

Review

Challenges, Techniques, and Trends of Simple Knowledge Graph Question Answering: A Survey

Mohammad Yani ^{1,2} and Adila Alfa Krisnadhi ^{1,*}¹ Faculty of Computer Science, Universitas Indonesia, Depok 16424, Indonesia; mohammad.yani81@ui.ac.id² Politeknik Negeri Indramayu, Indramayu 45252, Indonesia

* Correspondence: adila@cs.ui.ac.id; Tel.: +62-21-786-3419

Abstract: Simple questions are the most common type of questions used for evaluating a knowledge graph question answering (KGQA). A simple question is a question whose answer can be captured by a factoid statement with one relation or predicate. Knowledge graph question answering (KGQA) systems are systems whose aim is to automatically answer natural language questions (NLQs) over knowledge graphs (KGs). There are varieties of researches with different approaches in this area. However, the lack of a comprehensive study to focus on addressing simple questions from all aspects is tangible. In this paper, we present a comprehensive survey of answering simple questions to classify available techniques and compare their advantages and drawbacks in order to have better insights of existing issues and recommendations to direct future works.

Keywords: knowledge graph; question answering; simple questions; survey



Citation: Yani, M.; Krisnadhi, AA. Challenges, Techniques, and Trends of Simple Knowledge Graph Question Answering: A Survey. *Information* **2021**, *12*, 271. <https://doi.org/10.3390/info12070271>

Academic Editors: Pierpaolo Basile and Annalina Caputo

Received: 12 May 2021
Accepted: 24 June 2021
Published: 30 June 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Knowledge graph questions answering (KGQA) systems enable users to retrieve data from a knowledge graph (KG) without requiring a complete understanding of the KG schema. Instead of formulating a precise query in a particular, formal query language, users can obtain data from a KG by formulating their information need in the form of a natural language question (NLQ).

Realizing such a system requires one to address the gap between the input NLQ, which is unstructured, and the desired answer represented by data in the KG, which are in a structured form and possibly specified under a complex schema. This challenge is unsurprisingly nontrivial since one needs to accomplish a number of subtasks to develop a well-performing KGQA system—for example, entity linking, relation linking, and answer retrieval. Research leading to various solutions to these tasks led to the emergence of KGQA as an exciting research direction. This is also indicated by the development of a number of KGQA systems in the last several years.

1.1. Motivation: Why Simple Questions?

This survey focuses on the problem of answering simple questions in KGQA, or, in short, KGSQA. Simple questions are distinguished from complex ones in that the former implicitly contain a single relation instance, while the latter contain more than one relation instances. For example, “Where is the capital of Indonesia?” is a simple question as it corresponds to just one relation instance *is capital of*, while “Who is the mayor of the capital of Japan?” is a complex question since it contains two relation instances, namely “is mayor of” and “is capital of”, assuming the aforementioned relation instances are part of the vocabulary of the target KG. Note that the number of relation instances in a question also represents the number of triple statements that need to be consulted from the KG to obtain an answer.

Despite the emergence of a number of recent KGQA systems designed to deal with complex questions [1–11], KGSQA remains a challenge that is not yet completely solved.

One reason is the fact that many KGQA systems designed for dealing with complex questions approach complex KGQA, i.e., answering complex question in KGQA, through decomposition of the input complex questions into simple questions [1–9]. For example, the question “Who is the mayor of the capital of Japan?” can be decomposed into two simple questions: “Who is the mayor of *_A_*?” and “*_A_* is the capital of Japan?”. Here, *_A_* represents an unknown or a variable and we view the second sentence as a question where *_A_* is the entity it asks, despite having no explicit question word. Since those KGQA systems employ solutions to KGSQA as key building blocks to the overall system, their end-to-end performance clearly depends on how well KGSQA is solved. Moreover, the decomposition approach generally outperforms other methods, hence putting an even stronger emphasis on the importance of solving KGSQA.

The importance of KGSQA solution for complex KGQA is further highlighted by the fact that even state-of-the-art KGQA systems designed to handle complex questions also suffer from performance issues when dealing with simple questions. For example, the proposed system from Yang et al. [1] yields errors in processing trivial questions composed of a single predicate or relation. Meanwhile, in the system proposed by Hu et al. [12], simple questions contribute to 31% error in the relation mapping. Meanwhile, references [13–15] recognized that ambiguity in both the input simple question, i.e., ambiguous question content, and generated answer candidates to the question causes performance issues. In addition, as our survey found, the currently best-performing KGSQA solution [16] only achieved an accuracy below 90%. This indicates that a further study focusing on KGSQA is well-justified and a focused survey on this area would be beneficial to it.

Another facet to research on KGSQA is the benchmark data. There are several benchmark data sets used to measure the performance of KGQA systems. A majority of these data sets such as WebQuestions [17], ComplexWebQuestions [18], and LC-QuAD 2.0 [19] include both simple and complex questions. An exception to this is the SimpleQuestions data set [20], which is the only one that solely contains simple questions. This data set is also the largest among all existing benchmark data sets with more than one hundred thousand questions with Freebase, hence taken as the de facto benchmark for KGSQA. However, Freebase has been discontinued since 2016, causing several practical difficulties in using the data set. Fortunately, there have been efforts to develop alternative versions of this data set using Wikidata [21] and DBPedia [22], respectively, as the underlying KG. Note also that even in data sets with a mixture of simple and complex questions, the proportion of simple questions that they contain may be significant. For example, WebQuestions contains 85% of questions that can be addressed directly through a single predicate/relation [2]. The availability of large data sets of simple questions pave the way for the adoption of deep learning by KGQA systems, which is a key point of our survey.

Finally, from a practical perspective, the 100 most frequently asked questions on Google search engines in 2019 and 2020 are simple questions. (Available online: <https://keywordtool.io/blog/most-asked-questions/> (accessed on 27 June 2021).) The most frequently asked question in that list is “When are the NBA playoffs?”, which was asked an average of 5 million times per month. Meanwhile, the question “Who called me?” ranked number 10 in the list was asked an average of 450K times per month. Both examples are simple questions. The former corresponds to the triple (*nba*, *playoffs*, *?*), while the latter can be captured by the triple (*?*, *called*, *me*). This and all of the aforementioned arguments indicate that KGSQA is a fundamental challenge that should be well addressed to achieve an overall high-performing KGQA system.

1.2. Related Surveys

There are three other surveys in the area of KGQA published in the last four years with a rather different coverage and focus compared to ours. First, Höffner et al. [23] presented a survey of question answering challenges in the Semantic Web. In their survey, Höffner et al. used the term (Semantic Web) knowledge base to refer to the knowledge graph. They surveyed 62 systems described in 72 publications from 2010 to 2015. They considered

seven open challenges that KGQA systems have to face, namely multilingualism, ambiguity, lexical gap, distributed knowledge, complex queries, nonfactoid (i.e., procedural, temporal, and spatial) questions, as well as templates. The survey then categorized the systems according to these challenges, instead of specifically discussing their strengths and weaknesses. The survey also did not include discussion on to what extent the surveyed systems achieved their end-to-end KGQA performance.

In comparison to the survey conducted by Höffner et al., our survey considered a wider range of publication year, namely from 2010 to 2021, although in the end, no system prior to the year 2015 is selected for a more detailed elaboration because we found that there is no specialized benchmark on KGSQA prior to the SimpleQuestions data set. Data sets from the QALD benchmark series do include a number of simple questions. However, those simple questions are not strictly separated as such; thus, the performance of KGQA systems specifically only on those simple questions cannot be obtained. As a result, all 17 KGQA systems selected in our survey are in fact published in 2015 or later.

Furthermore, progress on deep learning in recent years influences a rapid development of KGQA systems through the adoption of various deep learning models, such as variants of recurrent neural networks and transformer architecture, on various subtasks of KGQA. This is a key point of our survey that is not considered at all in the survey of Höffner et al.

Diefenbach et al. [24] presented another survey of QA over (Semantic Web) knowledge bases. Unlike the survey conducted by Höffner et al., the survey conducted by Diefenbach et al. mostly focused on KGQA systems that have been explicitly evaluated on the QALD benchmark series; this benchmark series is briefly described in Appendix A—covering 33 systems published from 2012 to 2017. In their survey, Diefenbach et al. did include a brief discussion of several KGQA systems evaluated on the WebQuestions and SimpleQuestions data sets, although this was not taken as the main focus. Only two systems published in 2017 were mentioned; the rest were published earlier. The survey itself was organized based on the different subtasks in KGQA, namely question analysis, phrase mapping, disambiguation, construction of query, and distributed knowledge. Comparisons on precision, recall, and *F*-measure are provided. For each discussed subtask, a brief discussion on relevant research challenges is summarized. In addition, a general discussion on the evolution of those challenges is included at the end.

Compared to the survey conducted by Diefenbach et al., our survey does have some overlap in that five KGQA systems discussed in our survey are also mentioned there. However, as their survey emphasized KGQA systems evaluated on the QALD benchmark series, those five systems are only summarized briefly. In contrast, besides providing more elaboration on those five systems, our survey also covers another dozen KGQA systems, including the ones published more recently, i.e., later than 2017.

The survey conducted by Diefenbach et al. also enumerates a long list of challenges of an end-to-end KGQA systems, including not just those directly related to the various steps of answering questions, but also a few challenges concerning deployment aspects of KGQA systems. Nevertheless, the approaches discussed in the survey mostly include those typically associated with traditional rule-based and information-retrieval-based methods. Our survey covers many approaches that employ various deep learning models. This is unsurprising since progress in that area has only been significant in the last five years. Hence, our survey fills an important gap not covered by the survey conducted by Diefenbach et al.

Finally, the survey from Fu et al. [25] is the most recent survey in KGQA, though at the time of our survey's writing, this survey is only available in the preprint repository arXiv; hence, it has not yet been published in any refereed journals or other publication venues. The scope is limited only to complex KGQA, instead of KGSQA. The survey presents a brief discussion on 29 KGQA systems for complex questions from 2013 to 2020, most of which are categorized into traditional-based methods, information retrieval-based methods, and neural semantic parsing-based methods. The survey then summarizes future challenges in this subject area through a brief frontier trend analysis.

Contrary to that of the survey conducted by Fu et al., the scope of our survey is KGSQA. Consequently, our survey considered a completely different set of KGQA systems. In addition, unlike a rather brief summary by Fu et al. on challenges due to the use of deep learning models, our survey presents a more complete discussion on this aspect.

Note that in addition to the specifically aforementioned three surveys, there are other older surveys in the area of knowledge graph/base question answering, such as [26–28]. These surveys covered many older KGQA systems, particularly those published prior to 2010.

1.3. Aim of the Survey

In this paper, we survey 17 different KGQA systems selected after reviewing the literature on KGSQA in the last decade, i.e., from 2010 to 2021. We study these systems and identify the challenges, techniques, as well as recent trends in KGSQA. Our aim is to present an overview on the state-of-the-art KGQA systems designed to address the challenges of answering simple questions. In particular, our survey

1. is the first one focusing on the state-of-the-art of KGSQA;
2. elaborates in detail the challenges, techniques, solution performance, and trends of KGSQA, and in particular, a variety of deep learning models used in KGSQA; and
3. provides new key recommendations, which are not only useful for developing KGSQA, but can also be extended to more general KGQA systems.

The rest of this article is structured as follows. In Section 2, we present the methodology used in this survey. In Section 3, we describe the results of the search and selection process and show the articles selected based on a thematic analysis. In Section 4, we discuss the techniques employed by existing KGSQA systems along with the level of performance they achieved and give insights into future challenges of KGQA systems. Finally, Section 5 concludes the survey.

2. Methodology

In general, our aim is to obtain a good selection of KGSQA systems for a detailed discussion by examining relevant articles published from 2010 until 2021. Thus, our methodology consists of an initial search of articles and three levels of article selections. In each of the selection step, we exclude a number of articles (and thus KGQA systems) based on certain criteria explained below.

2.1. Initial Search

We initially use three reputable databases: ACM Digital Library (<https://dl.acm.org> (accessed on 27 June 2021)), IEEE Xplore (<https://ieeexplore.ieee.org> (accessed on 27 June 2021)), and Science Direct (www.sciencedirect.com (accessed on 27 June 2021)) to retrieve related articles describing KGQA systems. We employ specific keywords to obtain the journals and proceedings that focus on question answering over knowledge graph systems from 2010 to 2021. Specifically, we search for articles from that period, whose title and abstract

- (i) matches “question answering”; and
- (ii) matches either “semantic web”, or “knowledge graph”, or “knowledge base”.

Note that we do not employ exact matching to the searched keywords; hence, for example, “question answering” would also match a title containing “question” and “answering”. Moreover, since we cannot always determine whether an article focuses on KGSQA solely from its title or abstract, we do not include “simple question” as a keyword. The detailed search command we use is provided in Figure A1.

From the initial search over three databases above, we get 2014 candidates of different KGQA systems. We use the Scopus index to ensure the inclusion of any other related articles that are not already included. From a Scopus search using the same criteria as

above, we initially 1343 articles. In total, our initial search yields 3357 articles that are potentially related to question answering over knowledge graph.

