Is GPT fit for KGQA? – Preliminary Results*

Gerhard G. Klager[1,*,†], Axel Polleres[1,†]

[1]WU Wien - Vienna University of Economics and Business, Welthandelsplatz 1, Vienna, 1020, Austria

Abstract

In this paper we report about preliminary results on running question answering benchmarks against the recently hyped conversational AI services such as ChatGPT: we focus on questions that are known to be possible to be answered by information in existing Knowledge graphs such as Wikidata. In a preliminary study we experiment, on the one hand, with questions from established KGQA benchmarks, and on the other hand, present a set of questions established in a student experiment, which should be particularly hard for Large Language Models (LLMs) to answer, mainly focusing on questions on recent events. In a second experiment, we assess how far GPT could be used for query generation in SPARQL. While our results are mostly negative for now, we hope to provide insights for further research in this direction, in terms of isolating and discussing the most obvious challenges and gaps, and to provide a research roadmap for a more extensive study planned as a current master thesis project.

Keywords

Question Answering, KGQA, LLMs, GPT

1. Introduction

With the ever-growing number of publicly available Knowledge Graphs and their increasing relevancy the task of question answering (KGQA from here on) has risen in popularity as well [1]. The purpose of a KGQA-system is to allow end-users to retrieve information stored in a KG by means of natural language questions, without being familiar with the KG's structure or the query language used to access said KG.

In order to achieve this goal, often some kind of translation of natural language questions into a query is taking place [2]. Many different KGQA approaches exist, ranging from template-based approaches [3] to approaches based on unsupervised message passing [4] or approaches using methods of machine learning [5]. The capabilities of KGQA-systems range from answering simple questions [6] to complex questions [7] as well to engage in single- and multi-turn (or conversational) question answering [8]. Additionally, some of these approaches even try to enable QA independent of a fixed KG or language [2].

To train and evaluate these models numerous benchmarks have been created, enabling a direct

TEXT2KG 2023: Second International Workshop on Knowledge Graph Generation from Text, May 28 - Jun 1, 2023, co-located with Extended Semantic Web Conference (ESWC), Hersonissos, Greece

*The text of this paper was hand-written without the support of text-generating AI, this – however – does not apply to the SPARQL query examples in this paper ;-)

*Corresponding author.

†These authors contributed equally.

gerhard.klager@gmail.com (G. G. Klager); axel.polleres@wu.ac.at (A. Polleres)

0009-0000-2816-219X (G. G. Klager); 0000-0001-5670-1146 (A. Polleres)

comparison between existing and new QA-systems [1].

At the same time, with the recent success of OpenAI's ChatGPT[9] and its many competitors [10], we see many applications of such large language models (LLMs), not only restricted to question answering alone, but also in producing more or less useful code in programming and query languages. Facing these developments, we may ask ourselves both (a) if such LLMs can act as serious contenders to bespoke KGQA systems, and (b) whether LLMS could be used as a supportive technology for query formulation in the context of KGQA. However, literature covering this subject is still scarce and end-to-end QA-systems using LLMs such as ChatGPT in a synergetic combination with KGQA have not yet been proposed in abundance.

The aim of this paper is therefore to fill this gap by exploring the possibilities of using LLMs such as ChatGPT in the task of KGQA and to challenge the status quo of existing benchmarks aimed at training and evaluating KGQA-systems.

In particular, we are interested in answering the following questions:

1. How does LLM-based QA differ from established KGQA approaches and what are the respective strengths, weaknesses and challenges of the two methods?
2. Which components used in KGQA-systems could be enhanced using LLMs?
3. What types of questions are found in existing benchmarks for KGQA approaches and in how far can these be used in benchmarking LLM-based QA approaches?
4. How can a comprehensive benchmark for LLM-based, KGQA-based but also combined QA approaches be constructed that is challenging the current weaknesses of both approaches?

In order to get closer to answering these questions, we have started with a comprehensive literature review. The goal of this literature review is to establish an overview of the already existing different QA-systems and benchmarks on the one hand and to lay the foundations of the approaches chosen to create our proposed new benchmark and QA-system as well as their characteristics and main components. Our initial collection of articles[2, 4, 11, 1, 8, 12, 13, 7, 6, 14, 15, 5, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41] contains both proposed solutions to
(KG)QA as well as existing benchmarks.

