only triple based retrieval operations. DEV-NLQ claims to be the only QA system able to solve chained, arbitrarily-nested, complex, prepositional phrases.

CubeQA \[73,74\] is a novel approach of SQA over multi-dimensional statistical Linked Data using the RDF Data Cube Vocabulary\[13\], which existing approaches cannot process. Using a corpus of questions with open domain statistical information needs, the au-

---

9URL prefixes are defined in Table 3.

<table>
<thead>
<tr>
<th></th>
<th>URL Prefix</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dbo</td>
<td><a href="http://dbpedia.org/ontology/">http://dbpedia.org/ontology/</a></td>
<td></td>
</tr>
<tr>
<td>dbp</td>
<td><a href="http://dbpedia.org/resource/">http://dbpedia.org/resource/</a></td>
<td></td>
</tr>
<tr>
<td>owl</td>
<td><a href="http://www.w3.org/2002/07/owl#">http://www.w3.org/2002/07/owl#</a></td>
<td></td>
</tr>
</tbody>
</table>

Table 3
URL prefixes used throughout this work.

\[10\] http://wiki.answers.com/
\[11\] https://www.freebase.com/
\[12\] http://www.w3.org/wiki/LargeTripleStores
\[13\] http://www.w3.org/TR/vocab-data-cube/
Challenges of Question Answering in the Semantic Web

Aschauer and Pfister [23, 34, 25] analyze how those questions differ from others, which additional verbalizations are commonly used and how this influences design decisions for SQA on statistical data.

QAKiS [23, 34, 25] queries several multilingual versions of DBpedia at the same time by filling the produced SPARQL query with the corresponding language-dependent properties and classes. Thus, it can retrieve correct answers even in cases of missing information in the language-dependent knowledge base.

Freitas and Curry [53] evaluate a distributional-compositional semantics approach that is independent from manually created dictionaries but instead relies on co-occurring words in text corpora. The vector space over the set of terms in the corpus is used to create a distributional vector space based on the weighted term vectors for each concept. An inverted Lucene index is adapted to the chosen model.

Instead of querying a specific knowledge base, Sun et al. [124] use web search engines to extract relevant text snippets, which are then linked to Freebase, where a ranking function is applied and the highest ranked entity is returned as the answer.

HAWK [134] is the first hybrid source SQA system which processes Linked Data as well as textual information to answer one input query. HAWK uses an eight-fold pipeline comprising part-of-speech tagging, entity annotation, dependency parsing, linguistic pruning heuristics for an in-depth analysis of the natural language input, semantic annotation of properties and classes, the generation of basic triple patterns for each component of the input query as well as discarding queries containing not connected query graphs and ranking them afterwards.

SWIP (Semantic Web intercase using Pattern) [107] generates a pivot query, a hybrid structure between the natural language question and the formal SPARQL target query. Generating the pivot queries consists of three main steps: (1) Named entity identification, (2) Query focus identification and (3) sub query generation. To formalize the pivot queries, the query is mapped to linguistic patterns, which are created by hand from domain experts. If there are multiple applicable linguistic patterns for a pivot query, the user chooses between them.

Hakimov et al. [65] adapt a semantic parsing algorithm to SQA which achieves a high performance but relies on large amounts of training data which is not practical when the domain is large or unspecified.

Several industry-driven SQA-related projects have emerged over the last years. For example, DeepQA of IBM Watson [63], which was able to win the Jeopardy! challenge against human experts.

YodaQA [12] is a modular open source hybrid approach built on top of the Apache UIMA framework [14] that is part of the Brmson platform and is inspired by DeepQA. YodaQA allows easy parallelization and leverage of pre-existing NLP UIMA components by representing each artifact (question, search result, passage, candidate answer) as a separate UIMA CAS. Yoda pipeline is divided in five different stages: (1) Question Analysis, (2) Answer Production, (3) Answer Analysis, (4) Answer Merging and Scoring as well as (5) Successive Refining.

Further, KAIST’s Exobrain [15] project aims to learn from large amounts of data while ensuring a natural interaction with end users. However, it is limited to Korean.

Answer Presentation

Another, important part of SQA systems outside the SQA research challenges is result presentation. Verbose descriptions or plain URIs are uncomfortable for human reading. Entity summarization deals with different types and levels of abstractions.

Cheng et al. [30] proposes a random surfer model extended by a notion of centrality, i.e., a computation of the central elements involving similarity (or relatedness) between them as well as their informativeness. The similarity is given by a combination of the relatedness between their properties and their values.

Ngonga Ngomo et al. [101] present another approach that automatically generates natural language description of resources using their attributes. The rationale behind SPARQL2NL is to verbalize RDF data by applying templates together with the metadata of the schema itself (label, description, type). Entities can have multiple types as well as different levels of hierarchy which can lead to different levels of abstractions. The verbalization of the DBpedia entity 

| http://dbpedia.org/datatype/squareKilometre | can be verbalized as 123 square kilometres.

5. Challenges

In this section, we address seven challenges that have to be faced by state-of-the-art SQA systems. All men-
5.1. Lexical Gap

In a natural language, the same meaning can be expressed in different ways. Natural language descriptions of RDF resources are provided by values of the rdfs:label property (label in the following). While synonyms for the same RDF resource can be modeled using multiple labels for that resource, knowledge bases typically do not contain all the different terms that can refer to a certain entity. If the vocabulary used in a question is different from the one used in the labels of the knowledge base, we call this the lexical gap. Because a question can usually only be answered if every referred concept is identified, bridging this gap significantly increases the proportion of questions that can be answered by a system. Table 4 shows the methods employed by the 72 selected publications for bridging the lexical gap along with examples. As an example of how the lexical gap is bridged outside of SQA, see Lee et al. [80].

### String Normalization and Similarity Functions

Normalizations, such as conversion to lower case or to base forms, such as “é” to “e”, allow matching of slightly different forms and some simple mistakes, such as “Déjà Vu” for “déjà vu”, and are quickly implemented and executed. More elaborate normalizations use natural language processing (NLP) techniques for stemming (both “running” and “ran” to “run”).

