Multilingual Question Answering Systems for Knowledge Graphs—A Survey

Aleksandr Perevalov [a,b,*], Andreas Both [a,c] and Axel-Cyrille Ngonga Ngomo [d]

[a] *Faculty of Computer Science and Media, Leipzig University of Applied Sciences, Germany*
E-mail: {aleksandr.perevalov,andreas.both}@htwk-leipzig.de
[b] *Department of Computer Science and Languages, Anhalt University of Applied Sciences, Germany*
[c] *Technology Innovation Unit, DATEV eG, Germany*
[d] *Data Science Group (DICE), University of Paderborn, Germany*
E-mail: axel.ngonga@upb.de

Abstract. This paper presents a survey on multilingual Knowledge Graph Question Answering (mKGQA). We employ a systematic review methodology to collect and analyze the research results in the field of mKGQA by defining scientific literature sources, selecting relevant publications, extracting objective information (e.g., problem, approach, evaluation values, used metrics, etc.), thoroughly analyzing the information, searching for novel insights, and methodically organizing them. Our insights are derived from 46 publications: 26 papers specifically focused on mKGQA systems, 14 papers concerning benchmarks and datasets, and 7 systematic survey articles. Starting its search from 2011, this work presents a comprehensive overview of the research field, encompassing the most recent findings pertaining to mKGQA and Large Language Models. We categorize the acquired information into a well-defined taxonomy, which classifies the methods employed in the development of mKGQA systems. Moreover, we formally define three pivotal characteristics of these methods, namely resource efficiency, multilinguality, and portability. These formal definitions serve as crucial reference points for selecting an appropriate method for mKGQA in a given use case. Lastly, we delve into the challenges of mKGQA, offer a broad outlook on the investigated research field, and outline important directions for future research. Accompanying this paper, we provide all the collected data, scripts, and documentation in an online appendix.

Keywords: Knowledge Graph Question Answering, Multilinguality, Question Answering Dataset, Survey, Systematic Review

1. Introduction

The most popular search engines on the Web process dozens of billions of queries per day[1]. Up to half of these queries are *informational*, i.e., are sent by users with an information need pertaining to a certain topic. Informational queries can be formulated as full-fledged natural language (NL) questions (e.g., “How old is Donald Trump?”), or as keyword-based questions (e.g., “Donald Trump age”). In both cases, users of the Web expect a search engine to provide a *direct answer* (or a fact) in a precise way instead of a list of relevant Web pages or documents [1, 2]. Direct answers or facts can be extracted based on unstructured data (e.g., text or HTML document) or based on structured data (e.g., database, knowledge base, or graph). The systems dealing with a direct answer extraction over unstructured data are referred to as *Information Retrieval-based Question Answering* (IRQA) systems [3]. On the other hand, systems that perform a direct answer extraction over structured data are called *Knowledge-based*

[*] Corresponding author. E-mail: {aleksandr.perevalov,andreas.both}@htwk-leipzig.de.

[1] https://www.internetlivestats.com/google-search-statistics/

Question Answering (KBQA) systems [3]. Let us *focus on the latter class* of systems. *The objective of KBQA* is to identify answers $\mathcal{A}$ that fulfill an informational need of a NL question q, utilizing a KB $\mathcal{K}$ [4]. Recent KBQA developments effort in two development paradigms [5–7]: (1) the information extraction style and (2) the semantic parsing style. The *first development paradigm of KBQA* is known for utilizing mostly neural approaches, as it usually retrieves a set of answer candidates that are ranked or filtered based on a particular feature space. The *second development paradigm of KBQA* mostly utilizes symbolic approaches, however, also combines the neural ones; there the task is to convert a NL question q into a formal query q' (e.g., SPARQL) that are to be executed on a KB $\mathcal{K}$ with the sake of retrieving an answer. Recent advances in the field of *Large Language Models* (LLMs) have shown that both of the aforementioned development paradigms can be implemented with their means. In particular, one may directly pose a NL question q to this model to expect a direct answer (paradigm 1) [7, 8] or create an instruction for an LLM model to generate a SPARQL query (paradigm 2) [9].

While speaking about the Web, the extraction of direct answers based on structured data is enabled by the introduction of *the Semantic Web* [10], which aims at making the Web data machine-readable. The Semantic Web corresponds to the formal definition of a Knowledge Graph (KG) [11] and, therefore, can be considered as a giant decentralized KG. A dominant share of all KGs as well as the Semantic Web itself are described with the *Resource Description Framework* (RDF) [12] (RDF-based KGs). A family of systems addressing the challenge of giving direct answers based on KGs is named *Knowledge Graph Question Answering (KGQA) systems*. *KGQA systems have the same objective as KBQA* ones, however, a KB $\mathcal{K}$ is replaced with a knowledge graph $\mathcal{KG}$. Despite the terms KBQA and KGQA being mostly used interchangeably, we prefer to consider *KGQA systems as a subset of KBQA*. The answer types supported by the majority of the KGQA systems are limited to the following: resource[2] (e.g., a URI of an entity), literal (e.g., a string or a number), boolean, or a list of the previously mentioned ones. These answers can be converted back into NL using natural language generation (NLG) frameworks for KGs [13]. The majority of research-oriented KGQA systems work on general-domain KGs such as Freebase [14] (now part of the Google's KG [15]), DBpedia [16], Wikidata [17], etc. However, the techniques implemented therein can commonly be ported to any KG. As the Semantic Web is a giant KG indexed by many dominant search engines, *KGQA systems and its underlying technologies are serving a major role in providing users with access to structured information available on the Web*. This statement is supported by a substantial number of papers related to KGQA and affiliated with the organizations responsible for the modern major search engines [18–20]. KGQA systems find its *industrial application* in any place where data is structured as a KG. For instance, a product or its development process documentation can be stored as RDF documents [21]. Another example is material science data [22] or chemical data [23]. Hence, such data structured as RDF enables its users to build a KGQA system on top of it. Recent work has provided the KGQA community with such *applicability domains* as tourism [24], biomedicine [25], and even "COVID-19 pandemic" [26]. Nevertheless, we have observed that *providing access to KGs in multiple languages is still a challenge* for such systems. In the following paragraphs, we contextualize this challenge in detail.

It is well known that the Web, which in this context is frequently confused with the Internet, is the major information source for people all over the world [27–29] in different domains of life. Despite that, the majority of the information on the Web (53.6%)[3] is only accessible to a minor fraction of people (25.9%)[4], namely English speakers. This information imbalance is also the reason for a "cultural gap" on the Web [30]. Consequently, users who cannot write or read English on a certain level have limited access to the major share of the information available on the Web. This phenomenon in popular science is referred to as the *digital language divide*[5].

The scope of this paper is limited to *multilingual and cross-lingual Knowledge Graphs Question Answering (mKGQA) systems*. mKGQA systems extend standard KGQA functionality by *providing a possibility of processing questions or searching for information in several different languages* $l \in \mathcal{L}$ [31]. In our case "several" means *more than one*. Hence, a user may pose questions in different languages: $q_{l_1}, \dots, q_{l_n}$, where $n = |\mathcal{L}|$. At the same time, a mKGQA system may search for answers in KGs in different languages: $\mathcal{KG}_{l_1}, \dots, \mathcal{KG}_{l_n}$ (if multilingual information is not merged into one KG instance). Let us assume that a user writes the question q in the language l_i and a system

[2] https://www.w3.org/DesignIssues/Generic.html

[3] https://w3techs.com/technologies/overview/content_language, retrieved 2023-08-27

[4] https://www.internetworldstats.com/stats7.htm, retrieved 2023-08-27

[5] http://labs.theguardian.com/digital-language-divide/

finds an answer $\mathcal{A}$ in the $\mathcal{KG}_{l_j}$ (instantiated for language l_j). As it was mentioned before, the *ability to handle more than one language is called multilinguality*, however, *if $l_i \neq l_j$ given the example above, such an ability is called cross-linguality*. Please note that in the scope of our paper, the terms "cross-lingual" and "multilingual" are interchangeable, and we will therefore utilize the latter one. Hence, we regard mKGQA systems as being a research topic of particular importance as *these systems can bridge the gap between the users and the information on the Semantic Web* published in different languages.

Our analysis of the related work published over the past decade suggests (see Section 3) that *currently available systematic surveys on KGQA barely address the aspect of multilinguality*. In particular, the majority of the related surveys dedicate one paragraph or less to multilinguality while mentioning it as a challenge for KGQA. In addition, none of the related work concentrates specifically on the multilingual aspect of these systems. As there are KGQA systems that explicitly focus on multilinguality, there is clearly a need for a survey on mKGQA. This paper is guided by the following *research questions*: $\mathcal{RQ}1$—identify, structure, and characterize methods used for dealing with mKGQA; $\mathcal{RQ}2$—analyze the evaluation results, distribution of languages, and presence of different language groups, alphabets, and writing system among the publications about mKGQA; $\mathcal{RQ}3$—discover chronological tendencies in benchmarking datasets size and number of languages. The *goals of this paper* are (1) to analyze the work from the past and provide an overview of the current advances of mKGQA systems, (2) to propose a generalized taxonomy of the methods for mKGQA and define their characteristics, (3) to find out current research challenges and formulate future research directions. Note that in this work, we consider publications describing mKGQA systems, relevant survey articles, and corresponding benchmarking datasets as our work aims to cover the research field as generally as possible.

In this work, we employ a systematic review methodology posited in [32–35] (see Section 2) to collect and analyze the research results in the field of mKGQA by defining scientific literature sources, selecting relevant publications, extracting objective information (e.g., evaluation values, used metrics, etc.), analyzing the information, searching for new insights as well as generalizing them in a structured manner (i.e., in the form of a taxonomy). Finally, we summarize our observations and present a general outlook on the investigated research direction. Our insights are mainly derived from 46 publications, which were selected from more than a thousand publications that were retrieved during the initial selection phase. After the manual verification of the formal selection criteria, we selected 26 papers about mKGQA systems, 14 papers about benchmarks and datasets, and 7 systematic survey articles. To ensure the transparency and reproducibility of this work, we provide all the collected data, scripts, and documentation as an online appendix.

This article is structured as follows. In Section 2, we describe the methodology of the systematic survey. Section 3 contains the overview of the related systematic surveys about KGQA. In Section 4, we review the mKGQA systems and propose the taxonomy of the methods. The benchmarks for the mKGQA are reviewed in Section 5. We analyze and discuss the results of the work in Section 6. The article is concluded in Section 7.

2. Systematic Review Methodology of this Survey

To ensure transparency and reproducibility, we followed a strict systematic review methodology, which is based on prior literature [32–35]. In this section, we describe the methodology explicitly within the context of the actual review execution process. *The methodology consists of the following three major phases*: selection of sources, initial publications' selection, as well as extraction and systematization of the information. The phases are described in the following subsections. It is worth mentioning that only the authors of this work were involved in the review process. The first author led the review process by conducting the respective steps (e.g., writing scripts for automated information extraction (Section 2.2), manual information extraction (Section 2.3) etc.). The other authors cross-checked the work of the first author. All the authors were making regular synchronization meetings to ensure mutual agreement.

2.1. Selection of Sources

For the sources, we used *well-established digital research databases related to computer science, which offer free access to the advanced search features*. While following our multilingual agenda, we went beyond the English language for literature search, namely, we used *sources in the following languages: English, German, and Russian*. We chose these languages as for each of them at least one of the authors is a native speaker. To identify the sources we used the 3 search engines – Google[6], Bing[7], Yandex[8] – with a specific search string in different languages. The following search queries were used to find the sources:

- English: `("digital library" or "research database") and "computer science"`;
- German: `("digitale Bibliothek" or "forschungsdatenbank") and "informatik"`[9];
- Russian: `("цифровая библиотека" or "индекс цитирования") and "информатика"`[10].

Thereafter, the search results were processed according to the following *acceptance criteria*: collect articles related to the computer science field, provide studies in at least one of the selected languages, and offer free access to the advanced command search features. Finally, the *following sources were selected* for the further phases: IEEE Xplore[11], ACM DL[12], Springer[13], DBLP[14], ACL Anthology[15], Cyberleninka[16].

2.2. Initial Publications Selection

To search for publications, we used digital research databases (sources) that were selected during the previous phase. With the advanced search functionality and the corresponding complex search queries, *the three main aspects of the publications had to be covered*:

1. System aspect—Question Answering systems;
2. Data aspect—RDF-based Knowledge Graphs;
3. Language aspect—Multilinguality and cross-linguality.

We considered the *publications of the following types*: it describes a system, a benchmarking dataset, or it is a survey publication. Thus, the publications needed to match the following *acceptance criteria*: it describes a QA system, a related benchmarking dataset, or is a systematic survey; the described system or the benchmark are intended to work on RDF-based KGs; the described system or the dataset are intended to work with multiple languages (at least two). Furthermore, only the *publications released in the period from 2011 to 2023 were considered*[17]. As we ensured that our review process is reproducible, we repeated it multiple times so as not to miss any relevant publications before finalizing this work. The last time *the publications list was updated is February 15th, 2024*.

For each of the sources, *we utilized a complex search query that covers all the three aspects described above*. The conceptual form of the search query is presented in Table 1. After that, *we automatically extracted the following publication properties*: authors, title, abstract, publication year, DOI/URL, source, and publisher. The script and the corresponding documentation are available in the online appendix[18]. Based on this information, we manually assessed and cross-checked the publications according to the acceptance criteria described in the beginning of this section. The statistics regarding the selected and accepted publications grouped by the sources are shown in Table 2.

Considering our research background with mKGQA, we identified that *our systematic review methodology has high specificity*. This means that some of the relevant publications, known to us before, were not included in the review process. Therefore, we integrated one *exception to the selection process*: *we included publications that we*

[6] https://google.com/

[7] https://www.bing.com/

[8] https://yandex.ru/

[9] Literally translated as: `("digital library" or "research database") and "informatics"`

[10] Literally translated as: `("digital library" or "citation index") and "informatics"`

[11] https://ieeexplore.ieee.org

[12] https://dl.acm.org

[13] https://www.springer.com

[14] https://dblp.org

[15] https://aclanthology.org

[16] https://cyberleninka.ru/

[17] This work started in 2021; the original intention was to cover publications of the past decade, which is why we started searching from 2011.

[18] https://github.com/Perevalov/multilingual-KGQA-survey

Table 1
The conceptual representation of the query and its corresponding parts. The parts are concatenated with the AND operator.

Aspect	Query part
System	("Semantic search" OR "Question Answer*" OR "Question-Answer*" OR "KBQA" OR "KGQA" OR "KB QA" OR "KB-QA" OR "KG-QA" OR "KG QA" OR "NLI" OR "NLIDB" OR "QA" OR "Natural Language Interface")
Data	("Knowledge Base*" OR "Knowledge Graph*" OR "DBpedia" OR "Wikidata" OR "YAGO" OR "Semantic Web" OR "Linked Data" OR "RDF*" OR "data web" OR "SPARQL" OR "Query Graph" OR "Web data" OR "WWW" OR "web of data" OR "QALD*" OR "SimpleQuestions" OR "WebQuestions" OR "WebQSP" OR "LC-QuAD" OR "RuBQ" OR "SimpleDBpediaQA" OR "ComplexWebQuestions" OR "MCWQ")
Language	("multilingual*" OR "multi-lingual" OR "crosslingual*" OR "cross-lingual" OR "internationalized" OR "multilingualism" OR "multilinguistic" OR "multilanguage" OR "bilingual")

Table 2
Statistics on the selected and accepted publications grouped by its sources

#	IEEE Xplore	ACM DL	Springer	DBLP	Cyberleninka	ACL Anthology	Related Work	Total
Selected	19	289	1366	140	38	16	12	1880
Accepted	2	7	16	11	1	4	5	46

previously were aware of and that matched the following criteria: match the three main aspects (see above), and cited at least five times or published through a peer-reviewed process (see column "Related Work" in Table 2). The share of the "Related Work" publications source is roughly 10%.

2.3. Extraction and Systematization of the Information

After the initial publications selection phase, all the *accepted publications were manually analyzed* in a more detailed way. In particular, along with the annotated information (authors, title, abstract, publication year, DOI/URL, source, publisher), we manually extracted factual information from the publications, which is needed to answer research questions, and transformed it into a *tabular format* with the following columns:

- Paper type—describing a system, a dataset, or a systematic survey;
- Problem—textual description of the problem;
- Approach—proposal of authors on how to resolve the problem in general terms;
- Solution—actual results of the authors towards solving the problem;
- Languages—a set of languages that were used regarding the multilingual aspect;
- Knowledge graphs—set of knowledge graphs that were directly used in the work;
- Datasets—set of datasets that were mentioned or directly used in a publication;
- Metrics—a set of metrics used for the evaluation in a publication;
- Technologies—a set of technologies that were mentioned explicitly in a paper or seen in a repository;
- Source code & demo URLs—the links to the source code or/and demo application;
- Comment—an optional brief comment or remarks on the publication.

Thereafter, we cross-checked[19] the selected publications and rejected the ones that did not satisfy the acceptance criteria described in Section 2.1. Finally, the number of the publications to be considered was narrowed down to 46, of which 26 describe mKGQA systems, 14 are about benchmarking datasets, and the other 7 are the survey papers. The full table is available in the online appendix.

[19] The first author did the manual extraction process, the other authors checked the resulted information.

Table 3

The overview of the survey papers that include the aspect of multilinguality

Authors	Year	Problem	Methodology	# Papers	Multilinguality
Höffner et al. [36]	2016	The SOTA methods are not systematically collected	✓	72	Multilingual systems understanding noisy, human natural language input
Diefenbach et al. [37]	2018	Making an "enormous amount of information in the form of KBs" available with KBQA	✓	n/a	The vocab in the user query and the KB vocab are lexicalized in different languages
Dimitrakis et al. [38]	2020	No dataset/benchmark surveys were available	✗	n/a	Mentioned as a challenge
Franco da Silva et al. [39]	2020	No surveys available with a focus on Deep Learning	✓	13	Mentioned as a challenge
Zhang et al. [5]	2021	Recent advances in Deep Learning in the KBQA	✗	n/a	Lack of data in languages other than English
Antoniou et al. [40]	2022	No taxonomy of the systems was available	✗	n/a	Mentioned as a challenge
Pereira et al. [35]	2022	KBQA for data accessibility in the biomedical domain	✓	66	Mentioned as a challenge

3. Systematic Surveys about Question Answering over Knowledge Graphs

In this section, we review the survey articles that are, first of all, related to the considered research field. Secondly, they have been chosen according to the methodology described in Section 2. In Table 3 we present an overview of the publications, described below.