While still ongoing, we provide an overview of our current status of identified benchmarks and their main characteristics, as well as (components of) KGQA systems in Section 2 below. In the next step, we plan to compare existing benchmarks and systems from the literature for KGQA with LLM based question-answering. This shall be done by systematically comparing the questions available in each benchmark, wrt. the (syntactic) structures of the questions and – if available – corresponding structured queries, topic domains, and other characteristics indicating the complexity of the question answering tasks (such as for instance aggregations needed, etc.) and comparing the results of established KGQA and LLM based QA on each of these benchmarks.

For this first preliminary study, we proceed by selecting a mini-benchmark from

- sample subsets from the SimpleQuestions [42] and QALD-5 [43] benchmark datasets,
- a small dataset we manually designed to challenge ChatGPT

both of which having in mind that – in principle – the respective answers should intuitively be findable in existing KGs such as Wikidata [44]. We present this mini-benchmark in Section 3.

While certainly not yet representative of the field, we aim at drawing some initial conclusions about LLMs/GPTs capabilities and challenges with respect to two separate subtasks of KGQA

- directly answering the natural language queries from our mini-benchmark vs.
- translating the natural language queries to SPARQL

We summarize the performance on both these subtasks in Section 3.4 and derive hypothesis for further investigation.

We conclude in Section 4 with an outlook for further work that also include a workplan and main tasks to be executed in an ongoing Master thesis, wherefore we eagerly look forward to feedback during the TEXT2KG workshop. As an overall goal in our research agenda, we plan to design and implement a new hybrid approach making use of LLMs, such as ChatGPT (or also other emerging, and hopefully open LLMs) to improve upon the weaknesses of existing KGQA-systems without "LLM support".
Additionally, we hope our findings will serve as a base for new QA benchmarks aimed at improving the training and evaluation of future, combined LLM-and-KGQA-systems.

2. Related work on KGQA

With the growing attention given to KGQA in recent years we can also observe a large growth of literature covering KGQA-systems and in terms of different methods and benchmarks to evaluate these systems. This section is dedicated to provide an overview of this literature and laying the foundation for our planned research.

2.1. KGQA Benchmarks

The topic of benchmarking in QA ranges from the methods of creating benchmarking datasets, to the types of questions and queries used in a dataset, to methods to evaluate those benchmarks themselves.

Probably the most widely used family of question answering datasets is represented by the Question Answering over Linked Data (*QALD* from here on) campaign. This series of challenges aims at providing benchmarks for all QA-systems designed at using natural language requests of a user to retrieve information stored as structured data such as the RDF data format. Additionally, the challenge aims at comparing current state-of-the-art QA-systems with regard to their individual strengths and shortcomings. In order to participate in the current QALD challenge users can simply run their QA-system using the current challenge's dataset before storing their results in an XML file and upload it to the challenge's website [45]. The QALD challenge is currently taking place in its 10^{th} iteration [39] and accordingly provides 10 datasets that could be used within further elaborations of our study [**?**], a detailed analysis of these datasets is on our agenda; for the moment, we have considered a sample from the 5^{th} QALD [43] in our preliminary experiments, see below.

As a more manageable starting point, *SimpleQuestions* is a dataset containing 100k questions aimed at training and evaluating QA-systems with regard to solving the simple question answering problem, which consists answering questions that can be rephrased as (single triple) queries that ask for all objects linked to a questions given subject by its given relationship. In this context, simple QA is a term used referring to the simplicity of the reasoning process necessary to answer questions [46]. While SimpleQuestions was originally designed to be run over FreebBase, Diefenbach et al. [42] have adapted/extended the original SimpleQuestions dataset to Wikidata recently, which we include in our preliminary study since it is possible to be tested against a large KG, available via a public SPARQL endpoint.

As for further relevant QA benchmarks, Berant et. al. [47] created a new QA dataset named *WebQuestions*. The WebQuestions dataset acts as an extension to the FREE917 dataset (again based on Freebase) aimed at evaluating QA-systems. The authors created this dataset due to the FREE917 dataset requiring logical forms, making it inherently more difficult to scale it up due to the requirement of having expertise in annotating logical forms. Using the Google Suggest API, the authors obtained questions beginning with a wh-word (where, who, when, etc.) and containing exactly one entity. For each question, five candidate queries have been created. After collecting 1M questions in this process, 100k randomly selected questions have been submitted to Amazon Mechanical Turk (AMT from here on) where workers answered questions detecting duplicates and filtering out questions that could not be answered. The remaining dataset contained 5.810 questions. In particular, the approach to crowd-source and to cross-check labeling in order to compare humans against QA systems may also be useful for us in further elaborations of our study.