2.2. Article Selection Process

We omit articles collected in the initial search if they are irrelevant to our survey objective. To this end, we apply three steps of filtering to obtain the final selection of articles as described below. Figure 1 illustrates the whole article selection process, including the number of articles filtered out in each step.

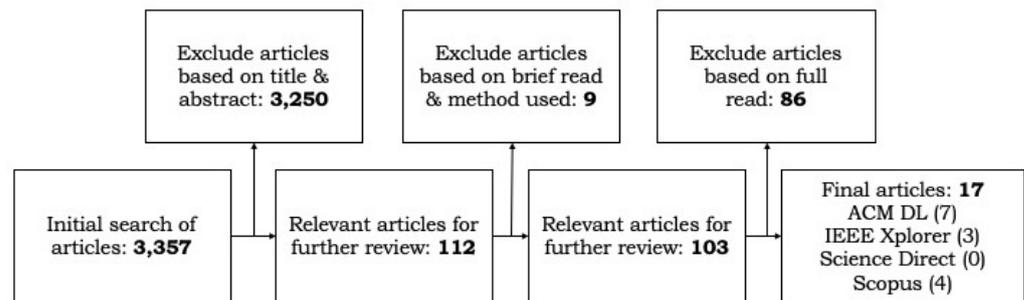


Figure 1. Articles are filtered through by three selection steps, namely a selection based on title and abstract, a selection through quick reading, and a selection through a full reading.

2.2.1. (Further) Selection Based on Title and Abstract

In the first selection step, we examine the title and abstract of each article, and then exclude it if the title or abstract clearly indicates that the content of the article is not about question answering over knowledge graph/bases. For example, articles in which the term knowledge base clearly does not refer to knowledge graph in the Semantic Web context are excluded in this step. Note that the result of this step is the selection of 112 relevant papers and the exclusion of 3250 other articles.

2.2.2. Selection through Quick Reading

In the second selection step, we did a quick reading over the articles obtained from the first filtering step. The aim is to filter out articles whose method(s) and evaluation section clearly do not involve simple questions. After this step, we obtain 103 articles that we consider relevant to our survey.

2.2.3. Selection through Full Reading

For the last selection step, we did a full reading of each article with the aim of gathering only articles that describe KGSQA systems. In addition, since we view the survey by Höffner et al. [23] as the most prominent recent survey in this area, we explicitly exclude articles that have been discussed in that survey. In this last selection step, we drop 86 articles and keep the remaining 17 to be fully discussed in Section 4.

3. Results

This section describes an overview of the terminologies in KGQA as well as a thematic analysis on the challenges and state-of-the-art of KGSQA as understood from the selected articles.

3.1. Terminology

To quote Hogan et al. [29], a knowledge graph (KG) is “a graph of data intended to accumulate and convey knowledge of the real world, whose nodes represent entities of interest and whose edges represent relations between these entities.” Though this definition is rather inclusive and does not point to a specific data model, KG in the KGQA literature mostly refers to a directed edge-labeled graph data model. In this data model, a KG is as a set \mathcal{K} of assertions, each of which is expressed as a triple of the form (s, p, o)

where the elements respectively represent the subject, predicate, and object of the assertion. The subject and object of a triple are some identifiers representing entities, while its predicate is an identifier that refers to a (binary) relation from the subject entity to the object entity. A graph data model sometimes allow literal values at the object position of a triple. Triples can be generalized into triple patterns, which are just like triples, except that its subject, object, and/or predicate can be a variable in addition to entity/relation identifiers, e.g., ($?x$, capital of, Indonesia) where $?x$ is a variable. A query, in its simplest form, is a set of such triple patterns and an answer to a query is a substitution of all variables in in the query with entity/relation identifiers (hence obtaining a set of triples) such that all the resulting triples hold in the KG.

The above data model was standardized as Resource Description Framework (RDF) [30] by W3C. The corresponding query language was also standardized as SPARQL [31]. Practical examples of large KGs include YAGO (<https://yago-knowledge.org> accessed on 27 June 2021), Freebase (FB), DBPedia (dbpedia.org accessed on 27 June 2021), Wikidata (wikidata.org accessed on 27 June 2021), LinkedGeoData (<http://linkedgeodata.org> accessed on 27 June 2021), OpenStreetMap (OSM) (<https://www.openstreetmap.org> accessed on 27 June 2021), and Acemap Knowledge Graph (AceKG) (<https://archive.acemap.info/app/AceKG/> accessed on 27 June 2021).

Let \mathcal{K} be a KG, $\mathcal{E}_{\mathcal{K}} = \{e_1, \dots, e_N\}$ the set of all entities defined in \mathcal{K} , and $\mathcal{R}_{\mathcal{K}} = \{r_1, \dots, r_M\}$ the set of all relations defined in \mathcal{K} . Entities and relations in \mathcal{K} form its vocabulary. Let q be a natural language question, conveniently represented as a bag-of-words $q = [w_1, \dots, w_t]$. For each such a question q , we assume its semantic representation with respect to \mathcal{K} is a query q' containing triple patterns using vocabulary of \mathcal{K} , which captures the intent of q . For example, the question “What is the capital of Indonesia” can be represented by a triple pattern ($?x$, capital of, Indonesia) where capital of and Indonesia are vocabulary elements of the KG.

Given an NLQ q , knowledge graph question answering (KGQA) refers to the problem of finding a set $\mathcal{A}_q \subseteq \mathcal{E}_{\mathcal{K}}$ of entities that are answers to q , i.e., each $e \in \mathcal{A}_q$ is an answer to a query q' that semantically represents q . In KGQA, we often need to consider an (N -gram) mention in q , which is just a sequence of $N \geq 1$ consecutive words in q . KGQA is different from information-retrieval-based question answering (IRQA) that uses unstructured text as data, that is, instead of a knowledge graph \mathcal{K} , a set of documents containing unstructured text is used. In addition, if q' contains only a single triple pattern, we call q a simple question. Otherwise, q is a complex question. KGQA where the question is simple is called knowledge graph simple question answering (KGSQA), which is the focus of this survey.

One traditional approach to KGQA is semantic parsing where the given question is first translated into its semantic representation, i.e., an “equivalent” query, which is then executed on the KG to retrieve an answer. However, rapid progress of deep learning research gave rise to an alternative strategy that avoids explicit translation to a query. Rather, it employs neural-network-based models where the question and the KG (or its parts) are represented in vector spaces and vector space operations are used to obtain the answers.

Generally, an end-to-end KGQA system consists of two main parts. Both of these parts are traditionally arranged in a pipeline with distinct separation between them, though with the adoption of deep learning, one can in principle have a truly end-to-end solution. The first part deals with the problem of understanding the intent of the input question through its key terms and structures, called question understanding [32,33] or entity identification [34]. The second part deals with the problem of employing the extracted terms and structures from the first part to obtain a matching answer from the KG. This is called query evaluation [32,33] or query selection [34] if a KG query is explicitly constructed, or joint search on embedding spaces [35] if vector space operations are used instead of explicit KG queries.

Performance evaluation of KGQA systems is customarily done through the use of benchmark data sets. This allows a reliable assessment and comparison between KGQA

systems. Commonly used performance measures include accuracy, recall, precision, and F -measure [24]. Appendix A summarizes existing benchmark data sets currently used in KGQA.

3.2. Tasks and Challenges in KGSQA

From the selected articles, we identify two categories of technical tasks in KGSQA coinciding with the two parts of KGSQA systems. They are entity/relation detection/prediction/linking and answer matching. Table 1 summarizes all variants of these aforementioned tasks and references to their proposed solutions, listed according to the year they were published. In Section 4, we will detail how these systems attempt to solve the tasks listed in the table. We now describe each of these tasks next.

Table 1. Published articles per KGSQA task per year. Some articles proposed solutions to multiple tasks.

Task (# Articles)	2015	2016	2017	2018	2019	2020	2021
Entity detection (9)		[36,37]	[16]	[38]	[14,35,39,40]		[41]
Entity prediction (9)		[36,42]	[13,43]		[35,44]	[45]	[41,46]
Entity linking (6)	[20]	[37]	[16]		[14,40]	[45]	
Relation prediction (12)		[42]	[13,16,43]	[38]	[14,35,39,40,44]		[41,46]
Answer matching (17)	[20]	[36,37,42]	[13,16,43]	[38]	[14,15,35,39,40,44]	[45]	[41,46]
Subgraph selection (7)		[36]	[43]	[38]	[35,39,40]		[46]

3.2.1. Entity and Relation Detection, Prediction, and Linking

The first category of tasks in KGQA concerns question understanding, that is, how KGQA systems need to capture the intent of the question through its key structure, possibly taking into account relevant information from the underlying KG. This can be in the form of entity and relation detection, prediction, and linking.

Let \mathcal{K} be a KG and q a question posed to it. Entity/relation detection refers to the task of determining which N -gram mentions in q represent entities/relations. Entity/relation prediction aims to classify q to a category labeled by an entity $e \in \mathcal{E}_{\mathcal{K}}$ (entity prediction) or a relation $r \in \mathcal{R}_{\mathcal{K}}$ (relation prediction) where e and r are an entity and a relation mentioned in q . Thus, $\mathcal{E}_{\mathcal{K}}$ (resp. $\mathcal{R}_{\mathcal{K}}$) forms the target label set for entity (resp. relation) prediction. Entity/relation linking aims to associate all entity/relation mentions in q to correct entities/relations in \mathcal{K} . In KGSQA, q will only contain one entity and one relation. Hence, all three tasks become relatively simpler compared to KGQA in general.

For instance, consider the question $q =$ "Where is the capital of the Empire of Japan?". In addition, suppose Japan is an entity and capital of is a relation in \mathcal{K} . In entity/relation detection, "the Empire of Japan" should be detected as entity and "is the capital of" should be detected as a relation. In entity/relation prediction, Japan and capital of are treated as one of possibly many class labels. One then needs to assign the question q to Japan and capital of, that is, the labels are for the question. In entity/relation linking, Japan and capital of are again treated as class labels, but unlike entity/relation prediction, the aim is to explicitly tag the mentions, that is, to assign "the Empire of Japan" with Japan and "is the capital of" with capital of.

A number of issues may arise in entity/relation prediction/linking. The two main ones are concerned with lexical gap and ambiguity, which also applies even to KGSQA. Lexical gap happens because of multiple ways to express the same or similar meaning [47]. For example, the relation "leader" in "Who is the leader of Manchester United?" may be expressed as the relation "manager" in the underlying KG. Lexical gap can be syntactic in nature, for example, due to misspellings and other syntactic errors, or semantic, for example, synonymy ("stone" and "rock") or hyper/hyponymy ("vehicle" and "car").

Meanwhile, ambiguity occurs when the same expression can be associated with different meanings. As an example, the question "Who was Prince Charles's family?" can have "Queen Elizabeth" as an answer due to a match between "family" in the question and the KG relation "parent", as well as "Prince William" through the KG relation "child".

Ambiguity may come in the form of homonymy (“Michael Jordan” the basketball player and “Michael Jordan” the actor) or polysemy (“man” referring to the whole human species and “man” referring to the males of the human species).

Another challenge related to entity/relation linking task is the presence of unknown structures in the question. In lexical gap and ambiguity challenges, one looks at how mentions of entities/relations in the input question can be associated with entities/relations already defined in the KG \mathcal{K} , i.e., elements of $\mathcal{E}_{\mathcal{K}}$ and $\mathcal{R}_{\mathcal{K}}$. However, note that q may contain mentions of entities/relations from outside the vocabulary of \mathcal{K} . Hence, such an association may not always be found. Consequently, the question cannot be answered by solely relying on \mathcal{K} — additional knowledge from other sources may be needed.

3.2.2. Answer Matching

Different from entity/relation detection/prediction/linking, the answer matching task concerns the actual process of obtaining the answer from the underlying KG. That is, given some representation of (the structure of) the input question q , the aim is to find the closest matching triples in \mathcal{K} .

Since a question in KGSQA only contains mentions of a single entity e and a single relation r , once e and r are correctly identified from \mathcal{K} (via entity/relation detection/prediction/linking), the answer can be easily be obtained via the union of two queries $\{(?x, r, e)\} \cup \{(e, r, ?x)\}$ where $?x$ is a variable representing an answer to q . However, correct identification of e and r is nontrivial due to the noise and incompleteness of information in \mathcal{K} . Errors in the identification may also propagate to the end result. So, an alternative view is to embed q and/or \mathcal{K} in a vector space and then build a high-quality association between them using deep learning models, without actually constructing an explicit query.

Frequently, \mathcal{K} contains a huge number of triples, hence naive answer matching may be too expensive. Moreover, q may contain mentions matching multiple entities/relations, e.g., in the question “Who is the leader of the Republic of Malta?” contains a mention “Republic of Malta” and “Malta”. The former matches best with the country Malta, while the latter may also be matched to the island Malta. So, entity/relation detection/prediction may not always yield only single detected/predicted entities/relations, but sometimes multiple of them, all with relatively high degree of confidence. Some KGSQA systems thus perform a subgraph selection as an additional task. In this task, given some anchor information such as entity/relation mentions, one selects only relevant parts of \mathcal{K} as the base for matching. The selected parts may be in the form of a set of triples, i.e., a smaller KG, or a set of entities and/or relations, i.e., a subset of $\mathcal{E}_{\mathcal{K}}$ or $\mathcal{R}_{\mathcal{K}}$.