If normalizations are not enough, the distance—and its complementary concept, similarity—can be quantified using a similarity function and a threshold. Common examples of similarity functions are Jaro-Winkler, an edit-distance that measures transpositions and n-grams, which compares sets of substrings of length $n$ of two strings. Also, one of the surveyed publications, Zhang et al. [150], uses the largest common substring, both between Japanese and translated English words. However, applying such similarity functions can carry harsh performance penalties. While an exact string match can be efficiently executed in a SPARQL triple pattern, similarity scores generally need to be calculated between a phrase and every entity label, which is infeasible on large knowledge bases [134]. There are however efficient indexes for some similarity functions. For instance, the edit distances of two characters or less can be mitigated by using the fuzzy query implementation of a Lucene Index [18] that implements a Levenshtein Automaton [112]. Furthermore, Ngonga Ngomo [59] provides a different approach to efficiently calculating similarity scores that could be applied to QA. It uses similarity metrics where a triangle inequality holds that allows for a large portion of potential matches to be discarded early in the process. This solution is not as fast as using a Levenshtein Automaton but does not place such a tight limit on the maximum edit distance.

### Automatic Query Expansion

While normalization and string similarity methods match different forms of the same word, they do not recognize synonyms. Synonyms, like design and plan, are pairs of words that, either always or only in a specific context, have the same meaning. In hyper-hyponym-pairs, like chemical process and photosynthesis, the first word is less specific than the second one. These word pairs, taken from lexical databases such as WordNet [22], are used as additional labels in Automatic query expansion (AQE). AQE is commonly used in information retrieval and traditional search engines, as summarized in Carpineto and Romano [28]. These additional surface forms allow for more matches and thus increase recall but lead to mismatches between related words and thus can decrease the precision.

In traditional document-based search engines with high recall and low precision, this trade-off is more common than in SQA. SQA is typically optimized for concise answers and a high precision, since a SPARQL query with an incorrectly identified concept mostly results in a wrong set of answer resources. However, AQE can be used as a backup method in case there is no direct match. One of the surveyed publications is an

---

17In linguistics, the term lexical gap has a different meaning, referring to a word that has no equivalent in another language.

18[http://lucene.apache.org](http://lucene.apache.org)
experimental study [115] that evaluates the impact of AQE on SQA. It has analyzed different lexical and semantic expansion features and used machine learning to optimize weightings for combinations of them. Both lexical and semantic features were shown to be beneficial on a benchmark dataset consisting only of sentences where direct matching is not sufficient.

**Pattern libraries** RDF individuals can be matched from a phrase to a resource with high accuracy using similarity functions and normalization alone. Properties however require further treatment, as (1) they determine the subject and object, which can be in different positions and (2) a single property can be expressed in many different ways, both as a noun and as a verb phrase which may not even be a continuous substring of the question. Because of the complex and varying structure of those linguistic patterns and the required reasoning and knowledge libraries to overcome this issues have been developed.

PATTY [97] detects entities in sentences of a corpus and determines the shortest path between the entities. The path is then expanded with occurring modifiers and stored as a pattern. Thus, PATTY is able to build up a pattern library on any knowledge base with an accompanying corpus.

BOA [61] generates linguistic patterns using a corpus and a knowledge base. For each property in the knowledge base, sentences from a corpus are chosen containing examples of subjects and objects for this particular property. BOA assumes that each resource pair that is connected in a sentence exemplifies another label for this relation and thus generates a pattern from each occurrence of that word pair in the corpus.

PARALEX [48] contains phrase to concept mappings in a lexicon that is trained from a corpus of paraphrases from the QA site WikiAnswers. The advantage is that no manual templates have to be created as they are automatically learned from the paraphrases.

**Entailment** A corpus of already answered questions or linguistic question patterns can be used to infer the answer for new questions. A phrase $A$ is said to entail a phrase $B$, if $B$ follows from $A$. Thus, entailment is directional: Synonyms entail each other, whereas hyper- and hyponyms entail in one direction only: “birds fly” entails “sparrows fly”, but not the other way around. Ou and Zhu [102] generate possible questions for an ontology in advance and identify the most similar match to a user question based on a syntactic and semantic similarity score. The syntactic score is the cosine-similarity of the questions using bag-of-words. The semantic score also includes hypernyms, hyponyms and denormalizations based on WordNet [22]. While the preprocessing is algorithmically simple compared to the complex pipeline of NLP tools, the number of possible questions is expected to grow superlinearly with the size of the ontology so the approach is more suited to specific domain ontologies. Furthermore, the range of possible questions is quite limited which the authors aim to partially alleviate in future work by combining multiple basic questions into a complex question.

**Document Retrieval Models** Blanco et al. [17] adapt entity ranking models from traditional document retrieval algorithms to RDF data. The authors apply BM25 as well as the tf-idf ranking function to an index structure with different text fields constructed from the title, object URIs, property values and RDF inlinks. The proposed adaptation is shown to be both time efficient and qualitatively superior to other state-of-the-art methods in ranking RDF resources.

**Composite Approaches** Elaborate approaches on bridging the lexical gap can have a high impact on the overall runtime performance of an SQA system. This can be partially mitigated by composing methods and executing each following step only if the one before did not return the expected results.

BELA [136] implements four layers. First, the question is mapped directly to the concept of the ontology using the index lookup. Second, the question is mapped based on Levenshtein distance to the ontology, if the Levenshtein distance of a word from the question and a property from an ontology exceed a certain threshold. Third, WordNet is used to find synonyms for a given word. Finally, BELA uses explicit semantic analysis (ESA) Gabrilovich and Markovitch [58]. The evaluation is carried out on the QALD 2 [132] test dataset and shows that the more simple steps, like index lookup and Levenshtein distance, had the most positive influence on answering questions so that many questions can be answered with simple mechanisms.

Park et al. [105] answer natural language questions via regular expressions and keyword queries with a Lucene-based index. Furthermore, the approach uses
Ambiguity is the phenomenon of the same phrase having different meanings; this can be structural and syntactic (like “flying planes”) or lexical and semantic (like “bank”). We distinguish between homonymy, where the same string accidentally refers to different concepts (as in money bank vs. river bank) and polysemy, where the same string refers to different but related concepts (as in bank as a company vs. bank as a building). We distinguish between synonymy and taxonomic relations such as metonymy and hypernymy. In contrast to the lexical gap, which impedes the recall of a SQA system, ambiguity negatively effects its precision. Ambiguity is the flipside of the lexical gap.

