mathematics

MDPI

Review

Semantic Protocol and Resource Description Framework Query Language: A Comprehensive Review

Essam H. Houssein [1,*], Nahed Ibrahem [1], Alaa M. Zaki [2] and Awny Sayed [3]

[1] Faculty of Computers and Information, Minia University, Minia 61519, Egypt
[2] Faculty of Science, Minia University, Minia 61519, Egypt
[3] Faculty of Computing and Information Technology, Information Technology Department, King Abdulaziz University, Jeddah 21589, Saudi Arabia
* Correspondence: essam.halim@mu.edu.eg

Abstract: This review presents various perspectives on converting user keywords into a formal query. Without understanding the dataset's underlying structure, how can a user input a text-based query and then convert this text into semantic protocol and resource description framework query language (SPARQL) that deals with the resource description framework (RDF) knowledge base? The user may not know the structure and syntax of SPARQL, a formal query language and a sophisticated tool for the semantic web (SEW) and its vast and growing collection of interconnected open data repositories. As a result, this study examines various strategies for turning natural language into formal queries, their workings, and their results. In an Internet search engine from a single query, such as on Google, numerous matching documents are returned, with several related to the inquiry while others are not. Since a considerable percentage of the information retrieved is likely unrelated, sophisticated information retrieval systems based on SEW technologies, such as RDF and web ontology language (OWL), can help end users organize vast amounts of data to address this issue. This study reviews this research field and discusses two different approaches to show how users with no knowledge of the syntax of semantic web technologies deal with queries.

Keywords: semantic web; SPARQL query; resource description framework (RDF); question answering system (QAS); semantic question answering (SQA); web ontology language (OWL)

MSC: 68Q55

Citation: Houssein, E.H.; Ibrahem, N.; Zaki, A.M.; Sayed, A. Semantic Protocol and Resource Description Framework Query Language: A Comprehensive Review. *Mathematics* **2022**, *10*, 3203. https://doi.org/10.3390/math10173203

Received: 26 July 2022
Accepted: 29 August 2022
Published: 5 September 2022

1. Introduction

As a result of the availability of varied RDF data as lined data (i.e., more and more relevant data from heterogeneous sources are interconnections and form a web), accessing information is becoming increasingly important. However, these methods face several challenges, such as the user's need to understand the underlying schema, vocabulary, and complexity of the semantic web (SEW). This was considered the main purpose of this review paper to show the many approaches discussed in many papers that have attempted to solve the problems that face the normal user when querying keywords without any experience with semantic web technologies. Here, the resource description framework (RDF) semantic web standard and its semantic query language, semantic protocol and RDF query language (SPARQL), are two key technologies. An RDF repository is a graph of triples (subject S, predicate P, and object O), with S and O representing vertices and P representing edges. By creating triple-based queries, SPARQL allows users to query data repositories that follow the RDF description of the World Wide Web Consortium (W3C).

SPARQL is a common approach to accessing RDF data. In a single query on an Internet search engine, such as Google, numerous matching documents are returned, with several items that are relevant and others that are not. End users typically suffer in organizing an enormous amount of information, much of which is likely to be irrelevant. As

a result, improved information retrieval systems based on SEW technologies, such as RDF and web ontology language (OWL), are being developed as solutions. The two primary categories of semantic search approaches are structured query language-based approaches and methods for inexperienced users who may not be familiar with query languages. For the latter, two general methods can be considered [1]: (i) keyword-based (KEWB) methods, in which users construct queries from a set of terms; and (ii) techniques based on natural language, in which users create inquiries using natural language. The second type of technique includes question answering (QA) approaches, where a QA system (QAS) [2] seeks the correct solution to a natural language inquiry from a given set of documents. The three components of QAS are question categorization, information retrieval, and response extraction, which all play a fundamental role in QAS. Semantic search becomes increasingly influential as data on the semantic web increase.

As a result, the user must understand the formal query's sophisticated syntax. Furthermore, the user must know the underlying structure and the literal terms used in the RDF data. Thus, we surveyed two semantic search approaches, KEWB and QA semantic approaches, that are used to help user queries without SPARQL language to underline the knowledge base. All QA pipelines include the necessary steps to convert a user-supplied natural language (NL) question into a formal language query (SPARQL), the results of which are returned from an underlying knowledge graph. To accurately answer a given input question, we require a QA pipeline that, ideally, includes those QA components that provide the maximum precision and recall for answering the query [3]. With the rapid growth of semantic information published online, the question of how web users can search and query massive amounts of heterogeneous and organized semantic data has become increasingly essential [4]. The need to query information content in several formats has increased, including structured and unstructured data such as plain language text, semi-structured web documents, and structured RDF data in SEW [5]. As a result, QAS is necessary to meet this demand. With QAS, a non-expert user unfamiliar with the underlining ontology and query language can ask a question in NL. The response is obtained from one or more dataset(s). The SQA system simplifies the search process for the user by hiding the intricate processes required to deliver an accurate and detailed answer. Thus, users can intuitively use QAS based on NL to communicate arbitrarily complex information requests. Converting the user's information requirements into a format that can be examined using basic SEW query processing and inference techniques is the most challenging component of developing such systems.

According to [5], the goal of QAS is to allow users to ask questions in NL, using their terminology, and receive a quick answer by querying an RDF knowledge store. As the main contribution of this review, we survey many of various QAS and KEWB systems. Furthermore, we offer statistics and analyses. This can help researchers to find the best solution to their problems. The main problem that faced our authors in discussed papers in this review is how normal users without any knowledge about SPARQL language and underlying ontology can make queries without any problem. In this review, we show many papers that have solved this problem and their techniques to solve this problem. The primary motivation of this review is to focus on optimizing and modularizing already-existing, higher-performing QA procedures. It will encourage researchers to employ high-quality modules and assist them in focusing on and isolating essential research issues. This review presents several methods to convert the query from the user into formal query language through two primary categories of semantic search approaches on the web. There are two approaches for naive users. For keyword-based methods, users write queries consisting of lists of keywords. For natural language-based approaches, users create queries using natural language; this includes QA approaches where QAS aims to determine the proper answer to a question posed in the natural language given a group of documents. We looked over QAS studies, classifying them according to each criterion and evaluating the field's future research requirements. The criteria for QAS classification, the categorization of QASs based on various criteria, and a comparison of the suggested types

of QAS with others are then presented. Looking at prior work on QASs, we may infer that most individuals have used the information retrieval approach for QASs.

The rest of this paper is structured as follows: basics and background for SEW search are discussed in depth in Section 2. The analysis of the current semantic search engines are discussed in Section 3. Several types of QAS are given in Section 4 and various techniques of the previous work for semantic search engines are presented in Section 5. Future trends and challenges are presented in Section 6. Section 7 presents an evaluation of semantic web techniques. Finally, the research review is concluded under Section 8.

2. Preliminaries

This section introduces several basic concepts and background on SEW and its search approaches.

2.1. Definitions and Concepts Overview

2.1.1. SEW Search

Recently, the Internet has enabled users to access information and services easily. As such, web search has become a critical web application to obtain results. However, these results may not be accurate. Thus, we discuss several limitations or problems that gave rise to research on returning the most relevant results using semantic search technology. This major web technology primarily depends on a set of keyword queries with text and significance ratings for documents based on the Internet links. Thus, web search has several limitations, and a slew of studies are underway to develop more intelligent methods, also known as SEW search. Such research has recently become the most widely explored idea [6,7].

2.1.2. Semantic Web Search Approaches

This subsection presents how standard inputs are transformed into formal ontological inquiries. The two primary categories of semantic search approaches on the web rely on structured query language (SQL). SQL is a domain-specific language used in programming and designed for managing data held in a relational database management system and methods for inexperienced individuals that may not be familiar with SPARQL. For such users, the two kinds of methods are as follows [1]:

- KEWB approaches, in which users input inquiries by using a group of keywords.
- Using NL from users to create queries, including QA approaches, wherein questions are directly entered in the search. Commonly, the user questions begin with WH pronouns, such as (where, which, what, and who). The query interpreter parses and converts it into logical form. From here, the query converter component converts the logical query into the knowledge representation language of the database. The answer generation component evaluates the inquiry, and results are returned to the user. Lexical and generic resources generated by professionals in the field are the only resources used to answer the questions.

2.2. Types of Search and Their Techniques

There are many types of searches we can categorize into two categories. The first is keyword search (KEWB), a type of search that looks for matching documents containing one or more words specified by the user. A single query to a search engine on the internet (such as Google) produces thousands of so-called "matched documents", some of which are relevant and some are not. End users typically suffer from picking through and understanding such massive amounts of information, much of which is likely to be irrelevant. Its techniques are divided into ranking result techniques, statistical techniques, and user-selection techniques. The second search type is semantic search, which gives more structure and computer-understandable meaning and provides a common framework for data sharing across applications, enterprises, and communities. This type (question answering semantic search) of multiple techniques are user interaction and semantic graph-

based techniques, depending on the number of the used dataset. We will discuss these types with their techniques in more detail with examples later in Section 5. We can add a summarization to types of search through Figure 1 to explain the techniques for each type.

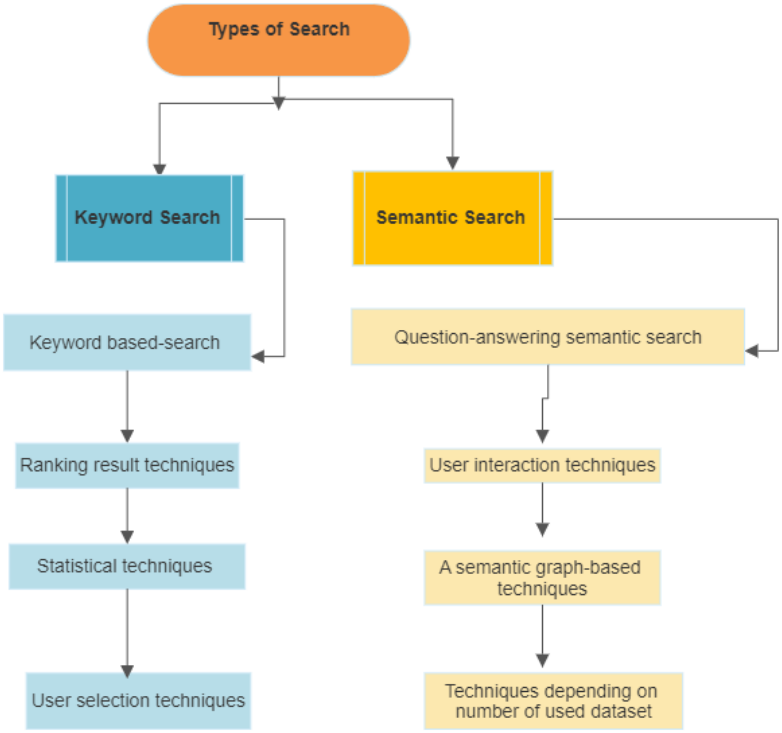

Figure 1. Types of search.

2.2.1. Question Answering System (QAS)

QA [2] is a kind of information retrieval (IR) that focuses on answering questions, and as such, the challenge is immediately responding in normal language to a user query in the area of IR and natural language processing (NLP). Question evaluation, document retrieval, and output results are the three basic subtasks of QA systems [8]; however, the systems may differ in how each subtask is carried out. QAS aims to determine the convenient result to the query posed in the normal language, given a group of documents. In QAS, question evaluation, document retrieval, and output results are the three important components.

2.2.2. Components of QAS

Any categorized QAS method has the following components:

- Question classification:is the initial phase in which a user query is classified, expected answer types are determined, and keywords are extracted. In addition, a question is reformulated into several semantically identical inquiries. Reformulating an inquiry to comparable signification inquiries is referred to as a matching process, increasing the retrieval system's recollection. Question classification in a QAS depends on the type of entity that questions represent.
- IR: The recall is critical for the output of results. If a document contains no correct answers, then no additional processing can be performed to discover a solution. In the IR phase, the precision and ranking of candidate passages can affect the QA ability. The IR approach achieves success through the output of relevant results, using intelligent QAS.
- Response extraction: The last step in QAS, where such modules are becoming more popular because these systems frequently require ranking and assessing a candidate's response.

2.2.3. Semantic Question Answering (SQA)

The main technology for allowing the final user to inquire about knowledge graphs utilized normal language forms is SQA systems, where user questions in natural language are automatically translated into semantic queries [9]. SQA eliminates two significant barriers to entry to the SEW: domination of SPARQL and knowledge about underlying ontologies, which means knowing what information structures are available to formulate queries, which also return results for enabling normal users to access the data web. As formal query language (SPARQL) is so complicated, users of SQA ask queries in natural language (NL), utilize their vocabulary, and they may ask a question and receive a quick response from an RDF content library.

SQA systems were usually divided into both stages. The query analyzer and retrieval are the first two steps. At the retrieval step, the analyst of queries creates the query which used to retrieve the response. They use Tokenizer, disambiguation, internationalization, Boolean forms, and semantics task identifiers. Moreover, inquiry restructuring, working perfectly, association identification, and representation-based identification are some techniques used in this research. These are just a few methods for use during the analysis step. With a topic relating to the fast rise of contextual information transmitted over the internet, people online may now discover as well as analyze massive volumes of homogeneous and organized data sources [4].