As an alternative to simple questions the Large-Scale Complex Question Answering Dataset 2.0 [40] (*LC-QuAD 2.0* from here on) is an extension to the original LCQuAD dataset [41] containing 30k complex questions as well as their corresponding paraphrased versions and SPARQL queries. This dataset is both compatible with Wikidata and DBpedia (2018). The dataset was created by generating a number of SPARQL queries before verbalizing them into natural language questions using the AMT. Afterward, these questions have been paraphrased to create additional natural language questions. The LC-Quad 2.0 dataset contains 10 different question types ranging from single fact questions which will be answered by returning either a subject or object to complex questions requiring complex patterns, temporal information to be answered, etc.

Another approach of how to create a benchmark has been taken by the creators of the WDAquaCore0Questions dataset which represents a collection of questions asked by users testing the demo of the WDAqua-core-0 QA-system for Wikidata [48].

Jiang and Usbeck analyzed 25 KGQA datasets with regard to five different KGs. Their study showed that many available KGQA datasets are unfit to train KGQA-systems due to their underlying assumptions or that these datasets are outdated and based on discontinued KGs. Additionally, the authors share light on the difficulties and high costs related to the generation of new datasets. Therefore, they propose an automated method to re-split datasets enabling their generalization as well as a method to analyze existing KGQA datasets with regard to their generalizability [11].

While many different benchmarks aimed at evaluating QA-systems for different KGs exist the question of which benchmark one should use can be a difficult one to answer. To answer this

question Orogat, Liu, and El-Roby proposed *CBench* [1], a suite that enables users to analyze existing benchmarks with regard to linguistic, syntactic, and structural properties of the dataset's questions and queries as well as to evaluate QA-systems. Additionally, the authors provide an overview of different creation methods for benchmarks ranging from manual creation based on heuristics to benchmarks created automatically from the KG in question.

In light of recent developments, and while social media is full of examples, there is — to the best of our knowledge — not yet a dedicated QA dataset originally tailored to LLMs and GPT specifically. In order to fill this gap, we asked students of the Digital Economy masters' program at the Vienna University of Economics and Business to generate a set of natural language questions aimed at asking ChatGPT to formulate queries GPT-3 would fail upon but suspected to be possible to answer with the information in publicly available KGs such as Wikidata. As a hint, we emphasized that we suspect LLMs to struggle with (a) recent events information beyond the training phase of the LLM (b) complex questions that require non-obvious conceptual understanding and reasoning. The students' task was to also find/formulate the corresponding SPARQL queries and – in the light of recent advances of LLMs for also code and query generation, attempt whether ChatGPT was able to create such queries. We report on a selected subset of these questions in section 3 below.

2.2. Question Answering Systems

With the existence of numerous QA-benchmarks it is no surprise that the literature presents an abundance of different QA-systems as well. These systems range from ones limited to single KGs to systems able to access multiple KGs, from language dependent to language independent systems, and from simpler template-based systems to complex systems incorporating elements of machine learning.

Diefenbach et. al. propose a QA-system capable of querying multiple KGs independent of the natural language used. Their approach has been evaluated on five well-known KGs and five different languages using three different benchmarks. Their proposed QA-system first performs *entity recognition* in terms of searching corresponding IRIs whose lexicalization is an n-gram (consecutive elements in a text) in the asked natural language questions question. After removing stop words from the set of IRIs queries that could represent possible interpretations of the question are constructed before being ranked based on multiple aspects such as the number of words matching the words in the original question. Next, a logistic regression based on labeled SPARQL queries will be trained to compute the confidence score for each query. Last, the highest ranked query above a certain threshold will be used to answer the question. If no query with confidence above the threshold is found, the whole question will be deemed unanswerable. During their study, the authors discovered performance differences in their approach wrt. different (natural) languages used and link these differences to the quality of the available data for each language [2].

Vakulenko et al. [4] take a quite different approach based on the usage of unsupervised message passing (QAmp from here on) which consists of two phases: in the first phase called question interpretation, the relevant sets of entities and predicates necessary for answering the

input question are again being identified and their confidence scores are being computed. In the second phase, the so-called answer inference phase, these confidence scores a propagated and aggregated over the underlying KG's structure, providing a confidence distribution over a set of possible answers which is then be used to locate the corresponding answer entities, rather than translating the query to SPARQL.