Besides lexical gap and ambiguity challenges, which may still occur in answer matching, one other challenge concerns answer uncertainty. This issue is due to the fact that answers are often produced by KGQA systems through a pipeline of models, each of which introduces a degree of uncertainty that may propagate along the pipeline and influence the final result. So, a good approach to estimate answer uncertainty would be helpful to reduce mistakes in answers or explain why such mistakes occur.

3.3. State-of-the-Art Approaches in KGSQA

A variety of KGSQA approaches have been proposed to address the tasks and challenges described in Section 3.2. Nevertheless, one key similarity between those approaches is the use of deep learning models as a core component. This is a trend motivated not just by advances in deep learning, but also by the availability of data sets of simple questions (particularly, SimpleQuestions) with annotated answers from some KG.

Different types of deep learning models have been used by the surveyed KGSQA systems. They include memory networks (MemNNs), convolutional neural networks (CNNs), gated recurrent unit (GRU) and its bidirectional variant (Bi-GRU), long short-term memory (LSTM) and its bidirectional variant (Bi-LSTM), as well as transformer architectures. More detailed explanation of these deep learning models is beyond the scope

of this survey. For more details on the basics, the reader may consult a number of deep learning textbooks, such as from [48–50] to name a few.

Table 2 lists the models/approaches used by any of the 17 surveyed KGSQA systems to solve entity detection, entity prediction, entity linking, and relation prediction tasks. Meanwhile, Table 3 list the models/approaches used to solve answer matching and sub-graph selection tasks. No solution was specifically proposed for relation detection and linking, which can be attributed to the fact that simple questions involve only one entity and one relation. For the former, once an entity mention has been detected in a question, one can infer that the remaining words (minus question word(s)) correspond to relation. As for the latter, once we can predict or link an entity in a question, rather than directly finding an association between possible relation mention in the question and a relation in the KG, it would be more straightforward to directly query the KG to obtain the relation (or candidates of them). So, by performing entity detection/prediction/linking in KGSQA, relation detection and linking can be skipped.

As seen in the Table 2, nine (out of 17) of the surveyed KGSQA systems contain a distinguishable entity detection component. The main idea is to view entity detection as a sequence labeling problem, which is naturally modeled using RNNs. In fact, all but one system from those nine employ Bi-GRU and Bi-LSTM. In addition, only Türe and Jojic [16] explored all combinations of vanilla and bidirectional variants of GRU and LSTM. References [14,36] proposed a combination of Bi-GRU and a layer of conditional random field (CRF), the latter can be understood as a classification layer similar to logistic/multinomial regression, but adapted to sequences. As an alternative to Bi-GRU, others proposed the use of Bi-LSTM [16,35,38,41] also in combination with CRF [37,40]. As concluded by Türe and Jojic [16], Bi-LSTM seems to perform better in this task. Meanwhile, the only known approach not based on RNNs is BERT [39], a transformer architecture proposed more recently than RNNs. BERT is known to outperform more traditional RNNs in many NLP tasks. Nevertheless, for entity detection in KGSQA, this improvement has not been demonstrated yet.

Similar as above, entity prediction component is found in nine (out of 17) of the surveyed KGSQA systems. This task is also viewed as a sequence labeling problem, though the label is for the whole question (not looking at mentions in the question separately). Like entity detection, various RNN models are used, particularly to encode the questions, and often, also the entities in KG. Prediction is done by pairing the embedding of the question with the embedding of the correct entity using some form of likelihood or distance optimization. Regarding the RNN models, Lukovnikov et al. [43] employs GRU, while Dai et al. [36] uses Bi-GRU. Li et al. [41] employs LSTM with attention to perform entity prediction (jointly with relation prediction) in an end-to-end framework. References [35,44] use Bi-LSTM with attention, while Zhang et al. [14] introduced a Bayesian version of Bi-LSTM, which is useful for measuring answer uncertainty. RNN-based encoder–decoder architectures were also proposed [13,42]. Finally, one exception to RNN is a BERT-based solution by Luo et al. [45]. These models for entity prediction do not require entity detection since they are directly trained on question–entity pairs. Even if both are present, e.g., in Dai et al. [36], the result of entity detection is not used by the entity prediction component.

Table 2. Models/approaches employed for entity detection, prediction, and linking, as well as relation prediction. No solution was proposed for relation detection and linking.

Entity detection (ED) employing the following models:

1. GRU in [16];
2. Bi-GRU in [16];
3. Bi-GRU with linear-chain conditional random field (CRF) in [14,36]—the latter adds a GRU on top of Bi-GRU;
4. LSTM in [16];
5. Bi-LSTM in [16,35,38], and the pipeline framework of [41];
6. Bi-LSTM with CRF in [37,40];
7. BERT in [39].

Entity prediction (EP) employs the following models:

1. GRU in [43];
2. Bi-GRU in [36];
3. LSTM with attention in the end-to-end framework of [41];
4. Bi-LSTM with attention in [35,44];
5. Bayesian Bi-LSTM in [46];
6. RNN-based encoder–decoder architecture with:
 - LSTM and temporal CNN as encoder and LSTM with attention as decoder in [42];
 - Bi-LSTM as encoder and LSTM as decoder in [13];
7. BERT in [45].

Entity linking (EL) employs the following approaches:

1. String/*N*-gram matching between questions and KG entities in [14,16,20,37,45];
2. CNN with adaptive maxpooling in [40].

Relation prediction (RP) employs the following models:

1. CNN with adaptive maxpooling [40];
2. GRU in [16,43];
3. Bi-GRU in [16,36,38], and the pipeline framework of [41];
4. LSTM in [16];
5. LSTM with attention in the end-to-end framework of [41];
6. Bi-LSTM in [16];
7. Bi-LSTM with attention [35,44];
8. Bayesian Bi-LSTM in [46];
9. RNN-based encoder–decoder architecture with:
 - LSTM and temporal CNN as encoder and LSTM with attention as decoder [42];
 - Bi-LSTM as encoder and LSTM as decoder [13];
 - GRU and Bi-GRU as encoder and GRU as decoder in [14];
10. BERT in [39].

Table 3. Models/approaches employed for answer matching and subgraph selection tasks.

<p>Answer matching (AM) employs the following approaches:</p> <ol style="list-style-type: none"> 1. String and structure similarity or direct query construction in [13,16,36,38,39,41–44]; 2. Optimized similarity in embedding space between KG entities and questions in [20,35,40,46]; 3. CNN with attentive maxpooling in [37]; 4. Support vector machines in [14]; 5. Matching aggregation framework consisting of ReLU linear layer, attention mechanism, and LSTM as aggregator in [15]; 6. A custom architecture consisting of BERT, relation-aware attention network, Bi-LSTM, and a linear layer in [45].
<p>Subgraph selection (SS) employs the following approaches:</p> <ol style="list-style-type: none"> 1. String matching mentions/detected entities vs. KG entities/reasons [35,36,38,39,43,46]; 2. Combination of string/<i>N</i>-gram matching and co-occurrence probability of mentions and KG entities in [40].

Meanwhile, a distinct entity linking component is only found in six systems. Five of them [14,16,20,37,45] use various forms of string or *N*-gram matching between the entity mention in the question and KG entities. The main reason is because either the entity mention has been identified via an entity detection model, or a KG entity has been predicted to be associated with the question; thus, one only needs to match it with the correct *N*-gram mention. The only solution with deep learning is from Zhao et al. [40] who used CNN with adaptive maxpooling, which is applied prior to computing a joint scoring to obtain a most likely answer.

Approaches employing a relation prediction component essentially follows a similar line of thinking as in entity prediction, namely by viewing it as a sequence labeling problem. So, similar RNN models are also used for this problem. Two exceptions are Zhao et al. [40] who used CNN with adaptive maxpooling and Lukovnikov et al. [39] who used BERT. Interestingly, a few systems [15,20,37,45] skipped the relation prediction task completely. That is, after performing entity detection and/or entity prediction, they directly proceed to answer matching task.

Meanwhile, solutions for answer matching task are more varied. A majority of the approaches use various forms of string matching, structure matching, or direct query construction [13,16,36,38,39,41–44]. In principle, these approaches perform an informed search over the KG (or parts of it) for triples that can be returned as answer. Criteria such as string label similarity or graph structure similarity are used. The latter is feasible because in simple questions, the relevant structure can be captured by a simple triple pattern. Several solutions [20,35,40,46] also conduct a similarity-based search, but they perform it in an embedding space that contain representations of both KG entities and questions. Zhang et al. [14] uses SVM. Yin et al. [37] uses CNN with attentive maxpooling, while [15,45] offer custom architectures as a solution—that of the former is mainly based on LSTM with attention, while that of the latter consists of BERT, attention network, and Bi-LSTM.

Finally, subgraph selection is only done by seven systems [35,36,38–40,43,46]. All of them use some form of string matching between mentions/detected entities and KG entities/reasons. In addition, Zhao et al. [40] also combines string matching with co-occurrence probability of mentions and KG entities into one criteria for subgraph selection. Notably, other surveyed KGSQA systems do not include specific subgraph selection approaches. This indicates that there is a room of exploration in this direction.

4. Discussion

This section discusses KGSQA systems and their models, strengths and weaknesses, and end-to-end performance. Summary of the strengths and weaknesses is given in

Tables 4 and 5 summarize the strengths and weaknesses of each of the proposed solutions. Table 6 presents the performance of those proposed solutions. Performance is given in terms of testing accuracy on the SimpleQuestions data set. The state-of-the-art performance is achieved by Türe and Jojic [16] with 88.3% accuracy. In the last part of this section, we describe the future challenges of KGSQA systems and its recommendation as a future direction.

4.1. Existing KGSQA Systems with its Models, Achievements, and Strengths and Weaknesses

4.1.1. Bordes et al.'s System

Bordes et al.'s system use n -gram matching in the entity linking task to link an entity extracted from a question to an entity in a KG. For answer matching, they measure the similarity score between embeddings of KG entities/reasons against embeddings of questions. To deal with the possibility that an input question contains fact(s) not known in the KG, Bordes et al. [20] augment the facts with triples extracted from a web corpus using ReVerb. Subjects and objects in ReVerb are linked to a KG using precomputed entity links proposed by Lin et al. [51]. This approach is helpful for a KGSQA system that wants to answer questions with relations expression are different from relations' name in a KG as the relations are represented in vectors value that possibly leads to obtain similar words used by a relation's name.

Although this model is claimed as a model that can work on natural language, the difference of pattern between a KG and ReVerbs leads to more effort to align. Nevertheless, this approach achieves a good baseline for end-to-end KGSQA performance with 63.9% accuracy.

4.1.2. Yin et al.'s System

Yin et al. [37] proposed two linkers called the active entity linker and the passive entity linker. The two linkers are used to produce top- N candidate entities. (1) Find the longest common consecutive subsequence between questions and entity candidates generated from KG based on question types. (2) Use Bi-LSTM + CRF to detect entity and find the longest common consecutive subsequence (1). The advantage of this approach is that besides the architecture used in this approach is simple. The entity linker can cover higher coverage of ground truth entities. However, it does not cover the semantic of an entity.

For answer matching, they use CNN and attentive max-pooling. Finally, a fact selector chooses the proper fact by calculating the similarity between an entity and a relation in a question and a fact in a KG. The best end-to-end performance of this model achieves 76.4% accuracy.

4.1.3. Dai et al.'s System

Dai et al.'s model [36] deals with entity and relation prediction, and subgraph selection. They use Conditional Focused (CFO), modeling $p(s, r|q) = p(r|q) \cdot p(s|q, r)$ in relation prediction. Meanwhile, they use Bi-GRU for entity prediction by choosing s, r that maximizes $p(s, r|q)$. In addition, since the search space is very large, they proposed a subgraph pruning in subgraph selection task to prune candidate space for (s, r) via entity detection with Bi-GRU and linear-chain CRF.

With the subgraph selection algorithm, this model can reduce the search space of subject mention during inference. However, the effect is that the entity representation obtained is less smooth. Hence, although this method is better in pruning the search space, the achievement of end-to-end accuracy is slightly less than the entity linker proposed by Zhu et al. [13], namely 75.7% accuracy. It is due to the fact that the learned entity representation of CFO is not smoother than the entity linker method from Zhu et al. [13].

4.1.4. He and Golub' System

He and Golub [42] perform entity and relation prediction via a combination of LSTM encoder (question), temporal CNN encoder (KG query), LSTM decoder with attention

and semantic relevance function (prediction). Once an entity and a relation are predicted, the system matches the answer directly via a query.

The strengths of this model are that (1) it uses fewer parameters and smaller data to learn, and (2) it can be used to generalize unseen entities. However, the model still cannot solve the lexical gap issue. The best end-to-end performance of this model achieves 70.9% accuracy.

4.1.5. Lukovnikov et al.'s System

The model deals with entity and relation prediction, and entity and relation selection. GRU is used in entity and relation prediction [43]. Meanwhile, in the entity and relation selection, Lukovnikov et al.'s use n -gram filtering and relation selection based on candidate entities, respectively.