The requirement to search information material in various formats has received greater attention, integrating data from different sources such as plain language text, semi-structured web documents, and organized RDF data on the SEW [10]. As a result, QAS is required to meet that demand. So we can say a non-expert user without knowledge about the underline ontology and query language could ask a question with a structure as a normal language statement.As it could hide complicated operations necessary to provide a precise and thorough result, the SQA system makes it simple for the user. Users of QAS can more easily request any sophisticated data by using normal language. The most challenging part of creating such systems is converting the individual's data requirements to a format that could be analyzed utilizing basic SEW query transmission and strategies for inference. According to [5], the purpose of QAS is to let individuals express inquires in normal language, utilizing their self vocabulary; they also obtain a relevant response by searching an RDF database.

2.3. Comparison between Classical Search and Semantic Search

The world wide web is a vast information repository with enormous potential. The retrieval of related information from the web is the main issue because it is hard for machines to process and integrate the information. KEWB search has become widely attractive, as it transmits terms to a web search, and the search engine returns a rank result set. A serious issue of KEWB search gives many irrelevant outputs. The procedure for looking over the classical engines was performed by matching the keywords without understanding the concept. In reality, the Internet is overgrowing, as pages are added at a breakneck pace. Searching on the web for a particular concept or term on hundreds of pages based on ranking algorithms or numbers is not an efficient solution because the results put the user in a maze to reach accurate information. For that reason, several initiatives are to reduce the current web's drawbacks and move toward a more intelligent machine. Semantic web is one of the concepts that leads to create an intelligent machine. Webpage absence on a basic framework for representing data, the misunderstanding of data stemming from inadequate connectivity of data, the inability of immediate data transmission and accessibility to handle with large numbers of users and topic while maintaining confidence at all stages, and computers' difficulty in understanding the available data are caused by the absence of a standardized language [11].

For SEW (WEB 3.0) [12], Tim Berners-Lee, the founder of the WWW Consortium, devised the concept of SEW in a Scientific American essay describing the web's future. More structure and computer-understandable meaning would be provided by the semantic web,

as well as a common framework for data exchange across apps, companies, and communities. As a result, the core definition of the SEW is a set of concepts that are connected rather than a collection of documents. At the same time, a notion can have multiple interpretations linked together from various publications. This means that concepts in one document are connected to similar concepts in other papers utilizing the RDF, a new standard (RDF). Terms in documents are linked by a collection of keywords on the traditional web.

2.4. Semantic Web Technologies and Languages Overview

The SEW architecture (W3C) that Tim Berners-Lee illustrated is known as the "SEW Stack" or "SEW Cake" [13]. It proposes that the SEW layer is an addition to, not a replacement for, the traditional hypertext web layer "traditional web" because it prioritizes the WWW of data "SEW" over the web of documents "traditional web". Rather than interlinking text content, it is built around interlinking data. Semantic web technologies and languages facilitate the development of semantic web applications by providing new techniques for managing information and analyzing semantic metadata This layer, according to George Abraham and Tim Berners, can be used to build a SEW using five standardized SEW technologies: RDF, RDFS, OWL, SPARQL, and rule interchange format (RIF).

RDF [14] is a model for describing real-world objects or entities. It has a controlled vocabulary containing several terms (e.g., words, phrases, or notations) that have been officially enumerated. Every term should have a clear and non-redundant definition. In addition, the RDF can be exchanged across various applications and systems without losing significance. RDF data were expressed in the form of triples, which are statements. A triple consists of three segments (S, P, and O) representing resource information in connected graphs. Every triple has a subject that represents a real-world entity or notion. It must also be recognized by a URI (uniform resource identifier), which may or may not be an actual URI on the Internet. The predicate follows the same rules as the O, except that the O could be represented by a URI, a textual, or a blank node. To avoid data ambiguity, URIs are unique identifiers that are used to describe unique concepts. Literals are data resources not recognized as entities and hence cannot be expressed as URIs, such as a textual description, a date, or an integer.

RDFS [15] is a framework for describing resources. RDF has a schema extension. A subject-based classification uses a hierarchy to organize the terms in a restricted vocabulary/RDF. The data model adds language and associated semantics for classes, subclasses, properties, and sub-properties.

OWL [16] describes the core objects and connections that contribute to the content of a specific subject vocabulary and the criteria for merging terms and relationships to form vocabulary expansions. Ontologies have also been characterized as explicit specifications of a conceptualization or shared knowledge of a particular interest area in other ways. Types of concepts employed, in addition to the limits imposed by their use, are fully described. The ontology includes a variety of valuable aspects that help systems become more intelligent and influential in determining the precise meaning of concepts.

SPARQL [12,17] is an inquiry language for RDF or a procedure for extracting data from an RDF graph developed by the W3C. The RDF diagram is made up of triples (S, P, and O), and SPARQL is adapted to extract data from them. It is a data access language that can be used locally or remotely. SPARQL is an RDF-inquired language governed by the W3C [18]. It defines a standard format for creating RDF data-targeting queries and a set of criteria for processing and returning the results. The syntax of SPARQL is discussed. A SPARQL query consists of the following order: prefix declarations are used to shorten URIs; schema specifications include an RDF graph and the sentence that contains the result, determining what data the inquiry should output; a pattern of query identifies inquiry qualifiers, cutting, sorting, and reordering query outputs, as well as what to inquire for in the source datasets [19,20].

Four different SPARQL query languages are illustrated in Figure 2. In a graph pattern, the **SELECT-query** is accustomed to getting several bindings for a variable. **CONSTRUCT**

returns an RDF graph, **DESCRIBE** outputs an RDF graph representing a resource, and **ASK** outputs a single zero or one result. It can be used to see if a graph pattern matches anything or not [21].

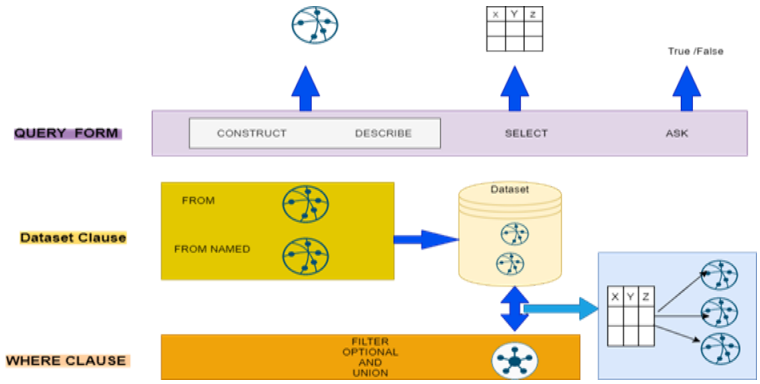

Figure 2. Forms of SPARQL query.

3. Analysis of the Semantic Search Engines

The current search engines, such as Google and others, are undoubtedly powerful and popular. They succeed as a result of page rankings and algorithms. However, many of these browsers are not based on the SEW notion. Rather than keyword matching, semantic search assures more relevant results by understanding the context of words or terms and their synonyms. As a result, the semantic search returns intelligent and relevant results. Kngine [22], and Alpha [23] are examples of browsers that use semantic technologies.

- Hakia: returns results based on matching meanings.
- Kngine: the results are split into two categories: web results and image results.
- Kosmix: makes use of semantics to try to link data from all across the web based on relevant search results.
- Powerset: concentrates on doing one thing well in order to return just pages that have the answer.
- DuckDuckGo: is a semantic search engine with a lot of features.
- XSEarch: uses a basic query language that may be used by a novice user [23]. We suggest some criteria and compare among different search engines based on the suggested criteria. Table 1 shows the results of this comparison. The compared search engines are classified into beta and non-beta engines.

Table 1. Comparison between semantic search engines.

Criteria	Search Engine	Semantic Web Technologies	Results
Beta Engines	Kngine	It is Own Semantic Technologies	Direct Answer or Link to Web Page
	Bing	Not Given	Link to Web Pages
Non Beta Engines	Swoogle	XML, RDF and OWL	URIs Ontologies
	Evi	Not Given	Direct Answer or Link to Web Page
	Watson	XML, RDF and OWL	URIs Ontologies
	Google	It is Own Semantic Technologies	Direct Answer or Link to Web Page

Different measures are used to judge the quality of search engines in real life [14]. The search results' usability and presentation are clear criteria. Search engine usability relates to an engine's simplicity for use, as seen in available links, instructions, and a user-friendly interface. Some engines, such as Swoogle and Watson, include links to search for terms, subjects, predicates, objects, documents, and ontologies in specialized fields. While other engines use a general search based on keyword matching and its technologies to obtain a semantic web engine, they utilize a thorough inquiry focused on the similarity of keywords and their technologies. A mechanism of providing outputs that differ from one search

engine to the next is the second point of quality of search engines. Swoogle and Watson, for example, use URI ontologies because their benchmark is based on XML, RDF, and OWL.

Furthermore, some engines, such as Google, Kngine, Siri, Evi, and Wolfram, use the most efficient semantic web technique, which is a direct response. The direct response includes not only text, but also photographs, videos, prices, and user reviews. While most systems continue to use the traditional method of representing output "links", semantic engines employ various techniques. Artificial intelligence, natural language processing, and machine learning are just a few examples. Swoogle as well as Watson utilize XML, OWL, and RDF as semantic technologies. Furthermore, Kngine takes advantage of the effectiveness of a knowledge-based strategy and the significance tests technique, whereas Google relies on self-knowledge graph innovation, known as the "Hummingbird algorithm". This concept stems from the fact that "precise and fast" are two of the most important aspects of any search engine.

Most of the engines discussed above offer a phone application that makes them more popular and portable for clients worldwide. Furthermore, many of them (Siri, Kngine, Evi, Wolfram, and Bing) offer advanced features, such as "speech recognition". They give the operating system the ability to transform spoken words into text. Apple, Microsoft, Samsung, and other companies produce these platforms. The data show that most search engines support English, but only Kngine supports Arabic. As a result, Kngine will be utilized as a part of the comparison when the suggested Arabic search engine is examined. Kngine is the world's 1st multilingual QA engine, with English, Arabic, German, and Spanish as supported languages.

Kngine [7] ("Knowledge Engine") is a WWW Kngine (i.e., a groundbreaking semantic search engine and question answer engine) that is aimed to deliver tailored and precise inquiry outputs. For example, details on the keywords' meaning, answering the customer's inquiries, creating several items, and discovering the relationships in the query's words are all examples. This search engine has an attractive feature in that it provides accurate results by connecting many types of linked information to display these to the customer, for example, videos, subtitles, pictures, and costs at sale stores as user reviews and influenced stories. Kngine presently has billions of concepts/terms, with more added daily. The site's strength is in this area. End users must understand complicated structured language expressions and the fundamental vocabularies, according to a number of papers introducing different solutions to the same problem. Among SEW search and final consumers, this has become a significant gap. Allowing users to conduct semantic searches by entering a keyword query is beneficial.

4. QAS vs. Traditional Information Retrieval System

QA is a complex IR characterized by data requirements communicated at least in part as normal language sentences or queries. It was once considered among the most normal ways to communicate with computers. In contrast to traditional IR, which considers accurate results identical to the required data, QA returns selected data fragments as an output. A single sentence, phrase, section, graphic, good segment, or entire text refers to a concise, intelligible, and correct response. The QAS primary goal is to figure out" where, when, how, and why did who do what to whom?" QAS uses IR and extraction techniques to establish a group of probable proposals and a rating methodology to provide the final answers. We can present the statistics for QAS search engine studies from 2010 to 2021 based on information from the Scopus databases in Figure 3.

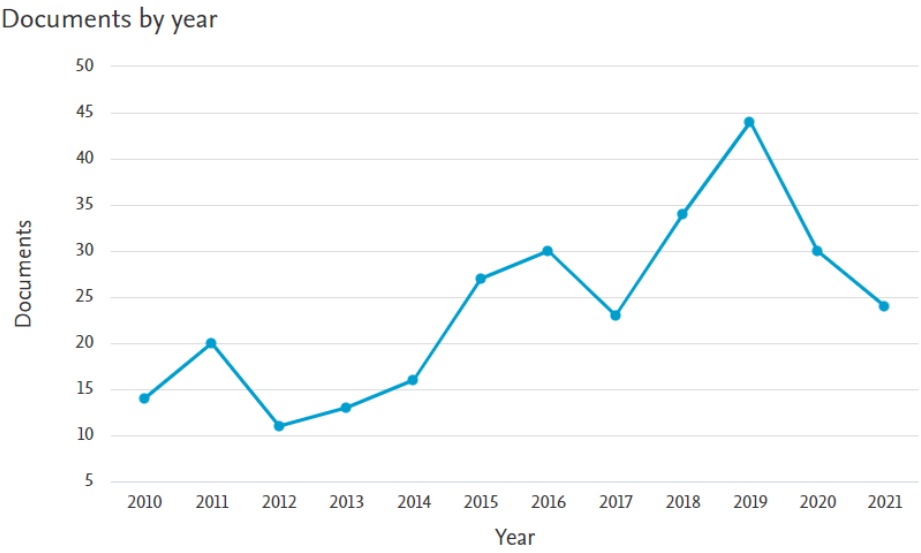

Figure 3. Histogram of question answering semantic search engine publications from 2010 to 2021.