This problem is aggravated by the very methods used for overcoming the lexical gap. The more loose the matching criteria become (increase in recall), the more candidates are found which are generally less likely to be correct than closer ones. Disambiguation is the process of selecting one of multiple candidate concepts for an ambiguous phrase. We differentiate between two types of disambiguation based on the source and type of information used to solve this mapping:

- **Corpus-based methods** are traditionally used and rely on counts, often used as probabilities, from unstructured text corpora. Such statistical approaches are based on the distributional hypothesis, which states that “difference of meaning correlates with difference of contextual distribution” [68]. The context of a phrase is identified here as its central characteristic [23]. Common context features used are word co-occurrences, such as left or right neighbours, but also synonyms, hyponyms, POS-tags and the parse tree structure. More elaborate approaches also take advantage of the context outside of the question, such as past queries of the user [119].

- In SQA, **Resource-based methods** exploit the fact that the candidate concepts are RDF resources. Resources are compared using different scoring schemes based of their properties and the connections between them. The assumption is that high score between all the resources chosen in the mapping implies a higher probability of those resources being related, and that this implies a higher probability of those resource being correctly chosen. RVT [62] uses Hidden Markov Models (HMM) to select the proper ontological triples according to the graph nature of DBpedia. CASIA [70] employs Markov Logic Networks (MLN): First-order logic statements are assigned a numerical penalty, which is used to define hard constraints, like “each phrase can map to only one resource”, alongside soft constraints, like “the larger the semantic similarity is between two resources, the higher the chance is that they are connected by a relation in the question”. Underspecification [128] discards certain combinations of possible meanings before the time consuming querying step, by combining restrictions for each meaning. Each term is mapped to a Dependency-based Underspecified Discourse Representation Structure (DUDE [37]), which captures its possible meanings along with their class restrictions. Treo [55,54] performs entity recognition and disambiguation using Wikipedia-based semantic relatedness and spreading activation. Semantic relatedness calculates similarity values between pairs of RDF resources. Determining semantic relatedness between entity candidates associated to words in a sentence allows to find the most probable entity by maximizing the total relatedness. EasyESA [29] is based on distributional semantic models which allow to represent an entity by a vector of target words and thus compresses its representation. The distributional semantic models allow to bridge the lexical gap and resolve ambiguity by avoiding the explicit structures of RDF-based entity descriptions for entity linking and relatedness. gAnswer [76] tackles ambiguity with RDF fragments, i.e., star-like RDF subgraphs. The number of connections between the fragments of the resource candidates is then used to score and select them. Wikimantic [19] can be used to disambiguate short questions or even sentences. It uses Wikipedia article interlinks for a generative model, where the probability of an article to generate a term is set to the terms relative occurrence in the article. Disambiguation is then an optimization problem to locally maximize each article’s (and thus DBpedia resource’s) term probability along with a global ranking method. Shekarpour et al. [113,116] disambiguate resource candidates using segments consisting of one or more words from a keyword query. The aim is to maximize the high textual similarity of keywords to resources along with relatedness between the resources (classes, properties and entities). The problem is cast as a Hidden Markov Model (HMM) with the states representing the set of candidate resources extended by OWL reasoning. The transition probabilities are based on the shortest path between the resources. The Viterbi algorithm generates an optimal path though the HMM that is used for disambiguation. DEANNA [145,146] manages phrase detection, entity recognition and entity disambiguation.
by formulating the SQA task as an integer linear programming (ILP) problem. It employs semantic coherence which measures co-occurrence of resources in the same context. DEANNA constructs a disambiguation graph, which encodes the selection of candidates for resources and properties. The chosen objective function maximizes the combined similarity while constraints guarantee that the selections are valid. The resulting problem is NP-hard but it is efficiently solvable in approximations by existing ILP solvers. The follow-up approach [47] uses DBpedia and Yago with a mapping of input queries to semantic relations based on text search. At QALD 2, it outperformed almost every other system on factoid questions and every other system on list questions. However, the approach requires detailed textual descriptions of entities and only creates basic graph pattern queries. LOD-Query [114] is a keyword-based SQA system that tackles both ambiguity and the lexical gap by selecting candidate concepts based on a combination of a string similarity score and the connectivity degree. The string similarity is the normalized edit distance between a labels and a keyword. The connectivity degree of a concept is approximated by the occurrence of that concept in all the triples of the knowledge base. Pomelo [66] answers biomedical questions on the combination of Drugbank, Diseases and Sider using owl:sameAs links between them. Properties are disambiguated using predefined rewriting rules which are categorized by context. Rani et al. [110] use fuzzy logic co-clustering algorithms to retrieve documents based on their ontology similarity. Possible senses for a word are assigned a probability depending on the context. Zhang et al. [150] translates RDF resources to the English DBpedia. It uses feedback learning in the disambiguation step to refine the resource mapping.

Instead of trying to resolve ambiguity automatically, some approaches let the user clarify the exact intent, either in all cases or only for ambiguous phrases: SQUALL [50,51] defines controlled, English-based vocabulary that is enhanced with knowledge from a given triple store. While this ideally results in a high performance, it moves the problem of the lexical gap and disambiguation fully to the user. As such, it covers a middle ground between SPARQL and full-fledged SQA with the author’s intent that learning the grammatical structure of this proposed language is easier for a non-expert than to learn SPARQL. A cooperative approach that places less of a burden on the user is proposed in [89], which transforms the question into a discourse representation structure and starts a dialogue with the user for all occurring ambiguities. CrowdQ [42] is a SQA system that decomposes complex queries into simple parts (keyword queries) and uses crowdsourcing for disambiguation. It avoids excessive usage of crowd resources by creating general templates as an intermediate step. FREyA (Feedback, Refinement and Extended Vocabulary Aggregation) [46] represents phrases as potential ontology concepts which are identified by heuristics on the syntactic parse tree. Ontology concepts are identified by matching their labels with phrases from the question without regarding its structure. A consolidation algorithm then matches both potential and ontology concepts. In case of ambiguities, feedback from the user is asked. Disambiguation candidates are created using string similarity in combination with WordNet synonym detection. The system learns from the user selections, thereby improving the precision over time. TBLS [131] uses both an domain independent and a domain dependent lexicon so that it performs well on specific topic but is still adaptable to a different domain. It uses AutoSPARQL [81] to refine the learned SPARQL using the QTL algorithm for supervised machine learning. The user marks certain answers as correct or incorrect and triggers a refinement. This is repeated until the user is satisfied with the result. An extension of TBLS is DEQA [83], which combines Web extraction with OXPath [52], interlinking with LIMES [100] and SQA with TBLS. It can thus answer complex questions about objects which are only available as HTML. Another extension of TBLS is ISOFT [104], which uses explicit semantic analysis to help bridging the lexical gap. NL-Graphs [47] combines SQA with an interactive visualization of the graph of triple patterns in the query which is close to the SPARQL query structure yet still intuitive to the user. Users that find errors in the query structure can either reformulate the query or modify the query graph. KOIOS [15] answers queries on natural environment indicators and allows the user to refine the answer to a keyword query by faceted search. Instead of relying on a given ontology, a schema index is generated from the triples and then connected with the keywords of the query. Ambiguity is resolved by user feedback on the top ranked results.