This model has a weakness, namely ambiguity issues are hard to solve using it. However, the strengths of this model are that (1) it can handle rare word and out-of-vocabulary problems, (2) it does not require a complete NLP pipeline construction, (3) it avoids error propagation, and (4) it can be adapted to different domains. The best end-to-end performance of this model is 71.2% accuracy.

4.1.6. Zhu et al.'s System

Zhu et al. [13] address three tasks of KGSQA, namely entity and relation prediction and answer matching. Entity and relation prediction tasks are accomplished jointly using seq2seq, which consists of Bi-LSTM encoder and a custom LSTM decoder with generating, copying, and paraphrasing modes. Meanwhile, answer matching is performed simply via a query from predicted entity-relation pairs.

The model can address the synonymy issue. It also improves the end-to-end performance of KGSQA systems since it achieves 11% higher accuracy than by using Bi-LSTM only, namely 77.4%. However, using a larger KG will cause a lower end-to-end performance.

4.1.7. Türe and Jojic's System

Türe and Jojic proposed an approach to address four tasks, namely entity detection and linking, relation prediction, and answer matching. They use a GRU+Bi+GRU and LSTM+Bi-LSTM in both entity detection and relation prediction. Meanwhile, for entity linking, they use inverted index and tf-idf weighting of each n -gram of entity aliases (against the whole entity name) in the KG. From entity text, all matching entities with tf-idf weights are computed. From each candidate entity, they construct a graph of single-hop neighbors. From these neighbors, all whose path are consistent with the predicted relation are selected. Finally, the entity with the highest score is chosen [16]. Although the employed model is simple, it achieves 88.3% accuracy.

As for the weaknesses, the model cannot identify entities or relations expressed in different ways, such as "la kings" and "Los Angeles Kings". An index that stores synonyms is implemented to tackle this issue. However, since the model is a supervised one, predicting relations not appearing at all as target class in training will be difficult.

4.1.8. Chao and Li's System

The model addresses three tasks, namely entity detection, relation prediction, and answer matching. Chao and Li [38] use Bi-LSTM and n -gram matching to detect an entity in a question. Moreover, GRU is used for relation prediction. Once an entity and a relation are obtained, the model constructs a ranking based on outdegree and Freebase notable types to match them with an answer.

Incorporating Freebase notable types is useful to help distinguishing polysemic entities. However, other ambiguities issues such as hypernymy and hyponymy are still hard to solve using this model. This is also seen from the performance (66.6%), which is lower than Bordes et al.'s model.

4.1.9. Zhang et al.'s System

Instead of using a Bi-LSTM as performed by Chao and Li, entity detection model by Zhang et al. [14] consists of a combination of Bi-GRU, GRU, and CRF. Moreover, they also use a combination approach for relation prediction, namely using GRU encoder, Bi-GRU encoder, and GRU decoder. For answer matching, the model uses SVM.

The model is robust in predicting the correct relation path connected with multitarget entity. However, it is vulnerable to errors in predicting relations in their paraphrase or synonym. Even so, in general, it outperforms the approaches of [36,37,42], achieving 81.7% accuracy.

4.1.10. Huang et al.'s System

Huang et al. [35] employs Bi-LSTM with attention to learn an embedding space representation of the relation from the input question. A similar model is used to learn an embedding space representation of the entity from the question. However, here, an entity detection model based on Bi-LSTM is employed to reduce the number of candidate entities of the question.

Once embedded representation of the entity and relation from a question are found, Huang et al. [35] uses translation-based embedding such as TransE [52] to compute candidate facts as an answer to the input question. The actual answer is obtained through optimizing a particular joint distance metric on the embedding space. The model can tackle entity and relation ambiguity issues in KGSQA. The model also can improve end-to-end performance to 1.2% accuracy from Chao2018's system, namely, 75.4% accuracy. However, this model is not good for non-KG embeddings.

4.1.11. Wang et al.'s System

Wang et al.'s model [44] use Bi-LSTM with attention pooling to predict an entity. The model is essential to learn a sequence of words or characters of entities mentioned in questions. By representing an entity as a vector, the similarity score between an entity in a KG and an entity in a question can be calculated by using similarity functions. Meanwhile, for the relation prediction, instead of using topic entities as used by He and Golub [42], Wang et al. adopt relation candidates as queries for the questions' attentive pooling using Multilevel Target Attention (MLTA) combined with Bi-LSTM. With the approach, Wang et al.'s model can capture relation-dependent question representations. However, the model encounters some difficulty in addressing the lexical gaps and synonymy issues. Even so, the end-to-end performance of Wang et al.'s model outperforms He and Golub's model, namely 82.2% accuracy compared with 70.9% accuracy.

4.1.12. Lan et al.'s System

Prior to Lan et al. [15], one issue faced by existing KGSQA systems is that specific question context cannot be captured well. To address this problem, Lan et al. proposed to use the "matching-aggregation" framework for sequence matching. Specifically, the framework consists of a single ReLU linear layer, attention mechanism, and LSTM as an aggregator in the answer matching task. In addition, to enrich the context, Lan et al. also model additional contextual information using LSTM with an attention mechanism.

Lan et al.'s model supports the contextual intention of questions. However, it is hard to choose the correct path of facts when a question is ambiguous. Even so, the model outperforms the previous works except for [14,16]. The accuracy achieved 80.9%.

4.1.13. Lukovnikov et al.'s System

Unlike previous KGSQA systems where RNNs are extensively used to obtain an embedding space representation of the input question as well as KG entities and relations, Lukovnikov et al. [39] makes use of BERT [53] to help detecting entities and predicting relations. In entity detection, the model predicts *start* and *end* position of entity span. This model uses an inverted index to generate entity candidates and ranks them using the

string similarity function and outdegree. Meanwhile, for predicting a relation, Lukovnikov et al. use BERT with a softmax layer. The predicted entity and relation are then used to obtain an answer by a query (entity predicate pairs) ranking based on label similarity.

The use of BERT makes the proposed model powerful for sequence learning in an NLQ. However, since BERT employs a huge number of parameters, its runtime is costly. Nevertheless, experiments showed that BERT is promising to be used in KGSQA. The end-to-end performance achieved 77.3% accuracy.

4.1.14. Zhao et al.'s System

Zhao et al.'s model [40] deal with four tasks of KGSQA, namely, entity detection, subgraph selection, entity linking, and relation prediction. For entity detection, they use Bi-LSTM and CRF. Since the search space of entity detection is too wide, they proposed a subgraph selection by using n -gram based selection. The subgraphs are ranked based on the length of the longest common subsequence of entity and mention and the co-occurrence probability between the two. Meanwhile, in the entity linking and relation prediction, they use CNN producing similarity scores, optimized jointly using well-order loss.

The model can address an inexact matching case using a literal and semantic approach. However, in case of ambiguity, the model has not solved the issue. Zhao et al.'s experiment shows that the end-to-end performance of the model outperforms all previous works except Türe and Jojic's model, namely 85.4% accuracy.

4.1.15. Luo et al.'s System

Different from Lukovnikov et al.'s model [39] that does not use BERT for entity linking, Luo et al. [45] use BERT not only for entity detection but also for entity linking. The entity linking task is performed by querying detected entities. With the output of entity linking, relations linked to it can be obtained. Meanwhile, answer matching is performed without explicit relation prediction using BERT. In addition, it also uses a relation-aware attention network, Bi-LSTM and linear layer for combining scores.

The model supports solving semantic gaps between questions and KGs. Although the model has implemented a semantic-based approach, the model has drawbacks: (1) the ambiguity issue has not been solved, and (2) it cannot identify KG redundancy problem. KG redundancy means that some entities have the same attributes but different ids (aka. polysemy). The best end-to-end performance of this model is 80.9% accuracy. It is a promising approach to extend for improving performance.

4.1.16. Zhang et al.'s System

Zhang et al. [46] proposed a model that addresses three tasks of KGSQA, namely, entity and relation prediction, and answer matching. The candidate entity set for each question is directly selected using n -gram matching. The candidate relations for each candidate entity are then selected. Each candidate entity and relation are encoded in Bayesian Bi-LSTM, implicitly trained as entity and relation prediction model because only representation is considered (not predicting an actual entity/relation). Finally, answer matching is performed through max-pooling the joint score (dot product) matrix calculation between an entity and predicate representation trained end-to-end.

The model is easy to use and retrain for different domains. However, ambiguity issue remains a challenge for the model. Nevertheless, as the first attempt to provide an explicit estimation of answer uncertainty, the model reached a good end-to-end performance, namely, 75.1% accuracy.

4.1.17. Li et al.'s System

Li et al. [41] proposed two solutions for KGSQA. The first one is a pipeline framework in which entity detection and relation prediction tasks are solved separately. For the former, Bi-LSTM is used, while Bi-GRU is employed for the latter. Similar to Chao and Li [38], this model also uses Freebase notable types to improve the performance of entity selection.

In addition, this model also employs outdegree of entities. The combination of context information improve the accuracy of KGSQA even though the end-to-end performance is still low, namely, 71.8%. However, although this model has low end-to-end accuracy, the model has three advantages: (1) it has a cheaper runtime, (2) it can address polysemy entities, and (3) it works on a semantic-based approach.

The second solution is a fully end-to-end framework in which LSTM with self-attention is used. Unlike other systems, this end-to-end framework has a semantic encoding layer that captures the representation of input questions and KG facts. The rest of the framework functions as semantic relevance calculator that allows one to obtain similarity scores between the input question and the subject entity as well as between the input question and the predicate relation. One advantage of this end-to-end framework is better accuracy for semantic representation. However, the framework is inefficient in model training.

Table 4. KGSQA systems proposed from 2015 to 2018 and its strengths and weaknesses.

KGSQA	Model (Addressed Task(s))	Pros (+) and Cons (-)
Bordes et al. [20]	<i>N</i> -gram matching, optimized similarity (EL, AM)	(+) Can work on natural language and an enormous memory (-) The difference of pattern between a KG and ReVerbs leads to more effort to align
Yin et al. [37]	Bi-LSTM with CRE, <i>n</i> -gram matching, CNN with attentive pooling (ED, EL, AM)	(+) A simple architecture (+) The entity linker cover higher coverage of ground truth entities (-) The entity linker does not cover the semantics of an entity
Dai et al. [36]	Bi-GRU with linear CRE, Bi-GRU, Bi-GRU, string or structure similarity, string matching mentions (ED, EP, RP, AM, SS)	(+) A reduced search space of subject mention during inference (-) Less smooth in entity representations
He and Golub [42]	LSTM and temporal CNN as encoder and LSTM with attention as decoder, direct query construction (EP, RP, AM)	(+) Using 16 times fewer parameters (+) Using a smaller data (+) Can be used to generalize unseen entities (-) Difficult to address the lexical gaps
Lukovnikov et al. [43]	GRU, GRU, string matching mentions, direct query construction (EP, RP, SS, AM)	(+) Can handle out-of-vocabulary and rare word problems (+) Not requiring NLP pipeline construction (+) Avoids error propagation (+) Reusable for different domains (-) Hard to solve ambiguity issues
Zhu et al. [13]	Bi-LSTM as encoder and LSTM as decoder, direct query construction (EP, RP, AM)	(+) Can address synonymy issues (-) A lower performance for a larger KB
Türe and Jojic [16]	GRU+Bi+GRU and LSTM+Bi-LSTM, <i>n</i> -gram matching, GRU+Bi+GRU and LSTM+Bi-LSTM, direct query construction (ED, EL, RP, AM)	(+) Simple architecture (-) Inherent ambiguity issues have not been solved
Chao and Li [38]	Bi-LSTM, Bi-GRU, String or structure similarity (ED, RP, AM)	(+) Useful to distinguish a polysemy entity (-) It has not addressed other ambiguity issues such as synonymy, hyponymy, and hypernymy

Table 5. KGSQA systems proposed from 2019 to 2021 and its strengths and weaknesses.