4.1. General Architecture of QAS

The user queries the system via forms. The inquiry is used to output all the available results to the input inquiry. The general architecture in Figure 4 discusses the steps or phases that pass through any question-answering system.

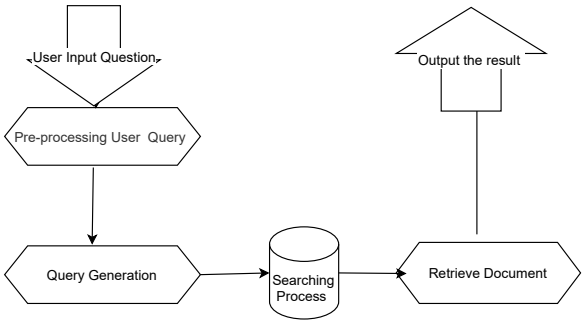

Figure 4. General architecture of QAS.

4.2. Types of QASs

Various QASs [2] are classified into two sets depending on their methodologies. The first set of QA systems is based on simple NLP and IR methods, whereas the second set is based on natural language reasoning as shown in Table 2.

Table 2. Characteristics of QASs.

Type	Method	Resources of Data	Domain	Responses	Questions Dealing To	Evaluations
QAS rely on NLP and IR	IR, Syntax Processing	Free text documents	Independent Domain	Snippets extracted	The wh-type	Uses existing IR
QASs rely on NLP	Semantic analysis	Knowledge Base	Oriented Domain	Synthesized	Variety of questions	N/A

4.2.1. Web-Based QAS

Across the widespread use of the web, a vast amount of information is obtainable; the Internet is one of the most effective sources of knowledge. Web-based QAS uses search engines such as Google and Yahoo to return webpages that may include solutions to the inquiries. Most of these web-based QA solutions work in an open domain, although some work in a domain-oriented environment. The Internet's wealth of knowledge creates an attractive warehouse to find rapid results to straightforward, factual questions [23]. Semi-

structure, heterogeneity, and distributivity characterize the data available on the Internet. W-H type questions are mostly handled by web-based QA systems seen in Figure 5.

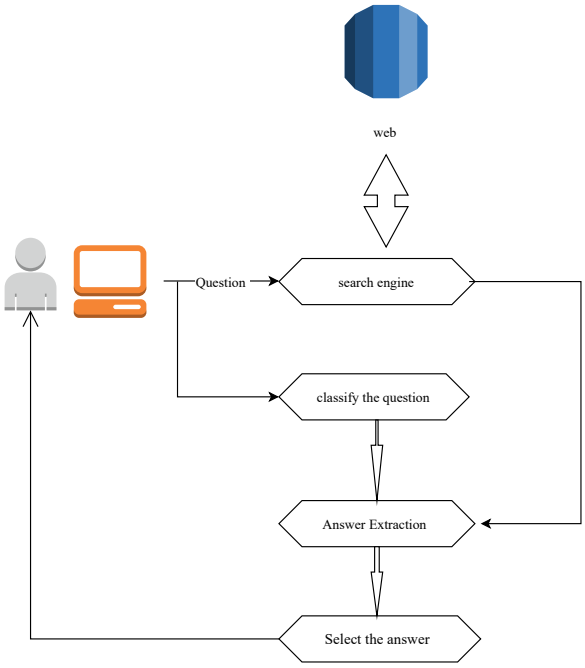

Figure 5. Design of web-based QASs.

4.2.2. IR/IE Based QASs

In response to the query, most IR-based QAS return a group of publications or paragraphs that have received the highest ranking. The information extraction (IE) system parses the query or publications retrieved by the IR system using natural language processing (NLP) technologies to determine the significance of each word. This system can only answer wh-type questions in Figure 6; other queries, such as the example, "What are the steps to putting a computer together?" will never be responded to. In this type of QAS, known as IE systems, its architecture is divided into two levels: at level 1, a named entity tagger is used to handle named entity elements in the text (who, when, what, etc.), and at level 2, NE tagging + adj (how far, how long, etc.).

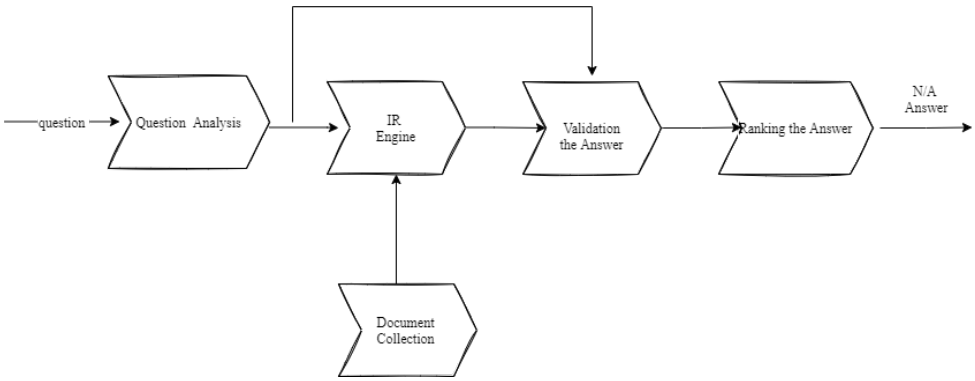

Figure 6. Design of an IR/IE-based QAS.

4.2.3. Restricted Domain QASs

This form of QAS requires academic assistance to comprehend the normal language content and appropriately answer the questions truthfully. The construction of a restricted domain QAS [24] is a good strategy to enhance the precision of QAS by reducing the volume of the ontology and the scope of the queries (RDQA), as seen in Figure 7. This

system has unique properties, such as "system must be accurate" and "reducing redundancy". By obtaining more precision, RDQA overcomes the limitations encountered in the open domain.

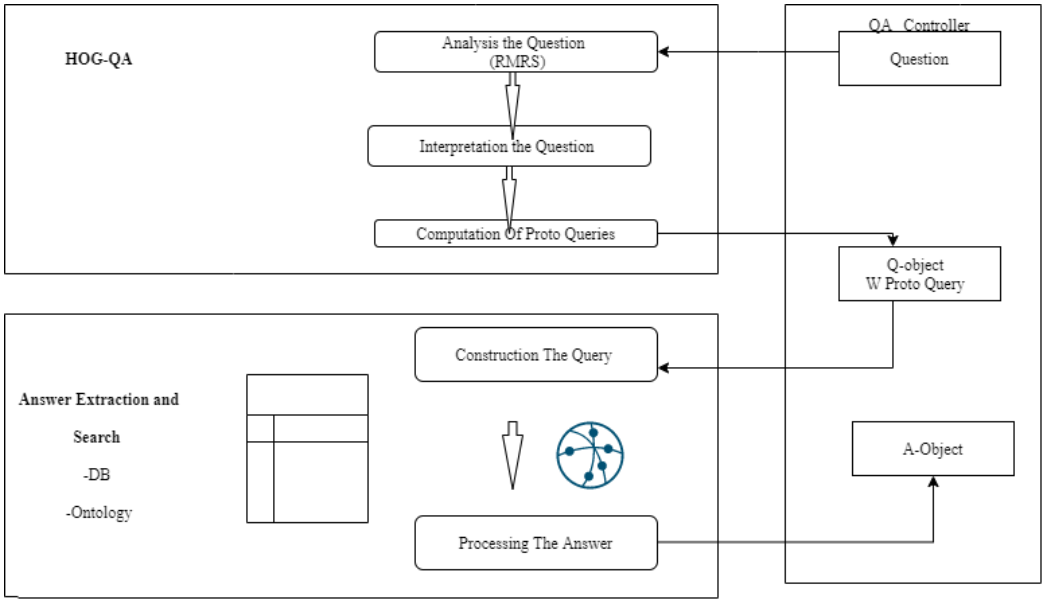

Figure 7. Design of domain-restricted QAS.

4.2.4. Rule-Based QASs

The rule-based QAS is a more complex version of the IR-based QAS. Rule-based question answering does not rely on sophisticated techniques or extensive language knowledge. In order to ensure accuracy in the responses retrieved, a wide range of NLP techniques are applied. Some common rule-based QAS use lexical and semantic information in questions to produce heuristic rules [2]. It produces laws for semantic groups, such as when, who, where, why, and what for each query form. The "who" rule searches for names, generally nouns of people or objects. The "what" laws concentrate on a general word equals function that is retrieved by all inquire kinds and is made up of DATE expressions or nouns. The "when" rules are primarily made up of time expressions. The majority of "where" rules consist of similar positions as "at" "in" "within" and "near". Some typical modules in this system include the IR module and the answer identifier or ranker module. The IR module feeds the ranker module a group of publications or phrases that involve the results to the supplied query. It then retrieves the outputs, ranker module assigning grades to phrases returned by the IR module, and answer identifier. It uses the score or rank of the answer substrings to identify them from the sentences.

5. State-of-the-Art Methods

This section presents previous techniques for semantic search engines and is divided into subsections. Before that, a summary of previous research on the QA semantic search engine and keyword-based search engines is presented. We can compare the different methods or semantic search techniques in Table 3. This comparison explained the results and the accuracy of each method, which helps us determine which is better than the others. After this table, we explain each technique in detail and know each method's strengths and weaknesses.

Table 3. Comparison among the different semantic search techniques or methods.

Used Methods	Reference	Main Purpose	Important Results	Accuracy
Ranking result techniques	[25]	Construct a score of 76 different languages' similarities.	English is in the top rank with nine other languages while Chinese is at the bottom	—
Natural language generation	[26]	Utilizing NLP techniques with semantic technologies	The recall rate of the system, based on impressions of 69 out of 11, is about 90%.	—
Semantic search technique	[27]	They are creating interactive interfaces so that users can annotate online documents	An event-based semantic web triple store with thousands of facts was used.	92%
	[28]	Described the natural language semantic processing of ECA algorithm	XLNet on the GLUE dataset is significantly better than BERT	—
Tokenization	[29]	Developing text preprocessing system	With a little quantity of input data, each algorithm provided in the solution produces accurate results	—
Semantic Analysis	[30]	Increase the accuracy of classifying with NLP	F-measure rate of 91%, average precision of 92%	
Question Answering technique	[31]	Answering questions effectively	Questions with Yes/No answers.	95%
	[32]	Answering the questions that are asked by the user	Overall Average over the entire dataset was 65% correct answer prediction	69.6%

5.1. Question-Answering Semantic Search Engine

Fazzinga and Lukasiewicz [12] presented a survey on evaluation methodology called QALD to assess and compare QAS that act as a bridge among semantic information and queries in normal language, particularly in terms of their capacity to manage vast quantities of diverse and structured information. In addition, a list of the advantages and disadvantages of present SQA systems, as well as its potential improvement through intervals, is provided. This evaluation methodology includes many QAS, such as Power Aqua [33], which performs on-the-fly QA on structured data in the case of a potential domain without making assumptions regarding the dataset's ontology or design. Thus, the method can take advantage of many vocabularies on the semantic web. Power Aqua uses the structure. The user's question is translated through linguistic processing to query triples of the type (S, P, and O) (not including comparisons and superlatives). A query triple is then provided to a converting component that searches several ontologies for adequate semantic resources that probably explain query keywords (involving a search on WordNet for synonyms, hyponyms, derived words, and meronyms). The group of vocabulary triples covering the user's question is obtained using these semantic resources. Finally, in only providing partial answers, each of the resulting triples must be concatenated to obtain a complete response. To this end, the several views given by various vocabularies are combined and evaluated.

The second QAS shows that users can submit inquiries in any format using Freya [34], which, to identify the answer type, first creates a syntactic parse tree. The processing begins with a lookup, utilizing an ontology-based gazetteer to annotate query phrases with related concepts. If confusing annotations are observed, the user is asked for clarification. The individual's choices are retained and utilized to train the framework and thus improve its performance over time. Subsequently, the triples are constructed based on ontological mappings, considering the characteristics' area and value. Lastly, the triples are assembled into a SPARQL expression. Moreover, Haemmerlé et al. [35] presented the semantic web interphase using patterns that transform NL questions to a nearly entirely keyword-based language rather than using them as input. The following steps list how the system converts the keyword-based query into an ontology. First, the keywords are converted into ideas, after which a group of ranking structures similar in meaning to those ideas is observed. The designs are chosen from a predetermined collection based on common user requests. Finally, the system prompts individuals to select the search style that most accurately represents the significance of a supplied word. The resultant query structure is created using the chosen representations.

Höffner et al. [36,37] suggested a new technique to SQA that uses the RDF Data Cube Vocabulary to handle multi-dimensional statistical linked data called CubeQA, which is inaccessible to previous approaches. Using a corpus of questions, the studies examine how queries that require open domains factual analysis differ from those that do not, what

extra verbalizations are typically used, and what their effect is on SQA design choice quantifiable data.