A different way to restrict the set of answer candidates and thus handle ambiguity is to determine the expected answer type of a factual question. The standard approach to determine this type is to identify the focus of the question and to map this type to an ontology class. In the example “Which books are written by Dan Brown?”, the focus is “books”, which is
mapped to \texttt{dbo:Book}. There is however a long tail of rare answer types that are not as easily alignable to an ontology, which, for instance, Watson \cite{b4} tackles using the TyCor \cite{b79} framework for type coercion. Instead of the standard approach, candidates are first generated using multiple interpretations and then selected based on a combination of scores. Besides trying to align the answer type directly, it is coerced into other types by calculating the probability of an entity of class A to also be in class B. DBpedia, Wikipedia and WordNet are used to determine link anchors, list memberships, synonyms, hyper- and hyponyms. The follow-up \cite{b137} compares two different approaches for answer typing. Type-and-Generate (TaG) approaches restrict candidate answers to the expected answer types using predictive annotation, which requires manual analysis of a domain. TyCor on the other hand employs multiple strategies using generate-and-type (GaT), which generates all answers regardless of answer type and tries to coerce them into the expected answer type. Experimental results hint that GaT outperforms TaG when accuracy is higher than 50%. The significantly higher performance of TyCor when using GaT is explained by its robustness to incorrect candidates while there is no recovery from excluded answers from TaG.

5.4. Complex Queries

Simple questions can most often be answered by translation into a set of simple triple patterns. Problems arise when several facts have to be found out, connected and then combined. Queries may also request a specific result order or results that are aggregated or filtered.

YAGO-QA \cite{b11} allows nested queries when the subquery has already been answered, for example “Who is the governor of the state of New York?” after “What is the state of New York?” YAGO-QA extracts facts from Wikipedia (categories and infoboxes), WordNet and GeoNames. It contains different surface forms such as abbreviations and paraphrases for named entities.

PYTHIA \cite{b129} is an ontology-based SQA system with an automatically build ontology-specific lexicon. Due to the linguistic representation, the system is able to answer natural language question with linguistically more complex queries, involving quantifiers, numerals, comparisons and superlatives, negations and so on.

IBM Watson \cite{b63} handles complex questions by first determining the focus element, which represents the searched entity. The information about the focus element is used to predict the lexical answer type and thus restrict the range of possible answers. This approach allows for indirect questions and multiple sentences.

Shekarpour et al. \cite{b113,b116}, as mentioned in Section 5.2, propose a model that use a combination of knowledge base concepts with a HMM model to handle complex queries.

Intui2 \cite{b43} is an SQA system based on DBpedia based on synfragments which map to a subtree of the syntactic parse tree. Semantically, a synfragment is a minimal span of text that can be interpreted as an RDF triple or complex RDF query. Synfragments interoperate with their parent synfragment by combining all combinations of child synfragments, ordered by syntactic and semantic characteristics. The authors assume that an interpretation of a question in any RDF query language can be obtained by the recursively interpretation of its synfragments. Intui3 \cite{b44} replaces self-made components with robust libraries such as the neural networks-based NLP toolkit SENNA and the DBpedia Lookup service. It drops the parser determined interpretation combination method of its predecessor that suffered from bad sentence parses and instead uses a fixed order right-to-left combination.

GETARUNS \cite{b41} first creates a logical form out of a query which consists of a focus, a predicate and arguments. The focus element identifies the expected answer type. For example, the focus of “Who
is the major of New York?” is “person”, the predicate “be” and the arguments “major of New York”. If no focus element is detected, a yes/no question is assumed. In the second step, the logical form is converted to a SPARQL query by mapping elements to resources via label matching. The resulting triple patterns are then split up again as properties are referenced by unions over both possible directions, as in 
\[\{?x \text{ ?p ?o} \} \cup \{?o \text{ ?p ?x}\}\]  
because the direction is not known beforehand. Additionally, there are filters to handle additional restrictions which cannot be expressed in a SPARQL query, such as “Who has been the 5th president of the USA”.

5.5. Distributed Knowledge

If concept information—which is referred to in a query—is represented by distributed RDF resources, information needed for answering it may be missing if only a single one or not all of the knowledge bases are found. In single datasets with a single source, such as DBpedia, however, most of the concepts have at most one corresponding resource. In case of combined datasets, this problem can be dealt with by creating sameAs, equivalentClass or equivalentProperty links, respectively. However, interlinking while answering a semantic query is a separate research area and thus not covered here.

Some questions are only answerable with multiple knowledge bases and we assume already created links for the sake of this survey. The ALOQUUS [78] system tackles this problem by using the PROTON [37] upper level ontology first to phrase the queries. The ontology is then aligned to those of other knowledge bases using the BLOOMS [77] system. Complex queries are decomposed into separately handled subqueries after coreference [24] are extracted and substituted. Finally, these alignments are used to execute the query on the target systems. In order to improve the speed and quality of the results, the alignments are filtered using a threshold on the confidence measure.

Herzig et al. [71] search for entities and consolidate results from multiple knowledge bases. Similarity metrics are used both to determine and rank results candidates of each datasource and to identify matches between entities from different datasources.