3.1. Overview of the Surveys

In 2017, Höffner et al. [36] dedicated their survey to the following problem: instead of a shared effort, *many essential components are redeveloped*. While *shared practices emerge over time, they are not systematically collected*. Moreover, as the authors describe it, *most systems focus on a specific aspect while the others are quickly implemented, which leads to low benchmark scores and thus undervalues the contribution*. The authors propose to mitigate these problems by *systematically collecting and structuring methods of dealing with common challenges faced by the used approaches*. The methodology consists of the following inclusion criteria for the publications: available via Google Scholar[20] through a predefined search query[21] or published at one of the major Semantic Web conferences within a predefined publication time span. Thereafter, the found publications are manually analyzed for compliance with the survey topic. The authors review 72 publications about 62 systems developed from 2010 to 2015. Secondly, they identified challenges faced by those approaches and collected solutions for them from the 72 publications. Finally, they draw conclusions and make recommendations on how to develop future semantic question answering (SQA) systems. The authors state that future research should be directed at more modularization, automatic reuse, self-wiring, and encapsulated modules with their own benchmarks and evaluations. They also notice *the movement towards multilingual, multi-knowledge source SQA systems* that are capable of understanding noisy, human natural-language input. However, the described survey dedicates only one paragraph to mKGQA systems.

The survey of Diefenbach et al. [37], which was published in 2017, targets *the problem of making an "enormous amount of information in the form of knowledge bases" available with the help of question answering systems*. The authors claim that they *focus on the techniques behind existing QA systems (unlike the other articles)*. They consider five tasks (question analysis, phrase mapping, disambiguation, query construction, and querying distributed knowledge) in the QA process and describe how QA systems solve them. The defined main goal of the authors is to describe, classify, and compare all techniques used by QA systems participating in the QALD[22] challenge [41]. The methodology of the survey has the following system selection process: the authors considered the QA systems that either directly participated in the QALD challenges or that were evaluated afterward, reusing the same evaluation set-up. To identify the latter systems, the authors search through Google Scholar for all publications mentioning or citing the publications of the QALD challenges. From among these, the authors take the publications referring to QA systems with the exclusion of controlled NL systems (i.e., publications that employ the approach of controlled NL are excluded from the review). The authors mention *multilingual functionality only in the case when the vocabulary*

[20] https://scholar.google.com/

[21] The search query: `'question answering' AND ('Semantic Web' OR 'data web')`

[22] Question Answering over Linked Data

in the user query and the KB vocabulary are expressed (lexicalized) in different languages. There are *only a few sentences mentioning multilinguality as an issue*.

In 2020, da Silva et al. [39] published the survey on end-to-end "simple QA systems". The authors claim that in *the traditional approaches, the process of answering a question can be divided into five steps corresponding to question analysis, phrase mapping, disambiguation, query construction, and querying distributed knowledge*. However, given the improvements in deep neural network models and higher availability of training data, end-to-end architectures have become the state of the art. To conduct a systematic survey, the authors decided to focus on deep learning-based QA systems designed to answer factoid questions. In particular, they describe how each existing system addresses its critical features in terms of training end-to-end models. The authors also make the evaluation process on these systems and discuss how each approach differs from the others in terms of the challenges tackled and the strategies employed. The methodology of the survey has the following inclusion criteria: 1. an initial search[23] for works in QA which adopt deep learning techniques; (2) the scope is reduced to the systems that are evaluated using the SimpleQuestions benchmark; (3) only those works providing clues for answering the research questions set by the authors were considered. The authors select publications published between 2015 and 2019. The initial search brought 59 papers. After applying the described criteria, 13 papers remained. Multilinguality is highlighted in the survey as a challenge that is focused on performing mediation between the users' need for information in their local languages and the semantic data that is often expressed in a culturally biased manner. However, *only one paragraph of this article is dedicated to multilinguality*.

The survey article of Dimitrakis et al. [38] was published in 2020. The authors claim that *the other surveys published up to 2018 are reviewing only the corresponding QA systems, while this survey contains a detailed list of available training/evaluation datasets for QA*. Another distinctive feature of the survey is that *it discusses how different types of QA systems and information sources can be combined into a unified pipeline to help researchers find combinatorial ways that can be more effective*. As a result, the authors review approaches covering text-based, data-based, and hybrid methods as well as the corresponding datasets. Note that no publication selection methodology was described by the authors. *The multilingual aspect is covered only by a small paragraph*.

In 2021, Zhang et al. [5] published their paper on deep learning in KBQA. The authors claim that the recent advances in deep learning are entering the KBQA field to improve the corresponding systems. The survey reviews *recent deep learning-based KBQA efforts for simple questions* in two main streams: (1) the information extraction style and (2) the semantic parsing style. Then, the authors switch to *the efforts that extend the neural architectures to answer more complex questions that require multi-hop deep reasoning*. Finally, *several well-known benchmarks for evaluating KBQA systems are reviewed* (e.g., WebQuestions [42], SimpleQuestions [43], LC-QuAD [44]). The following challenges are mentioned by the authors as remaining: compositional generalizability, the gap between the natural language and a knowledge base, lack of training data, limited coverage of KBs, and lack of data in languages other than English. The publication selection protocol was not described in the survey published. *There is only one sentence dedicated to the multilingual aspect*.

The survey on KBQA by Pereira et al. [35] was published in 2022. The authors tackle the problem of the KB data accessibility as *the visual navigation approach is not rich enough to answer more complex questions*, and *querying using SPARQL is not suitable for users who have not mastered the use of formal querying languages*. The survey is mainly focused on the biomedical data domain. The authors follow a strict methodology—PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) guidelines [45] to report the protocol's execution and present the findings. In this survey, 66 documents were analyzed to classify KBQA systems according to their architectural styles. The survey reviews 25 semantic parsing pipeline systems, 12 using subgraph matching, 7 based on templates, and 22 performing information extraction. The authors believe that on the one hand, it is necessary to answer increasingly complex questions, and on the other hand, there is a need to deal with the inherent incompleteness of KBs. There is only one paragraph dedicated to the multilingual functionality.

In 2022, Antoniou et al. [40] published a survey on SQA systems. The authors claim that no categories of SQA systems have been identified (no typology/taxonomy) before this survey. Hence, at the date of publication, there

[23] The search query: `(("simple questions answering" OR "end to end" OR "user to end" OR "machine learning") and ("natural language" OR "nlp" OR "natural language understand") and (application OR system OR program OR reviews OR techniques OR frameworks OR "practical applications"))`

are no surveys for the categorization of SQA systems. This survey distinguishes categories of SQA systems based on criteria in order to lay the groundwork for a collection of common practices, as no categories of SQA systems have been identified. The authors believe that the categorization and systematization can help developers, or anyone interested to find out directly the technique or steps used by each system or to benchmark their own system against existing ones. The classification created in the survey is based on the following properties: domain, data source, types of questions, types of analysis done on questions, types of representations used for questions, characteristics of the KB, techniques used for retrieving answers, user interaction, and answers. The methodology of the survey is not described. *The authors dedicate only one paragraph to the multilingual functionality.*

3.2. Summary

From the survey articles considered above, we can clearly see that none of them target the multilingual aspect of KGQA specifically. However, multilinguality is mentioned in all of the publications as an important challenge. *The absence of a survey paper dedicated to mKGQA specifically is the main motivation for conducting this work.* It was also noted that four out of seven of the papers do not publish their methodology or review protocol, hence, the reproducibility of their work is questionable. Therefore, *we encourage the research community to include the methodology or the protocol in their survey papers.*

4. Multilingual Question Answering Systems over Knowledge Graphs

In this section, we review the 21 mKGQA systems that were discovered in the 26 publications[24] obtained during our selection process (see Section 2). In Section 4.1, we summarize every publication that considers a particular mKGQA system. There, we identify particular attributes of the systems (e.g., supported languages or knowledge graphs, code availability) that help us to structure this data systematically. Therefore, based on the findings from the aforementioned section, we present a structured overview of the mKGQA systems in Table 4. In Section 4.2, we derive individual methods and method groups that are used for the development of mKGQA systems and use this information to assign the reviewed systems to particular methods or method groups according to their description. Finally, Section 4.3 proposes a blueprint for the evaluation of mKGQA systems and assigning them to particular "performance quadrants" according to their quality results.

4.1. Review of the Selected Systems

In this section, we group the reviewed systems by methods that were used for their implementation. These method groups are described in detail in Section 4.2. Despite some systems may belong to multiple groups, we still assign them to one according to the dominance of a particular method group within a system.

4.1.1. Systems Predominantly Using Methods Based on Rules and Templates (G1)

The QALL-ME system [46] was published in 2011 and is designed to provide relevant information and answers to arbitrary questions of its users. The authors regard this task as a challenge because of the "the exponential growth of digital information". Therefore, the authors of QALL-ME propose a reusable architecture for building multilingual QA systems that answer questions with the help of structured answer data sources from freely specifiable domains. The workflow of the system is managed by a software module named QA Planner, which orchestrates the QA components and thus passes the input question through the whole system until an answer is found. In particular, such components are language identifier, entity annotator, term annotator, temporal expression annotation, query generator, and answer retriever. The authors do not explicitly mention what methods were used for the implementation, however, for the query generation pattern mappings are used. The system implements a Service-Oriented Architecture (SOA). The authors claim that *QALL-ME works with German, Spanish, English, and Italian*. The system is intended to work on a non-public RDF-based knowledge graph and also evaluated on a benchmark that was not

[24] Some systems are described in multiple publications as they present minor improvements

published on the web, hence, it is not possible to compare QALL-ME with other systems. Nevertheless, the authors present results based on the accuracy (72.89% average for all languages) and Recognizing Textual Entailment (RTE) component performance measures (86.97% average for all languages). The authors claim that more attention is needed regarding the acquisition of minimal question patterns and interactive QA process. It is worth mentioning that the authors provide the source code of the system, written in Java, which is currently outdated. The QA components used in the system mostly work in a dictionary-based setting and thus are challenging to port to other datasets.

The KGQA system authored by Aggarwal [47] was published in 2012. The author targets the problem of the poor accuracy of multilingual natural-language interfaces that provide access to the Semantic Web data. The approach of Aggarwal is a cross-lingual semantic search method, which aims to retrieve all relevant pieces of information even if they are available in languages different from the initial question's language. Similarly to the previous paper – QALL-ME – this system is implemented with a multilingual QA pipeline that performs entity search (exact match between the entity and ontology label), parse tree generation (using the Stanford Parser[25] [48]), and computes cross-lingual semantic similarity and relatedness. The solution is implemented in Java. *Aggarwal's system provides answers for English and German questions* using the DBpedia knowledge graph. The evaluation is carried out on the QALD-2 dataset while measuring Precision (44%), Recall (48%), and F1 score (46%). Despite the acceptable quality, the author sees room for improvement pertaining to the existing semantic relatedness measures. In this regard, we also see that the semantic relatedness measures are mostly corpus-based, while nowadays it could be implemented with LMs [49] or graph neural networks [50].

The QAKiS system [51, 52] was originally published in 2013 with an extension in 2014 [53]. The authors of QAKiS focus on the inequality of the information in the multilingual DBpedia chapters (i.e., chapters can contain different information from one language to another) providing more specificity on certain topics. Thus, the ability to utilize all the information across the languages would be beneficial for QA systems. The approach is targeted at enhancing users' interactions with the Web of Data by providing query interfaces that provide flexible mappings between NL expressions, concepts, and relations in structured KBs. The implementation of the QAKiS systems contains four main multilingual modules: Named Entity Recognition—NER (Stanford Core NLP NER[26]), pattern matcher, query generator, SPARQL package. The main idea of the solution is to utilize the "relational patterns" that capture different ways to express a certain relation in a given language. *The QAKiS works with English, French, and German* languages while answering questions over DBpedia. The system is evaluated on the QALD-2 dataset and the reported precision is 50%. The authors still claim their mapping extension approach has room for improvement. It has to be mentioned that the mappings have to be created for each language individually and thus limiting the generalizability of the system.

The SWSNL system [54] was published in 2013. The authors aim at simplifying form-based search by introducing a system that is able to search over domain-specific data with NL- or keyword-based input. The approach of the authors is to create a component-based KGQA system, which is similar to QALL-ME and Aggarwal et al., that is able to answer questions over domain-specific RDF-based KG. The resulting solution is a Java-based application that converts a textual question into a KG independent query (with preprocessing, NER, and semantic parsing) and thereafter transforms it to SPARQL using a rule-based interpretation approach. The authors create the target KG by crawling one of the accommodation websites and integrating its information into a custom ontology. For the QA evaluation, the authors collect and annotate a dataset of 68 questions in *English and Czech* languages. The results regarding Precision, Recall, and F1 score demonstrate the following values respectively: 66%, 34%, 45%. The authors set possible follow-up contributions as follows: extension of the evaluation corpus, integration of a full-text search, and improvement of a NER module as well as the performance of the system.

The AMUSE system was published in 2017 [55]. The authors target the "lexical gap" while mapping natural-language questions to SPARQL queries, especially in a multilingual setting. The approach within the AMUSE system is based on probabilistic inference, which is aimed at predicting the query that has the highest probability of being the correct interpretation of the given question string. The actual implementation has two levels and is

[25] http://nlp.stanford.edu/software/lex-parser.shtml
[26] https://stanfordnlp.github.io/CoreNLP/ner.html

built with the help of Universal Dependencies (UD) [56] and Java. The first layer (L2KB) is trained using an entity linking objective that learns to link parts of the query to identifiers. The second layer is a query construction layer that takes the top k results from the L2KB layer and assigns semantic representations to the words to yield a logical representation of the complete question with the help of a semantic parse tree. The final output of the system is a SPARQL query. *The AMUSE systems works with English, German, and Spanish* over DBpedia KG. The evaluation is performed on the QALD-6 dataset where the following macro F1 score values were obtained: 34%, 37%, 42% for the supported languages respectively. The authors see that questions that require modifiers (e.g., filtering) to be present in the corresponding SPARQL queries may become an improvement for their system.

The WDAqua-core0 system [57] was released in 2017. The authors aim at the problem of handling the growing amount of structured Semantic Web data. The system uses a combinatorial approach based on the semantics encoded in the underlying KG. The implementation of the WDAqua-core0 is carried out using the Qanary framework [58, 59]. It can answer questions that require not only `SELECT` queries, but also `ASK`, and `COUNT`. Moreover, it can answer both NL and keyword-based questions. *WDAqua-core0 supports English, French, German, and Italian.* It is designed to support DBpedia and Wikidata and is evaluated on the QALD-7 dataset [60]. The questions over DBpedia are evaluated only using the English language, resulting in the F1 score of 51.1%. The system achieves the following F1 score values on Wikidata: 32.2%, 12.7%, 24.0%, and 17.3% for the supported languages respectively. The paper does not reveal a large number of implementation details. However, the corresponding follow-up papers (WDAqua-core1 [61] and QAnswer [62]) do provide further details pertaining to the internal workings of the framework.

The KGQA system UDepLambda [63] was released in 2017. The authors underline the problem of the particular focus on the English language in the publications related to the KGQA. Similarly to the AMUSE system, the proposed approach is to convert the NL questions to logical forms which are thereafter converted to machine-interpretable representations. The actual solution is also based on the universal dependencies [64] and maps NL to logical forms, representing underlying predicate-argument structures, in an almost language-independent manner. *The system UDepLambda works with English, German, and Spanish languages* over Freebase KG. The evaluation is done on two benchmarks WebQuestions and GraphQuestions. For the first benchmark, the following F1 score values are obtained: 49.5%, 46.1%, and 47.5% respectively for the supported languages. Given the second benchmark, the reported F1 score values are 17.7%, 9.5%, and 12.8% respectively to the supported languages. Despite the reasonable results of the system, the questions in languages other than English were machine-translated.

The system MuG-QA [65] was released in 2018 and is targeting the problem of handling the data within the rapid development of RDF, KGs, and the increase of non-English data. The approach of answering questions in the multilingual setting is focused on forming abstract conceptual grammar from the questions. Once a question is parsed, the resulting abstract grammar tree is matched with a KG to formulate a SPARQL query. The MUG-QA grammar is formed using the Grammatical Framework (GF) [66] and GF Resource Grammar Library [67], the entities and classes are linked using "interlanguage-links-dataset" [68]. The system works with English, French, Italian, and German languages. The MuG-QA is evaluated on the QALD-7 benchmark, which contains queries over DBpedia. The resulting micro F1 score values are as follows: 67.7%, 56.6%, 65.6%, and 61.3% respectively for the supported languages. The authors define that the "semantic flexibility" of the system and adding more languages are possible improvement directions for their system. The grammar-based methods require experts and increased labor costs for creating them. Despite the abstract grammar tree being language-agnostic, one is still required to create mappings for introducing a new language.

The WDAqua-core1 system [61] was published in 2018 and extending its predecessor—WDAqua-core0. The authors claim that a KGQA solution that would be freely available will allow the setup of the corresponding services across many new data sources and will likely boost the publication of new RDF datasets. The approach of WDAqua-core1 is based on the assumption that the questions can be understood by ignoring the syntax while focusing only on the semantics of the words. The implementation consists of a modular pipeline that contains query expansion, query construction, query ranking, and answer decision. The system is also integrated into the Qanary framework. Query expansion finds all concepts related to a particular n-gram substring in a question, query construction combines the concepts using a pre-defined algorithm for query patterns, query ranking ranks the generated queries according to a set of manually constructed features, and answer decision utilizes a binary classifier for additional filtering of queries. *WDAqua-core1 supports English, German, French, Italian, and Spanish* languages.

The set of supported KGs includes DBpedia, Wikidata, MusicBrainz, and DBLP. The WDAqua-core1 system was evaluated on the QALD-{3,. . . ,7} benchmarks. For the reasons of compactness, we list here the F1 score results only for QALD-7: 42%, 25%, 18%, 18%, 18% for the supported languages respectively. The authors aim to apply their approach to new KGs and query multiple KGs simultaneously in a follow-up contribution.