Usbeck et al. [38] proposed the first hybrid source SQA system to handle both linked data and textual information to respond to a single input question. This system, HAWK, uses an eight-step pipeline that includes grammatical pruning techniques in the examination of the NL form, semantic annotation of attributes and class, construction of fundamental triple structures on every element of the source question, eliminating requests with unconnected inquiry graphs, and grading the queries.

5.2. Techniques or Methods Used in QAS

5.2.1. User Interaction Techniques

There are some question-answering semantic search engines that depend on user interactions, as in Zafar et al. [9] proposed IQA, an SQA pipeline interaction scheme that uses option gain. This new metric promotes effective and simple human interactions, allowing for the fast connectivity of customer reviews into the QA operation. The suggested schema on several consumer connections can significantly increase the performance of SQA systems. The purpose was to provide a new user interaction strategy to solve the constraints of current SQA techniques to respond to complicated questions. Although user interaction is frequently used in other fields, such as IR and searching keywords on organized data [39], these models are not yet generally used in SQA. The planned semantic meaning of inquiry cannot be reliably determined utilizing automated approaches. For answering difficult questions, the suggested IQA technique can be beneficial. A thorough experimental evaluation and user research show the performance and adequacy of the suggested corporate approach for SQA. Figure 8 shows the findings on LC-QUAD, a well-known dataset for evaluating SQA systems, where the IQA can significantly enhance the effects, reliability, and usefulness of SQA systems for complicated queries.

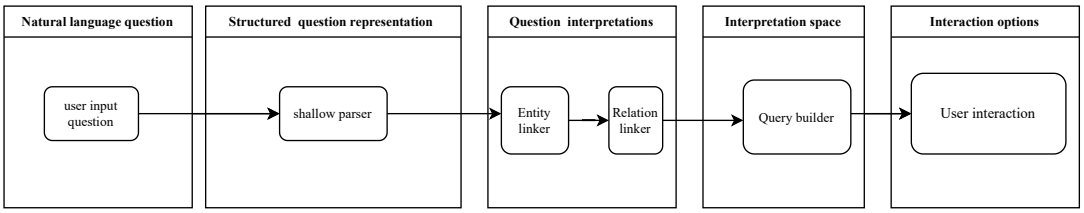

Figure 8. IQA architecture.

Aggarwal and Buitelaar [40] presented a QAS (SemSek) that matches NL keywords to ontology objects using several steps, namely, language analytics, inquiry description, and semantic similarity measurement. Query annotation mainly searches a DBpedia index for entities and classes that compare phrases in the NL query. The linguistic analysis provides a syntactic parse tree as guidance. SemSek obtains an arranged set of words following the dependence trees, beginning with the most probable of the detected objects and subclasses. SemSek uses two semantic similarity metrics, one based on Wikipedia and the other on WordNet structures, to match these phrases to DBpedia ideas.

Cheng et al. [41] presented a technique of a randomized surfer with computing of central parts, or a concept of centrality that considers their connectedness and informativeness. The similarity was determined as the degree to which its characteristics and values are related.

5.2.2. Machine Learning Depending Approaches

Singh et al. [3] proposed the Frankenstein framework that uses machine learning approaches to dynamically choose appropriate QA modules and compose related procedures that depend on the provided query, resulting in an all F-Score improvement. The QA optimization pipeline method is used to find the pathways with the best performance depending on the provided question properties and purpose. The objective is to reuse the

29 primary components available to the QA community rather than creating a new system from the start. This previous study also displays the capacity to incorporate elements that the scientific community has supplied into a unity platform. The framework's vaguely linked structure allows for learning from the features derived from incoming inquiries to lead the user query to the top-ranked element per QA task and allows for a quick merging of new elements. As a result of its component-agnostic design, the Frankenstein framework was readily adapted to various databases and knowledge sectors, becoming a foundation to create sophisticated QA processes automatically. Thus, a comprehensive summary is presented of the achievements of cutting-edge QA components in various activities and pipelines. Furthermore, the performance of the Frankenstein framework is assessed and compared in various scenarios, demonstrating its advantages.

Frost et al. [42] suggested a solely triple-based retrieval operation called DEV-NLQ, which relies on lambda calculus and triple storage based on events. DEV-NLQ promises to be the only QA system available for solving prepositional phrases that are chained, arbitrarily nested, and sophisticated.

5.2.3. A Semantic Graph-Based Techniques

Liu and Chen [43] presented a graph-based text method. A text's resemblance to another is separated into vertices and links, with each component being measured separately. The similarity in text accounts for the total of two portions. Hakimov et al. [44] suggested an SQA system based on input question syntactic dependency trees. This method has three basic steps: (1) The queries' dependence graph and POS tagging are used to extract triple patterns. (2) The retrieved entities, attributes, and classes are transferred to the underlying knowledge base. (3) Finally, query terms are assigned to an appropriate result, such as "who" to an "individual" or "firms" and "where" to a location. Subsequently, once the outcomes are rated, the highest-ranking answer is provided as the solution. Suppose the *S* or *O* of *P*–*O* or *S*–*P* pairings are answered by PARALEX [45]. In that case, they are transformed from sentences to concepts in a vocabulary that was developed utilizing a corpus of rephrases gathered from the Wiki results question-and-answer website. If one rephrase can be converted into an inquiry, that inquiry is also the right response to the rest of the rephrases, among which the language patterns are extended by mapping.

Another strategy presented by Ngomo et al. [46] is to produce NL descriptions of sources based on their qualities automatically. The goal of SPARQL2NL was to pronounce 16 RDF data using structures in conjunction and its metadata (label, specification, and kind). Entities can have various kinds and hierarchy levels, resulting in different levels of abstraction. The DBpedia object dbr: verbalization of Microsoft varies based on whether the kind was dbo: Agent or dbo: Company. Chergui et al. [47] proposed a practical and general semantic linguistic similarity method for QAS. A semantic graph-based technique is integrated to deal with the shifting semantic relevance of the query phrases and a Bayesian inference method for dealing with the semantic unpredictability represented by NL documents. For new challenges, case-based reasoning is used based on similar past issues. The most similar questions are determined by running a similarity calculation among the new and old texts.

Freitas et al. [48] reported that in their system, Treo, the query processing begins with the identification of the NL inquiry, pivoted items that can be translated to classes or categories and thus provide a beginning point for expanding activating query across the integrated data. Starting with these pivot entities, this technique iterates across surrounding nodes, calculating the mutual information among query phrases and discovered ontology keywords. Thus, a set of prioritized tripled pathways is produced from the pivoting object to the solution's final source, scored by the mean of similarity over each tripled path.

Shin et al. [49] proposed a predicate constraints-based QA technique that docks invalid element matches to save the inquiry area, improving accuracy. They generate a graph of inquiry that can be used to answer various questions. When creating a query graph, instead of connecting things, they place greater emphasis on matched associations.

5.2.4. Techniques Depending on the Number of Used Datasets

We classified QAS search engines as being dependent on more than one dataset, and others as depending on one dataset. Shekarpour et al. [50] proposed SINA, a QAS that can respond to user inquiries by converting terms specified by the user or NL questions into simultaneous SPARQL searches across a range of related information resources. SINA uses a mystery Markov system to find the best sources for a user-provided question from various datasets. Furthermore, this framework can construct federated queries by utilizing the related framework ontology to inquire and use disambiguated resources. SINA was tested on three different datasets to demonstrate how its several phases are carried out. In contrast to previous approaches, SINA may utilize the topology of linked data to create federated SPARQL searches by utilizing links between resources. As a result, data may be obtained from a single dataset or numerous interconnected datasets and thus utilize the entire LOD Cloud. A comprehensive analysis of SINA was also carried out using three various ontologies. Figure 9 explains the SINA search engine's high-level architecture and implementation.

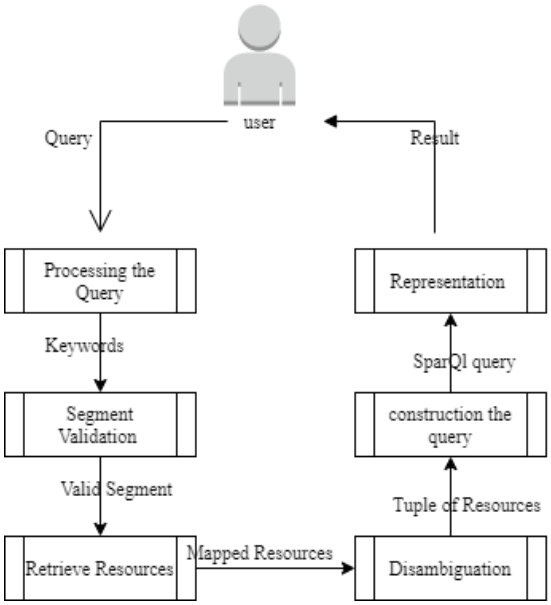

Figure 9. Architecture of SINA.

Sun et al. [51] proposed the QUASE system, a 3-step open domain strategy that uses online search and free knowledge base. For answering the input question, QUASE uses entity linkage, meaning feature building, and candidacy rating. The system then uses an online search to find publications and keywords that have the highest likelihood of matching the inquiry and displays these to the user as responses.

Cojan et al. [4,52] suggested a DBpedia-based question answering system (QAKIS) that uses the Wiki framework repository to bring NL words and ontologies topic tags closer together. This resource is created by obtaining lexicalizations of linked data OWL features from Internet textual data. For constructing a SPARQL query, QAKIS first defines the response category and also the category of the named object, and then compares the produced inquiry to patterns in the Wiki framework repository to find the more probable association.

Wendt et al. [53] presented a German QAS based on Alexandria, a domain vocabulary that comprises data on people, places, things, and activities, including timed elements. The vocabulary was constructed chiefly with data from Freebase, sections of DBpedia, personally contributed material, and n-ary relationships among objects. Alexandria treats the challenge of translating NL inquiries to SPARQL queries as a graph transforming difficulty. A dependency tree is used to depict the question's syntactic structure. NL

tokens are converted to ontological objects using a custom vocabulary for characteristics and indexing for object recognition. Second, using a compositional approach, SPARQL graph designs are created by aggregating the arcs of the dependence parse tree. Thanks to the ontology schema's representation of n-ary relations, Alexandria can compare small language inquiries with complicated triple designs.

Abacha and Zweigenbaum [54] proposed a semantic method for solving QA problems in the medical field that relies on NLP techniques and SEW technologies at the presentation and questioning phases. The suggested method was evaluated using actual queries retrieved from MEDLINE articles.

In addition, Zhang et al. [55] proposed Xser, an SQA comprising two separate phases. First, Xser uses phrase-level dependence architecture to identify the query design, leverages the generated pattern, and moves to the desired knowledge base. Changing domains and relying on a diverse knowledge base, for example, only changes strategy elements, thereby reducing the transformation. While, Hakimov et al. [56] applied a semantic parsing method to SQA, which obtains good results but necessitates a huge amount of analysis inputs, which is thus impractical for broad or undefined areas. Over the last few years, several SQA-based activities that are supported by the industry have arisen. For example, IBM Watson's DeepQA [57] was able to defeat human experts in the Jeopardy! Challenge.

Ren et al. [58] proposed a method for answering multiple-choice questions that include a retrieval methodology that considers the value of knowledge from a semantic standpoint. A proposed knowledge revision mechanism enables the model to edit retrieved knowledge from local and global viewpoints.

Unger et al. [59] suggested that a QAS, called TBSL, concentrates on converting NL questions into SPARQL inquiries that represent the question's semantic structure. TBSL produces a SPARQL format that reflects the internal design of the question, which is subsequently instantiated with URIs using statistic entity detection and predicate verification.

5.3. Keyword Based-Semantic Search Engine

In [60], Syed Naqi Hussain solved the recent challenge of retrieving documents in heterogeneous format. This survey discussed several challenges in retrieving semantic information from the web and presented solutions, namely, Swoogle and SBIRS have a mechanism of ranking the returned retrieval results. In [61], Jain et al. introduced a new framework based on fuzzy ontology for IR to overcome a present web system's weakness and make the most of its benefits, such as query expansion. Fuzzy ontologies are used to find the most semantically equal terms for a query, and the question is enlarged. Queries are analyzed using this suggested methodology as Google, Yahoo, Bing, and Exalead are four major search engines. With the advancement of database storage technology, a massive amount of data has been generated, and users now have access to it. In the scientific community, accurately extracting and interpreting such large amounts of data has become critical.

5.4. Techniques of Selecting More Relevant Result

5.4.1. Statistical Techniques

In [19], Jos'e A. Royo et al. proposed a system that takes a list of keywords and converts them into a semantic query for searching the web. The suggested system converts the list into semantic queries and retrieves more relevant results using the following steps. First, a list of keywords is taken from the user, and synonyms of keywords are selected to remove redundancy. Second, mapping keywords are generated from the first step with ontology terms, and finally, the SPARQL query is generated according to its synonyms probability. This approach also has more features, such as the system-generated user keywords from word net, and helps users to select the more relevant result. In addition, the optimized number of comparisons between terms depends on statistical techniques and generates queries that better match the original user query. However, this approach does not process queries on a highly dynamic global information system.