5.6. Procedural, Temporal and Spatial Questions

Procedural Questions  Factual, list and yes-no questions are easiest to answer as they conform directly to SPARQL queries using SELECT and ASK. Others, such as why (causal) or how (procedural) questions require more additional processing. Procedural QA can currently not be solved by SQA, since, to the best of our knowledge, there are no existing knowledge bases that contain procedural knowledge. While it is not an SQA system, we describe the document-retrieval based KOMODO [26] to motivate further research in this area. Instead of an answer sentence, KOMODO returns a Web page with step-by-step instructions on how to reach the goal specified by the user. This reduces the problem difficulty as it is much easier to find a Web page which contains instructions on how to, for example, assemble an “Ikea Billy bookcase” than it would be to extract, parse and present the required steps to the user. Additionally, there are arguments explaining reasons for taking a step and warnings against deviation. Instead of extracting the sense of the question using an RDF knowledge base, KOMODO submits the question to a traditional search engine. The highest ranked returned pages are then cleaned and procedural text is identified using statistical distributions of certain POS tags.

In basic RDF, each fact, which is expressed by a triple, is assumed to be true, regardless of circumstances. In the real world and in natural language however, the truth value of many statements is not a constant but a function of either or both the location or time.

Temporal Questions  Tao et al. [126] answer temporal question on clinical narratives. They introduce the Clinical Narrative Temporal Relation Ontology (CNTRO), which is based on Allen’s Interval Based Temporal Logic [5] but allows usage of time instants as well as intervals. This allows inferring the temporal relation of events from those of others, for example by using the transitivity of before and after. In CNTRO, measurement, results or actions done on patients are modeled as events whose time is either absolutely specified in date and optionally time of day or alternatively in relations to other events and times. The framework also includes an SWRL [75] based reasoner that can deduce additional time information. This allows the detection of possible causalties, such as between a therapy for a disease and its cure in a patient.

Melo et al. [89] propose to include the implicit temporal and spatial context of the user in a dialog in order
to resolve ambiguities. It also includes spatial, temporal and other implicit information.

QALL-ME [49] is a multilingual framework based on description logics and uses the spatial and temporal context of the question. If this context is not explicitly given, the location and time are of the user posing the question are added to the query. This context is also used to determine the language used for the answer, which can differ from the language of the question.

Spatial Questions In RDF, a location can be expressed as 2-dimensional geocoordinates with latitude and longitude, while three-dimensional representations (e.g. with additional height) are not supported by the most often used schema. Alternatively, spatial relationships can be modeled which are easier to answer as users typically ask for relationships and not exact geocoordinates.

Younis et al. [149] employ an inverted index for named entity recognition that enriches semantic data with spatial relationships such as crossing, inclusion and nearness. This information is then made available for SPARQL queries.

5.7. Templates

For complex questions, where the resulting SPARQL query contains more than one basic graph pattern, sophisticated approaches are required to capture the structure of the underlying query. Current research follows two paths, namely (1) template based approaches, which map input questions to either manually or automatically created SPARQL query templates or (2) template-free approaches that try to build SPARQL queries based on the given syntactic structure of the input question.

For the first solution, many (1) template-driven approaches have been proposed like TBSL [131] or SINA [13][136]. Furthermore, Casia [69] generates the graph pattern templates by using the question type, named entities and POS tags techniques. The generated graph patterns are then mapped to resources using WordNet, PATTY and similarity measures. Finally, the possible graph pattern combinations are used to build SPARQL queries. The system focuses in the generation of SPARQL queries that do not need filter conditions, aggregations and superlatives.


Damova et al. [58] return well formulated natural language sentences that are created using a template with optional parameters for the domain of paintings. Between the input query and the SPARQL query, the system places the intermediate step of a multilingual description using the Grammatical Framework [111], which enables the system to support 15 languages.

Rahoman and Ichise [109] propose a template based approach using keywords as input. Templates are automatically constructed from the knowledge base.

However, (2) template-free approaches require additional effort of making sure to cover every possible basic graph pattern [134]. Thus, only a few SQA systems tackle this approach so far.

Xser [134] first assigns semantic labels, i.e., variables, entities, relations and categories, to phrases by casting them to a sequence labelling pattern recognition problem which is then solved by a structured perceptron. The perceptron is trained using features including n-grams of POS tags, NER tags and words. Thus, Xser is capable of covering any complex basic graph pattern.

Going beyond SPARQL queries is TPSM, the open domain Three-Phases Semantic Mapping [60] framework. It maps natural language questions to OWL queries using Fuzzy Constraint Satisfaction Problems. Constraints include surface text matching, preference of POS tags and the similarity degree of surface forms. The set of correct mapping elements acquired using the FCSP-SM algorithm is combined into a model using predefined templates.

An extension of gAnswer [151] (see Section 5.2) is based on question understanding and query evaluation. First, their approach uses a relation mining algorithm to find triple patterns in queries as well as relation extraction, POS-tagging and dependency parsing. Second, the approach tries to find a matching subgraph for the extracted triples and scores them based on a confidence score. Finally, the top-k subgraph matches are returned. Their evaluation on QALD 3 shows that mapping NL questions to graph pattern is not as powerful as generating SPARQL (template) queries with respect to aggregation and filter functions needed to answer several benchmark input questions.

6. Conclusion

In this survey, we analyzed 62 systems and their contributions to seven challenges for SQA systems. SQA is
Challenges of Question Answering in the Semantic Web

an active research field with many existing and diverse approaches covering a multitude of research challenges, domains and knowledge bases.

We only cover QA on the Semantic Web, that is, approaches that retrieve resources as Linked Data from RDF knowledge bases. As similar challenges are faced by QA unrelated to the Semantic Web, we refer to Section 3. We choose to not go into detail for approaches that do not retrieve resources from RDF knowledge bases. Moreover, our consensus can be found in Table 6 for best practices. The upcoming HOBBIT project will clarify, which modules can be aligned with state-of-the-art performance and will quantify the impact of those modules. To cover the field of SQA in depth, we excluded works solely about similarity and paraphrases. The existence of common SQA challenges implies that a unifying architecture can improve the precision as well as increase the number of answered questions. Research into such architectures, includes openQA [87], QAQA [148], QALLME [49] and QANUS [98] (see Section 3.3). Our goal, however, is not to quantify submodule performance or interplay. That will be the task of upcoming projects of large consortiums. A new community is forming in that field and did not find a satisfying solution yet. In this section, we discuss each of the seven research challenges and give a short overview of already established as well as future research directions per challenge, see Table 6.