The LAMA system [69] was published in 2018. The authors of the system target the conventional problem of the RDF data accessibility in the context of providing NL interface to RDF, s.t., a user does not have to learn a query language. As an approach, it is proposed to develop a QA system that is based on analyzing lexico-syntactic patterns that can help generate corresponding SPARQL queries, i.e., they search for generalized linguistic structures that denote semantic relationships between concepts. The actual KGQA solution contains several processing phases: preprocessing (syntax parsing and question classification), generation of additional intermediate structures (dependency tree, POS tags, question type), and core processing module, which transforms the syntax tree into an intermediate representation, and finally the intermediate representation is parsed to generate one or more triple patterns used in the final SPARQL. The solution is implemented using the following tools: SyntaxNet[27], Penn Treebank [70], OntoNotes [71], and Universal Dependencies. It is important to underline that the system uses the Google Translate API while working with languages other than English. Despite that, the authors claim that *the LAMA system works with French and English languages*. The system works over the DBpedia KG and is evaluated on QALD-7 and LC-QuAD 1.0 [44] benchmarks. The authors report the following F1 score evaluation values for English: 90.5% and 81.6% respectively for the benchmarks. The authors of LAMA set the following tasks as possible system extensions: enrichment of the dependency and POS patterns, checking for logical coherence between the system's output and the expected answers, and inclusion of the audio modality. It is worth mentioning that the lexico-syntactic patterns used in the system are handcrafted.

4.1.2. Systems Predominantly Using Statistical Methods (G2)

The UTQA system [72] was released in 2016. The authors highlight the particular focus of the KGQA research field on the English language only. In the authors' view, this happens because of several reasons: lack of multilingual tools and resources on the one side and "vocabulary gap" between source and target languages. The approach exploited in the UTQA system is based on a set of multilingual components that sequentially process a question: keyword extraction (using maximum-entropy Markov model, non-English ones are translated with Google Translate API[28]), keyword type detection (using an SVM classifier), entity linking and ontology type extraction (using custom queries over multiple data sources, semantic similarity, and Babelfy tool [73]), and answer extraction. *The UTQA system works with English, Persian, and Spanish* on the DBpedia KG. The evaluation of the system is performed on the QALD-5 benchmark using a (1) language-specific approach and (2) using machine-translations of the given questions with the following results: (1) F1 score for English 65.2%, Persian 52.4%, and Spanish 54.2%; the approach significantly outperforms the full machine translation to English, where the following F1 score values are computed: (2) for Persian 29.5% and for Spanish 32.2%. The authors of UTQA define the following extension directions: improving relation extraction, adapting the approach to the monolingual KGs, and to a cross-dialect setting. Despite the multilingual support, it is not clear whether the multilingual SVM model is used for keyword type detection or not.

The Platypus system [74] was released in 2018 and is available online[29]. The authors support the paradigm of answering NL questions over structured repositories of machine-readable facts. The main approach of the Platypus system is to represent questions not directly in SPARQL, but rather in a custom logical representation. This is implemented with the help of two analyzer components. The (1) grammatical analyzer translates a NL question into a logical representation with manually designed rules. The (2) template-based analyzer does the same while finding a template that best matches the question and thereafter fills the logical representation slots. Finally, both representations are converted into a SPARQL query and executed. The solution is implemented with the help of Stanford Core NLP, SpaCy [75], and Rasa NLU. *The Platypus system works with French and English languages* while answering questions over Wikidata. The system is trained on the WikidataSimpleQuestions dataset. However,

[27] https://ai.googleblog.com/2016/05/announcing-syntaxnet-worlds-most.html

[28] https://cloud.google.com/translate

[29] https://qa.askplatyp.us/

the evaluation results are not presented. The authors see the template-based analyzer as a possible system's limitation since it works only in English. We also assume that the custom logical representations can be considered as a bottleneck of the approach.

The QAnswer system, which is a follow-up of WDAqua-core0 and WDAqua-core1 was released in 2019 [62, 76]. The authors target the problem of the limited accessibility of a large amount of LOD datasets. This problem is based on the fact that the majority of the systems allow accessing only one dataset and one language. The proposed approach is the same as for the WDAqua-core1 system—it is multilingual and KG-agnostic. The QA process consists of the following 4 steps: question expansion, query construction, query ranking, and answer decision. The system is extended by introducing the feedback and re-training functionality based on a user's data. *The QAnswer supports English, German, French, Italian, Spanish, Portuguese, Arabic, and Chinese languages*. By default, the system is able to answer questions over Wikidata, DBpedia, MusicBrainz, DBLP, and Freebase. The evaluation results on the QALD-{3, ..., 7} and LC-QuAD 1.0 benchmarks are presented in the paragraph about the WDAqua-core1 system. In addition, the evaluation on unpublished "cocktail", "HR", and "EU" benchmarks reports the following F1 score values respectively: 92%, 97%, 90%. The authors would like to improve the relation identification component by introducing non-dictionary-based methods (e.g., word embeddings). It is worth highlighting the flexibility and production-orientedness of the approach. However, the query generation algorithm may be a bottleneck as it's not possible to foresee all possible query structures for all KGs.

The DeepPavlov system [77] was published in 2020. The authors of the system target the answering of complex questions with "logical or comparative reasoning". As an approach, it is proposed to decompose the task of KBQA into multiple steps or components: query template prediction, entity detection, entity linking, relation ranking, path ranking, constraint extraction (if the question has constraints), and generation of query from extracted entities, relations, and constraints. The components' pipeline is based on deep-learning neural networks. Classification of questions by query template type using the BERT [78] large language model (CLS token), Entity Detection with BERT-based sequence labeling, Entity Linking is implemented using fuzzy matching of the string extracted at Entity Detection step with inverted index, relation ranking implemented with extracting relation candidates from the linked entities, the question's token embeddings are passed to the 2-layer Bi-LSTM to obtain hidden states which are taken for the dot product of relation embeddings (of their title) and passed to Softmax layer (the model is trained to maximize the product of token embedding and right relation embedding), BERT is used for path ranking of relation candidates, regular expressions are used to extract modifiers. The solution is implemented using the Python programming language. *The DeepPavlov KBQA system supports only English and Russian languages*. The system is compatible with the Wikidata KG. The evaluation is done on the LC-QuAD 2.0 dataset, the authors reported the following values for Precision 60%, Recall 66%, and F1 score 63%. It is worth mentioning that the used models and therefore the whole system is quite resource-intensive, for proper functionality on a CPU machine it requires around 32 GB of RAM.

The authors of *the Tiresias system* [79], which was published in 2022, focus on improving the multilingual accessibility of the KGQA systems. In addition to the structured DBpedia information, the authors propose to use multilingual DBpedia abstracts as an additional information source. The Tiresias systems process a question in a sequential mode, in particular, (1) the main named entity is recognized with DBpedia Spotlight, (2) a DBpedia abstract is retrieved for the entity using a SPARQL query, (3) the question text is translated into English using Bing or Helsinki MT, and (4) the final answer is produced with a pre-trained BERT-like QA model. The authors evaluate their system on a custom bilingual dataset (*English and Greek*) with a manually defined approach that splits the results into correct, partially correct, and wrong. Hence, the evaluation results can not be compared to the other systems as no standard metrics are used in this work. The authors see the technical accessibility of the Tiresias, more information sources in the QA process, and the set of supported languages as possible extension areas.

The *DeepPavlov 2023 system* [80] was released in 2023. The authors' main objective is to provide a user with full NL answers verbalized with KG triplets. Among that, the previous version of this system [77] was improved w.r.t. QA quality and now represents state-of-the-art for KGQA on Russian RuBQ 2.0 benchmark [81]. The system conducts the following tasks: entity detection, entity linking, relation ranking, SPARQL template prediction, SPARQL slot filling, and path ranking. As a result of the latter step, a complete SPARQL query over Wikidata is generated, which can be executed to get an answer. In the answer generation step, the system takes the query paths with answer URIs and uses the JointGT model [82] to produce the answer text. The components of the system that conduct the aforementioned tasks are BERT-based models trained on different KGQA datasets, such as LC-QuAD 1.0 [44]. The DeepPavlov 2023 system works on *English and Russian* languages, however, those are two different system instances as it uses monolingual neural models. The evaluation of the English version is provided with the LC-QuAD dataset (47% F1 score), and the Russian version was evaluated on RuBQ 2.0 dataset (53.1% F1 score). The system is accompanied by a working source code. The authors set the future objectives as combining knowledge for the systems from both structured and unstructured sources.

The authors of *XSemPLR approach* [83] tackle the task of cross-lingual semantic parsing (CLSP) over SQL, lambda calculus, and other meaning representations (eight in total) including SPARQL. The authors claim that their main contribution is a unified benchmark for CLSP constructed from nine existing datasets. However, for this survey, the most important contribution is the evaluation of multilingual LLMs on KGQA benchmarks. In particular, for the CLSP over SPARQL the authors used MCWQ benchmark [84], which contains questions in *English, Hebrew, Kannada, and Chinese* with queries over Wikidata. The following LMs were evaluated: LSTM [85], mBERT+Pointer-based Decoders (PTR) [86], XLM-R+PTR [87], mBART [88], Codex [89], BLOOM [90], mT5 [91]. The aforementioned models were used with the following settings: monolingual, monolingual few-shot, multilingual, cross-lingual zero-shot transfer, cross-lingual few-shot transfer. The highest results were provided by the mT5 model in the monolingual setting. Based on the *exact match* metric, the results are as follows: 39.29%, 33.02%, 23.74%, and 24.56% (for the aforementioned languages). Nevertheless, the authors claim that multilingual LLMs (e.g., BLOOM) are still inadequate to perform CLSP tasks. This work is provided with the source code for the evaluation. The authors define a challenge of a performance gap between monolingual training and cross-lingual transfer learning.

The CLRN system [92] represents a new approach to engage with the challenges of Cross-lingual KGQA (CLKGQA). Traditional methods typically revolve around the melding of multiple CLKGs into one consolidated KG. However, the authors challenge this approach, emphasizing shortcomings in the ability of existing Entity Alignment (EA) models to accurately align entity pairs in CLKGs. The authors suggest two important challenges to address: dependency of a QA model on a unified KG, and enhancement of an EA model's performance. To tackle these issues, they propose the Cross-lingual Reasoning Network (CLRN), a revolutionary multi-hop QA model that allows for flexible shifting between knowledge graphs at any point in the multi-hop reasoning process. Further, they establish an iterative framework that couples the CLRN and EA models to extract potential alignment triple pairs from the CLKGs during the QA procedure, thus enhancing the performance of the EA model. Their experimental results demonstrate that the CLRN outperforms other baselines. The experiments were conducted on the MLPQ [93] benchmark that incorporates language-specific DBpedia KGs in *English, Chinese, and French*. The authors particularly note meaningful improvement in the EA model's performance through iterative enhancement, leading to a statistically significant 1.0% increase in Hit@1 and Hit@10. Additionally, they open up an interesting discourse on the relationship between QA and EA from the QA perspective. The authors make their dataset and code publicly available, furthering the scope for future explorations.

4.1.3. Systems Predominantly Using Machine Translation Methods (G3)

The system authored by Y. Zhou et al. [94] was published in 2021. The authors aim to meet the rising demand for KGQA systems by answering multilingual questions. On the other hand, building a large-scale KG, as well as annotating QA data, is costly for each new language. Therefore, there is a considerable KGQA performance gap between source and target languages, which is consistent with the empirical results on a wide range of other tasks by prior works. The idea of the approach is to pre-train a multilingual transformer encoder in a self-supervised manner. Thereafter, fine-tune the multilingual encoder on the data of a data-rich (source) language. The assumption is that the fine-tuned model is generalizable enough to perform inference in other low-resource (target) languages. This paradigm can be adapted to KGQA in order to construct symbolic logical forms for KG queries. It is also proposed to replace the full-supervised machine translator with unsupervised bilingual lexicon induction (BLI) [95] for word-level translation. The actual implementation is using a BLI-based augmentation for multilingual training data. Thereafter, the encoder is adapted to the augmented data. The adversarial learning strategy coupled with BLI-based augmentation is proposed for robust cross-lingual transfer. *The system by Y. Zhou et al. is capable of working with English, Farsi, German, Romanian, Italian, Russian, French, Dutch, Spanish, Hindi, and Portuguese languages*. As the system works with the DBpedia KG, the evaluation is done on LC-QuAD 1.0 (translated to multiple languages with Google Translate) and QALD-9 benchmarks. The average F1 score values across the languages are 75.9% and 63.0% for the benchmarks, respectively. While considering the fact that the LC-QuAD 1.0 is machine-translated for the evaluation, the performance of the system may be questionable.

The system by A. Perevalov et al. [96] was published in 2022. The authors focus on the problem of unequal language distribution on the Web and therefore unequal content accessibility. In addition, only a few research initiatives are targeting the problem of multilingual access in the KGQA field. Therefore, the authors propose to combine well-known KGQA systems with machine translation (MT) tools in order to see the impact of machine translation on question-answering quality. In addition, determine whether machine translation could be an alternative to multilingual solutions. In the actual solution, the authors combine QAnswer, DeepPavlov, and Platypus with Yandex Translate API[30] and Helsinki NLP [97]. The evaluation is done on the QALD-9-plus [98] benchmark over DBpedia and Wikidata. *The following languages are used in the evaluation: English, German, French, Russian, Ukrainian, Lithuanian, Belarusian, Bashkir, and Armenian*. The highest F1 score of 44.59% is achieved by QAnswer on the English language (without translation). Based on the evaluation results, the authors come to the conclusion that given the current state of the art, it is always better (in terms of QA quality) to translate any source language to English despite a system that may natively support this source language. The authors would see possible improvement directions: to extend the evaluation w.r.t. languages, the number of questions, KGQA systems, MT systems, and introduce named entity-aware MT solutions. It is worth mentioning that in this work, no detailed error type analysis was given, e.g., regarding question types and other features.

The *Lingua Franca approach* [99] has been aiming at improving the method by Perevalov et al. (see paragraph above) by introducing named entity-aware MT approach combined with mKGQA systems. Lingua Franca leverages symbolic information about named entities stored in Wikidata to preserve their correct translation to the target language. In particular, the developed solution has the following processing steps: (1) named entity recognition and linking for identifying the names entities in a question, and (2) MT with entity-replacement technique using the entity labels from Wikidata in a target language. The approach was evaluated on QAnswer and Qanary KGQA systems and QALD-9-plus dataset using German, French, and Russian questions. The majority of the experimental cases (19 out of 24) show that the KGQA systems that were using Lingua Franca outperformed the ones that used standard MT tools.

4.1.4. Summary

In Table 4 we summarize the reviewed systems ordered by publication date with their characteristics such as:

- *publication year* of a paper;
- *languages* that were used in the evaluation or supported by the described system;
- *knowledge graphs* that were used in the evaluation;
- *datasets* (or benchmarks) used in the evaluation;
- *metrics* used to measure the QA quality of a system;
- *technologies* used to implement the described system;
- *code/demo* availability;
- *methods* that were used according to the taxonomy described in Section 4.2.

In the next subsection, we present the taxonomy of the methods that are used to develop mKGQA systems.

4.2. A Taxonomy of the Methods to mKGQA

While answering $\mathcal{RQ}1$, we organize the information presented before and help to derive more general insights on the systems and the methods they utilize. Therefore, we created a taxonomy of the methods used for the development of mKGQA systems. It is worth mentioning that the following applies also to the monolingual KGQA systems. However, in this section, we intentionally highlight the multilingual capabilities of the methods.

[30] https://yandex.com/dev/translate/

Table 4

The overview on the multilingual KGQA systems published between 2011 and 2023.

System name	Year	Languages	Knowledge Graphs	Datasets	Metrics	Technologies	Code/Demo	Methods (Taxonomy)
QALL-ME [46]	2011	en, de, es, it	Custom KG	Custom data	Accuracy, RTE	Java	✗	M1.4
N. Aggarwal [47]	2012	en, de	DBpedia	QALD-2	Precision, Recall, F1 score	Stanford Parser, Java	✗	M1.1
QAKiS [51–53, 100]	2013	en, fr, de	DBpedia	QALD-2	Precision, Recall	Stanford Core NLP	✗	M1.4, M2.1
SWSNL [54]	2013	en, cs	Custom KG (accommodation domain)	Custom data, ConnectionsCZ, ATIS	Precision, Recall, F1 score	Prague Dependency Treebank, MaxEntNER, LINGVOParser, OntologyNER, WSim, Java	✗	M1.2, M1.3
UTQA [72]	2016	en, fa, es	DBpedia	QALD-5	Precision, Recall, F1 score	Google Translate	✗	M2.1
AMUSE [55]	2017	en, de, es	DBpedia	QALD-6	Macro F1 score	Universal Dependencies, Java	✓	M1.1, M1.3
WDAqua-core0 [57]	2017	en, fr, de, it	DBpedia, Wikidata	QALD-7	Precision, Recall, F1 Score	Qanary	✗	M1.4, M2.1
UDepLambda [63]	2017	en, de, es	Freebase	WebQuestions, GraphQuestions	F1 Score	Universal Dependencies, Java	✓	M1.3, M1.4
MuG-QA [65]	2018	en, de, it, fr	DBpedia	QALD-7	Precision, Recall, F1 Score	Grammatical Framework (GF), The GF Resource Grammar Library	✗	M1.1, M1.2
WDAqua-core1 [61]	2018	en, de, fr, it, es	DBpedia, Wikidata, MusicBrainz, DBLP	QALD-{3-7}, LC-QuAD 1.0	Precision, Recall, F1 score	Qanary	✗	M1.4, M2.1
LAMA [69, 101]	2018	fr, en	DBpedia	QALD-7, LC-QuAD 1.0	F1 score	Google Translate, SyntaxNet, Penn Treebank, OntoNotes, Universal Dependencies	✗	M1.1, M1.3, M2.1, M3.1
Platypus [74]	2018	fr, en	Wikidata	WikidataSimpleQuestions		Core NLP, Spacy, RasaNLU	✓	M1.2, M1.3, M2.1
QAnswer [62, 76]	2019	en, de, fr, it, es, pt, ar, zh	Wikidata, DBpedia, MusicBrainz, DBLP, Freebase	QALD-{3-7}, LC-QuAD 1.0	Precision, Recall, F1 score, Runtime	Java, HDT	✗	M1.4, M2.1
DeepPavlov [77]	2020	en, ru	Wikidata	LC-QuAD {1.0, 2.0}	Precision, Recall, F1 Score	Python	✓	M1.4, M2.2
Y. Zhou et al. [94]	2021	en, fa, de, ro, it, ru, fr, nl, es, hi, pt	DBpedia	LC-QuAD 1.0, QALD-9	ICA, F1 score	Google Translate	✗	M2.2, M3.2
A. Perevalov et al. [96]	2022	en, de, fr, ru, uk, lt, be, ba, hy	Wikidata, DBpedia	QALD-9-Plus	Precision, Recall, F1 Score	Python, Yandex Translate	✓	M3.1
Tiresias [79]	2022	en, gr	DBpedia	Custom data	Custom metric	Python, Transformers, RDFLib, Spark NLP	✓	M2.2, M3.1
DeepPavlov 2023 [80]	2023	en, ru	Wikidata	RuBQ 2.0	Precision, Recall, F1 Score	Python	✓	M1.4, M2.2
XSemPLR [83]	2023	en, zh, he, kn	Wikidata	MCWQ	Exact Match	Python, OpenAI	✓	M2.2
CLRN [92]	2023	en, zh, fr	Wikidata	MLPQ	Hits@k	Python, Transformers	✓	M2.2
Lingua Franca [99]	2023	de, fr, ru	Wikidata	QALD-9-plus	Macro F1 Score, BLEU, NIST	Python, Transformers	✓	M3.2

4.2.1. Overview

The taxonomy is based on our review and materials from the previous survey articles [35–37, 39]. Note that not all of the methods to be presented below are working in an end-to-end manner, meaning that not all of them directly produce an answer or a SPARQL query. Some of the methods require the use of other methods to form a complete mKGQA system.