In [62], Mariana Soller Ramada et al. discussed the approach that converts semantic keywords into a SQL structured query language on a relational database. Several functions, such as query approximation, segmentation, and aggregation, are integrated into the proposed architecture for implementing semantic queries. By comparison, EL-GAYAR et al. [20] proposed a hybrid framework that integrates the benefits of keyword-based and semantic queries. In a fuzzy ontological model in IR, Fazzinga and Lukasiewicz utilized semantic query expansion [12] and proposed a structure depending on fuzzy vocabulary. This structure was evaluated on Google, Yahoo, Bing, and Exiled.

5.4.2. Ranking Result Techniques

In [63], Wen et al. focused on the challenge of effectively converting terms to SPARQL queries and developed a KAT technique. By creating a keyword index that captures the relevant information of terms in the RDF graph, KAT considers the background and decreases the uncertainty of incoming terms. A graph index is created to investigate RDF data graphs effectively. Furthermore, a situation ally rating mechanism for locating the more appropriate SPARQL query is suggested. Several contributions are summarized into two main points: new keywords in transforming input terms into correct SPARQL inquiries. Using semantic features between terms can increase the relevance of the results. A two-facet indicator is also shown. The keyword index includes RDF relevant information and aids in term disambiguation, whereas the graph index is built to aid graphs research. These two indicators [63] are created by KAT during the offline procedure. Whenever a customer inputs a term for inquiry, the keyword indicator translates terms to related nodes and links on the RDF graph, known as keywords components. The graph index is then searched for subgraphs containing these keywords. A context-aware approach ranks the subgraphs, and the items at the top are converted to SPARQL inquiries. A few keywords are used to construct correct SPARQL queries. For disambiguation, RDF class data are stored in a keyword index and examined in a graph. For scoring, a context-aware technique is applied.

Seaborne and Prud'hommeaux [64] discussed the same problem of end users in utilizing SPARQL languages, with the familiar official logic expression and the supporting vocabularies. This has become a significant gap between the semantic search and the final user. Allowing users to carry out semantic searches by entering a keyword query is beneficial. These two searches are not the same. A solution is presented in which keyword inquiries are automatically converted into SPARQL queries so that the final customer can perform a semantic search using familiar keywords. The study contributes by introducing a prototype system called 'SPARK' [64] that retrieves a rating set of SPARQL query as a translation output when provided a keyword query. In this translation, the three primary processes are comparison terms, inquiry design creation, and inquiry ranting. These processes face three challenges to match term and semantic queries: Vocabulary deficit is where the underlying ontology is frequently unknown to casual online users. Lack of relationship is where, in formal logic inquiries, relationships between concepts/instances must be explicitly expressed, which is often lacking in keyword searches. Figure 10 shows that the question of how to detect these missing relationships automatically has become a significant concern. Inquiry ranking is where, given ambiguous keyword inquiries, numerous formal queries may be generated from a single keyword.

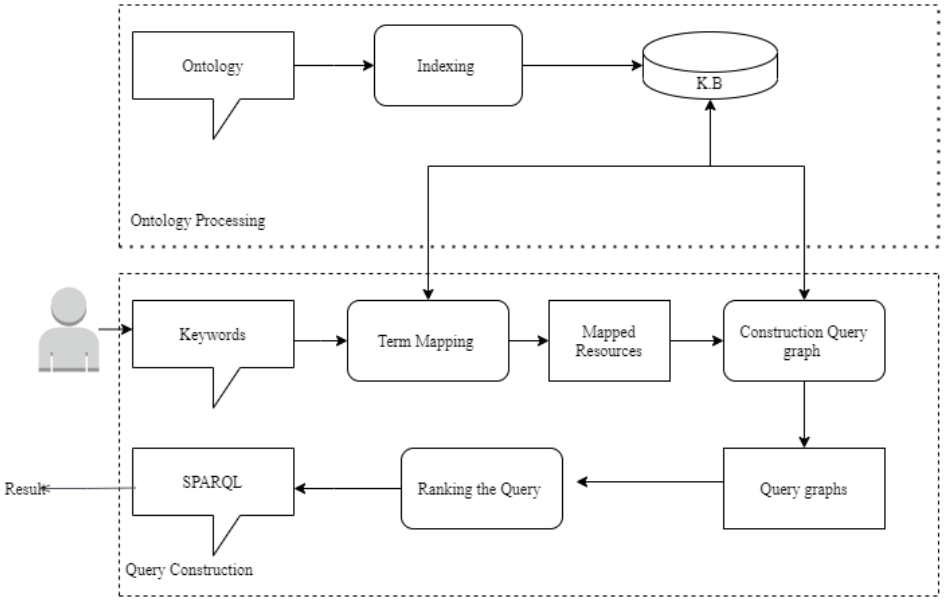

Figure 10. Spark framework.

The authors created a framework dealing with a keyword called Q2Semantic to present their method [65]. They used words excluded from Wikipedia to provide literal determines in the source RDF data. They use various query ranking techniques that account for various parameters such as query length and ontology element relevance. They also suggested an exploration technique as well as a new graph design known as a clustered graph that merely is an exclusion of the main RDF, allowing for scalability. Additionally, the developed algorithm enables the creation of top-k questions that could aid in faster completion of the interpretation process. Q2Semantic has a search interface that is based on keywords. Auto-completion is available in this interface. This feature takes advantage of the underlying RDF literals and enhanced Wiki words to help the customer type his terms. The user can choose from a list of sentences generated by Q2Semantic that contain these keywords. Their drawback was that this approach only supported terms that matched concepts and literals in RDF data (as concepts were shown as special K-Nodes in existing implementation). However, they did not add keyword support as relations and attributes to the present query capabilities.

Additionally, Gkirtzou et al. [66] presented an overview of their RDF keyword search progress. Their method generates a list of SPARQL queries and their NL specifications based on a list of keywords, which can then be assessed on the RDF structure. This approach consists of three steps: (a) convert term elements to structured data; (b) link term elements by looking for structural components in the data; and (c) consider a score function and give the found structural features. This study intended to work in the following directions in the future: investigate alternate approaches for constructing query pattern graphs, search diversification semantics and keyword-to-node matching string-matching methods.

5.4.3. User Selection Techniques

In [67], Zenz et al. faced a problem building semantic queries as an exacting job for customers because SPARQL language demands control, and the vocabularies have been utilized to collect data. As a solution, QUICK [67], a revolutionary system, is applied to assist users in constructing semantic inquiries in a particular area. The expressivity of semantic questions is combined with the convenience of keyword search in QUICK. Users are guided with advanced refining procedures to determine the query intention, starting with a keyword inquiry. The users must also overcome various challenges, such as learning SPARQL query language and gaining sufficient knowledge of the data source's ontology or schema. A structure for gradually building SPARQL query from user words

is among the suggested techniques to output near-exact inquiry structure guides, which allow users to build a SPARQL query quickly. Tests were carried out to determine the effectiveness of QUICK and the efficiency of the suggested algorithms. The QUICK system works in the following ways: the user enters keywords that can indicate a wide range of semantic searches in addition to the one intended. QUICK then computes all potential semantic inquiries and displays their subset in the query window. In addition, a list of query-building choices is created.

If the desired query is unavailable, the user can build the query progressively by using the construction pane. The query pane adjusts to the new settings, zooming in on the subset of semantic searches that meet the criteria. A fresh set of construction choices is produced and shown within the construction pane. This set of building options is known as a guide that directs users toward the requested inquiry. QUICK executes the query chosen by the user and outputs the answers. For a query that accurately matches the user intent, the answers can be retrieved by converting to SPARQL and are analyzed against an RDF store.

In [65], Wang et al. discussed the problem with improving data quantity on SEW for semantic inquiry. As a result, the user must understand the formal query's complex syntax. Furthermore, the user must know the underlying structure and the RDF literals. One answer to this problem is the keyword interface that has become familiar to users, given its frequent use. Over formal questions, keyword queries provide the following advantages: Syntax is not complex due to the sample set of term phrases, and Open ontologies allow customers to describe their information demands using their own terms. The study faced several challenges or obstacles: How do you handle keyword phrases that are stated in the user's language but do not occur in RDF data? How do you identify the appropriate query when keywords are confusing (ranking)? How can relevant queries be returned as rapidly as possible (scalability)?

In addition, ref. [68] Djebali and Raimbault introduced utilizing terms in addition to SPARQL components to search the Web of Data through present SPARQL endpoints [7]. As a result, the user can create more expressive pseudo-SPARQL searches without knowing an RDF-based vocabulary (classes and attributes) or resource IDs. The limits of the methodologies presented in prior papers were explored in this work. Those techniques restrict the investigation of historically and locally referenced (and frequently local preserved) RDF base sources, at the risk of losing data synchronization: it is the sacrifice to make for a pleasant experience researching data into the databases. Its objective is to expand the SPARQL expressiveness by enabling it to search an RDF base without understanding its design (i.e., ontological vocabulary) or resource IDs. They narrowed the gap between a machine's interpretation capabilities and how a user comprehends data.

6. Challenges of Modern SQA Systems

This section presents the challenges that faced modern SQA systems in the following. Table 4 summarizes the challenges that face some important methods through creating SQA with its used techniques and presents the output of each method.

Table 4. Challenges of QAS.

Challenges That Faced Modern SQA Systems	Papers That Solve This Problem	Techniques or Approaches Used	Outputs or Outcomes
Lexical Gap	[69]	Automatically Query Expansion (AQE) Normalization and string similarity algorithms.	Keyword Query Expansion on linked data using linguistic and semantic features framework.
	[70]	A corpus and a knowledge base.	Build linguistic patterns and generated template.
	[45]	A corpus of plagiarisms from the QA site Answers from the Wiki.	Proposed a lexicon that was educated using a corpus of plagiarisms from the QA site Answers from the Wiki.
Multilingualism	[71]	Using existing mappings between multiple Wikipedia language versions, which are handed over to DBpedia, are automatically extended.	QAKiS system.
Complex Queries	[72]	Using grammatically rules as Comparisons, negations, and quantifiers.	Ontology-based SQA system.
	[73]	Synfragments that transfer to a sub tree of the syntactic parse tree.	Outputs its Synfragments can yield a mapping of a question in any SPARQL query.
Distributed Knowledge	[74]	Using several knowledge bases to search for entities and combine the results.	System used PROTON ontology and BLOOMS System.
Temporal, Spatial and Procedural Questions	[75]	Document retrieval technique.	KOMODO system.
	[76]	Allen's Interval Based Temporal Logic but allowed both in intervals and temporally instants to be used.	Clinical Narrative Temporal. Relation Ontology (CNTRO).
Templates	[77]	Used the question type, named entities, and POS tagging methodologies to create graph pattern templates. WordNet, PATTY, and similarity measurements are used to map the resulting graph patterns to resources.	well-formed NL phrases for the domain.

6.1. Lexical Gap

In NL, the exact meaning can be expressed in various ways. The RDF label property provides natural language descriptions of RDF resources. Since synonyms for much the same RDF resource might be specified using a variety of labels, knowledge bases seldom include all of the available expressions for a particular item. A lexical gap arises whenever the language used during a query is different from that used in the knowledge base labels. Normalization and similarity functions for string normalizations are used to bridge the lexical gap, such as converting from uppercase to lowercase or from base forms to base forms, as "é" to "e". If normalizations are not enough, similarity, a similarity function, and a threshold can be utilized to measure distance and its complementary notion. Automatically query expansion (AQE) normalization and string similarity algorithms identify distinct variants of the same word but not synonyms. Synonyms, such as design and plan, are terms that have the same meaning, whether they are used together or separately. In hyper-hyponym pairings, such as biological process and photosynthesis, the first term is less detailed than the second. These word combinations are utilized as supplementary labels in automatic query expansion and are derived from WordNet and other lexical databases [69].

AQE is often employed in traditional search engines and information retrieval; if there is not a direct match, it can be used as a fallback alternative. One of the publications surveyed is experimental research [78] that looked at the influence of AQE on SQA. With pattern libraries using only similarity functions and normalization, from a term to a resource, RDF people may be precisely matched. However, because (1) they determine the S and O that could be in different positions and (2) a particular P can be stated in several ways, including as a noun and a verb phrase that could or could not be a continuous segment of the query, the properties require additional treatment. Libraries to overcome these challenges have been produced due to the intricate and variable nature of those linguistic patterns and the required thinking and expertise. Nakashole et al. [79] presented the impact of pattern libraries on SQA; PATTY finds items in a corpus of texts and calculates the shortest path among them. After that, the path is extended with any modifiers and recorded as a template. As a result, PATTY might construct a pattern library from any corpus-based knowledge base.

In addition, Teije et al. [70] used a corpus and a knowledge base to build linguistic patterns. Sentences from a corpus providing instances of S and O for this special attribute are selected for each P in the knowledge base. Each resource couple linked in a phrase, according to BOA, illustrates another label for this relationship, and each happening of that word pair in the corpus generates a template.