Overall, the authors of this survey cannot observe a research drift to any of the challenges. The number of publications in a certain research challenge does not decrease significantly, which can be seen as an indicator that none of the challenges is solved yet – see Table 5. Naturally, since only a small number of publications addressed each challenge in a given year, one cannot draw statistically valid conclusions. The challenges proposed by Cimiano and Minock and reduced within this survey appear to be still valid.

Bridging the (1) lexical gap has to be tackled by every SQA system in order to retrieve results with a high recall. For named entities, this is commonly achieved using a combination of the reliable and mature natural language processing algorithms for string similarity and either stemming or lemmatization, see Table 6. Automatic Query Expansion (AQE), for example with WordNet synonyms, is prevalent in information retrieval but only rarely used in SQA. Despite its potential negative effects on precision, we consider it a net benefit to SQA systems. Current SQA systems duplicate already existing efforts or fail to decide on the right technique. Thus, reusable libraries to lower the entrance effort to SQA systems are needed. Mapping to RDF properties from verb phrases is much harder, as they show more variation and often occur at multiple places of a question. Pattern libraries, such as BOA [61], can improve property identification, however they are still an active research topic and are specific to a knowledge base.

The next challenge, (2) ambiguity is addressed by the majority of the publications but the percentage does not increase over time, presumably because of use cases

---

Table 5
Number of publications per year per addressed challenge. Percentages are given for the fully covered years 2011–2014 separately and for the whole covered timespan, with 1 decimal place. For a full list, see Table 7.

<table>
<thead>
<tr>
<th>Year</th>
<th>Total</th>
<th>Lexical Gap</th>
<th>Ambiguity</th>
<th>Multilingualism</th>
<th>Complex Operators</th>
<th>Distributed Knowledge</th>
<th>Procedural, Temporal or Spatial Templates</th>
</tr>
</thead>
<tbody>
<tr>
<td>2010</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2011</td>
<td>16</td>
<td>11</td>
<td>12</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2012</td>
<td>14</td>
<td>6</td>
<td>7</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2013</td>
<td>20</td>
<td>18</td>
<td>12</td>
<td>2</td>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2014</td>
<td>13</td>
<td>7</td>
<td>8</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2015</td>
<td>6</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>all</td>
<td>70</td>
<td>46</td>
<td>42</td>
<td>6</td>
<td>12</td>
<td>4</td>
<td>6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Year</th>
<th>2011</th>
<th>68.8</th>
<th>75.0</th>
<th>6.3</th>
<th>18.8</th>
<th>6.3</th>
<th>12.5</th>
<th>12.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>2012</td>
<td>42.9</td>
<td>50.0</td>
<td>7.1</td>
<td>14.3</td>
<td>7.1</td>
<td>7.1</td>
<td>28.6</td>
<td></td>
</tr>
<tr>
<td>2013</td>
<td>85.0</td>
<td>60.0</td>
<td>10.0</td>
<td>25.0</td>
<td>5.0</td>
<td>5.0</td>
<td>25.0</td>
<td></td>
</tr>
<tr>
<td>2014</td>
<td>53.8</td>
<td>61.5</td>
<td>7.7</td>
<td>15.4</td>
<td>7.7</td>
<td>7.7</td>
<td>0.0</td>
<td></td>
</tr>
</tbody>
</table>

| all  | 65.7 | 60.0 | 8.6  | 17.1| 5.7  | 8.6 | 15.7 |

---

27. https://www.w3.org/community/nli/
29. Synonyms and other related words almost never have exactly the same meaning.
Table 6
Established and actively researched as well as envisioned techniques for solving each challenge.

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Established</th>
<th>Future</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lexical Gap</td>
<td>stemming, lemmatization, string similarity, synonyms, vector space model, indexing, pattern libraries, explicit semantic analysis</td>
<td>combined efforts, reuse of libraries</td>
</tr>
<tr>
<td>Ambiguity</td>
<td>user information (history, time, location), underspecification, machine learning, spreading activation, semantic similarity, crowdsourcing, Markov Logic Network</td>
<td>holistic, knowledge-base aware systems</td>
</tr>
<tr>
<td>Multilingualism</td>
<td>translation to core language, language-dependent grammar</td>
<td>usage of multilingual knowledge bases</td>
</tr>
<tr>
<td>Complex Operators</td>
<td>reuse of former answers, syntactic tree-based formulation, answer type orientation, HMM, logic</td>
<td>non-factual questions, domain-independence</td>
</tr>
<tr>
<td>Distributed Knowledge and</td>
<td>temporal logic</td>
<td>domain specific adaptors, procedural SQA</td>
</tr>
<tr>
<td>Procedural, Temporal, Spatial</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Templates</td>
<td>fixed SPARQL templates, template generation, syntactic tree based generation</td>
<td>complex questions</td>
</tr>
</tbody>
</table>

with small knowledge bases, where its impact is minuscule. For systems intended for longtime usage by the same persons, we regard as promising the integration of previous questions, time and location, as is already common in web of document search engines. There is a variety of established disambiguation methods, which use the context of a phrase to determine the most likely RDF resource, some of which are based on unstructured text collections and others on RDF resources. As we could make out no clear winner, we recommend system developers to make their decisions based on the resources (such as query logs, ontologies, thesauri) available to them. Many approaches reinvent disambiguation efforts and thus—like for the lexical gap—holistic, knowledge-base aware, reusable systems are needed to facilitate faster research.

Despite its inclusion since QALD 3 and following, publications dealing with (3) multilingualism remain a small minority. Automatic translation of parts of or the whole query requires the least development effort, but suffers from imperfect translations. A higher quality can be achieved by using components, such as parsers and synonym libraries, for multiple languages. A possible future research direction is to make use of various language versions at once to use the power of a unified graph [34]. For instance, DBpedia [34] provides a knowledge base in more than 100 languages, which could form the base of a multilingual SQA system.

Complex operators (4) seem to be used only in specific tasks or factual questions. Most systems either use the syntactic structure of the question or some form of knowledge-base aware logic. Future research will be directed towards domain-independence as well as non-factual queries.