In a nutshell, there are two system development paradigms. *The first class of the KGQA systems relies on a sequence of predefined task-oriented components*. This paradigm is named "Semantic Parsing" and is often referred to as "QA pipelines" [58]. In such systems, a question is processed in a multi-step setting respectively to the used components, for example, NER, Relation Prediction (REL), Query Builder (QB), and Query Executor (QE). The aim of such systems is to convert a NL question to a SPARQL query. *The second paradigm is named as "end-to-end KGQA" and aims at answering a question in a single step*. These systems are mainly based on neural network-related approaches [102] e.g., ranking of an answer candidate given a question, or translation of a question to a query (end-to-end semantic parsing) [103]. Both of the aforementioned paradigms may utilize one or more methods from the taxonomy defined below.

We organize the methods as follows, the high-level general groups (denoted as "G") contain the low-level concrete methods (denoted as "M"):

G1 — *methods based on rules and templates*:

- M1.1 — *syntax tree parsing* is used to convert NL to a machine-readable syntax tree;
- M1.2 — *grammar rules* are used to extract structured information from NL with manually defined rules;
- M1.3 — *logical representations* are used as a machine-readable intermediate form to represent the semantics of a given NL text;
- M1.4 — *dictionaries, indexes, and templates* are used for generating queries or matching entities and relations;

G2 — *statistical methods*:

- M2.1 — *classical machine learning and statistical methods* are used for the downstream tasks of KGQA (e.g., NER, REL, etc.);
- M2.2 — *deep learning methods* are mainly used in the context of language modeling, graph embedding models, and encoder-decoder architectures;

G3 — *machine translation methods*:

- M3.1 — *end-to-end machine translation methods* are used for direct translation of a source language to the target one that is supported by the system;
- M3.2 — *enhanced machine translation methods* are used for machine translation with intermediate improvements (e.g., KG enriched [104]) of a source language to the target one that is supported by the system.

The taxonomy is demonstrated in Figure 1. It is worth mentioning that the methods are not mutually exclusive within one system. Thus, multiple of them can be used within one system, for example, M1.4 and M2.2 are used within the QAnswer system. The methods used by the corresponding systems are listed in Table 4.

4.2.2. Overview of the Methods from the Taxonomy

Let us review the methods while specifically focusing on their multilingual capability. It is worth underlining that the methods of groups G1 and G2 are widely used in monolingual KGQA, i.e., are not multilingual by definition, however, they may provide this functionality to some extent. For example, the method M2.2 (deep learning) covers multilingual language models as well as monolingual ones. The methods of group G3 are mostly used in the multilingual context. Technically, the methods from the G3 could be placed under the groups G1 or G2, however, in the context of mKGQA, G3 represents a completely different approach on the ideological level.

Most of the *syntax tree parsing (M1.1)* implementations are grammar-based (e.g., Stanford NLP Parser [105] or BLLIP [106]). Hence, those are closely dependent on the language-specific grammar rules, which are related to method M1.2. However, it is possible to extend syntax tree parsing methods to multiple languages while implementing it with multilingual language models or the ones trained separately on different languages (method M2.2), as demonstrated in the following publications [107–109]. One of the major state-of-the-art multilingual syntactic

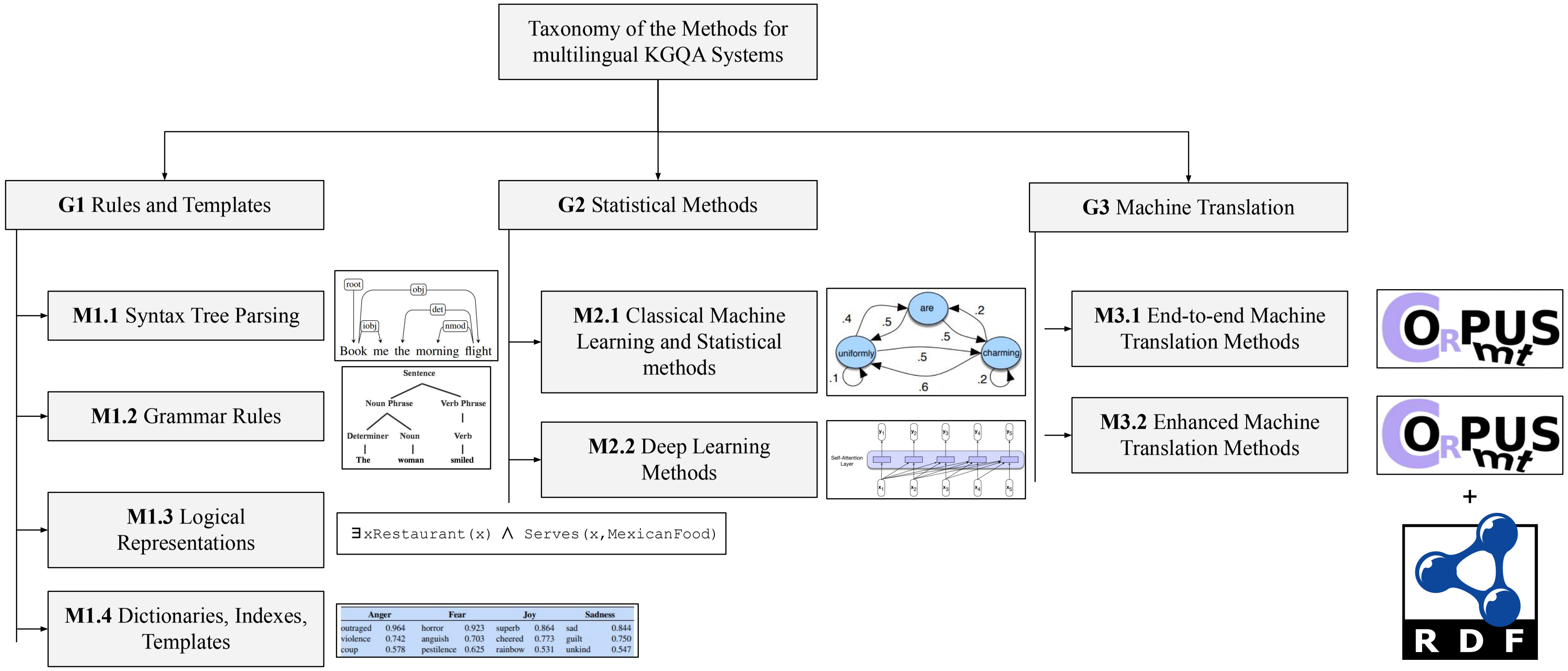

Taxonomy of the Methods for multilingual KGQA Systems
G1 Rules and Templates
G2 Statistical Methods
G3 Machine Translation
M1.1 Syntax Tree Parsing
M1.2 Grammar Rules
M1.3 Logical Representations
M1.4 Dictionaries, Indexes, Templates
M2.1 Classical Machine Learning and Statistical methods
M2.2 Deep Learning Methods
M3.1 End-to-end Machine Translation Methods
M3.2 Enhanced Machine Translation Methods
root
obj
det
iobj
nmod
Book me the morning flight
Sentence
Noun Phrase
Verb Phrase
Determiner
Noun
Verb
The
woman
smiled
∃xRestaurant(x) ∧ Serves(x,MexicanFood)
Anger
Fear
Joy
Sadness
outraged 0.964 horror 0.923 superb 0.864 sad 0.844
violence 0.742 anguish 0.703 cheered 0.773 guilt 0.750
coup 0.578 pestilence 0.625 rainbow 0.531 unkind 0.547
are
uniformly
charming
.4
.5
.2
.5
.5
.1
.6
.2
Self-Attention Layer
OPUS mt
+
R D F

Fig. 1. The taxonomy of the methods used for the development of multilingual KGQA systems. The method example pictures are taken from [111–113]. The surveyed systems are classified according to this taxonomy in Table 4.

parsers Stanza [110] is based on Universal Dependencies [64], which is a large tree bank for many languages. The method M1.1 can be used for building machine-readable representations of a NL question, NER, and REL.

The method based on *grammar rules (M1.2)*, as described above, is language-dependent by definition. In most cases, context-free grammars (CFG) are commonly used in NLP due to their efficient implementation [114]. One needs to define a set of language-dependent rules to extract particular structures from a NL text to implement this method. The Grammatical Framework (GF) [66] provides a syntax for creating pseudo-multilingual grammars (one still needs to define general rules for multiple languages, although it appears to be more convenient). Such tools as POTATO [115] and Yargy parser [116] are providing grammar-based functionality. The method M1.2 can be used for NER and REL, in addition, the structural elements of a NL text, such as subject-predicate-object structure, can be extracted. The extracted information leads to the building of machine-readable representations of a text, which are related to the following method M1.3.

The methods using *logical representations (M1.3)* are aimed at creating machine-readable meaning representations of NL with the means of description logics (DLs). For example, a question "In what city was Angela Merkel born?" is represented in a logical form as "$\exists x\, isBirthPlaceOf(x, Angela\_Merkel) \sqcap City(x)$". Such representations are human and machine-readable, unambiguous, and language-agnostic. However, the transformation between NL and logical representations is a non-trivial process: earlier it was solved with the rule-based methods [117, 118] (language-dependent), while in the current research, mostly statistical [119, 120] and neural network-based methods are used [121, 122] (can be multilingual). The method M1.3 is used for building meaningful representations of a question and SPARQL query building.

The methods based on *dictionaries, indexes, and templates (M1.4)* are mainly focused on the lookup tasks and the query generation. One of the examples could be a named entity linking (NEL) task while looking up the label-URI dictionary of a KG. The dictionaries can be exploited for mapping between language-specific terms. The templates can be used for the SPARQL query generation process via fulfilling the slots with the extracted information on the previous steps, e.g., the DeepPavlov system uses several query templates that correspond to the different query types. Hence, method M1.4 is language-dependent by default but can be extended to serve multilingual functionality, e.g., by introducing multilingual dictionaries that link all the language-specific labels of one entity together. The method M1.4 is used for such tasks as NEL, term translation, and QB.

The methods based on *classical machine learning and statistical methods (M2.1)* are solving a variety of classification and sequence labeling tasks. One can utilize logistic regression in combination with TF-IDF for detecting the expected answer type (classification task) [123]. Another example can be the NER task using Hidden Markov Models (HMMs) [124]. Conventionally, these methods are known as lightweight, transparent, and explainable in comparison to deep learning (method M2.2). Nevertheless, their multilingual functionality is limited. For example, while considering TF-IDF as a method for text-to-vector transformations, the usage of this method in multiple languages leads to extremely sparse feature sets as the vocabulary increases with each language. Consequently, one needs to develop different language-specific models in order to process multiple languages. The method M2.1 can be used for NER, answer type detection, POS tagging, intent detection, REL, and other similar tasks.

The methods based on *deep learning M2.2* are applied to the wide range of KGQA tasks. The two main applications of this method class are graph embeddings and language modeling. The graph embedding direction is used in the KGQA field to search and extract the sub-graphs, relations, or entities given a textual question [125]. The language modeling direction is aimed at solving the KGQA downstream tasks with better quality and generalizability than the classical machine learning methods [126]. LLMs are well-suited for working in the multilingual setting (e.g., multilingual BERT supports 104 languages [127]), however, they require fine-tuning for the downstream tasks. The paper [83] demonstrated that LLMs are able to work in a zero- and few-shot setting with multiple languages for SPARQL query generation. Hence, the method class M2.2 is suitable for all the downstream KGQA tasks (e.g., NER, REL), as well as for the end-to-end KGQA (i.e., producing a SPARQL query directly or an answer). Nevertheless, one needs to take into account the resource consumption factor of the deep learning-based methods, especially seeing the latest LLMs such as PaLM [128], Chinchilla [129], LLaMa [130], and others.

The *end-to-end machine translation M3.1* methods are utilized for translating source languages to the target ones that are supported by a particular KGQA system. In this case, the machine translation tool is treated as a "black box", hence, a developer does not influence its working process. The majority of the neural machine translation models (e.g., [97]) provide the corresponding functionality for one language pair per model, i.e., multiple models required for translating different language pairs. However, the state-of-the-art large generative models [131] provide the ability to handle multiple languages for the translation tasks, despite the majority of the training data being in English[31]. This method is simple to integrate into an existing KGQA system, however, it does not ensure the precise translation of named entities.

The *methods related to the enhanced machine translation M3.2* are serving the same functionality as the M3.1, however, in this context, more background knowledge is used in the process. These machine translation methods can take into account the KG embeddings of the named entities (e.g., KG-NMT approach [104]), tag the named entities in an input text before translating it [132], or use bilingual lexicon induction for training data augmentation (e.g., [94]). Therefore, the translation process is improved based on the used background knowledge. For instance, the additional information regarding named entities ensures that they are not corrupted during the translation process.

4.3. Quality Measurement for mKGQA Systems

To completely cover $\mathcal{RQ}1$, we elaborate the procedure for a comprehensive quality measurement of the mKGQA systems.

Multilinguality denotes the usage of several different languages [31]. Formally, we consider a KGQA system as multilingual if it supports more than one language by default (i.e., without additional efforts to re-train, clone, or fine-tune it). In a more general definition, the languages handled by a multilingual system must belong to different language groups (e.g., Finnish vs Russian), alphabets (e.g., Bulgarian vs French), or writing systems (e.g., German vs Arabic). It is worth underlining that *the quality deviation among the different languages handled by a multilingual method should tend to zero, while demonstrating the absolute results comparable with the monolingual state-of-the-art method*. Let us formally state when a KGQA system handles multilingual questions well using the following definitions:

- L is a set of languages: $\{l_0, \dots, l_n\}$;
- $Q_l : A \times \Theta \times L \to \mathbb{R}$ is a quality function that maps a method $a \in A$ parametrized by $\theta \in \Theta$ and an input language $l \in L$ to a real-valued quality measure (e.g., F1 score value);
- μ_L is a function that maps a set of quality results from Q_l to its average value.
- σ_L is a function that maps a set of quality results from Q_l to its standard deviation value.

Therefore, a system $\mathcal{S}$ parametrized by θ with the highest multilingual capabilities is denoted as having the highest average quality across the languages:

$$\max_{\mathcal{S}} \mu_L(Q_l(\mathcal{S}, \theta, l_0), \dots, Q_l(\mathcal{S}, \theta, l_n)) \tag{1}$$

and the lowest standard deviation:

$$\min_{\mathcal{S}} \sigma_L(Q_l(\mathcal{S}, \theta, l_0), \dots, Q_l(\mathcal{S}, \theta, l_n)) \tag{2}$$

Obviously, such a definition represents a *multi-objective optimization problem* that results in a set of solutions produced by different methods, the most optimal method regarding the multilinguality can be found by building a *Pareto front* (see Figure 2 for the visual representation). Each system from the figure is associated with a *quality quadrant* according to its μ and σ values. We name the quality quadrants as follows: Q_1 – inconsistent high quality, Q_2 – consistent high quality, Q_3 – consistent low quality, and Q_4 – inconsistent low quality. Naturally, the best-performing systems are located in the Q_2 area.

We encourage researchers to use this procedure and our findings for comprehensive quality measurement of mKGQA systems.

[31] https://github.com/openai/gpt-3/blob/master/dataset_statistics/languages_by_document_count.csv

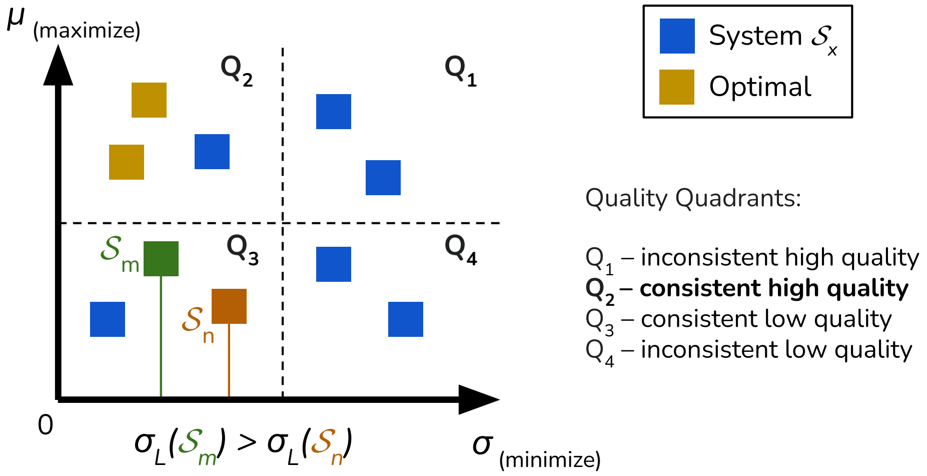

μ (maximize)
Q2
Q1
$\mathcal{S}_m$
Q3
Q4
$\mathcal{S}_n$
0
$\sigma_L(\mathcal{S}_m) > \sigma_L(\mathcal{S}_n)$
σ (minimize)
System $\mathcal{S}_x$
Optimal
Quality Quadrants:
Q_1 – inconsistent high quality
Q_2 – consistent high quality
Q_3 – consistent low quality
Q_4 – inconsistent low quality

Fig. 2. Visual representation of the multi-objective quality function for mKGQA systems, the gold-colored results represent the Pareto front (optimal solution). The systems are associated with the quality quadrants that help to easily interpret the values.