Moreover, Fader et al. [45] proposed a lexicon that was educated using a corpus of plagiarisms from the QA site, Answers from Wiki. The advantage was that no manual templates were needed because they are automatically learned from paraphrasing.

6.2. Ambiguity

Because of the particular character of NL inquiries and semantic resources, ambiguity is a vital issue in any SQA system [80]. Ambiguity occurs when the same term has multiple meanings, which may be lexical and meaningful or structural and grammatical. They differentiated among homonymy, which occurs when the exact string denotes two distinct but related concepts. For example, riverbank vs. money bank and polysemy. Whenever two different but related ideas are referenced by the same string (for example, money bank vs. riverbank) (as in bank as a building vs. bank as a company), synonymy is distinguished from taxonomic relationships as metonymy and hypernymy. In opposite to the lexical gap that hinders an SQA system's recall, ambiguity has a detrimental impact on its precision. The lexical gap's absolute opposite is ambiguity [81].

In SQA, ambiguity occurs when a word or a statement has many meanings. As a result, when linguistic triples match numerous KB concepts, when linguistic triples do not suit any KB notion, the SQA system requires disambiguation solutions to determine the

proper meaning and enumerate similar terms. Depending on the information source and type used to answer this mapping, we differentiated among two kinds of disambiguation: Firstly, traditional corpus-based approaches rely on counts from unstructured text corpora, which are commonly utilized as probabilities, like statistical approaches in [82]. Secondly, resource-based approaches in SQA use the reality that candidate ideas are RDF resources to your advantage. Different scoring techniques are used to compare resources based on their attributes and relationships. A rating for all resources selected in the mapping is assumed where those resources are more likely to be related, which means those resources are more likely to be chosen correctly.

In the literature, there are 13 different forms of ambiguity. Semantic or syntactic/structural ambiguity can cause ambiguity at the word, word (group), and sentence level. Semantic ambiguity means ambiguity in meaning. The definition of the sentence(s) and word(s) is the subject of semantic ambiguity. In a particular scenario, a sentence, a word, or a phrase might have several meanings [80]. There are many cases of semantic ambiguity, such as homonym and contrastive polysemy. Another name for homonym is a term with more than one meaning [83]. Another example of a term with multiple meanings is polysemous. Both senses of the terminology, unlike homonyms, relate to the same concept. Hypernym [84] is an instance or a noun with a subtype that is also a noun. Meronym [85] is a concept that is both a portion of and a description of the whole. Morphology is a word that refers to ambiguity. Synonym [86] is when two or more words have the same meaning. Approach and method are two different terms with the same sense, for example. Pragmatism relates to how language is used to convey a speaker's meaning in given contexts, primarily when the terms used represent something distinct, indeterminacy, and generality. We analyzed numerous definitions of generality and found that the most relevant meaning is as shown in: "a term is generally in regard with another word if the former's connotation is a genus of the latter's connotation". The term "parents" has ambiguous generality because it might refer to either a father or a mother.

6.3. Multilingualism

On the internet, information is defined in several various languages. As language tags could be used to describe RDF resources in multiple languages simultaneously, web publications do not necessarily utilize the same language. Furthermore, users speak a variety of native languages. As a result, SQA systems that can accept a variety of incoming languages, including ones that are not the same as the languages used to express the knowledge, are more flexible. In [87], authors discussed this challenge to address it through only a portion of the query must be translated appropriately, after which semantic relevance and similarity measurements between resources in the knowledge base associated with the first one aid in recognition of the last things. Additionally, for the QAKiS system in [52,71], existing mappings between multiple Wikipedia language versions, which are handed over to DBpedia, are automatically extended.

6.4. Complex Queries

The most common way to answer simple questions is to translate them to a group of not complex triple patterns. Numerous facts must be merged, discovered, and then problems arise. Queries can also specify a particular return order and aggregated or filtered results. Adolphs et al. discussed this challenge through the YAGO-QA system in [88] allowed nested questions. Unger and Cimiano [72] developed an ontology-based SQA system with an autonomously produced ontology-specific vocabulary. Thanks to the linguistic representation, the system can answer NL questions with grammatically more complicated inquires, such as comparisons, negations, quantifiers and numerals, superlatives, and so on.

Moreover, Dima [73] discussed Built on DBpedia; the SQA system relies on synfragments that transfer to a subtree of the syntactic parse tree. A synfragment is a tiny fragment of text which could be understood as an RDF triple or a complicated RDF inquiry in terms

of semantics. Combining all possible child synfragment groupings, ordered by semantic and syntactic properties, synfragments communicate with their parent synfragment. The authors assumed that iteratively interpreting its synfragments can yield the mapping of a question in any SPARQL query. Furthermore, in [89], Delmonte started by converting a question into a logical form, which includes arguments, a predicate, and concentration. The emphasis element identifies the desired answer type. For example, "Who is the major of New York?" focuses on the word "person", the P is "be", and the argument "major of New York".

A yes/no question is assumed if no focus element is found. The logic form is then translated to a SPARQL query in the second stage, which involves label matching to map items to resources. Because the direction is unknown, when unions refer to properties in both possible directions, as in (? x? p? o UNION? o? p? x), the resultant triple patterns are divided up again. Additional limitations that cannot be specified in a SPARQL query, such as "Who was the fifth president of the United States?" are handled by various filters.

6.5. Distributed Knowledge

If distributed RDF resources show a query's concept information, data needed to result in it could be lacking if only one or none of the knowledge bases are determined. Most ideas have one related resource in single datasets with one resource, DBpedia. This difficulty can be solved in the case of combined datasets by using the equivalent class, or equivalent property connections. Some systems addressed this challenge, such as the proposed system solution of Joshi et al. [74]. This challenge can be solved by first formulating inquiries regarding the PROTON [90] higher level ontology. The vocabulary is then aligned with those of additional knowledge sources using the BLOOMS [91] system. In addition, ref. [92] used several knowledge bases to search for entities and combine the results. Similarity measures are used to determine and rank each data source's results candidates and to find matches between entities from other data sources.

6.6. Temporal, Spatial and Procedural Questions

- **Procedural questions:** inquiries about the procedure. Factual, yes–no, and list questions were not complex to answer because they responded to SPARQL SELECT and ASK queries. Other questions, such as how (procedural)or why (causal), require more time and effort. SQA cannot currently tackle procedural QA because no ontologies include procedural knowledge currently available to the best of our knowledge. They illustrated the KOMODO [75] system, which is based on document retrieval to encourage more study in this area, even if it is not an SQA system. Instead of a statement, KOMODO provides a web page with step-by-step directions for accomplishing the user's goal.
- **Temporary questions [76]:** responding to a temporal query about clinical narratives. They proposed the clinical narrative temporal relation ontology (CNTRO), which is modeled on Allen's interval-based temporal logic [93] but allowed both intervals and temporally instants to be used. This enables you to infer the temporal relationship between events based on the transitivity after and before events. For example, in CNTRO, patient measurements, findings, and operations are modeled as events, with time provided directly in date and in time of day or other events and timings.
- **Spatial questions:** In RDF, questions concerning space and location can be stated as two-dimensional geo-coordinates with latitude and longitude, but three-dimensional formulations seem more challenging to convey (for example, with additional height). Spatial relationships, on the other hand, could be expressed as easier to respond to because users often inquire about relationships rather than precise geo-coordinates. As an inverted index, suggested in [94], Younis et al. added spatial interactions such as crossing, inclusion, and proximity to semantic data for named entity recognition. After that, SPARQL searches can use this information.

6.7. Templates

To understand the underlying inquiry's design, sophisticated algorithms are necessary for complex questions using more than one basic graph structure in the SPARQL query. Current research is divided into two categories: template-based techniques, which seek to construct SPARQL queries based on the input question's syntactic structure, and template-free techniques, which aim at developing SPARQL queries depending on the input question's syntactic structure. Additionally, some systems have attempted to address this difficulty through many template-driven techniques, such as Unger et al. [59] and Shekarpour et al. [95] have offered the first solution. In addition, Liu et al. [77] used the question type, named entities, and POS tagging methodologies to create graph pattern templates. WordNet, PATTY, and similarity measurements are used to map the resulting graph patterns to resources. Finally, SPARQL queries are built using the various graph pattern combinations.

Moreover, in [96], Ben Abacha and Zweigenbaum focused on a specific biomedical patient-treatment area and used a combination of manual template generation and machine learning to achieve their goal. Damovo et al. [97] produced well-formed NL phrases for the domain of paintings utilizing a template with settings that may be changed. The system uses the grammatical framework [98] to place an intermediate step of a multilingual description among the input keyword and the formal query, enabling the system to support 15 languages. Additionally, in TPSM, the open domain three-phase semantic mapping [99] framework went beyond SPARQL searches. It uses fuzzy constraint satisfaction problems to map natural language inquiries to OWL queries. Surface text matching, POS tag preference, and the degree of surface form similarity are all constraints. The collection of specific mapping components acquired using the FCSP-SM method is combined into a model using specified templates.

7. Evaluation of Semantic Web Techniques

This review paper evaluates and presents a thorough overview of semantic web technologies, their associated domains, and those specifically related domain research directions, which provides future directions and guidance for researchers. More automated reuse, self-wiring, and enclosed modules with benchmarks and evaluations should include all focus on future study. The development of SQA systems that can comprehend noisily human natural language input from multiple languages and knowledge sources is currently underway. Distributed, dynamic, incoherent, and very sensitive to privacy issues is the semantic web data space. Three domains contribute to the development of the semantic web. Computational intelligence is the first area, followed by evolutionary and swarm computing, and finally, methods for knowledge representation. We store massive amounts of data and offer reasoning over the web with the use of swarm computing [100].

The techniques for addressing challenges with ambiguity and uncertainty are provided by computational intelligence. Data are stored more consistently and coherently thanks to knowledge representation techniques by using the aforementioned criteria: WDAqua-core1 [101], ganswer2 [102], WDAqua [103], Frankenstein [3], and system in [104]. The following are the reasons why we chose these SPARQL-based systems to compare. The two systems, WDAqua and ganswer2, achieved the best results on the QALD-7 dataset, according to the QALD-7 study [105]. The system WDAqua-core1 has the greatest performance on the LC-QuAD dataset [106], according to [107]. The performance of the system in [104] is compared to published systems WDAqua-core1 and Frankenstein in Table 5. On the LC-QuAD dataset, this comparison shows that the QA system in [104] extensively performs better QA systems. We used the LC-QuAD dataset to test this QA system in [104] on 2430 questions that still apply to the latest SPARQL endpoint version (2019-06).

Table 6 shows that the QA system in [104] performs better than systems WDAqua and ganswer2 on the QALD-7 dataset. For 968 questions in the LC-QuAD dataset and 80 questions in the QALD-7 dataset, no SPARQL query was generated, according to an in-depth study of the unsuccessful questions to the system in [104]. Most of these errors

occurred during the phrase mapping when the required resources, attributes, or classes were not found. So from our understanding, the only QA systems that could be compared and reviewed were those whose source code was publicly available or whose methodology had been examined using established benchmark datasets such as LC-QuAD or QALD.

Table 5. Performance evaluation on the LC-QuAD dataset.

Metrics	WDAqua-Core1 [34]	Frankenstein [3]	[108]
Precision	0.59	0.20	0.88
Recall	0.38	0.21	0.56
F1-score	0.46	0.20	0.68

Table 6. Performance evaluation on the QALD-7 dataset.

Metrics	WDAqua [25]	Ganswer2 [26]	[108]
Precision	0.488	0.557	0.813
Recall	0.535	0.592	0.527
F1-score	0.511	0.556	0.639

8. Conclusions and Future Work

The present study briefly introduces techniques for semantic search on the Web. As the volume of data on the SEW increases, so do the opportunities for semantic search. This topic is among the most popular in the SEW and Web search communities. As a result, the user must understand the formal query's sophisticated syntax. Furthermore, the user must be aware of the underlying structure and the literals used in the RDF data to enable the semantic web technologies, such as OWL, RDFS, RDF, and rule and query languages, such as SPARQL. These technologies aid in the resolution of challenges in numerous sectors. This review paper discussed the semantic search approaches to convert the query from the native user to SPARQL query language by introducing some keyword-based semantic search engines and QAS. However, each used algorithms and techniques to achieve this goal.

Several keyword-based semantic search engines and QAS begin with an analysis of the semantic web's nature and requirements and then show different approaches with the same goal: to help users with NL post queries by SPARQL language on ontologies or RDF schema without a thorough knowledge of this technology. However, each method used algorithms and techniques to achieve this goal. Thus, the current problem to explore is how to convert the normal user keywords into SPARQL query language without knowledge of the language of semantic Web and its underlying ontology.

This review discussed various papers presenting techniques such as quick, spark, TBSL, and SINA frameworks. To apply semantic search, each of them faced several limitations such as lexical gap, temporary questions, templates, temporal spatial and procedural questions, complex queries, multilingualism, and ambiguity.