KGSQA	Model (Addressed Task(s))	Pros (+) and Cons (-)
Zhang et al. [14]	Bi-GRU with linear CRF, <i>n</i> -gram matching, GRU and Bi-GRU as encoder and GRU as decoder, SVM (ED, EL, RP, AM)	(+) Robust in predicting the correct relation path connected with multiple target entities (-) Vulnerable to error in predicting relations in their paraphrase or synonym
Huang et al. [35]	Bi-LSTM, Bi-LSTM with attention, Bi-LSTM with attention, optimized similarity, string matching (ED, EP, RP, AM, SS)	(+) Can address entity and relation ambiguity (-) Not good for non-KG embeddings
Wang et al. [44]	Bi-LSTM with attention, direct query construction (EP, RP, AM)	(+) Can capture relation-dependent questions representations (-) Difficult to address the lexical gaps and synonymy issue
Lan et al. [15]	Matching aggregation framework: ReLU linear layer; attention mechanism; and LSTM as aggregator (AM)	(+) Support for the contextual intention of questions (-) Hard to choose the correct path of facts when a question is ambiguous
Lukovnikov et al. [39]	BERT, BERT, string or structure similarity, string matching mentions (ED, RP, AM, SS)	(+) Powerful for sequence learning (of entity and relation) in an NLQ (-) Costly and can overfit easily when learning facts from a KG
Zhao et al. [40]	Bi-LSTM with CRF, CNN with adaptive max-pooling, CNN with adaptive max-pooling, optimized similarity in embedding space between KG entities and questions (ED, EL, RP, AM)	(+) Can address inexact matching using literal and semantic approach (-) The ambiguity issue is unsolved
Luo et al. [45]	BERT, <i>n</i> -gram matching, a custom architecture: BERT; relation-aware attention network; Bi-LSTM; and a linear layer (EP, EL, AM)	(+) Support the solution for semantic gaps between questions and KBs (-) Questions ambiguity and KGs redundancy issue has not been solved
Zhang et al. [46]	Bayesian Bi-LSTM, Bayesian Bi-LSTM, optimized similarity, string matching mentions (EP, RP, AM, SS)	(+) Easy to use and retrain for different domains (-) Cannot address ambiguous questions
Li et al. [41]	Bi-LSTM, LSTM with attention, LSTM with attention in the end-to-end and pipeline framework, direct query construction (ED, EP, RP, AM)	(+) Can address polysemy entity (-) Inefficient model training

Table 6. Trends and state-of-the-art performance of KGSQA systems on the SimpleQuestions benchmark data set and Freebase KG. T.A. refers to Testing Accuracy in percent.

KGSQA System	Year	Model (Addressed Task(s))	T.A
Bordes et al. [20]	2015	<i>N</i> -gram matching, optimized similarity (EL, AM)	63.9
Yin et al. [37]	2016	Bi-LSTM with CRF, <i>n</i> -gram matching, CNN with attentive pooling (ED, EL, AM)	76.4
Dai et al. [36]	2016	Bi-GRU with linear CRF, Bi-GRU, Bi-GRU, String and structure similarity, string matching mentions (ED, EP, RP, AM, SS)	75.7
He and Golub [42]	2016	LSTM and temporal CNN as encoder and LSTM with attention as decoder, or direct query construction (EP, RP, AM)	70.9
Lukovnikov et al. [43]	2017	GRU, GRU, string matching mentions, direct query construction (EP, RP, SS, AM)	71.2
Zhu et al. [13]	2017	Bi-LSTM as encoder and LSTM as decoder, direct query construction (EP, RP, AM)	77.4
Türe and Jojic [16]	2017	GRU+Bi+GRU and LSTM+Bi-LSTM, <i>n</i> -gram matching, GRU+Bi+GRU and LSTM+Bi-LSTM, direct query construction (ED, EL, RP, AM)	88.3
Chao and Li [38]	2018	Bi-LSTM, Bi-GRU, String or structure similarity (ED, RP, AM)	66.6
Zhang et al. [14]	2019	Bi-GRU with linear CRF, <i>n</i> -gram matching, GRU and Bi-GRU as encoder and GRU as decoder, SVM (ED, EL, RP, AM)	81.7
Huang et al. [35]	2019	Bi-LSTM, Bi-LSTM with attention, Bi-LSTM with attention, optimized similarity, string matching (ED, EP, RP, AM, SS)	75.4
Wang et al. [44]	2019	Bi-LSTM with attention, direct query construction (EP, RP, AM)	82.2
Lan et al. [15]	2019	Matching aggregation framework consisting of ReLU linear layer, attention mechanism, and LSTM as aggregator (AM)	80.9
Lukovnikov et al. [39]	2019	BERT, BERT, string or structure similarity, string matching mentions (ED, RP, AM, SS)	77.3
Zhao et al. [40]	2019	Bi-LSTM with CRF, CNN with adaptive max-pooling, CNN with adaptive max-pooling, optimized similarity in embedding space between KG entities and questions (ED, EL, RP, AM)	85.4
Luo et al. [45]	2020	BERT, <i>n</i> -gram matching, a custom architecture consisting of BERT, relation-aware attention network, Bi-LSTM, and a linear layer (EP, EL, AM)	80.9
Zhang et al. [46]	2021	Bayesian Bi-LSTM, Bayesian Bi-LSTM, optimized similarity, string matching mentions (EP, RP, AM, SS)	75.1
Li et al. [41]	2021	Bi-LSTM, LTSM with attention, LSTM with attention in the end-to-end and pipeline framework, direct query construction (ED, EP, RP, AM)	71.8

4.2. Open Challenges in the Future Research

This section presents the drawbacks of the existing works and the open challenges in the future. We present the existing works' remaining challenges that address the simple questions challenges as insights of the potential future research.

The existing works put the remaining challenges on addressing SimpleQuestions issue. However, there are four main unsolved issues, i.e., paraphrase issue, ambiguity issue, benchmark data set issue, and cost issue.

4.2.1. Paraphrase Issue

Although the problem of using different words can be solved by using a word embedding approach (such as the system proposed by [35,38,43]), it cannot easily work with a question paraphrase. The reasons are twofold. First, the SimpleQuestions data set does not

provide paraphrases of questions, and hence, one cannot easily train a model to capture them. Second, the existing models did not build or generate paraphrases of questions from the SimpleQuestions data set in their training data. So, those models cannot correctly predict an answer when given a question with the same meaning as one in the data, but expressed using unseen paraphrasing.

4.2.2. Ambiguity Issue

The ambiguity issue is also revealed in the existing works. At least there are two types of ambiguity faced by the current works, namely, entity ambiguity and question ambiguity.

Entity ambiguity can be divided into three main issues. Although these types of ambiguity have been revealed by Lukovnikov et al. [43], these issues remain as drawbacks to KGSQA systems.

The first issue concerns indistinguishable entities. This occurs when two (actually different) entities have exactly the same name and type. For instance, the question “what song was on the recording indiana?” will return multiple entities in the KG with the label name “indiana” and the label type “musical recording”.

Second issue concerns hard ambiguity. This happens when two entities have the same label but different types. Besides, a clear indication of which entity is expected is not provided by the question. For instance, for the question, “where was Thomas born?”, *Thomas, baseball player* and *Thomas, film actor* are the two best ranked candidates and the question itself does not indicate which one is preferred.

Third issue concerns soft ambiguity. This is similar to hard ambiguity, except that the question provides some hints as to which entity is expected. This is intuitively less challenging than hard ambiguity since one can try to extract relevant context from the question to help the question answering process. Nevertheless, this issue remains largely unsolved.

Meanwhile, question ambiguity is an issue that happens when the question itself contains ambiguous descriptions. For instance, the question “who was Juan Ponce de Leon family” may output a different entity between a prediction and the ground truth. In this example, a prediction may output “Barbara Ryan”, while the ground truth is “Elizabeth Ryan”. It is due to the fact that the relation used to obtain an answer is different from the expected one. The former is obtained from “parents”, while the latter is obtained from “children”.

4.2.3. Benchmark Data Set Issue

SimpleQuestions contains questions that follow a rigid pattern. This leads to two drawbacks. First, rigid templates lead to an overfitting issue when it is used to train machine learning models. Second, the template cannot be used to generalize question patterns into more advanced question types.

4.2.4. Computational Cost Issue

Normally, a KG contains billions of facts. For instance, the two larger KGs, namely, Wikidata (<https://dumps.wikimedia.org> accessed on 27 June 2021) and Freebase (<https://developers.google.com/freebase> accessed on 27 June 2021), though the latter is now discontinued. The sheer size of the KGs implies that existing approaches need to work harder in encoding/decoding KG facts. Moreover, recent deep learning advances bring newer models whose size is getting larger. Consequently, the issue of computational cost becomes even more pronounced.

One effort that can be considered regarding this issue is to make useful (and much smaller) subsets of the full version of the KG available. Freebase versions of such subsets are available (FB2M and FB5M). However, since Freebase is discontinued, research has shifted to using Wikidata and/or DBpedia. Unfortunately, no similar subsets from Wikidata or DBpedia is known.

4.3. Recommendation

Future research should be directed at more semantics-based approach rather than syntactic-based approach. Some embedding methods can be used to encode and decode facts of a KG and questions in words or sentence representation. Thus, novel research can be tackled by using the embedding methods to address entity ambiguity and question ambiguity. Table 7 provides the survey papers about embedding methods that can give insights into existing techniques used for vector-based representations. Based on Table 5, we briefly provide an overview of the potential approaches for building KGQA systems toward a semantics-based approach.

Table 7. List of survey papers of embedding methods.

Survey Focus	Original Paper	Year
Knowledge graph embedding	Wang et al. [54]	2017
Representation of meaning	Camacho-Collados and Pilehvar [55]	2018
Graph embedding	Goyal and Ferrara [56]	2018

Semantic matching models such as RESCALL [57] and TATEC [58] can be extended for future KGQA for matching facts by exploring similarity-based scoring functions. Although the time complexity of those methods is little bit high [54], they can match latent semantics of relations and entities in their representation vectors.

Moreover, other variants of translation models such as TransH [59] and KG2E [60] can also be considered as methods other than TransE and TransR. Besides the space complexity and time complexity is smaller [54], those methods have an advantage that TransE and TransR do not have. By using different roles between entities and relations, TransH can determine the proper relations by defining if a relation is an incoming relation or the outgoing relation. Thus, a machine can understand a pair of subject-predicate and a pair of predicate-object desired by a question. KG2E models relations and entities as random variables rather than as deterministic points in vector spaces. Thus, it leads to improving the effectiveness model for uncertainties relations and entities in KGs.

To address the ambiguity issue, a hybrid approach can be considered to learn the sense of semantic of given texts or facts rather than Bi-LSTM and seq2seq proposed by Zhu et al. [13]. A hybrid model is a technique that uses KGs and text corpora for the data [55]. The more representation resources are used, a machine can determine the semantic meaning of facts well. A potential non-KG source for concept representations is Wikipedia because its concepts of entities usually are linked to any KGs, such as Wikidata and DBPedia. For this purpose, the methods of [61–63] can be considered.

The usage of pre-trained models for natural language processing (NLP) tasks that can comprehensively learn a sentence's context can also be considered to obtain mentions (entities and relations) of questions that will be used to link them into entities and relations in a KG. To wrap the models, existing libraries, such as HuggingFace's Transformers [64], can be used for particular tasks of KGQA systems. Table 8 describes a summary of tasks of Transformers proposed by Wolf et al. [64] that can be used to entity and relation linking tasks of KGQA systems. (<https://huggingface.co> accessed on 27 June 2021.)

Table 8. Summary of tasks of Transformers [64] that relevant to use for entity and relation linking on KGQA.

Task	Caption	Input	Output	Used in
Named entity recognition	To classify tokens according to a class	Text	NER	Entity and relation linking
Extractive question answering	To extract an answer from a text given a question	Context	Answers	Entity and relation linking
Masked language model	To mask tokens in a sequence	Text	Filled mask	Entity missing detection

The tasks mentioned in Table 8 can be explored for future KGQA, (i.e., question understanding and answer retrieval or query evaluation stage [32,33] in particular). Masked language model (MLM) and NER can be used for extracting and defining relations and entities and mentioned in a question. Meanwhile, context extraction can be applied to question to obtain a richer representation of the question. This can lead to a better precision in the answers. A nice wrapper such as SimpleTransformer can be a useful tool to perform to simplify the development (<https://simpletransformers.ai> accessed on 27 June 2021).

Furthermore, on a benchmark data set construction side, a novel data set should include a large variety of question paraphrases. The paraphrases can be constructed at either word level or question level. Thus, future research can work with richer sets of questions to obtain models capable of capturing the sense of the questions in a more fine-grained way. Table 9 provides an example of possible paraphrases that can be constructed for both word-level and question-level paraphrasing. Question generator tools can be used to generate paraphrased questions from a set of original questions instantly. A survey of Madnani and Dorr [65] provides a comprehensive survey to paraphrase generation using data-driven phrasal.

Table 9. Example of question paraphrasing. In the left column, the original question is an example of SimpleQuestions from Bordes et al. [20]. In the mid-column, the paraphrase is constructed by word level by changing the verb *die* into its synonyms (e.g., *decease*). In the last column, the paraphrase is built by question-level.

Original Question [20]	Word-level Paraphrasing	Question-Level Paraphrasing
"Where did John Drainie die?"	"Where did John Drainie <u>decease</u> ?"	"John Drainie died in which city?"

Additionally, constructing a benchmark data set with more varied question templates and ensuring balanced data are essential to avoid overfitting when training a model. Besides, a breakthrough benchmark data set containing complex questions and their constituting sub-questions can help the development of novel approaches use sub-questions (simple questions) to answer complex questions. Moreover, such a novel data set also can be used to evaluate the question decomposition task where approaches along this line have shown promising results for addressing complex questions in the future KGQA [3,7–9].

As language models such as BERT are getting larger, an obvious counter-approach is to try to come up with simpler architectures with smaller number of parameters so that training cost can be reduced. From a data perspective, creating useful subsets of large KGs (such as Wikidata or DBPedia) by selecting facts that are representative to the whole facts in the KG can also reduce the size of data. This can then lead to a lower training cost.