Yani et al. [7] propose yet another a method to detect entities and their position on triples that have been mentioned in a complex question. Their approach is capable of not only detecting the entity name but also of determining in which triple the entity is located and if the given entity is a head or tail of the triple.

Shin et. al. [15] notice that QA systems suffer notably from the divergence of the unstructured data composing natural language questions and the structured data composing a KG. Existing approaches to deal with this issue use lexicons in order to cover differently represented data. Since these lexicons only consider representations for entity and relation mentions the authors propose a new predicate constraint lexicon restricting subject and object types for a predicate. This so-called Predicate Constraints based Question Answering (PCQA from here on) lexicon does not make use of any templates. Rather the authors generated query graphs focusing on matching relations in order to cover diverse types of questions.

Another QA-system proposed by Liang et. al. [5] is based on the idea of splitting the process of translating natural language questions into SPARQL queries into five sub-tasks. First, a random forest model is trained to identify a question's type. Next, various entity recognition and property mapping tools are used to map the question's phrases before all possible triple patterns are created based on these mapped resources. Afterward, possible SPARQL are generated by combining these triple patterns before a Tree-LSTM based ranking model is used to select the most plausible SPARQL query representing the correct intention behind the natural language question. Possible SPARQL queries are then constructed by combining these triple patterns in the query generation step. In order to select the correct SPARQL query among a number of candidate queries for each question, a ranking model based on Tree-LSTM is used in the query ranking step. The ranking model takes into account both the syntactical structure of the question and the tree representation of the candidate queries to select the most plausible SPARQL query representing the correct intention for the respective question.

As this short overview shows, the main tasks in many KGQA systems firstly involve entity/property recognition and matching to respective IRIs in the KG. Secondly, some but not all QA systems proceed by formulating SPARQL queries from these entities. Our following preliminary experiment is therefore tailored to mainly challenge GPT in terms of whether these tasks can be adequately supported by (currently existing) LLMs.

3. Benchmarking LLMs

In order to evaluate the raw performance of large language models we decided to use two of OpenAI's large language models. The first model we used is the GPT 3.5-based ChatGPT. Additionally, we used OpenAI's older GPT 3-based davinci model to give a comparison to ChatGPT's results and to possibly detect structural characteristics of LLMs in the context of question answering.

For this we first let each system/model answer all natural questions directly and secondly indirectly by first generating corresponding SPARQL queries for Wikidata, before subsequently attempting to retrieve their results. This process was done for (i) the student dataset aimed at providing questions that cannot be answered by ChatGPT, (ii) a subset of the SimpleQuestions adaptation for Wikidata [42], and (iii) a subset of the QALD-5 dataset [43], with the student dataset consisting of 14 questions and the two subsets consisting of 15 randomly drawn questions from the original datasets. Extending this study to analogously test further KGQA benchmarks is on our agenda.

We limited our study to the Wikidata KG. This decision has been made for multiple reasons. First, while other popular KGs, such as Freebase, stopped their operations, Wikidata is one of the most popular, and most actively maintained KGs. Besides accounting for the KGs relevancy, this could also mean that Wikidata is better suited to be used when answering information on current events, an expected weakness of LLMs wrt. QA. Secondly, this study aims at uncovering LLMs limitations wrt. KGQA. This renders Wikidata specifically challenging since the task of entity recognition can be assumed to be harder for Wikidata than for other KGs such as DBpedia. This is due Wikidata's numeric identifiers, LLMs should not be able to derive the correct identifiers directly from the question asked, which could be blurred by the inherent semantics of language-based URIs [49] as used for instance in DBpedia. While limiting ourselves to one KG for this preliminary study allowed us to obtain first insights results, expanding and comparing our research wrt. to a comparison with other/multiple KGs is on our agenda.