Our next step is creating QAS that attempts to solve many problems facing state-of-the-art systems and improve their performance. The diversity of training data is limited because the currently available training datasets only contain three categories of questions. In actuality, many more types of inquiries are frequently asked. FILTER, LIMIT, ORDER, MIN, MAX, UNION, and other commonly used SPARQL operators were not considered. One obvious route for future research is to collect complicated questions containing the listed operators to increase the training dataset's size and quality. In the training dataset, the number of questions for each class should be fairly consistent. Additionally, numerous expressions for the same question should be produced to expand the training dataset's amount and variety and improve system performance.

Author Contributions: E.H.H.: Supervision, methodology, conceptualization, formal analysis, investigation, visualization, Writing—review and editing. N.I.: Conceptualization, Methodology, Resources, Writing—original draft. A.M.Z.: Visualization, conceptualization, formal analysis, methodology, Writing—review and editing. A.S.: Visualization, conceptualization, formal analysis, methodology, Writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: No funding was received for this work.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: This article does not contain any studies with human participants or animals performed by any of the authors.

Data Availability Statement: Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Unni, M.; Baskaran, K. Overview of approaches to semantic web search. *Int. J. Comput. Sci. Commun. (IJCSC)* **2011**, *2*, 345–349.
2. Gupta, P.; Gupta, V. A survey of text question answering techniques. *Int. J. Comput. Appl.* **2012**, *53*. [CrossRef]
3. Singh, K.; Radhakrishna, A.S.; Both, A.; Shekarpour, S.; Lytra, I.; Usbeck, R.; Vyas, A.; Khikmatullaev, A.; Punjani, D.; Lange, C.; et al. Why reinvent the wheel: Let's build question answering systems together. In Proceedings of the 2018 World Wide Web Conference, Lyon, France, 23–27 April 2018; pp. 1247–1256.
4. Lopez, V.; Unger, C.; Cimiano, P.; Motta, E. Evaluating question answering over linked data. *J. Web Semant.* **2013**, *21*, 3–13. [CrossRef]
5. Hirschman, L.; Gaizauskas, R. Natural language question answering: The view from here. *Nat. Lang. Eng.* **2001**, *7*, 275–300. [CrossRef]
6. Baeza-Yates, R.; Raghavan, P. Next generation Web search. In *Search Computing*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 11–23.
7. Rodrigo, Á.; Perez-Iglesias, J.; Peñas, A.; Garrido, G.; Araujo, L. A Question Answering System based on Information Retrieval and Validation. In *Proceedings of the CLEF (Notebook Papers/LABs/Workshops)*; Universidad Nacional de Educación a Distancia: Madrid, Spain, 2010.
8. Lopez, V.; Uren, V.; Sabou, M.; Motta, E. Is question answering fit for the semantic web? A survey. *Semant. Web* **2011**, *2*, 125–155. [CrossRef]
9. Zafar, H.; Dubey, M.; Lehmann, J.; Demidova, E. IQA: Interactive query construction in semantic question answering systems. *J. Web Semant.* **2020**, *64*, 100586. [CrossRef]
10. Bouziane, A.; Bouchiha, D.; Doumi, N.; Malki, M. Question answering systems: Survey and trends. *Procedia Comput. Sci.* **2015**, *73*, 366–375. [CrossRef]
11. Lee, J.; Min, J.K.; Oh, A.; Chung, C.W. Effective ranking and search techniques for Web resources considering semantic relationships. *Inf. Process. Manag.* **2014**, *50*, 132–155. [CrossRef]
12. Fazzinga, B.; Lukasiewicz, T. Semantic search on the Web. *Semant. Web* **2010**, *1*, 89–96. [CrossRef]
13. Horrocks, I.; Parsia, B.; Patel-Schneider, P.; Hendler, J. Semantic web architecture: Stack or two towers? In *Proceedings of the International Workshop on Principles and Practice of Semantic Web Reasoning*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 37–41.
14. Harris, S.; Seaborne, A. SPARQL 1.1 Overview. W3C Recommendation, 21 March 2013. 2013. Available online: https://www.w3.org/TR/sparql11-overview/ (accessed on 28 August 2022).
15. d'Aquin, M.; Gridinoc, L.; Angeletou, S.; Sabou, M.; Motta, E. Watson: A gateway for next generation semantic web applications. In Proceedings of the 6th International Semantic Web Conference (ISWC 2007), Busan, Korea, 11–15 November 2007.
16. Al Muqrishi, A.; Sayed, A.; Kayed, M. Caseng: Arabic Semantic Search Engine. *J. Theor. Appl. Inf. Technol.* **2015**, *75*. Available online: https://www.researchgate.net/publication/282284960_Caseng_Arabic_semantic_search_engine (accessed on 28 August 2022).
17. Madhu, G.; Govardhan, D.A.; Rajinikanth, D.T. Intelligent semantic web search engines: A brief survey. *arXiv* **2011**, arXiv:1102.0831.
18. Seaborne, A.; Manjunath, G.; Bizer, C.; Breslin, J.; Das, S.; Davis, I.; Harris, S.; Idehen, K.; Corby, O.; Kjernsmo, K.; et al. SPARQL/Update: A language for updating RDF graphs. *W3c Memb. Submiss.* **2008**, *15*. Available online: https://www.researchgate.net/publication/316898305_SPARQL_Update_-_A_Language_for_Updating_RDF_Graphs (accessed on 28 August 2022).
19. Royo, J.A.; Mena, E.; Bernad, J.; Illarramendi, A. Searching the web: From keywords to semantic queries. In Proceedings of the Third International Conference on Information Technology and Applications (ICITA'05), Sydney, NSW, Australia, 4–7 July 2005; Volume 1, pp. 244–249.

20. El-Gayar, M.; Mekky, N.E.; Atwan, A.; Soliman, H. Enhanced search engine using proposed framework and ranking algorithm based on semantic relations. *IEEE Access* **2019**, *7*, 139337–139349. [CrossRef]
21. Antoniou, G.; Van Harmelen, F. *A Semantic Web Primer*; MIT Press: Cambridge, MA, USA, 2004.
22. Amazon Alexa, C.D.C. Evi Search Engine. 2005. Available online: http://www.evi.com (accessed on 28 August 2022).
23. Alpha, W. Alpha Search Engine. 2020. Available online: https://www.wolframalpha.com/ (accessed on 28 August 2022).
24. Frank, A.; Krieger, H.U.; Xu, F.; Uszkoreit, H.; Crysmann, B.; Jörg, B.; Schäfer, U. Question answering from structured knowledge sources. *J. Appl. Log.* **2007**, *5*, 20–48. [CrossRef]
25. Şenel, L.K.; Utlu, I.; Yücesoy, V.; Koç, A.; Çukur, T. Generating semantic similarity atlas for natural languages. In Proceedings of the 2018 IEEE Spoken Language Technology Workshop (SLT), Athens, Greece, 18–21 December 2018; pp. 795–799.
26. Gunasekara, L.; Vidanage, K. Uniontbot: Semantic natural language generation based api approach for chatbot communication. In Proceedings of the 2019 National Information Technology Conference (NITC), Colombo, Sri Lanka, 8–10 October 2019; pp. 1–8.
27. Peelar, S.; Frost, R. A Compositional Semantics for a Wide-Coverage Natural-Language Query Interface to a Semantic Web Triplestore. In Proceedings of the 2020 IEEE 14th International Conference on Semantic Computing (ICSC), San Diego, CA, USA, 3–5 February 2020; pp. 257–262.
28. Maksutov, A.A.; Zamyatovskiy, V.I.; Vyunnikov, V.N.; Kutuzov, A.V. Knowledge base collecting using natural language processing algorithms. In Proceedings of the 2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), St. Petersburg and Moscow, Russia, 27–30 January 2020; pp. 405–407.
29. Li, W. Analysis of Semantic Comprehension Algorithms of Natural Language Based on Robot's Questions and Answers. In Proceedings of the 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA), Dalian, China, 25–27 August 2020; pp. 1021–1024.
30. Gupta, P.; Goswami, A.; Koul, S.; Sartape, K. IQS-intelligent querying system using natural language processing. In Proceedings of the 2017 International Conference of Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 20–22 April 2017; Volume 2, pp. 410–413.
31. Sarker, J.; Billah, M.; Al Mamun, M. Textual question answering for semantic parsing in natural language processing. In Proceedings of the 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), Dhaka, Bangladesh, 3–5 May 2019; pp. 1–5.
32. Sadhuram, M.V.; Soni, A. Natural language processing based new approach to design factoid question answering system. In Proceedings of the 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 15–17 July 2020; pp. 276–281.
33. Lopez, V.; Fernández, M.; Motta, E.; Stieler, N. Poweraqua: Supporting users in querying and exploring the semantic web. *Semant. Web* **2012**, *3*, 249–265. [CrossRef]
34. Damljanovic, D.; Agatonovic, M.; Cunningham, H. FREyA: An interactive way of querying Linked Data using natural language. In *Proceedings of the Extended Semantic Web Conference*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 125–138.
35. Comparot, C.; Haemmerlé, O.; Hernandez, N. An easy way of expressing conceptual graph queries from keywords and query patterns. In *Proceedings of the International Conference on Conceptual Structures*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 84–96.
36. Höffner, K.; Lehmann, J. Towards question answering on statistical linked data. In Proceedings of the 10th International Conference on Semantic Systems, Leipzig, Germany, 4–5 September 2014; pp. 61–64.
37. Höffner, K.; Lehmann, J.; Usbeck, R. CubeQA—Question answering on RDF data cubes. In *Proceedings of the International Semantic Web Conference*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 325–340.
38. Usbeck, R.; Ngomo, A.C.N.; Bühmann, L.; Unger, C. Hawk–hybrid question answering using linked data. In *Proceedings of the European Semantic Web Conference*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 353–368.
39. Demidova, E.; Zhou, X.; Nejdl, W. Efficient query construction for large scale data. In Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, Dublin, Ireland, 28 July–1 August 2013; pp. 573–582.
40. Aggarwal, N.; Buitelaar, P. A System Description of Natural Language Query over DBpedia. In Proceedings of the ILD@ ESWC, Heraklion, Greece, 28 May 2012; pp. 96–99.
41. Cheng, G.; Tran, T.; Qu, Y. Relin: Relatedness and informativeness-based centrality for entity summarization. In *Proceedings of the International Semantic Web Conference*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 114–129.
42. Frost, R.A.; Donais, J.; Mathews, E.; Agboola, W.; Stewart, R. A demonstration of a natural language query interface to an event-based semantic web triplestore. In *Proceedings of the European Semantic Web Conference*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 343–348.
43. Liu, Z.; Chen, X. A graph-based text similarity algorithm. In *Proceedings of the 2012 National Conference on Information Technology and Computer Science*; Atlantis Press: Amsterdam, The Netherlands, 2012; pp. 614–617. [CrossRef]
44. Hakimov, S.; Tunc, H.; Akimaliev, M.; Dogdu, E. Semantic question answering system over linked data using relational patterns. In Proceedings of the Joint EDBT/ICDT 2013 Workshops, Genoa, Italy, 18–22 March 2013; pp. 83–88.
45. Fader, A.; Zettlemoyer, L.; Etzioni, O. Paraphrase-driven learning for open question answering. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Sofia, Bulgaria, 4–9 August 2013; pp. 1608–1618.