Approaches using (5) distributed knowledge as well as those incorporating (6) procedural, temporal and spatial data remain niches. Procedural SQA does not exist yet as present approaches return unstructured text in the form of already written step-by-step instructions. While we consider future development of procedural SQA as feasible with the existing techniques, as far as we know there is no RDF vocabulary for and knowledge base with procedural knowledge yet.

The (7) templates challenge which subsumes the question of mapping a question to a query structure is still unsolved. Although the development of template based approaches seems to have decreased in 2014, presumably because of their low flexibility on open domain tasks, this still presents the fastest way to develop a novel SQA system but the limitation to simple query structures has yet to be overcome.

Future research should be directed at more modularization, automatic reuse, self-wiring and encapsulated modules with their own benchmarks and evaluations. Thus, novel research field can be tackled by reusing already existing parts and focusing on the research core problem itself. A step in this direction is QANARY [20], which describes how to modularize QA systems by providing a core QA vocabulary against which existing vocabularies are bound. Another research direction are SQA systems as aggregators or framework for other systems or algorithms to benefit of the set of existing approaches. Furthermore, benchmarking will move to single algorithmic modules instead of benchmarking a system as a whole.
Challenges of Question Answering in the Semantic Web

The target of local optimization is benchmarking a process at the individual steps, but global benchmarking is still needed to measure the impact of error propagation across the chain. A Turing test-like spirit would suggest that the latter is more important, as the local measure are never fully representative. Additionally, we foresee the move from factual benchmarks over common sense knowledge to more domain specific questions without purely factual answers. Thus, there is a movement towards multilingual, multi-knowledge-source SQA systems that are capable of understanding noisy, human natural language input.

Acknowledgements This work has been supported by the FP7 project GeoKnow (GA No. 318159), the BMWI Project SAKE (Project No. 01MD15006E) and by the Eurostars projects DIESEL (E!9367) and QAMEL (E!9725) as well as the European Union’s H2020 research and innovation action HOBBIT (GA 688227).
Table 7: Surveyed publications from November 2010 to July of 2015, inclusive, along with the challenges they explicitly address and the approach or system they belong to. Additionally annotated is the use of intermediate templates. In case the system or approach is not named in the publication, a name is generated using the last name of the first author and the year of the first included publication.