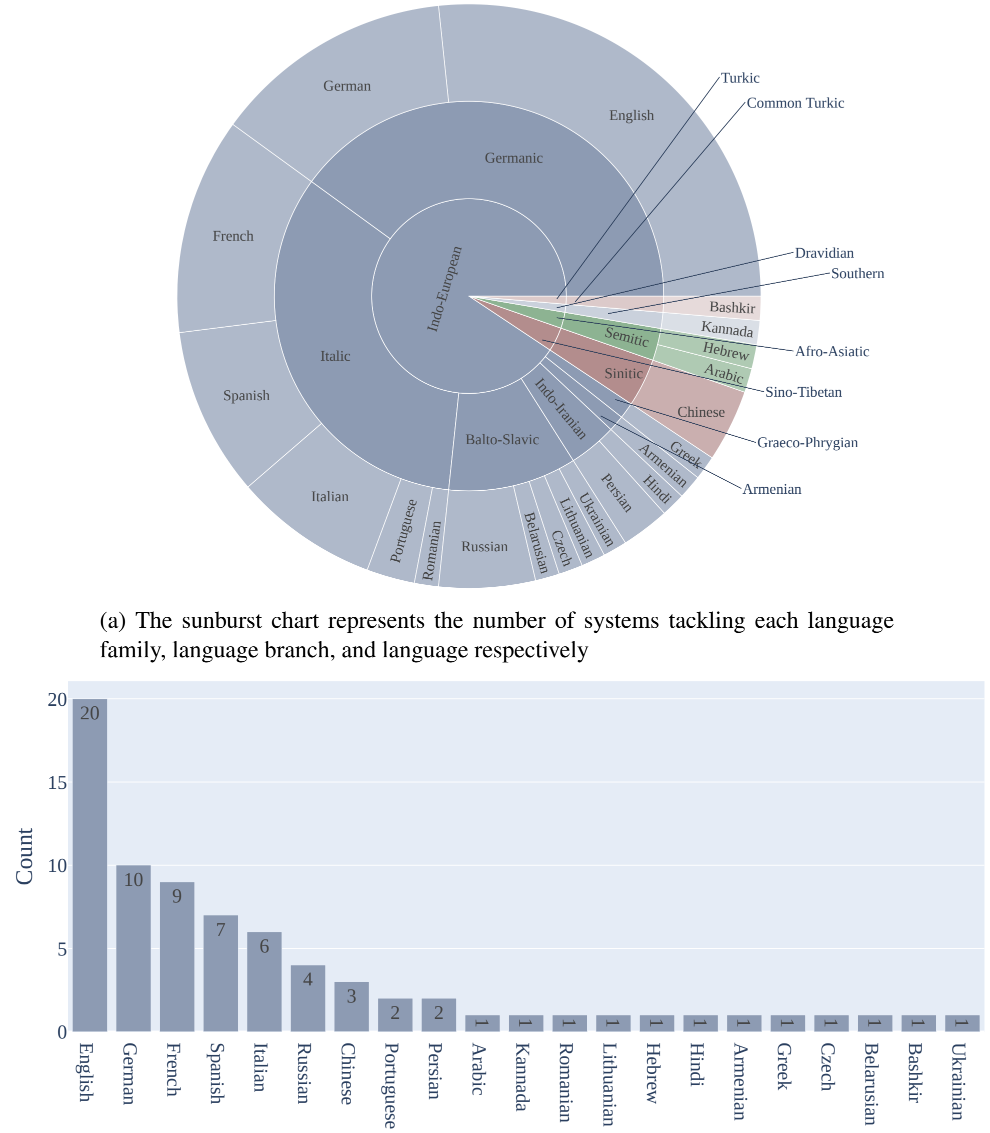
4.4. *Evaluation Results of the Reviewed Systems*

While working on $\mathcal{RQ}2$, we collected and structured the evaluation results of the reviewed systems on different benchmarks. Only the latest results for the corresponding system benchmark combinations are included. Despite the majority of the papers reporting the evaluation values using common metrics such as Precision, Recall, and F1 score, some of the authors also explicitly specify averaging strategy (e.g., micro, macro) or use other metrics (e.g., Accuracy, Hits@k, or Exact match for SPARQL queries). For the purpose of data consistency, we explicitly mention the used metrics (see the table from online appendix[32]). The reported performance values from our review in Section 4.1, demonstrate that such approaches as KGQA pipelines based on multilingual LMs (Group 2) and MT (Group 3) are outperforming the other ones (Group 1) within the context of multilinguality. While analyzing the values, it is clear that the English language significantly outperforms other ones regarding QA quality. We contributed the full set of values to the KGQA leaderboard [133] which is available online[33].

4.5. *Language Coverage by the mKGQA Systems*

We analyzed the data on reviewed mKGQA systems (see Table 4) regarding the languages and language families that are covered by them to answer $\mathcal{RQ}2$ completely. The visual representation of the analysis is presented in Figure 3. In particular, Figure 3a shows that the Indo-European language family tremendously dominates among the others (Turkic, Sino-Tibetan, Afro-Asiatic) in the field of mKGQA. Among the reviewed systems, our estimation for the share of the mKGQA systems targeting one or more languages from the Indo-European language family is 95%. Figure 3b demonstrates the data on how many systems support input in a particular language. The English language is the most supported one as all the reviewed systems (21) process the questions written in it. The second most supported language is German: 10 out of 21 support input in writing it. The majority of the listed languages are supported by only one system, those are, namely, Arabic, Romanian, etc. It is worth mentioning that there are monolingual KGQA systems supporting a language other than English, which are, however, not covered in our work since it focuses on multilinguality.

[32] https://github.com/Perevalov/multilingual-KGQA-survey/blob/main/data/review_tables/evaluation-unified-metric-column.tsv
[33] https://kgqa.github.io/leaderboard/



(a) The sunburst chart represents the number of systems tackling each language family, language branch, and language respectively

(b) The bar chart represents the number of systems supporting a concrete language

Fig. 3. The visual representation of language and language family coverage among the mKGQA systems

4.6. Summary

To the best of our knowledge, during the past decade, there were only 21 mKGQA systems developed within the research context. We observed that in most of the cases, namely 95%, the mKGQA systems *target Indo-European language family*. Therefore, *the mKGQA systems mostly work within one writing system*, namely Alphabetic (while using Latin, Cyrillic, Armenian, or Greek script). Hence, the actual generalizability and scalability of the used methods across diverse languages are unclear. In addition, our survey demonstrated that *all systems except QAnswer target general-domain KGs only*. We recalled that *the researches are not using the standard metrics or evaluation tools that ensure the comparative evaluation results*. Finally, the analysis showed that a *significant share of the systems, namely 11 out of 21, is not accessible due to the outdated demo websites and source code or their absence*. Therefore, the experimental results are not reproducible.

We highlighted *three main groups of the methods for the mKGQA*: rules and templates (G1), statistical methods (G2), and machine translation (G3). The analysis of the taxonomy, which was created in this work, demonstrated that *the researchers prefer to reuse monolingual methods that are adapted to the other languages rather than working towards the language-agnostic ones*. The analysis of the KGQA systems showed that *the assignment of a system to only one method group is not possible, as most of them combine multiple methods*.

Based on our observations, we foresee the following research challenges and research directions for the mKGQA. Developing *methods and systems that work with diverse languages*, in particular the ones that:

- originate from different language groups (e.g., Finno-Ugric and Balto-Slavic);
- have different alphabets/scripts (e.g., Cyrillic and Latin);
- use different writing systems (e.g., Alphabetical and Abjad).

Addressing *how well zero- or few-shot transfer methods work* w.r.t. training and evaluation on the diverse languages (see the list above). Providing *case studies on domain-specific applications of mKGQA systems* (e.g., Material Science, Chemistry, Business, Government, Law etc.); Searching for the *advanced mKGQA evaluation metrics*, namely:

- How well a system performs w.r.t. different languages (e.g., does it have a high-quality variance among different languages)?
- What is a criterion of high-quality w.r.t. mKGQA?

Designing and developing *general domain large-scale multilingual benchmarking datasets* for trustworthy evaluation of mKGQA systems by

- incorporate diverse languages (see the above paragraph about diverse languages)
- gold-standard answers on multiple KGs for wider applicability (e.g., at least Wikidata, DBpedia).

Exploring *capabilities of LLMs for mKGQA* by:

- Using Retrieval-augmented Generation (RAG): providing LLMs with relevant triples from KGs by verbalizing the triples and using them as a part of a prompt.
- Fine-tuning an LLM on verbalized triples from a KG for learning the facts.

5. Benchmarks for Multilingual Question Answering over Knowledge Graphs

This section describes the existing benchmarks that have been developed for mKGQA. The overview of the benchmarks will showcase their unique characteristics and contributions to the field, shedding light on the respective progress and challenges of mKGQA.

5.1. Overview

The research in the field of KGQA is strongly dependent on data, nevertheless, the particular *challenge is the lack of the benchmarks for trustworthy evaluation of the KGQA systems in multiple languages* [134, 135]. In the field of OpenQA, several works related to machine translation of existing benchmarking datasets were done (e.g., [134, 136]). However, this is not the case for KGQA. To the best of our knowledge, *only five benchmarks (or benchmark series*[34]*) exist that tackle multiple languages*: Question Answering over Linked Data (QALD) [41, 98, 137], EventQA [138]—event-centric questions over knowledge graphs, the RuBQ dataset for Question Answering over Wikidata [81, 139], Multilingual Compositional Wikidata Questions (MCWQ)[84], Mintaka [140], and MLPQ (A Dataset for Path Question Answering over Multilingual Knowledge Graphs) [93]. The overview of the benchmarks is demonstrated in Table 5.

[34] Some of the benchmarks (e.g., RuBQ 1-2, or QALD 1-10) have multiple versions. Therefore, we refer to them in general as *benchmark series*.

The QALD is a well-established benchmark series for mKGQA. It has several multilingual versions, namely QALD-{3,6,7,8,9,9-plus,10}. The QALD-3 includes 199 questions and ground truth SPARQL queries over DBpedia and MusicBrainz[35]. The QALD-6 contains 450 questions and queries over DBpedia. It follows the QALD JSON format where for each question the following is given: a textual representation in multiple languages, the corresponding SPARQL query, the answer entity URI, and the answer type. The QALD-7 contains 258 questions with queries over DBpedia and Wikidata. It follows the QALD JSON format. The QALD-8 includes 260 questions with queries over DBpedia and follows the QALD JSON format. The QALD-9 contains 558 questions and queries over DBpedia. The QALD-9-plus dataset [98] has improved and extended translations to eight languages, and also covers the Wikidata KG[36]. The translations and their validation were done using the crowd-sourcing approach, where the participating crowd-workers were native speakers of the respective languages. The rewordQALD9 [141] has improved the translations in the Italian language. The translations were done by Italian native speakers proficient in English. The authors of QALD-9-ES [142] have discovered significant flaws in the Spanish questions of the QALD-9. Therefore, they created new improved translations in Spanish with the help of native speakers. The new Spanish translations were integrated into the QALD-9-plus benchmarking dataset. The task for the translators also involved reformulation of the English questions, which results in multiple paraphrases of the English questions. The newest version—QALD-10 [143] introduces 402 new questions in English, Chinese, German, and Russian. The benchmark follows the QALD JSON structure. The benchmark series has become a benchmark for many packageresearch studies in KGQA (e.g., [59, 144–146]).

EventQA is the benchmark for answering event-centric questions (e.g., "In which tournament, known as major, did Jason Dufner win?"). The benchmark contains 1000 questions in the corresponding languages: English, German, and Portuguese. The SPARQL queries are targeting the EventKG [147]—an event-centric KG that incorporates 690,247 events. The benchmark is represented using a newly developed JSON structure.

RuBQ KGQA benchmarking series includes two versions. The latest one—RuBQ 2.0—contains 2910 questions and is based on its previous version RuBQ 1.0. Similarly to the latest QALD versions, the SPARQL queries within the RuBQ are written for Wikidata. The creation of this benchmark was done in a semi-automatic way: the automatically collected question-answer pairs in textual format were annotated using an entity linking tool; thereafter, the linked entities were checked by crowd-workers; finally, based on the linked question and answer entities, the SPARQL queries were generated and manually validated. The questions are represented in the native Russian language and machine-translated English language. Additionally, it contains a list of entities, relations, answer entities, SPARQL queries, answer-bearing paragraphs, and query-type tags. The benchmark uses a newly developed JSON structure.

MCWQ is a KGQA benchmarking dataset over Wikidata (similarly to QALD and RuBQ) that is based on the previously created CFQ dataset [148]. MCWQ contains questions in the Hebrew, Kannada, Chinese, and English languages. All the non-English questions were obtained using machine translation with several rule-based adjustments. It has a well-detailed structure including questions with highlighted entities, original SPARQL query over Freebase [14] (from CFQ), SPARQL query over Wikidata (introduced in MCWQ), textual representations of a question in four aforementioned languages, and additional fields. The benchmark newly developed JSON structure.

Mintaka is a recent KGQA benchmark that provides 20,000 questions in the following languages: English, Arabic, French, German, Hindi, Italian, Japanese, Portuguese, and Spanish. Similarly to QALD, RuBQ, and MCWQ, Mintaka's SPARQL queries are over Wikidata. The structure of Mintaka includes annotated named entities, gold standard answers, and internal fields such as question category (e.g., geography) and complexity (e.g., ordinal). The benchmark does not contain the gold standard SPARQL queries, instead, the Wikidata entity IDs are provided as an answer and the corresponding language-specific labels. The questions, annotated named entities, and gold standard answers were created and annotated by crowd-workers respectively. Mintaka follows a newly created JSON structure.

MLPQ is a large-scale KGQA benchmark that contains 300,000 questions in English, Chinese, and French. MLPQ SPARQL is over DBpedia similarly to the early QALD versions. MLPQ has its own RDF structure that

[35] https://musicbrainz.org/

[36] https://www.wikidata.org/

contains a question's text with a language tag, ground truth answer URIs, and a SPARQL query. As the benchmark was generated automatically, the authors provided query templates that were used for the generation process. In addition, MLPQ is accompanied by the DBpedia triples covering all the used questions.

Table 5

Overview on the mKGQA benchmarks

Name	Domain	# questions	Format	Languages	KGs	Comment
QALD-3 [137]	General	199	RDF XML & Turtle	en, de, es, it, fr, nl	DBpedia, Musicbrainz	Translations quality is poor
QALD-6 [149]	General	450	QALD JSON	en, fa, de, es, it, fr, nl, ro	DBpedia	
QALD-7 [60]	General	258	QALD JSON	en, pt, de, es, it, fr, nl, hi	DBpedia, Wikidata	
QALD-8 [150]	General	260	QALD JSON	en, de, es, it, fr, nl, hi, ro	DBpedia	
QALD-9 [41]	General	558	QALD JSON	en, de, ru, pt, hi, fa, it, fr, ro, es, nl	DBpedia	
QALD-9-plus [98]	General	558 (507)	QALD JSON	en, de, fr, ru, uk, lt, be, ba, hy, es*	DBpedia, Wikidata	Not all questions are covered by Wikidata
rewordQALD9 [141]	General	558	QALD JSON	en, it	DBpedia	English questions were also paraphrased
QALD-10 [143]	General	909	QALD JSON	en, de, zh, ru	Wikidata	Test set has only four languages
EventQA [138]	Events	1,000	EventQA	en, de, pt	EventKG	Lack of event-centric KGQA systems
RuBQ 1.0 [139]	General	1,500	RuBQ	en, ru	Wikidata	Machine-translated questions
RuBQ 2.0 [81]	General	2,910	RuBQ	en, ru	Wikidata	
MCWQ [84]	General	124,187	MCWQ	en, he, kn, zh	Wikidata	Rule-based generation, translations obtained with MT
Mintaka [140]	General	20,000	Mintaka	en, ar, de, ja, hi, pt, es, it, fr	Wikidata	Named entities are annotated in the English questions
MLPQ [93]	General	300,000	MLPQ	en, zh, fr	DBpedia	Uses multilingual DBpedia with inter-language links

5.2. Summary

Given the number and characteristics of the reviewed benchmarks, it is clear that there is little attention paid to the mKGQA research field. To the best of our knowledge, only 14 benchmarks tackle the multilingual aspect (some of them have different versions, which are the subsets of the newest ones). The total number of benchmarks (mono- and multilingual) according to the KGQA leaderboard is 48 [133]. Hence, the *share of multilingual benchmarks is 29.1%*. We observed that the crowd-sourcing approach for benchmark creation is gaining traction. The crowd-sourcing tasks vary from verification linked entities [139], translation and validation questions [98] to creation and annotation of questions from scratch [140]. While answering $\mathcal{RQ}3$, we identified that there is a chronological trend toward an increase in the size of the benchmarks, however, the number of languages still varies (see Figure 4).

We also discovered some flaws in the existing benchmarks:

- The maximum order of magnitude for the number of questions among the manually and semi-automatically created benchmarks is 10^4 (e.g., QALD, RuBQ);
- There is no standardized format used across the research groups for the KGQA benchmarks (except QALD JSON);
- The majority of the benchmarks stick to only one KG, namely Wikidata (e.g., RuBQ, Mintaka);
- The benchmarks created with machine translation tools or automatically have unclear questions' quality (RuBQ, MCWQ, MLPQ).

The aforementioned flaws may become the objectives for future research in this field. Especially, it is worth developing a standardized format for the benchmarks, enlarging them regarding the number of questions, languages, and knowledge graphs using the crowd-sourcing setting.