46. Ngonga Ngomo, A.C.; Bühmann, L.; Unger, C.; Lehmann, J.; Gerber, D. SPARQL2NL: Verbalizing SPARQL queries. In Proceedings of the 22nd International Conference on World Wide Web, Rio de Janeiro, Brazil, 13–17 May 2013; pp. 329–332.
47. Chergui, O.; Begdouri, A.; Groux-Leclet, D. Integrating a Bayesian semantic similarity approach into CBR for knowledge reuse in Community Question Answering. *Knowl.-Based Syst.* **2019**, *185*, 104919.
48. Freitas, A.; Oliveira, J.G.; O'riain, S.; Da Silva, J.C.; Curry, E. Querying linked data graphs using semantic relatedness: A vocabulary independent approach. *Data Knowl. Eng.* **2013**, *88*, 126–141. [CrossRef]
49. Shin, S.; Jin, X.; Jung, J.; Lee, K.H. Predicate constraints based question answering over knowledge graph. *Inf. Process. Manag.* **2019**, *56*, 445–462. [CrossRef]
50. Shekarpour, S.; Marx, E.; Ngomo, A.C.N.; Auer, S. Sina: Semantic interpretation of user queries for question answering on interlinked data. *J. Web Semant.* **2015**, *30*, 39–51. [CrossRef]
51. Sun, H.; Ma, H.; Yih, W.T.; Tsai, C.T.; Liu, J.; Chang, M.W. Open domain question answering via semantic enrichment. In Proceedings of the 24th International Conference on World Wide Web, Florence, Italy, 18–22 May 2015; pp. 1045–1055.
52. Cojan, J.; Cabrio, E.; Gandon, F. Filling the gaps among DBpedia multilingual chapters for question answering. In Proceedings of the 5th Annual ACM Web Science Conference, Paris, France, 2–4 May 2013; pp. 33–42.
53. Wendt, M.; Gerlach, M.; Düwiger, H. Linguistic modeling of linked open data for question answering. In *Proceedings of the Extended Semantic Web Conference*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 102–116.
54. Abacha, A.B.; Zweigenbaum, P. MEANS: A medical question-answering system combining NLP techniques and semantic Web technologies. *Inf. Process. Manag.* **2015**, *51*, 570–594. [CrossRef]
55. Xu, K.; Zhang, S.; Feng, Y.; Zhao, D. Answering natural language questions via phrasal semantic parsing. In *Proceedings of the CCF International Conference on Natural Language Processing and Chinese Computing*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 333–344.
56. Hakimov, S.; Unger, C.; Walter, S.; Cimiano, P. Applying semantic parsing to question answering over linked data: Addressing the lexical gap. In *Proceedings of the International Conference on Applications of Natural Language to Information Systems*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 103–109.
57. Gliozzo, A.M.; Kalyanpur, A. Predicting lexical answer types in open domain QA. *Int. J. Semant. Web Inf. Syst. (IJSWIS)* **2012**, *8*, 74–88. [CrossRef]
58. Ren, M.; Huang, H.; Gao, Y. SKR-QA: Semantic ranking and knowledge revise for multi-choice question answering. *Neurocomputing* **2021**, *459*, 142–151. [CrossRef]
59. Unger, C.; Bühmann, L.; Lehmann, J.; Ngonga Ngomo, A.C.; Gerber, D.; Cimiano, P. Template-based question answering over RDF data. In Proceedings of the 21st International Conference on World Wide Web, Lyon, France, 16–20 April 2012; pp. 639–648.
60. Shah, U.; Finin, T.; Joshi, A.; Cost, R.S.; Matfield, J. Information retrieval on the semantic web. In Proceedings of the Eleventh International Conference on Information and Knowledge Management, McLean, VA, USA, 4–9 November 2002; pp. 461–468.
61. Jain, S.; Seeja, K.; Jindal, R. A fuzzy ontology framework in information retrieval using semantic query expansion. *Int. J. Inf. Manag. Data Insights* **2021**, *1*, 100009. [CrossRef]
62. Ramada, M.S.; da Silva, J.C.; de Sá Leitão-Júnior, P. From keywords to relational database content: A semantic mapping method. *Inf. Syst.* **2020**, *88*, 101460. [CrossRef]
63. Wen, Y.; Jin, Y.; Yuan, X. KAT: Keywords-to-SPARQL translation over RDF graphs. In *Proceedings of the International Conference on Database Systems for Advanced Applications*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 802–810.
64. Prudhommeaux, E. SPARQL Query Language for RDF. 2008. Available online: http://www.w3.org/TR/rdf-sparql-query/ (accessed on 26 March 2013).
65. Wang, H.; Zhang, K.; Liu, Q.; Tran, T.; Yu, Y. Q2semantic: A lightweight keyword interface to semantic search. In *Proceedings of the European Semantic Web Conference*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 584–598.
66. Gkirtzou, K.; Papastefanatos, G.; Dalamagas, T. RDF keyword search based on keywords-to-SPARQL translation. In Proceedings of the First International Workshop on Novel Web Search Interfaces and Systems, Melbourne, Australia, 23 October 2015; pp. 3–5.
67. Zenz, G.; Zhou, X.; Minack, E.; Siberski, W.; Nejdl, W. From keywords to semantic queries—Incremental query construction on the Semantic Web. *J. Web Semant.* **2009**, *7*, 166–176. [CrossRef]
68. Djebali, S.; Raimbault, T. SimplePARQL: A new approach using keywords over SPARQL to query the web of data. In Proceedings of the 11th International Conference on Semantic Systems, Vienna, Austria, 16–17 September 2015; pp. 188–191.
69. Fellbaum, C. *Wordnet: An Electronic Lexical Database*; MIT Press: Cambridge, MA, USA, 1998. [CrossRef]
70. Janowicz, K.; Schlobach, S.; Lambrix, P.; Hyvönen, E. Knowledge Engineering and Knowledge Management. In *Proceedings of the 19th International Conference on Knowledge Engineering and Knowledge Management*; Springer: Berlin/Heidelberg, Germany, 2014.
71. Cabrio, E.; Cojan, J.; Aprosio, A.P.; Magnini, B.; Lavelli, A.; Gandon, F. QAKiS: An open domain QA system based on relational patterns. In Proceedings of the International Semantic Web Conference (ISWC 2012), Athens, Greece, 2 July 2015. Available online: https://hal.inria.fr/hal-01171115 (accessed on 28 August 2022).
72. Unger, C.; Cimiano, P. Pythia: Compositional meaning construction for ontology-based question answering on the semantic web. In *Proceedings of the International Conference on Application of Natural Language to Information Systems*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 153–160.
73. Dima, C. Intui2: A Prototype System for Question Answering over Linked Data. In *Proceedings of the CLEF (Working Notes)*; Citeseer: Valencia, Spain, 2013.

74. Joshi, A.K.; Jain, P.; Hitzler, P.; Yeh, P.Z.; Verma, K.; Sheth, A.P.; Damova, M. Alignment-based querying of linked open data. In *Proceedings of the OTM Confederated International Conferences on the Move to Meaningful Internet Systems*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 807–824.
75. Canitrot, M.; Roger, P.Y.; de Filippo, T.; Saint-Dizier, P. The KOMODO system: Getting recommendations on how to realize an action via Question-Answering. In Proceedings of the KRAQ11 Workshop, Chiang Mai, Thailand, 12 November 2011; pp. 1–9.
76. Tao, C.; Solbrig, H.R.; Sharma, D.K.; Wei, W.Q.; Savova, G.K.; Chute, C.G. Time-oriented question answering from clinical narratives using semantic-web techniques. In *Proceedings of the International Semantic Web Conference*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 241–256.
77. He, S.; Liu, S.; Chen, Y.; Zhou, G.; Liu, K.; Zhao, J. CASIA@ QALD-3: A Question Answering System over Linked Data. In *Proceedings of the CLEF (Working Notes)*; Citeseer: Valencia, Spain, 2013.
78. Shekarpour, S.; Höffner, K.; Lehmann, J.; Auer, S. Keyword query expansion on linked data using linguistic and semantic features. In Proceedings of the 2013 IEEE Seventh International Conference on Semantic Computing, Irvine, CA, USA, 16–18 September 2013; pp. 191–197.
79. Nakashole, N.; Weikum, G.; Suchanek, F. PATTY: A taxonomy of relational patterns with semantic types. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Jeju Island, Korea, 12–14 July 2012; pp. 1135–1145.
80. Hazrina, S.; Sharef, N.M.; Ibrahim, H.; Murad, M.A.A.; Noah, S.A.M. Review on the advancements of disambiguation in semantic question answering system. *Inf. Process. Manag.* **2017**, *53*, 52–69. [CrossRef]
81. Höffner, K.; Walter, S.; Marx, E.; Usbeck, R.; Lehmann, J.; Ngonga Ngomo, A.C. Survey on challenges of question answering in the semantic web. *Semant. Web* **2017**, *8*, 895–920. [CrossRef]
82. Shirai, K.; Inui, K.; Tanaka, H.; Tokunaga, T. An empirical study on statistical disambiguation of japanese dependency structures using a lexically sensitive language model. In *Proceedings of the Natural Language Pacific-Rim Symposium*; National Electronics and Computer Technology Center: Bangkok, Thailand, 1997; pp. 215–220.
83. Cleland, A.A.; Gaskell, M.G.; Quinlan, P.T.; Tamminen, J. Processing Semantic Ambiguity: Different Loci for Meanings and Senses. In Proceedings of the Annual Meeting of the Cognitive Science Society, San Diego, CA, USA, 20–26 June 2006; Volume 28.
84. Snow, R.; Jurafsky, D.; Ng, A.Y. Learning syntactic patterns for automatic hypernym discovery. *Adv. Neural Inf. Process. Syst.* **2004**, *17*. Available online: https://proceedings.neurips.cc/paper/2004 (accessed on 28 August 2022).
85. Girju, R.; Badulescu, A.; Moldovan, D. Automatic discovery of part-whole relations. *Comput. Linguist.* **2006**, *32*, 83–135.
86. Rindflesch, T.C.; Aronson, A.R. Ambiguity resolution while mapping free text to the UMLS Metathesaurus. In *Proceedings of the Annual Symposium on Computer Application in Medical Care*; American Medical Informatics Association: Boston, MA, USA, 1994; p. 240.
87. Aggarwal, N.; Polajnar, T.; Buitelaar, P. Cross-lingual natural language querying over the web of data. In *Proceedings of the International Conference on Application of Natural Language to Information Systems*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 152–163.
88. Adolphs, P.; Theobald, M.; Schafer, U.; Uszkoreit, H.; Weikum, G. YAGO-QA: Answering questions by structured knowledge queries. In Proceedings of the 2011 IEEE Fifth International Conference on Semantic Computing, Palo Alto, CA, USA, 18–21 September 2011; pp. 158–161.
89. Delmonte, R. *Computational Linguistic Text Processing–Lexicon, Grammar, Parsing and Anaphora Resolution*; Nova Science Publishers: New York, NY, USA, 2008.
90. Damova, M.; Kiryakov, A.; Simov, K.; Petrov, S. Mapping the central LOD ontologies to PROTON upper-level ontology. *Ontol. Matching* **2010**, *61*.
91. Jain, P.; Hitzler, P.; Sheth, A.P.; Verma, K.; Yeh, P.Z. Ontology alignment for linked open data. In *Proceedings of the International Semantic Web Conference*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 402–417.
92. Herzig, D.M.; Mika, P.; Blanco, R.; Tran, T. Federated entity search using on-the-fly consolidation. In *Proceedings of the International Semantic Web Conference*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 167–183.
93. Allen, J.F. Maintaining knowledge about temporal intervals. *Commun. ACM* **1983**, *26*, 832–843. [CrossRef]
94. Younis, E.M.; Jones, C.B.; Tanasescu, V.; Abdelmoty, A.I. Hybrid geo-spatial query methods on the Semantic Web with a spatially-enhanced index of DBpedia. In *Proceedings of the International Conference on Geographic Information Science*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 340–353.
95. Shekarpour, S.; Ngomo, A.C.N.; Auer, S. Query Segmentation and Resource Disambiguation Leveraging Background Knowledge. In Proceedings of the 14th International Semantic Web Conference, Bethlehem, PA, USA, 11–15 October 2015; pp. 82–93.
96. Ben Abacha, A.; Zweigenbaum, P. Medical question answering: Translating medical questions into sparql queries. In Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium, Miami, FL, USA, 28–30 January 2012; pp. 41–50.
97. Damova, M.; Dannélls, D.; Enache, R.; Mateva, M.; Ranta, A. Multilingual natural language interaction with semantic web knowledge bases and linked open data. In *Towards the Multilingual Semantic Web*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 211–226.
98. Ranta, A. Grammatical framework. *J. Funct. Program.* **2004**, *14*, 145–189. [CrossRef]
99. Gao, M.; Liu, J.; Zhong, N.; Chen, F.; Liu, C. Semantic mapping from natural language questions to OWL queries. *Comput. Intell.* **2011**, *27*, 280–314. [CrossRef]

100. Patel, A.; Jain, S. Present and future of semantic web technologies: A research statement. *Int. J. Comput. Appl.* **2021**, *43*, 413–422. [CrossRef]
101. Diefenbach, D.; Singh, K.; Maret, P. Wdaqua-core1: A question answering service for rdf knowledge bases. In Proceedings of the Companion Proceedings of the The Web Conference 2018, Lyon, France, 23–27 April 2018; pp. 1087–1091.
102. Zou, L.; Huang, R.; Wang, H.; Yu, J.X.; He, W.; Zhao, D. Natural language question answering over RDF: A graph data driven approach. In Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data, Snowbird, UT, USA, 22–27 June 2014; pp. 313–324.
103. Diefenbach, D.; Singh, K.; Maret, P. Wdaqua-core0: A question answering component for the research community. In *Proceedings of the Semantic Web Evaluation Challenge*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 84–89.
104. Liang, S.; Stockinger, K.; de Farias, T.M.; Anisimova, M.; Gil, M. Querying knowledge graphs in natural language. *J. Big Data* **2021**, *8*, 1–23. [CrossRef] [PubMed]
105. Usbeck, R.; Ngomo, A.C.N.; Haarmann, B.; Krithara, A.; Röder, M.; Napolitano, G. 7th open challenge on question answering over linked data (QALD-7). In *Proceedings of the Semantic Web Evaluation Challenge*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 59–69.
106. Trivedi, P.; Maheshwari, G.; Dubey, M.; Lehmann, J. Lc-quad: A corpus for complex question answering over knowledge graphs. In *Proceedings of the International Semantic Web Conference*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 210–218.
107. Diefenbach, D.; Both, A.; Singh, K.; Maret, P. Towards a question answering system over the semantic web. *Semant. Web* **2020**, *11*, 421–439. [CrossRef]
108. Kanakaraj, M.; Guddeti, R.M.R. Performance analysis of Ensemble methods on Twitter sentiment analysis using NLP techniques. In Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC 2015), Anaheim, CA, USA, 7–9 February 2015; pp. 169–170.