Furthermore, deep learning-based approaches that model nonlinear structures in data, while is also less costly in encoding KG data such as Graph Convolutional Networks (GCN) could be a good choice to explore [66]. Besides reducing the dimensionality, GCN can prevent unnecessary computation by using information from previous iterations to obtain a new embedding [56,66].

5. Conclusions

Based on articles collected from January 2010 until June 2021, we obtained 17 articles strongly relevant to KGSQA published in 2015–2021. During the past five years, we have seen an increasing interest in addressing the existing challenges of KGSQA. The most challenging tasks considered by these articles correspond to entity detection/prediction/linking and relation prediction, followed by answer matching, and subgraph selection. A Bi-GRU and Bi-LSTM used by Türe and Jojic [16] leads to insofar the best performance of KGSQA systems, namely, a 88.3% accuracy. In contrast, LSTM used by He and Golub [42] gives the lowest performance at 70.9% accuracy.

Also, there are four prominent unsolved issues during the period, namely, a paraphrase problem, ambiguity problems, benchmark data set problems, and computational cost problems. Future research should be directed at a more semantics-based approach rather than a syntactic-based approach to overcome the issues. The usage of embedding methods can address or at least reduce the entity or question ambiguity issue. The usage of a pre-trained NLP model, such as Transformer, can facilitate different tasks in KGQA systems. In addition, a hybrid approach can be considered to enrich semantic representations of data (i.e., KGs and questions) to help a machine better determine the sense of given questions corresponding to KG facts. Additionally, simple models of low complexity are needed to address the computational cost issue caused by the larger size of data in the future. Future KGQA can also explore and may adopt existing works such as RESCALL and TATEC in order to improve the performance of matching accuracy in semantic matching with low complexity. An embedding model that models entities and relations as random variables rather than deterministic points in vector space can also be considered to encode and decode uncertainties entities and relations in KGs.

Meanwhile, on the data set construction side, a novel data set should include paraphrased questions rather than just the original questions. It can help novel KGQA systems to learn questions in various forms. Word-level and question-level variations can be used for generating paraphrased questions. In addition, the data set should also contain questions with a more varied pattern rather than a rigid pattern to help avoiding overfitting during training.

Author Contributions: Methodology, M.Y.; writing—original draft, M.Y.; supervision, A.A.K.; editing—revised version, A.A.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by a funding from Universitas Indonesia under a 2020 Publikasi Terindeks Internasional (PUTI) Q3 grant no. NKB-1823/UN2.RST/HKP.05.00/2020.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors wish to thank all anonymous reviewers for the positive feedback and helpful comments for correction or revision.

Conflicts of Interest: The authors have no conflict of interest in writing this manuscript.

Appendix A

Table A1. Data set used for evaluation of KGQA systems.

Data Set	Number of Questions	KG
QALD-1 ¹	Unavailable	Unavailable
QALD-2 ²	Unavailable	Unavailable
QALD-3 [67]	100 for each KG	DBPedia, MusicBrainz
QALD-4 [68]	200	DBPedia
QALD-5 [69]	300	DBPedia
QALD-6 [70]	350	DBPedia
QALD-7 [71]	215	DBPedia
QALD-8 [72]	219	DBPedia
QALD-9 [73]	408	DBPedia
SimpleQuestions	>100 k 49 k 43 k	Freebase [20] Wikidata [21] DBPedia [22]
LC-QuAD 1.0 [74]	5 k	DBPedia
LC-QuAD 2.0 [19]	30 k	DBPedia and Wikidata
WebQuestions [17]	6 k	Freebase
Free197 [75]	917	Freebase
ComplexQuestions [76]	150	Freebase
ComplexWebQuestionsSP [18]	34 k	Freebase
ConvQuestions [77]	11 k	Wikidata
TempQuestions [78]	1 k	Freebase
NLPCC-ICCPOL [79]	24 + k	NLPCC-ICCPOL
BioASQ ³	100 per-task	Freebase

¹ <http://qald.aksw.org/index.php?x=home&q=1> accessed on 27 June 2021; ² <http://qald.aksw.org/index.php?x=home&q=2> accessed on 27 June 2021; ³ <http://bioasq.org> accessed on 27 June 2021.

```

TITLE-ABS-KEY("question answering")
AND
(TITLE-ABS-KEY("semantic web") OR TITLE-ABS-KEY("knowledge graph")
OR TITLE-ABS-KEY("knowledge base"))
AND
(LIMIT-TO(PUBSTAGE, "final"))
AND
(LIMIT-TO(DOCTYPE, "cp") OR LIMIT-TO(DOCTYPE, "ar"))
AND
(LIMIT-TO(PUBYEAR, 2021) OR
LIMIT-TO(PUBYEAR, 2020) OR
LIMIT-TO(PUBYEAR, 2019) OR
LIMIT-TO(PUBYEAR, 2018) OR
LIMIT-TO(PUBYEAR, 2017) OR
LIMIT-TO(PUBYEAR, 2016) OR
LIMIT-TO(PUBYEAR, 2015) OR
LIMIT-TO(PUBYEAR, 2014) OR
LIMIT-TO(PUBYEAR, 2013) OR
LIMIT-TO(PUBYEAR, 2012) OR
LIMIT-TO(PUBYEAR, 2011) OR
LIMIT-TO(PUBYEAR, 2010)) AND
(LIMIT-TO(LANGUAGE, "English"))

```

Figure A1. Query for collecting article from 2010 to 2021.

QALD

QALD is a series of evaluation benchmark that answer questions over related data. It has so far been organized as an ESWC workshop and an ISWC workshop and a part of CLEF's Question Answering lab. Table A2 provides QALD challenges of each version.

Table A2. QALD Challenges.

Series	Challenges
QALD-1	Heterogeneous and distributed interlinked data
QALD-2	Linked data interaction
QALD-3	Multilingual QA and ontology lexicalization
QALD-4, 5, and 6	Multilingual on hybrid of interlinked data sets (structured and unstructured data)
QALD-7	HOBBIT for big linked data
QALD-8 and 9	Web of data (GERBIL QA challenge)

SimpleQuestions

SimpleQuestions consists of more than 100k questions written by human annotators and associated with Freebase facts [20]. SimpleQuestions contains single relation factoid questions where the facts are written in triple forms (subject, relation, object). Since Freebase is no longer available for free, facts of SimpleQuestions is also associated with Wikidata.

LC-QuAD

LC-QuAD is built and organized by the University of Bonn, German. There are two series of LC-QuAD, namely LC-QuAD 1.0 and LC-QuAD 2.0. LC-QuAD 1.0 contains 5000 question pairs and SPARQL queries corresponding to it. Specifically, in the April

2016 edition, the target Knowledge Graph is DBpedia [74]. In 2018, LC-QuAD 2.0 was established. It is the extension of the previous one, namely LC-QuAD 1.0. LC-QuAD 2.0 consists of 30,000 questions with their corresponding SPARQL. Wikidata and DBpedia are the target Knowledge Graph; specifically, the version for 2018 [19].

WebQuestions

This data set contains more than 6k questions/answer pairs associated with Freebase. The questions are mostly centered around a single named entity [17].

Free917

Free917 contains 917 data separated into 641 training examples and 276 testing examples. Each sample contains two fields: utterances in natural language and its logical forms [75].

ComplexQuestions

Abujabal et al. [76] constructed this data set to address issues appeared in complex questions. It consists of 150 test questions that exhibit multiple clauses in a compositional way.

ComplexWebQuestionsSP

Like ComplexQuestions data set, this data set is also used to evaluate a KGQA systems that address complex questions. The difference is that this data set contains questions that refer to event multiple entities such as (“Who did Brad Pitt play in Troy?”) [17].

ConvQuestions

ConvQuestions is a benchmark that contains 11k questions for a real conversation evaluated over Wikidata. It consists of a variety of complex questions such as comparisons, aggregations, compositionality, and reasoning. It also covers five domains: Movies, Books, Soccer, TV Series, and Music [77].

TempQuestions

TempQuestions is used to evaluate temporal issues in a KGQA system. The temporal questions with gold-standard answers in this data set are about 1k questions where the questions are collected from Free917, WebQuestions, and ComplexQuestions data sets by selecting time-related questions from them [78].

NLPCC-ICCPOL

NLPCC-ICCPOL is the largest public Chinese KGQA benchmark data set. There exists 14,609 and 9870 pairs of question–answer in training and testing data, respectively [79].

BioASQ

BioASQ contains challenges of semantic biomedical questions. There are three challenges of BioASQ, namely a large indexing of biomedical semantic, information retrieval, summarization, and QA. Task MESINESP On Medical Semantic Indexing In Spanish (<http://www.bioasq.org> accessed on 27 June 2021).

Table A3. The number of articles collected from ACM Digital Library, IEEE Xplore, and Science Direct.

Publication Venue	Step 1	Step 2	Step 3
SIGIR	4	3	0
WWW	12	12	1
TOIS	1	1	1
CIKM	10	9	1
VLDB	2	2	0
SIGMOD	3	3	0
IMCOM	2	2	0
SBD	1	1	0
EDBT	1	1	0
SEM	2	1	0
GIR	1	1	0
WSDM	1	1	1
SIGWEB	1	0	0
IHI	1	1	0
TURC	1	1	0
CoRR	3	3	3
Number of articles from ACM Digital Library	46	42	7
BESC	1	1	1
DCABES	1	1	0
ICCI*CC	1	1	0
ICCIA	2	2	0
ICDE	1	1	0
ICINCS	1	1	0
ICITSI	1	1	0
ICSC	2	2	0
IEEE Access	3	3	1
IEEE Intelligent Systems	1	0	0
IEEE TKDE	1	1	0
IEEE/ACM TASLP	1	1	1
IJCNN	1	0	0
ISCID	1	1	0
PIC	1	1	0
TAAI	1	1	0
Number of articles from IEEE Xplore	20	18	3
Artificial Intelligent in Medicine	1	0	0
Experts Systems with Applications	3	2	0
Information Processing and Management	1	1	0
Informations Sciences	2	1	0
Journal of Web Semantics	2	2	0
Neural Networks	1	1	0
Neurocomputing	1	1	0
Number of articles from Science Direct	11	8	0

Table A4. The number of articles collected from Scopus.

Publication Venue	Step 1	Step 2	Step 3
EMNLP	2	2	1
Turkish Journal of Electrical Engineering and Computer Sciences	1	1	0
Data Technologies and Applications	1	1	0
ICDM	1	1	0
IHMSC	1	1	0
AICPS	1	1	0
WETICE	1	1	0
Cluster Computing	1	1	0
ICWI	1	1	0
Data Science and Engineering	1	1	0
LNCS	3	3	1
NAACL	2	2	0
CEUR Workshop Proceedings	3	3	0
LNI	1	1	0
SWJ	1	1	0
PLOS ONE	1	1	0
Information (Switzerland)	1	1	0
SEKE	1	1	0
ICSC	1	1	0
ISWC	2	2	0
COOLING	1	1	0
ACL	2	2	2
Knowledge-Based Systems	1	1	0
IJCNW	1	1	1
WWW	1	1	1
Number of articles from Scopus	35	35	7
Total (# ACM Digital Library + # IEEE Xplore + # Scopus)	112	103	17

References

1. Yang, M.; Lee, D.; Park, S.; Rim, H. Knowledge-based question answering using the semantic embedding space. *Expert Syst. Appl.* **2015**, *42*, 9086–9104, doi:10.1016/j.eswa.2015.07.009.
2. Xu, K.; Feng, Y.; Huang, S.; Zhao, D. Hybrid Question Answering over Knowledge Base and Free Text. In Proceedings of the COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, Osaka, Japan, 11–16 December 2016; Calzolari, N., Matsumoto, Y., Prasad, R., Eds.; ACL: Stroudsburg, PA, USA, 2016; pp. 2397–2407.
3. Zheng, W.; Yu, J.X.; Zou, L.; Cheng, H. Question Answering Over Knowledge Graphs: Question Understanding Via Template Decomposition. *Proc. VLDB Endow. PVLDB* **2018**, *11*, 1373–1386, doi:10.14778/3236187.3236192.
4. Hu, S.; Zou, L.; Yu, J.X.; Wang, H.; Zhao, D. Answering Natural Language Questions by Subgraph Matching over Knowledge Graphs. *IEEE Trans. Knowl. Data Eng.* **2018**, *30*, 824–837, doi:10.1109/TKDE.2017.2766634.
5. Zhang, X.; Meng, M.; Sun, X.; Bai, Y. FactQA: Question answering over domain knowledge graph based on two-level query expansion. *Data Technol. Appl.* **2020**, *54*, 34–63, doi:10.1108/DTA-02-2019-0029.
6. Bakhshi, M.; Nematbakhsh, M.; Mohsenzadeh, M.; Rahmani, A.M. Data-driven construction of SPARQL queries by approximate question graph alignment in question answering over knowledge graphs. *Expert Syst. Appl.* **2020**, *146*, 113205, doi:10.1016/j.eswa.2020.113205.
7. Zhang, H.; Cai, J.; Xu, J.; Wang, J. Complex Question Decomposition for Semantic Parsing. In Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, 28 July– 2 August 2019; Volume 1: Long Papers; Korhonen, A., Traum, D.R., Márquez, L., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 4477–4486, doi:10.18653/v1/p19-1440.
8. Ji, G.; Wang, S.; Zhang, X.; Feng, Z. A Fine-grained Complex Question Translation for KBQA. In Proceedings of the ISWC 2020 Demos and Industry Tracks: From Novel Ideas to Industrial Practice Co-Located with 19th International Semantic Web Conference (ISWC 2020), Globally Online, 1–6 November 2020 (UTC); Taylor, K.L., Gonçalves, R.S., Lécué, F., Yan, J., Eds.; 2020; Volume 2721, pp. 194–199.
9. Shin, S.; Lee, K. Processing knowledge graph-based complex questions through question decomposition and recomposition. *Inf. Sci.* **2020**, *523*, 234–244, doi:10.1016/j.ins.2020.02.065.