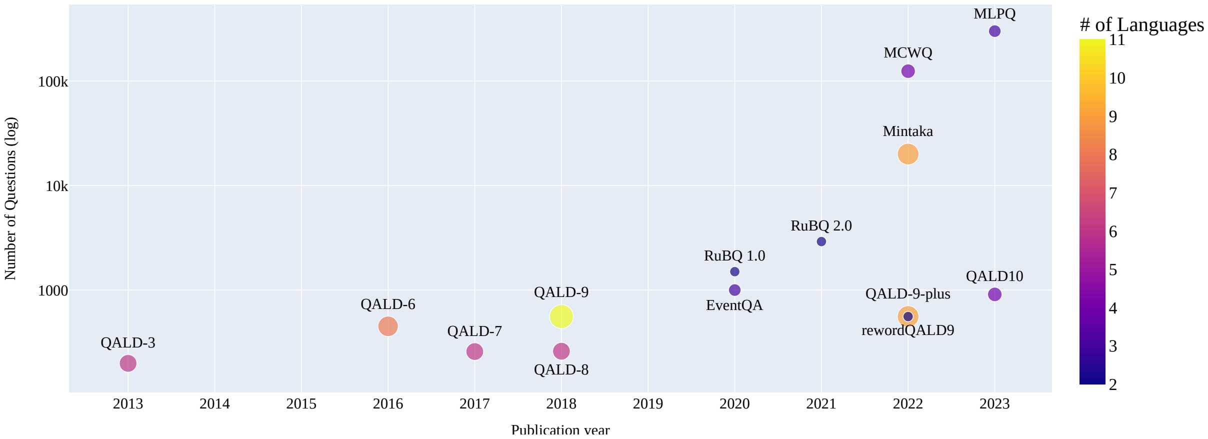


Fig. 4. The bubble chart represents the chronological order of the benchmarking datasets, their number of questions, and languages.

6. Discussion

This section discusses the analyzed work on mKGQA. The first subsection focuses on the challenges encountered in this field, highlighting the obstacles and limitations that researchers face when developing and evaluating KGQA systems in multiple languages. The second subsection explores potential future research directions in mKGQA, highlighting areas that require further exploration and innovation. By examining both the challenges and future research directions, this section aims to provide valuable insights and guide future advancements in mKGQA.

6.1. Challenges

While reviewing the related survey papers, mKGQA systems, and corresponding datasets, we identified several challenges that currently exist in this research field. In the following paragraphs, we discuss the most remarkable challenges in the mKGQA field.

6.1.1. Noisy Human Natural Language Input

One of the challenges in mKGQA is *effectively handling noisy human natural language input*. This challenge is amplified when dealing with multiple languages because different languages have diverse structures, grammatical rules, and vocabulary [151]. Moreover, questions can range from well-formed NL, where the syntax and grammar align with the language's rules, to keyword questions, which consist of a few crucial terms without a proper sentence structure.

Grammatical and orthographic errors further contribute to the noisy input in mKGQA. Since users may not be fluent in all the languages they interact with, they are prone to making mistakes in constructing sentences, selecting appropriate words, or adhering to correct grammar rules. *Grammatical and orthographic errors add complexity to understanding and interpreting the user's intent accurately*. The mKGQA systems that internally use methods from the group G1 “Rules and Templates” (e.g., SWSNL [54], AMUSE [55], UDepLambda [63]) are especially sensitive to this kind of noise.

Another aspect of noisy input in mKGQA is the wrong spelling of named entities. Named entities are essential components for understanding the context and semantics of a question [134]. However, users may misspell or incorrectly transliterate named entities from one language to another. For example, a user asking a question about the “Eiffel Tower” in French may mistakenly spell it as “Eiffle Tower” in English. This presents a challenge in *effectively mapping and resolving named entities in different languages*. This challenge was first mentioned by Perevalov et al. [96] and addressed within the Lingua Franca system [99] by Srivastava et al.

6.1.2. A User Question and a Knowledge Graph are Expressed in Different Languages

One of the critical challenges highlighted in the existing literature is the specific issue of handling mKGQA process, particularly *when a user question and a KG are expressed in different languages* [92]. This is commonly referred to as cross-linguality.

To address the cross-linguality challenge, the conventional approach involves *enriching KGs with multilingual labels*. By incorporating multilingual labels, KGs become capable of accommodating and understanding different languages. This approach aims to provide a more seamless and efficient user experience, regardless of the language in which the question is asked.

Another approach for dealing with cross-linguality is centered around *leveraging text and graph vector representations* [152]. Often, the availability of multilingual training data is limited, which makes it challenging to directly train models for mKGQA. In such cases, text and graph vector representations offer a way to bridge this gap. These representations capture semantic similarities and relationships between words or entities across different languages, allowing effective knowledge retrieval and alignment even in the absence of extensive multilingual training data. The context of cross-linguality was tackled within the LAMA system [69] explicitly on language-specific DBpedia KGs. However, current versions of KGs including DBpedia are not divided by languages.

6.1.3. The Use of Cultural Traits and Country-Specific Aspects

In addition to the previously mentioned challenges, another significant aspect in mKGQA is the *consideration of cultural traits and country-specific aspects when searching for information*. This challenge arises due to the diverse preferences, interests, and expectations of users across different cultures and countries.

When users search for information or ask questions, their *expectations may vary based on their cultural background and country of origin*. For instance, if a user from the United States asks for recommendations for popular TV series, their preferences and expectations might differ from those of a user from Japan or France. The concept of popularity and the perception of what constitutes an engaging TV series can vary significantly across cultures.

Addressing this challenge requires a nuanced understanding of cultural differences and the ability to provide contextually relevant answers to users from different backgrounds. The approaches to deal with this challenge should *involve real-user feedback on ranking different entities* pertaining to a common search topic (e.g., TV series). This may include setting up crowd-sourcing tasks, gathering its results, and building a dataset with the corresponding ranks. To our best knowledge, this challenge was not tackled by any work in the mKGQA field.

6.1.4. Quality Discrepancies Among Different Languages

One significant challenge evident from evaluation values in the field of mKGQA is that the *quality of QA is notably lower in languages other than English*. In particular, the evaluation results from [96] for the QAnswer, DeepPavlov, and Platypus systems demonstrate high deviation among the QA quality results in different languages. This observation underlines the need for the research community to develop effective solutions aimed at enhancing the performance of multilingual question answering systems and bridging this gap in quality between languages.

The existing disparities in QA quality arise due to various factors. Firstly, the availability of high-quality training data and resources for languages other than English may be limited, resulting in inadequate training for multilingual models. Moreover, the linguistic complexities, divergent language structures, and semantic nuances present in different languages further contribute to the lower performance in non-English languages.

To address this challenge, substantial efforts are to be made to develop techniques that align multilingual KGs, curate larger and more diverse benchmarking datasets, and improve the training methodologies for multilingual QA models. In particular, a promising strategy is to leverage zero- or few-shot inference while working with LLMs. These approaches aim to reduce biases toward English and provide a more equitable distribution of performance across different languages.

6.1.5. The Lack of Language Diversity

The landscape of mKGQA reveals certain *limitations when it comes to targeting languages that belong to different language groups, employ distinct alphabets or writing systems*. The existing systems predominantly focus on widely spoken languages (e.g., QAKiS [51], MuG-QA, [65]), which may overlook the linguistic diversity represented by low-resource and endangered languages. The systems from our review evaluated on the most diverse sets of languages are QAnswer [62], Zhou et al. [94], and Perevalov et al. [96].

In practice, the availability of mKGQA systems catering to low-resource and endangered languages is severely limited, with very few systems specifically designed for these language varieties (e.g., Bashkir language in [96]). This dearth of attention towards such languages is concerning, as it hinders the inclusivity and accessibility of KG-based information retrieval for diverse linguistic communities.

A common approach adopted in the development of mKGQA is the *usage of machine translation approaches or the adaptation of monolingual methods* to accommodate multiple languages (cf. [96, 99], thereby pursuing a multilingual capability. However, this adaptation of methods primarily focuses on language pairs with substantial linguistic resources, neglecting the unique challenges associated with low-resource languages.

6.1.6. Size and Translation Quality of Benchmarking Datasets

It is important to acknowledge that the existing *benchmarks for mKGQA are relatively small in scale*. In terms of magnitude, the non-automatically generated benchmarks typically consist of around 10^3 instances (e.g., QALD-9-plus [98]).

The limited size of these benchmarks poses challenges in accurately evaluating and benchmarking the performance of such systems. The smaller scale restricts the diversity of queries and contexts that can be covered. This, in turn, limits the generalizability of the performance results and may hinder the development of robust mKGQA systems.

Additionally, the quality of translations within these benchmarks is often a concern. In some cases, the questions are machine-translated (e.g., RuBQ 2.0 [81]), leading to questionable translation quality. Poor translation quality (e.g., QALD-9 [41]) introduces noise and inaccuracies into the benchmark data, potentially affecting the reliability of evaluations and comparisons between different multilingual question answering systems.

6.1.7. Effective Usage of Large Language Models

In recent times, there has been a surge of attention and interest in leveraging LLMs for mKGQA as well as other NLP tasks. However, despite the initial excitement and optimism surrounding these powerful models, *reports from the research community suggest that the obtained results do not always align with the high expectations set forth.*

In particular, cross-lingual semantic parsing has faced challenges in achieving the desired performance with LLMs. Zhang et al. [83] demonstrated that the results obtained for cross-lingual semantic parsing did not meet the anticipated levels of accuracy and quality. Similarly, even in monolingual settings, Klager et al. [9] reported underwhelming outcomes with LLMs for semantic parsing tasks.

One promising approach, which addresses these shortcomings, involves integrating structured data directly into the mKGQA process, building upon the practices established in the domain of monolingual KGQA [153, 154].

6.1.8. General Challenges

The field of mKGQA encounters several challenges that require attention and resolution. In contrast to the aforementioned challenges, these extend beyond the multilingual aspect and also encompass the broader field of KGQA:

1. The majority of KGQA systems predominantly target a single domain, often limited to the general domain. This domain-specificity restricts the applicability and scope of these systems, hindering their potential to address diverse domains effectively.
2. The lack of advanced metrics for evaluating the performance of KGQA systems poses a significant challenge. Without such evaluation metrics, it becomes difficult to comprehensively measure, compare, and benchmark the effectiveness of different multilingual KGQA approaches.
3. Ensuring the reproducibility of KGQA systems is crucial for building upon existing research. However, we have observed low reproducibility rates in some studies within the field.
4. The lack of a standardized format for KGQA benchmarks poses another significant challenge. Without a common benchmark format (e.g., QALD-JSON), it becomes technically arduous to compare the performance of different systems or share and replicate research findings.
5. It is prevalent for KGQA benchmarks to focus solely on a single KGs (e.g., DBpedia or Wikidata). This narrow focus limits the generalizability and applicability of the developed systems, as they may struggle when applied to different KGs.

6.2. Future Research Directions

Future research directions in the field of mKGQA hold substantial potential for advancements. Analyzing the challenges, the following research directions have been identified for further exploration:

1. Study on how people ask questions in different languages: understanding how individuals formulate questions in various languages, including their native language (L1), second language (L2), etc. is crucial. This research direction entails analyzing the syntactic structures employed, common errors made, and prevalent misspellings encountered in multilingual question posing on the Web.
2. mKGQA with unequal data distribution across languages: KGs often exhibit disparities in data availability across different languages. Investigating efficient mKGQA techniques that tackle this unequal distribution of language-specific data within KGs is an important research direction.
3. mKGQA with cultural context: consideration of a user's cultural background can significantly influence their information needs and preferences. Therefore, incorporating and adapting the ranking of answer candidates in accordance with a user's cultural context represents an essential research direction for enhancing the effectiveness of mKGQA.
4. Improving mKGQA quality in languages other than English: while much progress has been made in English KGQA, there is a need to extend the focus to other languages. A crucial objective in this research direction is to minimize the standard deviation of QA quality among different languages, ensuring comparable performance across all supported languages.
5. Generalization of methods to handle diverse languages: enhancing the applicability of mKGQA techniques to languages originating from different language groups, alphabets, writing systems, low-resource, and endangered languages is a vital research direction. It involves developing approaches that can effectively address the unique challenges posed by each language type.
6. Extending benchmarks for mKGQA: Expanding the existing benchmarks used for evaluating mKGQA systems is essential. This research direction involves augmenting the number of questions and translations available in different languages and ensuring the quality of these resources to ensure comprehensive evaluation and comparison of mKGQA approaches.
7. Encouraging publication of negative results: as a majority of publications tend to report positive results, advocating for the publication of negative results is vital. This research direction promotes transparency and prevents duplication of efforts by enabling researchers to learn from unsuccessful attempts and focus on more promising avenues.
8. Ensuring high-level reproducibility and comparability: to establish a strong foundation in the field of mKGQA, it is imperative to ensure the reproducibility and comparability of research results. This research direction emphasizes the adoption of standardized evaluation metrics, the release of open-source code, and the sharing of datasets to facilitate fair comparison and validation of proposed approaches.

We suggest the research community to consider these research directions, as further progress can be made in the field of mKGQA, leading to improved systems and comprehensive solutions for addressing multilingual information needs.

7. Conclusion

In conclusion, this systematic survey provides a comprehensive overview of the current state of research in mKGQA. Our work is primarily motivated by the lack of surveys that focus specifically on mKGQA (see Section 1). We followed the strict review methodology (see Section 2) to ensure the reproducibility and transparency of the results produced by us. Therefore, in the first step 1875 publications were retrieved according to the defined criteria. The publication search was conducted in three languages: English, German, and Russian. Finally, 46 publications were accepted for the review, where 26 of them are related to the systems, 14 to the benchmarks, and 7 to the related survey papers.

We systematically reviewed all the aforementioned publications and proposed the taxonomy of the methods for mKGQA systems. We formally defined three characteristics of the methods for mKGQA systems, namely: resource efficiency, multilinguality, and portability. We prepared the online repository[37] that contains structured information on the models and tools reviewed in this survey, as well as the source data and scripts. Finally, we highlighted the current challenges and future research directions of mKGQA, which also incorporates information on recent advances in the field of LLMs. *To the best of our knowledge, this is the first systematic survey that focuses specifically on the mKGQA.*

To answer $\mathcal{RQ}1$, we have organized prior contributions on mKGQA and have derived generalized insights on the systems and the methods they utilize with our taxonomy of the methods used for mKGQA systems. The taxonomy divides the methods for mKGQA into three major groups: Rules and Templates, Statistical Methods, and Machine Translation. In addition, we have proposed a novel procedure for evaluating the quality of mKGQA systems by using a multi-objective function that takes into account average quality across languages and the corresponding standard deviation. Based on the resulting values, this allows researchers to categorize systems into the quality quadrants.

While answering $\mathcal{RQ}2$, we have identified that most of the publications use Precision, Recall, and F1 score as metrics for QA quality evaluation. However, in some cases, such metrics as Accuracy, Hits@k, and exact match for SPARQL queries are used. We have found that the Indo-European language family tremendously dominates among the other families, such as Turkic, Sino-Tibetan, and Afro-Asiatic when it comes to mKGQA systems. Among the reviewed publications, we estimate that the Indo-European language family is used in 95% of work.

Finally, to answer $\mathcal{RQ}3$, we have systematically analyzed the available mKGQA benchmarks and identified that there is a chronological tendency towards an increase in the size of the benchmarks. Despite that, the number of languages and their diversity still varies from one to another. Moreover, the absence of a standardized benchmark format makes the process of creating and accessing this data fuzzy as no clear requirements and procedures are defined.

It is worth mentioning that our review methodology favors precision over recall when selecting the publications. Therefore, some relevant papers (e.g., [155, 156] have been left out due to a mismatch of our selection criteria. The most common mismatch reason is caused by the absence of the peer review process (e.g., preprint only, see Section 2.2) or lack of citations for preprints (our threshold is at least five, see Section 2.2). Also, some papers use multilingual benchmarks but focus only on the English language [157].

In the future, we will regularly update the survey results through the online KGQA leaderboard to keep track of the state-of-the-art in the mKGQA field. We believe that our review methodology enables other researchers to conduct similar work more efficiently. We also encourage researchers to deal with the challenges and research directions mentioned in Section 6.

Acknowledgements

This work has been partially supported by grants for the ITZBund[38]-funded research project “Entwicklung und Erforschung von IT-basierten Lösungen im Rahmen des ChatBot-Frameworks des Bundes (Question-Answering-Komponenten zur Erweiterung des ChatBot-Frameworks)” at the Leipzig University of Applied Sciences. The first author also thanks the Department of Computer Science and Languages of the Anhalt University of Applied Sciences for supporting this work.

References

[1] A. Strzelecki and P. Rutecka, Direct answers in Google search results, *IEEE Access* **8** (2020), 103642–103654.

[2] M. Potthast, M. Hagen and B. Stein, The dilemma of the direct answer, in: *ACM SIGIR Forum*, Vol. 54, ACM New York, NY, USA, 2021, pp. 1–12.

[3] D. Jurafsky and J.H. Martin, *Speech and Language Processing (3rd Edition draft).*, Prentice-Hall, Inc., USA, 2020, Chapter Question Answering and Information Retrieval. https://web.stanford.edu/~jurafsky/slp3/.

[37] https://github.com/Perevalov/multilingual-KGQA-survey

[38] https://www.itzbund.de/

[4] R. Dutt, S. Khosla, V.B. Kumar and R. Gangadharaiah, Designing harder benchmarks for evaluating zero-shot generalizability in question answering over knowledge bases, in: *ACL 2023 Workshop on Natural Language Reasoning and Structured Explanations*, 2023. https://www.amazon.science/publications/designing-harder-benchmarks-for-evaluating-zero-shot-generalizability-in-question-answering-over-knowledge-bases.

[5] C. Zhang, Y. Lai, Y. Feng and D. Zhao, A review of deep learning in question answering over knowledge bases, *AI Open* (2021).

[6] S. Xu, T. Culhane, M.-H. Wu, S.J. Semnani and M.S. Lam, Complementing GPT-3 with Few-Shot Sequence-to-Sequence Semantic Parsing over Wikidata, *arXiv preprint arXiv:2305.14202* (2023).

[7] N. McKenna and P. Sen, KGQA without retraining, in: *ACL 2023 Workshop on SustaiNLP*, 2023. https://www.amazon.science/publications/kgqa-without-retraining.

[8] J. Baek, A.F. Aji and A. Saffari, Knowledge-Augmented Language Model Prompting for Zero-Shot Knowledge Graph Question Answering, *arXiv preprint arXiv:2306.04136* (2023).

[9] G.G. Klager and A. Polleres, Is GPT fit for KGQA?–Preliminary Results, in: *Joint Proceedings of TEXT2KG 2023 and BiKE 2023*, S. Tiwari, N. Mihindukulasooriya, F. Osborne, D. Kontokostas, J. D'Souza, M. Kejriwal and E. Marx, eds, 2023, pp. 171–191. https://ceur-ws.org/Vol-3447/Text2KG_Paper_11.pdf.