<table>
<thead>
<tr>
<th>Publication</th>
<th>System or Approach</th>
<th>Year</th>
<th>Lexical Gap</th>
<th>Ambiguity</th>
<th>Multilingualism</th>
<th>Complex Operators</th>
<th>Distributed Knowledge</th>
<th>Procedural, Temporal or Spatial Templates</th>
<th>QALD 1</th>
<th>QALD 2</th>
<th>QALD 3</th>
<th>QALD 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tao et al. [126]</td>
<td>Tao10</td>
<td>2010</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adolphs et al. [1]</td>
<td>YAGO-QA</td>
<td>2011</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Blanco et al. [17]</td>
<td>Blanco11</td>
<td>2011</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Canitrot et al. [26]</td>
<td>KOMODO</td>
<td>2011</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Damljanovic et al. [36]</td>
<td>FREyA</td>
<td>2011</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ferrandez et al. [49]</td>
<td>QALL-ME</td>
<td>2011</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Freitas et al. [55]</td>
<td>Treo</td>
<td>2011</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gao et al. [60]</td>
<td>TPSM</td>
<td>2011</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kalyanpur et al. [79]</td>
<td>Watson</td>
<td>2011</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Melo et al. [88]</td>
<td>Melo11</td>
<td>2011</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Moussa and Abdel-Kader [95]</td>
<td>QASYO</td>
<td>2011</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ou and Zhu [102]</td>
<td>Ou11</td>
<td>2011</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shen et al. [119]</td>
<td>Shen11</td>
<td>2011</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unger and Cimiano [128]</td>
<td>Pythia</td>
<td>2011</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unger and Cimiano [129]</td>
<td>Pythia</td>
<td>2011</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bicer et al. [15]</td>
<td>KOIOS</td>
<td>2011</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Freitas et al. [54]</td>
<td>Treo</td>
<td>2011</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Joshi et al. [78]</td>
<td>ALOQUUS</td>
<td>2012</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lehmann et al. [83]</td>
<td>DEQA</td>
<td>2012</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yahya et al. [145]</td>
<td>DEANNA</td>
<td>2012</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yahya et al. [146]</td>
<td>DEANNA</td>
<td>2012</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shekarpour et al. [113]</td>
<td>SINA</td>
<td>2012</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unger et al. [131]</td>
<td>TBSL</td>
<td>2012</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Walter et al. [136]</td>
<td>BELA</td>
<td>2012</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Younis et al. [149]</td>
<td>Younis12</td>
<td>2012</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Welty et al. [137]</td>
<td>Watson</td>
<td>2012</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Elbedweihy et al. [46]</td>
<td>Elbedweihy12</td>
<td>2012</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cabrio et al. [22]</td>
<td>QAKiS</td>
<td>2012</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Demartini et al. [42]</td>
<td>CrowdQ</td>
<td>2013</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Year</td>
<td>Name</td>
<td>Year</td>
<td>Title</td>
<td>Year</td>
<td>Name</td>
<td>Year</td>
<td>Title</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>---------------------</td>
<td>------</td>
<td>------------------------------</td>
<td>------</td>
<td>---------------------</td>
<td>------</td>
<td>------------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2013</td>
<td>Aggarwal et al.</td>
<td>2013</td>
<td>GermanNLI</td>
<td>2013</td>
<td>Dima</td>
<td>2013</td>
<td>Intui2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2013</td>
<td>Deines and Krechel</td>
<td>50</td>
<td>SQUALL2SPARQL</td>
<td>2013</td>
<td>Giannone et al.</td>
<td>62</td>
<td>RTV</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2013</td>
<td>Dima</td>
<td>43</td>
<td>Hakimov13</td>
<td>2013</td>
<td>He et al.</td>
<td>64</td>
<td>CASIA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2013</td>
<td>Fader et al.</td>
<td>48</td>
<td>CASIA</td>
<td>2013</td>
<td>Herzig et al.</td>
<td>71</td>
<td>CRM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2013</td>
<td>Ferre</td>
<td>50</td>
<td>SQUALL2SPARQL</td>
<td>2013</td>
<td>Hakimov et al.</td>
<td>64</td>
<td>CASIA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2013</td>
<td>Fader et al.</td>
<td>48</td>
<td>SQUALL2SPARQL</td>
<td>2013</td>
<td>He et al.</td>
<td>64</td>
<td>CASIA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2013</td>
<td>Giannone et al.</td>
<td>62</td>
<td>SQUALL2SPARQL</td>
<td>2013</td>
<td>Herzig et al.</td>
<td>71</td>
<td>CRM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2013</td>
<td>Hakimov et al.</td>
<td>64</td>
<td>SQUALL2SPARQL</td>
<td>2013</td>
<td>Huang and Zou</td>
<td>76</td>
<td>gAnswer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2013</td>
<td>He et al.</td>
<td>64</td>
<td>SQUALL2SPARQL</td>
<td>2013</td>
<td>Shekarpour et al.</td>
<td>116</td>
<td>SINA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2013</td>
<td>He et al.</td>
<td>64</td>
<td>SQUALL2SPARQL</td>
<td>2013</td>
<td>Shekarpour et al.</td>
<td>115</td>
<td>SINA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2013</td>
<td>He et al.</td>
<td>64</td>
<td>SQUALL2SPARQL</td>
<td>2013</td>
<td>Shekarpour et al.</td>
<td>114</td>
<td>SINA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2013</td>
<td>Pradel et al.</td>
<td>107</td>
<td>SQUALL2SPARQL</td>
<td>2013</td>
<td>Rahoman and Ichise</td>
<td>110</td>
<td>SINA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2013</td>
<td>Rahoman and Ichise</td>
<td>110</td>
<td>SQUALL2SPARQL</td>
<td>2013</td>
<td>Shekarpour et al.</td>
<td>116</td>
<td>SINA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2013</td>
<td>Shekarpour et al.</td>
<td>116</td>
<td>SQUALL2SPARQL</td>
<td>2013</td>
<td>Shekarpour et al.</td>
<td>115</td>
<td>SINA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2013</td>
<td>Shekarpour et al.</td>
<td>115</td>
<td>SQUALL2SPARQL</td>
<td>2013</td>
<td>Shekarpour et al.</td>
<td>114</td>
<td>SINA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2013</td>
<td>Delmonte</td>
<td>41</td>
<td>GETARUNS</td>
<td>2013</td>
<td>Cojan et al.</td>
<td>134</td>
<td>QAKIS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2013</td>
<td>Cojan et al.</td>
<td>134</td>
<td>GETARUNS</td>
<td>2013</td>
<td>Yahya et al.</td>
<td>147</td>
<td>SPOX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2013</td>
<td>Yahya et al.</td>
<td>147</td>
<td>GETARUNS</td>
<td>2013</td>
<td>Zhang et al.</td>
<td>150</td>
<td>Kawamura13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2014</td>
<td>Carvalho et al.</td>
<td>29</td>
<td>GETARUNS</td>
<td>2013</td>
<td>Rani et al.</td>
<td>110</td>
<td>SINA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2014</td>
<td>Rani et al.</td>
<td>110</td>
<td>GETARUNS</td>
<td>2013</td>
<td>Zou et al.</td>
<td>151</td>
<td>SINA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2014</td>
<td>Zou et al.</td>
<td>151</td>
<td>GETARUNS</td>
<td>2013</td>
<td>Stewart</td>
<td>123</td>
<td>DEV-NLQ</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2014</td>
<td>Stewart</td>
<td>123</td>
<td>GETARUNS</td>
<td>2013</td>
<td>Höflner and Lehmann</td>
<td>73</td>
<td>CubeQA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2014</td>
<td>Höflner and Lehmann</td>
<td>73</td>
<td>GETARUNS</td>
<td>2013</td>
<td>Cabrio et al.</td>
<td>25</td>
<td>QAKIS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2014</td>
<td>Cabrio et al.</td>
<td>25</td>
<td>GETARUNS</td>
<td>2013</td>
<td>Freitas and Curry</td>
<td>53</td>
<td>QAKIS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2014</td>
<td>Freitas and Curry</td>
<td>53</td>
<td>GETARUNS</td>
<td>2013</td>
<td>Dima</td>
<td>44</td>
<td>Intui3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2014</td>
<td>Dima</td>
<td>44</td>
<td>GETARUNS</td>
<td>2013</td>
<td>Hamon et al.</td>
<td>60</td>
<td>POMELO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2014</td>
<td>Hamon et al.</td>
<td>60</td>
<td>GETARUNS</td>
<td>2013</td>
<td>Park et al.</td>
<td>104</td>
<td>ISOFT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2014</td>
<td>Park et al.</td>
<td>104</td>
<td>GETARUNS</td>
<td>2013</td>
<td>He et al.</td>
<td>70</td>
<td>CASIA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2014</td>
<td>He et al.</td>
<td>70</td>
<td>GETARUNS</td>
<td>2013</td>
<td>Xu et al.</td>
<td>144</td>
<td>Xser</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2014</td>
<td>Xu et al.</td>
<td>144</td>
<td>GETARUNS</td>
<td>2013</td>
<td>Elbedwehly et al.</td>
<td>47</td>
<td>NL-Graphs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2014</td>
<td>Elbedwehly et al.</td>
<td>47</td>
<td>GETARUNS</td>
<td>2013</td>
<td>Sun et al.</td>
<td>128</td>
<td>QuASE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2014</td>
<td>Sun et al.</td>
<td>128</td>
<td>GETARUNS</td>
<td>2013</td>
<td>Park et al.</td>
<td>105</td>
<td>Park15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2015</td>
<td>Park et al.</td>
<td>105</td>
<td>GETARUNS</td>
<td>2015</td>
<td>Damova et al.</td>
<td>38</td>
<td>MOLTO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2015</td>
<td>Damova et al.</td>
<td>38</td>
<td>GETARUNS</td>
<td>2015</td>
<td>Sun et al.</td>
<td>124</td>
<td>QuASE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2015</td>
<td>Sun et al.</td>
<td>124</td>
<td>GETARUNS</td>
<td>2015</td>
<td>Usbeck et al.</td>
<td>134</td>
<td>HAWK</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2015</td>
<td>Usbeck et al.</td>
<td>134</td>
<td>GETARUNS</td>
<td>2015</td>
<td>Hakimov et al.</td>
<td>65</td>
<td>Hakimov15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
References


Challenges of Question Answering in the Semantic Web