10. Lu, X.; Pramanik, S.; Roy, R.S.; Abujabal, A.; Wang, Y.; Weikum, G. Answering Complex Questions by Joining Multi-Document Evidence with Quasi Knowledge Graphs. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, 21–25 July 2019; Piwowarski, B., Chevalier, M., Gaussier, É., Maarek, Y., Nie, J., Scholer, F., Eds.; ACM: New York, NY, USA, 2019; pp. 105–114, doi:10.1145/3331184.3331252.
11. Vakulenko, S.; Garcia, J.D.F.; Polleres, A.; de Rijke, M.; Cochez, M. Message Passing for Complex Question Answering over Knowledge Graphs. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, 3–7 November 2019; Zhu, W., Tao, D., Cheng, X., Cui, P., Rundensteiner, E.A., Carmel, D., He, Q., Yu, J.X., Eds.; ACM: New York, NY, USA, 2019; pp. 1431–1440, doi:10.1145/3357384.3358026.
12. Hu, S.; Zou, L.; Yu, J.X.; Wang, H.; Zhao, D. Answering Natural Language Questions by Subgraph Matching over Knowledge Graphs (Extended Abstract). In Proceedings of the 34th IEEE International Conference on Data Engineering, ICDE 2018, Paris, France, 16–19 April 2018; IEEE Computer Society: Washington, DC, USA, 2018; pp. 1815–1816, doi:10.1109/ICDE.2018.00265.
13. Zhu, S.; Cheng, X.; Su, S.; Lang, S. Knowledge-based Question Answering by Jointly Generating, Copying and Paraphrasing. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, 6–10 November 2017; Lim, E., Winslett, M., Sanderson, M., Fu, A.W., Sun, J., Culpepper, J.S., Lo, E., Ho, J.C., Donato, D., Agrawal, R., et al., Eds.; ACM: Washington, DC, USA, 2017; pp. 2439–2442, doi:10.1145/3132847.3133064.
14. Zhang, R.; Wang, Y.; Mao, Y.; Huai, J. Question Answering in Knowledge Bases: A Verification Assisted Model with Iterative Training. *ACM Trans. Inf. Syst.* **2019**, *37*, 40:1–40:26, doi:10.1145/3345557.
15. Lan, Y.; Wang, S.; Jiang, J. Knowledge Base Question Answering With a Matching-Aggregation Model and Question-Specific Contextual Relations. *IEEE/ACM Trans. Audio, Speech Lang. Process.* **2019**, *27*, 1629–1638, doi:10.1109/TASLP.2019.2926125.
16. Türe, F.; Jovic, O. No Need to Pay Attention: Simple Recurrent Neural Networks Work! In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, 9–11 September 2017; Palmer, M., Hwa, R., Riedel, S., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2017; pp. 2866–2872, doi:10.18653/v1/d17-1307.
17. Berant, J.; Chou, A.; Frostig, R.; Liang, P. Semantic Parsing on Freebase from Question-Answer Pairs. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, Grand Hyatt Seattle, Seattle, Washington, USA, 18–21 October 2013; A Meeting of SIGDAT, a Special Interest Group of the ACL: Stroudsburg, PA, USA, 2013; pp. 1533–1544.
18. Talmor, A.; Berant, J. The Web as a Knowledge-Base for Answering Complex Questions. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, 1–6 June 2018, Volume 1 (Long Papers); Walker, M.A., Ji, H., Stent, A., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2018; pp. 641–651, doi:10.18653/v1/n18-1059.
19. Dubey, M.; Banerjee, D.; Abdelkawi, A.; Lehmann, J. LC-QuAD 2.0: A Large Dataset for Complex Question Answering over Wikidata and DBpedia. In Proceedings of the Semantic Web—ISWC 2019—18th International Semantic Web Conference, Auckland, New Zealand, 26–30 October 2019, Proceedings, Part II; Lecture Notes in Computer Science; Ghidini, C., Hartig, O., Maleshkova, M., Svátek, V., Cruz, I.F., Hogan, A., Song, J., Lefrançois, M., Gandon, F., Eds.; Springer: Berlin/Heidelberg, Germany, 2019; Volume 11779, pp. 69–78, doi:10.1007/978-3-030-30796-7_5.
20. Bordes, A.; Usunier, N.; Chopra, S.; Weston, J. Large-scale Simple Question Answering with Memory Networks. *arXiv* **2015**, arXiv:1506.02075.
21. Diefenbach, D.; Tanon, T.P.; Singh, K.D.; Maret, P. Question Answering Benchmarks for Wikidata. In Proceedings of the ISWC 2017 Posters & Demonstrations and Industry Tracks co-located with 16th International Semantic Web Conference (ISWC 2017), Vienna, Austria, 23–25 October 2017; CEUR Workshop Proceedings, Nikitina, N., Song, D., Fokoue, A., Haase, P., Eds.; 2017; Volume 1963.
22. Azmy, M.; Shi, P.; Lin, J.; Ilyas, I.F. Farewell Freebase: Migrating the SimpleQuestions Dataset to DBpedia. In Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, NM, USA, 20–26 August 2018; Bender, E.M., Derczynski, L., Isabelle, P., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2018; pp. 2093–2103.
23. Höffner, K.; Walter, S.; Marx, E.; Usbeck, R.; Lehmann, J.; Ngomo, A.N. Survey on challenges of Question Answering in the Semantic Web. *Semant. Web* **2017**, *8*, 895–920, doi:10.3233/SW-160247.
24. Diefenbach, D.; López, V.; Singh, K.D.; Maret, P. Core techniques of question answering systems over knowledge bases: A survey. *Knowl. Inf. Syst.* **2018**, *55*, 529–569, doi:10.1007/s10115-017-1100-y.
25. Fu, B.; Qiu, Y.; Tang, C.; Li, Y.; Yu, H.; Sun, J. A Survey on Complex Question Answering over Knowledge Base: Recent Advances and Challenges. *arXiv* **2020**, arXiv:2007.13069.
26. Cimiano, P.; Minock, M. Natural Language Interfaces: What Is the Problem?—A Data-Driven Quantitative Analysis. In Proceedings of the Natural Language Processing and Information Systems, 14th International Conference on Applications of Natural Language to Information Systems, NLDB 2009, Saarbrücken, Germany, 24–26 June 2009; Revised Papers; Lecture Notes in Computer Science; Horacek, H., Métais, E., Muñoz, R., Wolska, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2009; Volume 5723, pp. 192–206, doi:10.1007/978-3-642-12550-8_16.
27. López, V.; Uren, V.S.; Sabou, M.; Motta, E. Is Question Answering fit for the Semantic Web? A survey. *Semantic Web* **2011**, *2*, 125–155, doi:10.3233/SW-2011-0041.
28. Freitas, A.; Curry, E.; Oliveira, J.G.; O’Riain, S. Querying Heterogeneous Datasets on the Linked Data Web: Challenges, Approaches, and Trends. *IEEE Internet Comput.* **2012**, *16*, 24–33, doi:10.1109/MIC.2011.141.

29. Hogan, A.; Blomqvist, E.; Cochez, M.; d'Amato, C.; de Melo, G.; Gutiérrez, C.; Gayo, J.E.L.; Kirrane, S.; Neumaier, S.; Polleres, A.; et al. Knowledge Graphs. *arXiv* **2020**, arXiv:abs/2003.02320.
30. Cyganiak, R.; Wood, D.; Lanthaler, M.; (Eds.). *RDF 1.1 Concepts and Abstract Syntax*; W3C Recommendation 25 February 2014. Available online: <http://travesia.mecd.es/portaln/jspui/bitstream/10421/2427/1/RDF%201.pdf> (accessed on 27 June 2021).
31. Harris, S.; Seaborne, A.; (Eds.). *SPARQL 1.1 Query Language*; W3C Recommendation 21 March 2013. Available online: <https://www.w3.org/TR/sparql11-query/> (accessed on 27 June 2021).
32. 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 International Conference on Management of Data, SIGMOD 2014, Snowbird, UT, USA, 22–27 June 2014; Dyreson, C.E., Li, F., Özsu, M.T., Eds.; ACM: New York, NY, USA, 2014; pp. 313–324, doi:10.1145/2588555.2610525.
33. Shin, S.; Jin, X.; Jung, J.; Lee, K. Predicate constraints based question answering over knowledge graph. *Inf. Process. Manag.* **2019**, *56*, 445–462, doi:10.1016/j.ipm.2018.12.003.
34. Unger, C.; Bühmann, L.; Lehmann, J.; Ngomo, A.N.; Gerber, D.; Cimiano, P. Template-based question answering over RDF data. In Proceedings of the 21st World Wide Web Conference 2012, WWW 2012, Lyon, France, 16–20 April 2012; Mille, A., Gandon, F.L., Misselis, J., Rabinovich, M., Staab, S., Eds.; ACM: New York, NY, USA, 2012; pp. 639–648, doi:10.1145/2187836.2187923.
35. Huang, X.; Zhang, J.; Li, D.; Li, P. Knowledge Graph Embedding Based Question Answering. In Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019, Melbourne, VIC, Australia, 11–15 February 2019; Culpepper, J.S., Moffat, A., Bennett, P.N., Lerman, K., Eds.; ACM: New York, NY, USA, 2019; pp. 105–113, doi:10.1145/3289600.3290956.
36. Dai, Z.; Li, L.; Xu, W. CFO: Conditional Focused Neural Question Answering with Large-scale Knowledge Bases. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, Berlin, Germany, 7–12 August 2016; Volume 1: Long Papers; The Association for Computer Linguistics: Stroudsburg, PA, USA, 2016, doi:10.18653/v1/p16-1076.
37. Yin, W.; Yu, M.; Xiang, B.; Zhou, B.; Schütze, H. Simple Question Answering by Attentive Convolutional Neural Network. In Proceedings of the COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, Osaka, Japan, 11–16 December 2016; Calzolari, N., Matsumoto, Y., Prasad, R., Eds.; ACL: 2016; pp. 1746–1756.
38. Chao, Z.; Li, L. The Combination of Context Information to Enhance Simple Question Answering. In Proceedings of the 5th International Conference on Behavioral, Economic, and Socio-Cultural Computing, BESC 2018, Kaohsiung, Taiwan, 12–14, November 2018; IEEE: Kaohsiung, Taiwan, 2018; pp. 109–114, doi:10.1109/BESC.2018.8697305.
39. Lukovnikov, D.; Fischer, A.; Lehmann, J. Pretrained Transformers for Simple Question Answering over Knowledge Graphs. In Proceedings of the Semantic Web—ISWC 2019—18th International Semantic Web Conference, Auckland, New Zealand, 26–30 October 2019, Proceedings, Part I; Lecture Notes in Computer Science; Ghidini, C., Hartig, O., Maleshkova, M., Svátek, V., Cruz, I.F., Hogan, A., Song, J., Lefrançois, M., Gandon, F., Eds.; Springer: Berlin/Heidelberg, Germany, 2019; Volume 11778, pp. 470–486, doi:10.1007/978-3-030-30793-6_27.
40. Zhao, W.; Chung, T.; Goyal, A.K.; Metallinou, A. Simple Question Answering with Subgraph Ranking and Joint-Scoring. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, 2–7 June 2019, Volume 1 (Long and Short Papers); Burstein, J., Doran, C., Solorio, T., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 324–334, doi:10.18653/v1/n19-1029.
41. Li, L.; Zhang, M.; Chao, Z.; Xiang, J. Using context information to enhance simple question answering. *World Wide Web* **2021**, *24*, 249–277, doi:10.1007/s11280-020-00842-7.
42. He, X.; Golub, D. Character-Level Question Answering with Attention. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, TX, USA, 1–4 November 2016; Su, J.; Carreras, X.; Duh, K., Eds. The Association for Computational Linguistics: Stroudsburg, PA, USA, 2016; pp. 1598–1607, doi:10.18653/v1/d16-1166.
43. Lukovnikov, D.; Fischer, A.; Lehmann, J.; Auer, S. Neural Network-based Question Answering over Knowledge Graphs on Word and Character Level. In Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, 3–7 April 2017; Barrett, R., Cummings, R., Agichtein, E., Gabrilovich, E., Eds.; ACM: New York, NY, USA, 2017; pp. 1211–1220, doi:10.1145/3038912.3052675.
44. Wang, R.; Ling, Z.; Hu, Y. Knowledge Base Question Answering With Attentive Pooling for Question Representation. *IEEE Access* **2019**, *7*, 46773–46784, doi:10.1109/ACCESS.2019.2909826.
45. Luo, D.; Su, J.; Yu, S. A BERT-based Approach with Relation-aware Attention for Knowledge Base Question Answering. In Proceedings of the 2020 International Joint Conference on Neural Networks, IJCNN 2020, Glasgow, UK, 19–24 July 2020; pp. 1–8, doi:10.1109/IJCNN48605.2020.9207186.
46. Zhang, L.; Lin, C.; Zhou, D.; He, Y.; Zhang, M. A Bayesian end-to-end model with estimated uncertainties for simple question answering over knowledge bases. *Comput. Speech Lang.* **2021**, *66*, 101167, doi:10.1016/j.csl.2020.101167.
47. Lee, J.; Kim, S.; Song, Y.; Rim, H. Bridging Lexical Gaps between Queries and Questions on Large Online Q&A Collections with Compact Translation Models. In Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, EMNLP 2008, Proceedings of the Conference, Honolulu, Hawaii, USA, 25–27 October 2008; A Meeting of SIGDAT, A Special Interest Group of the ACL; ACL: Stroudsburg, PA, USA, 2008; pp. 410–418.
48. Goodfellow, I.J.; Bengio, Y.; Courville, A.C. *Deep Learning*; Adaptive Computation and Machine Learning, MIT Press: Cambridge, MA, USA, 2016.