[10] T. Berners-Lee, J. Hendler and O. Lassila, The Semantic Web, *Scientific American* **284**(5) (2001), 34–43. http://www.jstor.org/stable/26059207.

[11] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. de Melo, C. Gutiérrez, S. Kirrane, J.E. Labra Gayo, R. Navigli, S. Neumaier, A.-C. Ngonga Ngomo, A. Polleres, S.M. Rashid, A. Rula, L. Schmelzeisen, J.F. Sequeda, S. Staab and A. Zimmermann, *Knowledge Graphs*, Synthesis Lectures on Data, Semantics, and Knowledge Vol. 22, Morgan & Claypool, 2021. ISBN 9781636392363. doi:10.2200/S01125ED1V01Y202109DSK022. https://kgbook.org/.

[12] F. Manola, E. Miller, B. McBride et al., RDF primer, *W3C recommendation* **10**(1–107) (2004), 6.

[13] D.M. Axel-Cyrille Ngonga Ngomo and L. Bühman, A Holistic Natural Language Generation Framework for the Semantic Web, in: *Proceedings of the International Conference Recent Advances in Natural Language Processing*, ACL (Association for Computational Linguistics), 2019, p. 8.

[14] K. Bollacker, C. Evans, P. Paritosh, T. Sturge and J. Taylor, Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge, in: *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, Association for Computing Machinery, New York, NY, USA, 2008, pp. 1247–1250–. ISBN 9781605581026. doi:10.1145/1376616.1376746.

[15] Google, Freebase Data Dumps, June 21, 2022 edn, 2022. https://developers.google.com/freebase/data.

[16] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak and Z. Ives, DBpedia: A nucleus for a web of open data, in: *The semantic web*, Springer, 2007, pp. 722–735.

[17] D. Vrandečić and M. Krötzsch, Wikidata: A Free Collaborative Knowledgebase, *Commun. ACM* **57**(10) (2014), 78–85–. doi:10.1145/2629489.

[18] K. Balog, J. Dalton, A. Doucet and Y. Ibrahim, Report on the Eighth Workshop on Exploiting Semantic Annotations in Information Retrieval (ESAIR '15), *SIGIR Forum* **50**(1) (2016), 49–57–. doi:10.1145/2964797.2964806.

[19] I. Miliaraki, R. Blanco and M. Lalmas, From "Selena Gomez" to "Marlon Brando": Understanding Explorative Entity Search, in: *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 2015, pp. 765–775–. ISBN 9781450334693. doi:10.1145/2736277.2741284.

[20] C. Qiu, G. Zhou, Z. Cai and A. Søgaard, A Global–Local Attentive Relation Detection Model for Knowledge-Based Question Answering, *IEEE Transactions on Artificial Intelligence* **2**(2) (2021), 200–212. doi:10.1109/TAI.2021.3068697.

[21] Y. Pikus, N. Weißenberg, B. Holtkamp and B. Otto, Semi-Automatic Ontology-Driven Development Documentation: Generating Documents from RDF Data and DITA Templates, in: *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, SAC '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 2293–2302–. ISBN 9781450359337. doi:10.1145/3297280.3297508.

[22] L. Himanen, A. Geurts, A.S. Foster and P. Rinke, Data-driven materials science: status, challenges, and perspectives, *Advanced Science* **6**(21) (2019), 1900808.

[23] M. Irmer, C. Bobach, T. Böhme, A. Püschel and L. Weber, Using a chemical ontology for detecting and classifying chemical terms mentioned in texts, *Proceedings of Bio-Ontologies 2013* (2013).

[24] S. Aghaei, E. Raad and A. Fensel, Question answering over knowledge graphs: A case study in tourism, *IEEE Access* **10** (2022), 69788–69801.

[25] A. Pereira, J.R. Almeida, R.P. Lopes and J.L. Oliveira, Querying semantic catalogues of biomedical databases, *Journal of Biomedical Informatics* **137** (2023), 104272.

[26] A. Both, P. Heinze, A. Perevalov, J.R. Bartsch, R. Iudin, J.R. Herkner, T. Schrader, J. Wunsch, R. Gürth and A.K. Falkenhain, Quality Assurance of a German COVID-19 Question Answering Systems Using Component-Based Microbenchmarking, in: *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, WSDM '22, Association for Computing Machinery, New York, NY, USA, 2022, pp. 1561–1564–. ISBN 9781450391320. doi:10.1145/3488560.3502196.

[27] M. Keskenidou, A. Kyridis, L.P. Valsamidou and A.-H. Soulani, The Internet as a source of information. The social role of blogs and their reliability, *Observatorio (OBS*)* (2014).

[28] H.N. De Zoysa, The Internet as an accessible source of knowledge with special reference to undergraduates of the University of Kelaniya (2016).

[29] J. Slomian, O. Bruyère, J.-Y. Reginster and P. Emonts, The internet as a source of information used by women after childbirth to meet their need for information: A web-based survey, *Midwifery* **48** (2017), 46–52.

[30] M. Miquel-Ribé and D. Laniado, Wikipedia culture gap: quantifying content imbalances across 40 language editions, *Frontiers in physics* **6** (2018), 54.

[31] A.S. Hornby and A.P. Cowie, Oxford advanced learner's dictionary of current English (1977).

[32] B. Kitchenham, Procedures for performing systematic reviews, *Keele, UK, Keele University* **33**(2004) (2004), 1–26.

[33] P. Mian, T. Conte, A. Natali, J. Biolchini and G. Travassos, A systematic review process for software engineering, in: *ESELAW'05: 2nd Experimental Software Engineering Latin American Workshop*, 2005.

[34] J. Biolchini, P.G. Mian, A.C.C. Natali and G.H. Travassos, Systematic review in software engineering, *System engineering and computer science department COPPE/UFRJ, Technical Report ES* **679**(05) (2005), 45.

[35] A. Pereira, A. Trifan, R.P. Lopes and J.L. Oliveira, Systematic review of question answering over knowledge bases, *IET Software* **16**(1) (2022), 1–13.

[36] K. Höffner, S. Walter, E. Marx, R. Usbeck, J. Lehmann and A.-C. Ngonga Ngomo, Survey on challenges of question answering in the semantic web, *Semantic Web* **8**(6) (2017), 895–920.

[37] D. Diefenbach, V. López, K.D. Singh and P. Maret, Core techniques of question answering systems over knowledge bases: a survey, *Knowledge and Information Systems* **55** (2017), 529–569.

[38] E. Dimitrakis, K. Sgontzos and Y. Tzitzikas, A survey on question answering systems over linked data and documents, *Journal of intelligent information systems* **55**(2) (2020), 233–259.

[39] J.W.F. da Silva, A.D.P. Venceslau, J.E. Sales, J.G.R. Maia, V.C.M. Pinheiro and V.M.P. Vidal, A short survey on end-to-end simple question answering systems, *Artificial Intelligence Review* **53**(7) (2020), 5429–5453.

[40] C. Antoniou and N. Bassiliades, A survey on semantic question answering systems, *The Knowledge Engineering Review* **37** (2022).

[41] R. Usbeck, R.H. Gusmita, A.N. Ngomo and M. Saleem, 9th Challenge on Question Answering over Linked Data (QALD-9), in: *Joint 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, California, United States of America, October 8th - 9th, 2018.*, 2018, pp. 58–64.

[42] J. Berant, A. Chou, R. Frostig and P. Liang, Semantic Parsing on Freebase from Question-Answer Pairs, in: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Seattle, Washington, USA, 2013, pp. 1533–1544. https://aclanthology.org/D13-1160.

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

[44] P. Trivedi, G. Maheshwari, M. Dubey and J. Lehmann, LC-QuAD: A corpus for complex question answering over knowledge graphs, in: *International Semantic Web Conference*, Springer, 2017, pp. 210–218.

[45] M.J. Page, J.E. McKenzie, P.M. Bossuyt, I. Boutron, T.C. Hoffmann, C.D. Mulrow, L. Shamseer, J.M. Tetzlaff, E.A. Akl, S.E. Brennan et al., The PRISMA 2020 statement: an updated guideline for reporting systematic reviews, *Systematic reviews* **10**(1) (2021), 1–11.

[46] Óscar Ferrández, C. Spurk, M. Kouylekov, I. Dornescu, S. Ferrández, M. Negri, R. Izquierdo, D. Tomás, C. Orasan, G. Neumann, B. Magnini and J.L. Vicedo, The QALL-ME Framework: A specifiable-domain multilingual Question Answering architecture ∗, *Journal of Web Semantics* **9**(2) (2011), 137–145, Provenance in the Semantic Web. doi:https://doi.org/10.1016/j.websem.2011.01.002. https://www.sciencedirect.com/science/article/pii/S1570826811000126.

[47] N. Aggarwal, Cross Lingual Semantic Search by Improving Semantic Similarity and Relatedness Measures, in: *The Semantic Web – ISWC 2012*, P. Cudré-Mauroux, J. Heflin, E. Sirin, T. Tudorache, J. Euzenat, M. Hauswirth, J.X. Parreira, J. Hendler, G. Schreiber, A. Bernstein and E. Blomqvist, eds, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 375–382. ISBN 978-3-642-35173-0.

[48] M.-C. De Marneffe, B. MacCartney, C.D. Manning et al., Generating typed dependency parses from phrase structure parses., in: *Lrec*, Vol. 6, 2006, pp. 449–454.

[49] B. Li, H. Zhou, J. He, M. Wang, Y. Yang and L. Li, On the Sentence Embeddings from Pre-trained Language Models, in: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, Online, 2020, pp. 9119–9130. doi:10.18653/v1/2020.emnlp-main.733. https://aclanthology.org/2020.emnlp-main.733.

[50] W. Wang, G. Li, B. Ma, X. Xia and Z. Jin, Detecting Code Clones with Graph Neural Network and Flow-Augmented Abstract Syntax Tree, in: *2020 IEEE 27th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, 2020, pp. 261–271. doi:10.1109/SANER48275.2020.9054857.

[51] E. Cabrio, J. Cojan, F. Gandon and A. Hallili, Querying Multilingual DBpedia with QAKiS, in: *The Semantic Web: ESWC 2013 Satellite Events*, P. Cimiano, M. Fernández, V. Lopez, S. Schlobach and J. Völker, eds, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 194–198. ISBN 978-3-642-41242-4.

[52] J. Cojan, E. Cabrio and F. Gandon, Filling the Gaps among DBpedia Multilingual Chapters for Question Answering, in: *Proceedings of the 5th Annual ACM Web Science Conference*, WebSci '13, Association for Computing Machinery, New York, NY, USA, 2013, pp. 33–42–. ISBN 9781450318891. doi:10.1145/2464464.2464500.

[53] E. Cabrio, J. Cojan and F. Gandon, *Mind the Cultural Gap: Bridging Language-Specific DBpedia Chapters for Question Answering*, in: *Towards the Multilingual Semantic Web: Principles, Methods and Applications*, P. Buitelaar and P. Cimiano, eds, Springer Berlin Heidelberg, Berlin, Heidelberg, 2014, pp. 137–154. ISBN 978-3-662-43585-4. doi:10.1007/978-3-662-43585-4_9.

[54] I. Habernal and M. Konopík, SWSNL: Semantic Web Search Using Natural Language, *Expert Systems with Applications* **40**(9) (2013), 3649–3664. doi:https://doi.org/10.1016/j.eswa.2012.12.070.

[55] S. Hakimov, S. Jebbara and P. Cimiano, AMUSE: Multilingual Semantic Parsing for Question Answering over Linked Data, in: *The Semantic Web – ISWC 2017*, C. d'Amato, M. Fernandez, V. Tamma, F. Lecue, P. Cudré-Mauroux, J. Sequeda, C. Lange and J. Heflin, eds, Springer International Publishing, Cham, 2017, pp. 329–346. ISBN 978-3-319-68288-4.

[56] J. Nivre, D. Zeman, F. Ginter and F. Tyers, Universal Dependencies, in: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Tutorial Abstracts*, Association for Computational Linguistics, Valencia, Spain, 2017. https://aclanthology.org/E17-5001.

[57] D. Diefenbach, K. Singh and P. Maret, WDAqua-core0: A Question Answering Component for the Research Community, in: *Semantic Web Challenges*, M. Dragoni, M. Solanki and E. Blomqvist, eds, Springer International Publishing, Cham, 2017, pp. 84–89. ISBN 978-3-319-69146-6.

[58] A. Both, D. Diefenbach, K. Singh, S. Shekarpour, D. Cherix and C. Lange, Qanary–a methodology for vocabulary-driven open question answering systems, in: *European Semantic Web Conference*, Springer, 2016, pp. 625–641.

[59] D. Diefenbach, K. Singh, A. Both, D. Cherix, C. Lange and S. Auer, The Qanary Ecosystem: Getting New Insights by Composing Question Answering Pipelines, in: *Web Engineering - 17th International Conference, ICWE 2017, Rome, Italy, June 5-8, 2017, Proceedings*, J. Cabot, R.D. Virgilio and R. Torlone, eds, Lecture Notes in Computer Science, Vol. 10360, Springer, 2017, pp. 171–189. doi:10.1007/978-3-319-60131-1_10.

[60] R. Usbeck, A.-C.N. Ngomo, B. Haarmann, A. Krithara, M. Röder and G. Napolitano, 7th open challenge on question answering over linked data (QALD-7), in: *Semantic web evaluation challenge*, Springer, 2017, pp. 59–69.

[61] D. Diefenbach, K. Singh and P. Maret, WDAqua-Core1: A Question Answering Service for RDF Knowledge Bases, in *WWW '18*, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 2018, pp. 1087–1091–. ISBN 9781450356404. doi:10.1145/3184558.3191541.

[62] D. Diefenbach, P.H. Migliatti, O. Qawasmeh, V. Lully, K. Singh and P. Maret, QAnswer: A Question Answering Prototype Bridging the Gap between a Considerable Part of the LOD Cloud and End-Users, in: *The World Wide Web Conference*, WWW '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 3507–3510–. ISBN 9781450366748. doi:10.1145/3308558.3314124.

[63] S. Reddy, O. Täckström, S. Petrov, M. Steedman and M. Lapata, Universal Semantic Parsing, in: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Copenhagen, Denmark, 2017, pp. 89–101. doi:10.18653/v1/D17-1009. https://aclanthology.org/D17-1009.

[64] J. Nivre, M.-C. De Marneffe, F. Ginter, Y. Goldberg, J. Hajic, C.D. Manning, R. McDonald, S. Petrov, S. Pyysalo, N. Silveira et al., Universal dependencies v1: A multilingual treebank collection, in: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, 2016, pp. 1659–1666.

[65] E. Zimina, J. Nummenmaa, K. Jarvelin, J. Peltonen and K. Stefanidis, MuG-QA: Multilingual Grammatical Question Answering for RDF Data, in: *2018 IEEE International Conference on Progress in Informatics and Computing (PIC)*, 2018, pp. 57–61. doi:10.1109/PIC.2018.8706310.

[66] A. Ranta, Grammatical framework, *Journal of Functional Programming* **14**(2) (2004), 145–189.

[67] A. Ranta, The GF resource grammar library, *Linguistic Issues in Language Technology* **2** (2009).

[68] Interlanguage-links dataset, DBpedia, 2021. https://databus.dbpedia.org/dbpedia/generic/interlanguage-links/2021.12.01.

[69] N. Radoev, A. Zouaq, M. Tremblay and M. Gagnon, A Language Adaptive Method for Question Answering on French and English, in: *Semantic Web Challenges*, D. Buscaldi, A. Gangemi and D. Reforgiato Recupero, eds, Springer International Publishing, Cham, 2018, pp. 98–113. ISBN 978-3-030-00072-1.

[70] A. Taylor, M. Marcus and B. Santorini, The Penn treebank: an overview, *Treebanks* (2003), 5–22.

[71] E. Hovy, M. Marcus, M. Palmer, L. Ramshaw and R. Weischedel, OntoNotes: the 90% solution, in: *Proceedings of the human language technology conference of the NAACL, Companion Volume: Short Papers*, 2006, pp. 57–60.

[72] A. Pouran Ben Veyseh, Cross-Lingual Question Answering Using Common Semantic Space, in: *Proceedings of TextGraphs-10: the Workshop on Graph-based Methods for Natural Language Processing*, Association for Computational Linguistics, San Diego, CA, USA, 2016, pp. 15–19. doi:10.18653/v1/W16-1403. https://aclanthology.org/W16-1403.

[73] A. Moro, A. Raganato and R. Navigli, Entity Linking meets Word Sense Disambiguation: a Unified Approach, *Transactions of the Association for Computational Linguistics* **2** (2014), 231–244. doi:10.1162/tacl_a_00179. https://aclanthology.org/Q14-1019.

[74] T. Pellissier Tanon, M.D. de Assunção, E. Caron and F.M. Suchanek, Demoing Platypus – A Multilingual Question Answering Platform for Wikidata, in: *The Semantic Web: ESWC 2018 Satellite Events*, A. Gangemi, A.L. Gentile, A.G. Nuzzolese, S. Rudolph, M. Maleshkova, H. Paulheim, J.Z. Pan and M. Alam, eds, Springer International Publishing, Cham, 2018, pp. 111–116. ISBN 978-3-319-98192-5.

[75] M. Honnibal, I. Montani, S. Van Landeghem and A. Boyd, spaCy: Industrial-strength Natural Language Processing in Python (2020). doi:10.5281/zenodo.1212303.

[76] D. Diefenbach, J. Giménez-García, A. Both, K. Singh and P. Maret, QAnswer KG: Designing a Portable Question Answering System over RDF Data, in: *The Semantic Web*, A. Harth, S. Kirrane, A.-C. Ngonga Ngomo, H. Paulheim, A. Rula, A.L. Gentile, P. Haase and M. Cochez, eds, Springer International Publishing, Cham, 2020, pp. 429–445. ISBN 978-3-030-49461-2.

[77] D. Evseev and M.Y. Arkhipov, SPARQL query generation for complex question answering with Bert and BiLSTM-based model, in: *Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference Dialogue 2020*, 2020, pp. 270–282.

[78] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, 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, Volume 1 (Long and Short Papers)*, Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. doi:10.18653/v1/N19-1423. https://aclanthology.org/N19-1423.

[79] M. Mountantonakis, M. Bastakis, L. Mertzanis and Y. Tzitzikas, Tiresias: Bilingual Question Answering over DBpedia, in: *Workshop at ISWC 2022 on Deep Learning for Knowledge Graphs*, CEUR, 2022.

[80] R. Turganbay, V. Surkov, D. Evseev and M. Drobyshevskiy, Generative Question Answering Systems over Knowledge Graphs and Text, 2023, pp. 1112–1126. doi:10.28995/2075-7182-2023-22-1112-1126.

[81] I. Rybin, V. Korablinov, P. Efimov and P. Braslavski, RuBQ 2.0: an innovated Russian question answering dataset, in: *European Semantic Web Conference*, Springer, 2021, pp. 532–547.

[82] P. Ke, H. Ji, Y. Ran, X. Cui, L. Wang, L. Song, X. Zhu and M. Huang, Jointgt: Graph-text joint representation learning for text generation from knowledge graphs, *arXiv preprint arXiv:2106.10502* (2021).

[83] Y. Zhang, J. Wang, Z. Wang and R. Zhang, XSemPLR: Cross-Lingual Semantic Parsing in Multiple Natural Languages and Meaning Representations, in: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, Toronto, Canada, 2023, pp. 15918–15947. https://aclanthology.org/2023.acl-long.887.

[84] R. Cui, R. Aralikatte, H. Lent and D. Hershcovich, Compositional Generalization in Multilingual Semantic Parsing over Wikidata, *Transactions of the Association for Computational Linguistics* **10** (2022), 937–955. doi:10.1162/tacl_a_00499.

[85] S. Hochreiter and J. Schmidhuber, Long Short-Term Memory, *Neural Comput.* **9**(8) (1997), 1735–1780–. doi:10.1162/neco.1997.9.8.1735.

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

[87] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer and V. Stoyanov, Unsupervised cross-lingual representation learning at scale, *arXiv preprint arXiv:1911.02116* (2019).

[88] H.A. Chipman, E.I. George, R.E. McCulloch and T.S. Shively, mBART: multidimensional monotone BART, *Bayesian Analysis* **17**(2) (2022), 515–544.

[89] M. Chen, J. Tworek, H. Jun, Q. Yuan, H.P.d.O. Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman et al., Evaluating large language models trained on code, *arXiv preprint arXiv:2107.03374* (2021).

[90] T.L. Scao, A. Fan, C. Akiki, E. Pavlick, S. Ilić, D. Hesslow, R. Castagné, A.S. Luccioni, F. Yvon, M. Gallé et al., Bloom: A 176b-parameter open-access multilingual language model, *arXiv preprint arXiv:2211.05100* (2022).

[91] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua and C. Raffel, mT5: A massively multilingual pre-trained text-to-text transformer, *arXiv preprint arXiv:2010.11934* (2020).

[92] Y. Tan, X. Zhang, Y. Chen, Z. Ali, Y. Hua and G. Qi, CLRN: A reasoning network for multi-relation question answering over Cross-lingual Knowledge Graphs, *Expert Systems with Applications* **231** (2023), 120721. doi:https://doi.org/10.1016/j.eswa.2023.120721. https://www.sciencedirect.com/science/article/pii/S095741742301223X.

[93] Y. Tan, Y. Chen, G. Qi, W. Li and M. Wang, MLPQ: A Dataset for Path Question Answering over Multilingual Knowledge Graphs, *Big Data Research* **32** (2023), 100381. doi:https://doi.org/10.1016/j.bdr.2023.100381. https://www.sciencedirect.com/science/article/pii/S221457962300014X.

[94] Y. Zhou, X. Geng, T. Shen, W. Zhang and D. Jiang, Improving Zero-Shot Cross-lingual Transfer for Multilingual Question Answering over Knowledge Graph, in: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, Online, 2021, pp. 5822–5834. doi:10.18653/v1/2021.naacl-main.465. https://aclanthology.org/2021.naacl-main.465.

[95] A. Irvine and C. Callison-Burch, A comprehensive analysis of bilingual lexicon induction, *Computational Linguistics* **43**(2) (2017), 273–310.

[96] A. Perevalov, A. Both, D. Diefenbach and A.-C. Ngonga Ngomo, Can Machine Translation Be a Reasonable Alternative for Multilingual Question Answering Systems over Knowledge Graphs?, in: *Proceedings of the ACM Web Conference 2022*, WWW '22, Association for Computing Machinery, New York, NY, USA, 2022, pp. 977–986–. ISBN 9781450390965. doi:10.1145/3485447.3511940.

[97] J. Tiedemann and S. Thottingal, OPUS-MT — Building open translation services for the World, in: *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal, 2020.

[98] A. Perevalov, D. Diefenbach, R. Usbeck and A. Both, QALD-9-plus: A Multilingual Dataset for Question Answering over DBpedia and Wikidata Translated by Native Speakers, in: *2022 IEEE 16th International Conference on Semantic Computing (ICSC)*, IEEE, 2022, pp. 229–234.

[99] N. Srivastava, A. Perevalov, D. Kuchelev, D. Moussallem, A.-C. Ngonga Ngomo and A. Both, Lingua Franca – Entity-Aware Machine Translation Approach for Question Answering over Knowledge Graphs, in: *Proceedings of the 12th Knowledge Capture Conference 2023*, K-CAP '23, Association for Computing Machinery, New York, NY, USA, 2023, pp. 122–130–. ISBN 9798400701412. doi:10.1145/3587259.3627567.

[100] E. Cabrio, J. Cojan, A.P. Aprosio, B. Magnini, A. Lavelli and F. Gandon, QAKiS: an open domain QA system based on relational patterns, in: *International Semantic Web Conference, ISWC 2012*, 2012.

[101] N. Radoev, A. Zouaq and M. Gagnon, French and English Question Answering using Lexico-Syntactic Patterns, Vol. 65.

[102] N. Chakraborty, D. Lukovnikov, G. Maheshwari, P. Trivedi, J. Lehmann and A. Fischer, Introduction to neural network-based question answering over knowledge graphs, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **11**(3) (2021), e1389.

[103] X. Yin, D. Gromann and S. Rudolph, Neural machine translating from natural language to SPARQL, *Future Generation Computer Systems* **117** (2021), 510–519.

[104] D. Moussallem, M. Arčan, A.-C.N. Ngomo and P. Buitelaar, Augmenting neural machine translation with knowledge graphs, *arXiv preprint arXiv:1902.08816* (2019).

[105] D. Klein and C.D. Manning, Accurate unlexicalized parsing, in: *Proceedings of the 41st annual meeting of the association for computational linguistics*, 2003, pp. 423–430.

[106] E. Charniak and M. Johnson, Coarse-to-fine n-best parsing and maxent discriminative reranking, in: *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, 2005, pp. 173–180.

[107] D. Chen and C.D. Manning, A fast and accurate dependency parser using neural networks, in: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 740–750.

[108] C. Dyer, M. Ballesteros, W. Ling, A. Matthews and N.A. Smith, Transition-Based Dependency Parsing with Stack Long Short-Term Memory, in: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Association for Computational Linguistics, Beijing, China, 2015, pp. 334–343. doi:10.3115/v1/P15-1033. https://aclanthology.org/P15-1033.

[109] T. Dozat and C.D. Manning, Deep Biaffine Attention for Neural Dependency Parsing, in: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, OpenReview.net, 2017. https://openreview.net/forum?id=Hk95PK9le.

[110] P. Qi, Y. Zhang, Y. Zhang, J. Bolton and C.D. Manning, Stanza: A Python natural language processing toolkit for many human languages, *arXiv preprint arXiv:2003.07082* (2020).

[111] D. Jurafsky and J.H. Martin, *Speech and Language Processing (3rd Edition draft)*, Prentice-Hall, Inc., USA, 2020.

[112] G. Klyne and J.J. Carroll, Resource Description Framework (RDF): Concepts and Abstract Syntax, W3C, 2004. http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/.

[113] J. Tiedemann and S. Thottingal, OPUS-MT — Building open translation services for the World, in: *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal, 2020.

[114] D. Grune and C.J. Jacobs, Parsing Techniques (Monographs in Computer Science), Springer-Verlag, 2006.

[115] Á. Kovács, K. Gémes, E. Iklódi and G. Recski, POTATO: exPlainable infOrmation exTrAcTion framewOrk, *arXiv preprint arXiv:2201.13230* (2022).

[116] A. Kukushkin, Yargy parser: Rule-based facts extraction for Russian language, GitHub, 2022. https://github.com/natasha/yargy.

[117] D.L. Waltz, An English Language Question Answering System for a Large Relational Database, *Commun. ACM* **21**(7) (1978), 526–539–. doi:10.1145/359545.359550.

[118] G.G. Hendrix, E.D. Sacerdoti, D. Sagalowicz and J. Slocum, Developing a Natural Language Interface to Complex Data, *ACM Trans. Database Syst.* **3**(2) (1978), 105–147–. doi:10.1145/320251.320253.

[119] L.S. Zettlemoyer and M. Collins, Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars, *arXiv preprint arXiv:1207.1420* (2012).

[120] T. Kwiatkowksi, L. Zettlemoyer, S. Goldwater and M. Steedman, Inducing probabilistic CCG grammars from logical form with higher-order unification, in: *Proceedings of the 2010 conference on empirical methods in natural language processing*, 2010, pp. 1223–1233.

[121] J. Liu, S.B. Cohen and M. Lapata, Discourse representation structure parsing, in: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 429–439.

[122] J. Cheng, S. Reddy, V. Saraswat and M. Lapata, Learning structured natural language representations for semantic parsing, *arXiv preprint arXiv:1704.08387* (2017).

[123] A. Perevalov and A. Both, Augmentation-based Answer Type Classification of the SMART dataset., in: *SMART@ ISWC*, 2020, pp. 1–9.

[124] L.E. Baum and T. Petrie, Statistical inference for probabilistic functions of finite state Markov chains, *The annals of mathematical statistics* **37**(6) (1966), 1554–1563.

[125] A. Saxena, A. Tripathi and P. Talukdar, Improving multi-hop question answering over knowledge graphs using knowledge base embeddings, in: *Proceedings of the 58th annual meeting of the association for computational linguistics*, 2020, pp. 4498–4507.

[126] Y. Yan, R. Li, S. Wang, H. Zhang, Z. Daoguang, F. Zhang, W. Wu and W. Xu, Large-Scale Relation Learning for Question Answering over Knowledge Bases with Pre-trained Language Models, in: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2021, pp. 3653–3660. doi:10.18653/v1/2021.emnlp-main.296. https://aclanthology.org/2021.emnlp-main.296.

[127] T. Pires, E. Schlinger and D. Garrette, How Multilingual is Multilingual BERT?, in: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Florence, Italy, 2019, pp. 4996–5001. doi:10.18653/v1/P19-1493. https://aclanthology.org/P19-1493.

[128] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H.W. Chung, C. Sutton, S. Gehrmann et al., Palm: Scaling language modeling with pathways, *arXiv preprint arXiv:2204.02311* (2022).

[129] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D.d.L. Casas, L.A. Hendricks, J. Welbl, A. Clark et al., Training compute-optimal large language models, *arXiv preprint arXiv:2203.15556* (2022).

[130] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar et al., Llama: Open and efficient foundation language models, *arXiv preprint arXiv:2302.13971* (2023).

[131] T. Brown, B. Mann, N. Ryder, M. Subbiah, J.D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever and D. Amodei, Language Models are Few-Shot Learners, in: *Advances in Neural Information Processing Systems*, Vol. 33, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan and H. Lin, eds, Curran Associates, Inc., 2020, pp. 1877–1901. https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[132] A. Ugawa, A. Tamura, T. Ninomiya, H. Takamura and M. Okumura, Neural machine translation incorporating named entity, in: *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 3240–3250.

[133] A. Perevalov, X. Yan, L. Kovriguina, L. Jiang, A. Both and R. Usbeck, Knowledge Graph Question Answering Leaderboard: A Community Resource to Prevent a Replication Crisis, in: *Proceedings of the Language Resources and Evaluation Conference*, European Language Resources Association, Marseille, France, 2022, pp. 2998–3007. https://aclanthology.org/2022.lrec-1.321.

[134] E. Loginova, S. Varanasi and G. Neumann, Towards End-to-End Multilingual Question Answering, *Information Systems Frontiers (ISF)* **22** (2020), 1–14.

[135] A. Neves, A. Lamurias and F. Couto, Biomedical Question Answering using Extreme Multi-Label Classification and Ontologies in the Multilingual Panorama, in: *Semantic Indexing and Information Retrieval for Health Held in conjunction with the 42nd European Conference on Information Retrieval (SIIRH@ECIR)*, 2020.

[136] C.P. Carrino, M. Ruiz Costa-Jussà and J.A. Rodríguez Fonollosa, Automatic Spanish translation of SQuAD dataset for multi-lingual question answering, in: *LREC 2020: 12th International Conference on Language Resources and Evaluation: Marseílle, France: May 13-15, 2020: conference proceedings*, European Language Resources Association (ELRA), 2020, pp. 5515–5523.

[137] E. Cabrio, P. Cimiano, V. Lopez, A.-C.N. Ngomo, C. Unger and S. Walter, QALD-3: Multilingual Question Answering over Linked Data., *CLEF (Working Notes)* **38** (2013).

[138] T. Souza Costa, S. Gottschalk and E. Demidova, Event-QA: A Dataset for Event-Centric Question Answering over Knowledge Graphs, in: *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 3157–3164–. ISBN 9781450368599. doi:10.1145/3340531.3412760.

[139] V. Korablinov and P. Braslavski, RuBQ: a Russian dataset for question answering over Wikidata, in: *International Semantic Web Conference*, Springer, 2020, pp. 97–110.

[140] P. Sen, A.F. Aji and A. Saffari, Mintaka: A Complex, Natural, and Multilingual Dataset for End-to-End Question Answering, in: *Proceedings of the 29th International Conference on Computational Linguistics*, International Committee on Computational Linguistics, Gyeongju, Republic of Korea, 2022, pp. 1604–1619. https://aclanthology.org/2022.coling-1.138.

[141] M. Sanguinetti, M. Atzori, N. Puddu et al., rewordQALD9: A Bilingual Benchmark with Alternative Rewordings of QALD Questions, in: *CEUR WORKSHOP PROCEEDINGS*, Vol. 3235, CEUR-WS, 2022.

[142] J. Soruco, D. Collarana, A. Both and R. Usbeck, QALD-9-ES: A Spanish Dataset for Question Answering Systems, in: *Knowledge Graphs: Semantics, Machine Learning, and Languages*, IOS Press, 2023, pp. 38–52.

[143] R. Usbeck, X. Yan, A. Perevalov, L. Jiang, J. Schulz, A. Kraft, C. Möller, J. Huang, J. Reineke, A.-C.N. Ngomo, M. Saleem and A. Both, QALD-10 — The 10th Challenge on Question Answering over Linked Data, *Under review in the Semantic Web Journal* (2023). https://www.semantic-web-journal.net/system/files/swj3357.pdf.

[144] K. Höffner, S. Walter, E. Marx, R. Usbeck, J. Lehmann and A.-C. Ngonga Ngomo, Survey on Challenges of Question Answering in the Semantic Web, *Semantic Web* **8** (2016). doi:10.3233/SW-160247.

[145] D. Sorokin and I. Gurevych, Modeling semantics with gated graph neural networks for knowledge base question answering, *arXiv preprint arXiv:1808.04126* (2018).

[146] D. Diefenbach, A. Both, K. Singh and P. Maret, Towards a question answering system over the semantic web, *Semantic Web* **11**(3) (2020), 421–439.

[147] S. Gottschalk and E. Demidova, EventKG–the hub of event knowledge on the web–and biographical timeline generation, *Semantic Web* **10**(6) (2019), 1039–1070.

[148] D. Keysers, N. Schärli, N. Scales, H. Buisman, D. Furrer, S. Kashubin, N. Momchev, D. Sinopalnikov, L. Stafiniak, T. Tihon, D. Tsarkov, X. Wang, M. van Zee and O. Bousquet, Measuring Compositional Generalization: A Comprehensive Method on Realistic Data, in: *International Conference on Learning Representations (ICLR)*, 2020. https://openreview.net/pdf?id=SygcCnNKwr.

[149] C. Unger, A.-C.N. Ngomo and E. Cabrio, 6th open challenge on question answering over linked data (qald-6), in: *Semantic Web Challenges: Third SemWebEval Challenge at ESWC 2016, Heraklion, Crete, Greece, May 29-June 2, 2016, Revised Selected Papers 3*, Springer, 2016, pp. 171–177.

[150] R. Usbeck, A.-C.N. Ngomo, F. Conrads, M. Röder and G. Napolitano, 8th challenge on question answering over linked data (QALD-8), *language* **7**(1) (2018), 51–57.

[151] K. Al Sharou, Z. Li and L. Specia, Towards a better understanding of noise in natural language processing, in: *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, 2021, pp. 53–62.

[152] N. Chakraborty, D. Lukovnikov, G. Maheshwari, P. Trivedi, J. Lehmann and A. Fischer, Introduction to neural network-based question answering over knowledge graphs, *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **11**(3) (2021). doi:10.1002/widm.1389. https://jens-lehmann.org/files/2021/neural_kgqa_intro.pdf.

[153] J. Baek, A.F. Aji and A. Saffari, Knowledge-augmented language model prompting for zero-shot knowledge graph question answering, in: *ACL 2023 Workshop on Matching Entities*, 2023. https://www.amazon.science/publications/knowledge-augmented-language-model-prompting-for-zero-shot-knowledge-graph-question-answering.

[154] J. Jiang, K. Zhou, Z. Dong, K. Ye, W.X. Zhao and J.-R. Wen, StructGPT: A General Framework for Large Language Model to Reason over Structured Data, 2023.

[155] Y. Zhan, Y. Li, M. Zhang and L. Zou, ADMUS: A Progressive Question Answering Framework Adaptable to Multiple Knowledge Sources, *arXiv preprint arXiv:2308.04800* (2023).

[156] M.F. Elahi, B. Ell, G. Nolano and P. Cimiano, Multilingual Question Answering over Linked Data building on a model of the lexicon-ontology interface, *Semantic Web Journal* (2023). https://www.semantic-web-journal.net/system/files/swj3619.pdf.

[157] B. Faria, D. Perdigão and H. Gonçalo Oliveira, Question Answering over Linked Data with GPT-3, in: *12th Symposium on Languages, Applications and Technologies (SLATE 2023)*, Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2023.