49. Aggarwal, C.C. *Neural Networks and Deep Learning—A Textbook*; Springer: Berlin/Heidelberg, Germany, 2018, doi:10.1007/978-3-319-94463-0.
50. Zhang, A.; Lipton, Z.C.; Li, M.; Smola, A.J. Dive into Deep Learning. 2020. Available online: <https://d2l.ai> (accessed on).
51. Lin, T.; Mausam.; Etzioni, O. Entity Linking at Web Scale. In Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-Scale Knowledge Extraction, AKBC-WEKEX@NAACL-HLT 2012, Montréal, QC, Canada, 7–8 June 2012; Fan, J., Hoffman, R., Kalyanpur, A., Riedel, S., Suchanek, F.M., Talukdar, P.P., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2012; pp. 84–88.
52. Bordes, A.; Usunier, N.; García-Durán, A.; Weston, J.; Yakhnenko, O. Translating Embeddings for Modeling Multi-Relational Data. In Proceedings of the Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013, Lake Tahoe, NV, USA, 5–8 December 2013; Burges, C.J.C., Bottou, L., Ghahramani, Z., Weinberger, K.Q., Eds.; MIT Press: Cambridge, MA, USA, 2013; pp. 2787–2795.
53. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, 2–7 June 2019, Volume 1 (Long and Short Papers); Burstein, J., Doran, C., Solorio, T., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 4171–4186, doi:10.18653/v1/n19-1423.
54. Wang, Q.; Mao, Z.; Wang, B.; Guo, L. Knowledge Graph Embedding: A Survey of Approaches and Applications. *IEEE Trans. Knowl. Data Eng.* **2017**, *29*, 2724–2743, doi:10.1109/TKDE.2017.2754499.
55. Camacho-Collados, J.; Pilehvar, M.T. From Word To Sense Embeddings: A Survey on Vector Representations of Meaning. *J. Artif. Intell. Res.* **2018**, *63*, 743–788, doi:10.1613/jair.1.11259.
56. Goyal, P.; Ferrara, E. Graph embedding techniques, applications, and performance: A survey. *Knowl. Based Syst.* **2018**, *151*, 78–94, doi:10.1016/j.knosys.2018.03.022.
57. Nickel, M.; Tresp, V.; Kriegel, H. A Three-Way Model for Collective Learning on Multi-Relational Data. In Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, WA, USA, 28 June–2 July 2011; Getoor, L.; Scheffer, T., Eds.; Omnipress: Madison, WI, USA, 2011; pp. 809–816.
58. García-Durán, A.; Bordes, A.; Usunier, N. Effective Blending of Two and Three-way Interactions for Modeling Multi-relational Data. In Proceedings of the Machine Learning and Knowledge Discovery in Databases—European Conference, ECML PKDD 2014, Nancy, France, 15–19 September 2014, Proceedings, Part I; Lecture Notes in Computer Science; Calders, T., Esposito, F., Hüllermeier, E., Meo, R., Eds.; Springer: Berlin/Heidelberg, Germany, 2014; Volume 8724, pp. 434–449, doi:10.1007/978-3-662-44848-9\ 28.
59. Wang, Z.; Zhang, J.; Feng, J.; Chen, Z. Knowledge Graph Embedding by Translating on Hyperplanes. In Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, 27–31 July 2014, Québec City, QC, Canada; Brodley, C.E., Stone, P., Eds.; AAAI Press: Menlo Park, CA, USA, 2014; pp. 1112–1119.
60. He, S.; Liu, K.; Ji, G.; Zhao, J. Learning to Represent Knowledge Graphs with Gaussian Embedding. In Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, 19–23 October 2015; Bailey, J., Moffat, A., Aggarwal, C.C., de Rijke, M., Kumar, R., Murdock, V., Sellis, T.K., Yu, J.X., Eds.; ACM: New York, NY, USA, 2015; pp. 623–632, doi:10.1145/2806416.2806502.
61. Wang, Z.; Zhang, J.; Feng, J.; Chen, Z. Knowledge Graph and Text Jointly Embedding. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, Doha, Qatar, 25–29 October 2014, A Meeting of SIGDAT, a Special Interest Group of the ACL; Moschitti, A., Pang, B., Daelemans, W., Eds.; ACL: 2014; pp. 1591–1601. doi:10.3115/v1/d14-1167.
62. Sherkat, E.; Milios, E.E. Vector Embedding of Wikipedia Concepts and Entities. In Proceedings of the Natural Language Processing and Information Systems—22nd International Conference on Applications of Natural Language to Information Systems, NLDB 2017, Liège, Belgium, 21–23 June 2017; Lecture Notes in Computer Science; Frasincar, F., Ittoo, A., Nguyen, L.M., Métais, E., Eds.; Springer: Berlin/Heidelberg, Germany, 2017; Volume 10260, pp. 418–428, doi:10.1007/978-3-319-59569-6_50.
63. Camacho-Collados, J.; Pilehvar, M.T.; Navigli, R. Nasari: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities. *Artif. Intell.* **2016**, *240*, 36–64, doi:10.1016/j.artint.2016.07.005.
64. Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; Brew, J. HuggingFace’s Transformers: State-of-the-art Natural Language Processing. *arXiv* **2019**, arXiv:abs/1910.03771.
65. Madnani, N.; Dorr, B.J. Generating Phrasal and Sentential Paraphrases: A Survey of Data-Driven Methods. *Comput. Linguist.* **2010**, *36*, 341–387, doi:10.1162/coli_a_00002.
66. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. In Proceedings of the 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, 24–26 April 2017.
67. Cabrio, E.; Cimiano, P.; López, V.; Ngomo, A.N.; Unger, C.; Walter, S. QALD-3: Multilingual Question Answering over Linked Data. In Proceedings of the Working Notes for CLEF 2013 Conference, Valencia, Spain, 23–26 September 2013; Forner, P., Navigli, R., Tufis, D., Ferro, N., Eds.; 2013; Volume 1179.
68. Unger, C.; Forascu, C.; López, V.; Ngomo, A.N.; Cabrio, E.; Cimiano, P.; Walter, S. Question Answering over Linked Data (QALD-4). In Proceedings of the Working Notes for CLEF 2014 Conference, Sheffield, UK, 15–18 September 2014; Cappellato, L., Ferro, N., Halvey, M., Kraaij, W., Eds.; 2014, Volume 1180, pp. 1172–1180.

69. Unger, C.; Forascu, C.; López, V.; Ngomo, A.N.; Cabrio, E.; Cimiano, P.; Walter, S. Question Answering over Linked Data (QALD-5). In Proceedings of the Working Notes of CLEF 2015—Conference and Labs of the Evaluation forum, Toulouse, France, 8–11 September 2015; Cappellato, L., Ferro, N., Jones, G.J.F., SanJuan, E., Eds.; 2015, Volume 1391.
70. Unger, C.; Ngomo, A.N.; Cabrio, E. 6th Open Challenge on Question Answering over Linked Data (QALD-6). In Proceedings of the Semantic Web Challenges—Third SemWebEval Challenge at ESWC 2016, Heraklion, Crete, Greece, 29 May–2 June 2016, Revised Selected Papers; Communications in Computer and Information Science; Sack, H., Dietze, S., Tordai, A., Lange, C., Eds.; Springer: Berlin/Heidelberg, Germany, 2016; Volume 641, pp. 171–177, doi:10.1007/978-3-319-46565-4_13.
71. Usbeck, R.; Ngomo, A.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 Challenges—4th SemWebEval Challenge at ESWC 2017, Portoroz, Slovenia, 28 May–1 June 2017, Revised Selected Papers; Communications in Computer and Information Science; Dragoni, M., Solanki, M., Blomqvist, E., Eds.; Springer: Berlin/Heidelberg, Germany, 2017; Volume 769, pp. 59–69, doi:10.1007/978-3-319-69146-6_6.
72. Usbeck, R.; Ngomo, A.N.; Conrads, F.; Röder, M.; Napolitano, G. 8th Challenge on Question Answering over Linked Data (QALD-8) (invited paper). In Proceedings of the 4th Workshop on Semantic Deep Learning (SemDeep-4) and NLIWoD4: Natural Language Interfaces for the Web of Data (NLIWOD-4) and 9th Question Answering over Linked Data challenge (QALD-9) Co-Located with 17th International Semantic Web Conference (ISWC 2018), Monterey, CA, USA, 8–9 October 2018; Choi, K., Anke, L.E., Declerck, T., Gromann, D., Kim, J., Ngomo, A.N., Saleem, M., Usbeck, R., Eds.; 2018; Volume 2241, pp. 51–57.
73. Usbeck, R.; Gusmita, R.H.; Ngomo, A.N.; Saleem, M. 9th Challenge on Question Answering over Linked Data (QALD-9) (invited paper). In Proceedings of the 4th Workshop on Semantic Deep Learning (SemDeep-4) and NLIWoD4: Natural Language Interfaces for the Web of Data (NLIWOD-4) and 9th Question Answering over Linked Data Challenge (QALD-9) Co-Located with 17th International Semantic Web Conference (ISWC 2018), Monterey, CA, USA, 8–9 October 2018; Choi, K., Anke, L.E., Declerck, T., Gromann, D., Kim, J., Ngomo, A.N., Saleem, M., Usbeck, R., Eds.; 2018; Volume 2241, pp. 58–64.
74. Trivedi, P.; Maheshwari, G.; Dubey, M.; Lehmann, J. LC-QuAD: A Corpus for Complex Question Answering over Knowledge Graphs. In Proceedings of the Semantic Web—ISWC 2017—16th International Semantic Web Conference, Vienna, Austria, 21–25 October 2017, Proceedings, Part II; Lecture Notes in Computer Science; d’Amato, C., Fernández, M., Tamma, V.A.M., Lécué, F., Cudré-Mauroux, P.; Sequeda, J.F.; Lange, C.; Heflin, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2017; Volume 10588, pp. 210–218, doi:10.1007/978-3-319-68204-4_22.
75. Cai, Q.; Yates, A. Large-scale Semantic Parsing via Schema Matching and Lexicon Extension. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, Sofia, Bulgaria, 4–9 August 2013, Volume 1: Long Papers; The Association for Computer Linguistics: Stroudsburg, PA, USA, 2013; pp. 423–433.
76. Abujabal, A.; Yahya, M.; Riedewald, M.; Weikum, G. Automated Template Generation for Question Answering over Knowledge Graphs. In Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, 3–7 April 2017; Barrett, R., Cummings, R., Agichtein, E., Gabrilovich, E., Eds.; ACM: New York, NY, USA, 2017; pp. 1191–1200, doi:10.1145/3038912.3052583.
77. Christmann, P.; Roy, R.S.; Abujabal, A.; Singh, J.; Weikum, G. Look before you Hop: Conversational Question Answering over Knowledge Graphs Using Judicious Context Expansion. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, 3–7 November 2019; Zhu, W., Tao, D., Cheng, X., Cui, P., Rundensteiner, E.A., Carmel, D., He, Q., Yu, J.X., Eds.; ACM: New York, NY, USA, 2019; pp. 729–738, doi:10.1145/3357384.3358016.
78. Jia, Z.; Abujabal, A.; Roy, R.S.; Strötgen, J.; Weikum, G. TempQuestions: A Benchmark for Temporal Question Answering. In Proceedings of the Companion of the Web Conference 2018 on the Web Conference 2018, WWW 2018, Lyon, France, 23–27 April 2018; Champin, P., Gandon, F.L., Lalmas, M., Ipeirotis, P.G., Eds.; ACM: New York, NY, USA, 2018; pp. 1057–1062, doi:10.1145/3184558.3191536.
79. Shen, C.; Huang, T.; Liang, X.; Li, F.; Fu, K. Chinese Knowledge Base Question Answering by Attention-Based Multi-Granularity Model. *Information* **2018**, *9*, 98, doi:10.3390/info9040098.