3.1. Question Answering, Query Generation and Query Execution

All necessary computations and all necessary programming in this study has been done by R scripts [50], using the openai package (version 0.4.0) [51] in combination with OpenAI's API to interact with ChatGPT and davinci. At this point, it must be noted that OpenAI's API allows the usage of different *temperature* options to control how deterministic the behavior of the LLM should be. While the chosen setting might potentially have a significant influence on the results, we used the default temperature setting of 1 in this study to replicate especially ChatGPTs behavior when accessed through its Web interface, to which we have been able to record differences regardless. Secondly, upon trying to execute our scripts with a temperature of 0, supposedly meaning fully deterministic behavior, a later mentioned problem of the LLMs getting stuck at certain questions and repeating previous answers or query structures occurred for most of the questions, rendering the results useless. At this point, we cannot confirm whether this is a result of high server load or if the temperature setting is at least partially responsible for this. Further investigation of the significance of the temperature setting is therefore on our agenda. In this context, note that question 12 and 15 in the sample of the QALD-5 dataset (cf. Table 3) are identical. Since we drew 15 random questions, all having different questions IDs, this indicates that the original dataset contains duplicates. We did deliberately *not* remove those for now, in order determine whether duplicates yield different answers: while slightly different in their wording, the direct natural language answers' content has been identical, and likewise the generated SPARQL queries were identical in our preliminary experimental run. A more in depth investigation in how far repeated "runs" of the experiment yield different results or improvements, also in the context of adapting GPT's temperature parameter, is on our agenda.

In order to generate both the answers to a natural language question (NLQ) and its corresponding SPARQL query the following natural language prompt was used:

> *"Please write me a SPARQL query on Wikidata without comments to answer the following Question: NLQ."*.

Since the used LLMs usually add comments within their queries the passage *"without comments"* has been added to eliminate these comments, which allows for easier processing and subsequently easier execution of these queries. Finally, we used the WikidataQueryServiceR package (version 1.0.0) [52] to execute the generated queries and to retrieve their results.

We note that, besides the elimination of unwanted comments the selected prompt has been carefully designed in a way that aims at preventing the prompt to influence the LLMs answers besides their structural representation: as it was the aim of this preliminary study to determine how LLM's as a standalone option lend themselves to QA and KGQA tasks, we started with a static, uniform prompt. The resulting findings should then form a foundation based on which further insights on the topic of prompt engineering could be derived, i.e., how advanced methods including specifically engineered prompts or hybrid QA approaches that dynamically generate prompts could perform KGQA tasks. While exploring these questions further the lies outside of the current study's scope, we consider it a potentially important direction for future work.

3.2. Performance Evaluation

In order to evaluate ChatGPT's and GPT 3's performance both their answers given in natural language and the results of the queries generated by the LLMs have been assigned one out of three possible grades: *Correct*, *Incorrect*, and *No answer*. Correct marks a case where the LLMs were in fact able to answer the given question correctly. Incorrect results mark cases where they were able to answer the question but did so incorrectly. Finally, no answer is assigned to cases where the LLM's where unable to generate an answer to the question asked. We assume that this will be the case when the LLMs are asked about events happening outside of their training period (hence after September 2021).

3.3. Results

In this section, we will show the results generated by ChatGPT and GPT 3 on the student dataset, as well as our subsamples of the SimpleQuestions dataset and the QALD-5 challenge dataset.

3.3.1. Student Dataset

As described earlier, we generated SPARQL queries for each of the questions in the student-generated dataset, retrieved their results as well as the direct answers NLAs) to the NLQs given by ChatGPT and GPT 3, and evaluated them. Table 1 shows the questions in this dataset.

Unsurprisingly, ChatGPT was unable to answer most of the dataset's questions correctly. However, ChatGPT acknowledged its limitations wrt. dates and added a disclaimer at the beginning of its answers stating that its knowledge is limited to dates up to September 2021.[1]

[1]Some further experiments also show that this behavior could seemingly – in the tested GPT versions – sometimes be worked around by prompt reformulation, typically leading to a factually wrong answer.

Table 1
A sample of hard questions for GPT from our student experiment

	Question
1	Who is the current president of the United States?
2	Who won the football worldcup 2022?
3	Give me all Austrian female actors that are aged over 50?
4	Give me all Austrian female actors aged over 50years that are also dancers or singers?
5	When did the famous Brazilian football player Pelé die?
6	For which team does Lionel Messi play?
7	What is the most recent MineCraft Java Edition version?
8	How many people do live on earth?
9	What was the average temperature in Vienna in 2022?
10	Who is the fastest person in the world?
11	What is the oldest painting in the world?
12	Where does the handball world cup take place this year (2023)?
13	Who is CEO of Twitter?
14	Which team won the 'Serie A' championship last season?

Out of 14 questions with our standardized prompt, ChatGPT was able to provide 8 NLAs, three out of which were correct.
As expected, incorrect answers have been given wrt. changes that happened after 2021, such as when ChatGPT was asked to name the "*most recent Minecraft Java Edition version*" (1.19.3 at the time of writing) to which it responded with "*As of September 2021, the most recent version of Minecraft Java Edition is 1.17.1.*".

Another interesting observation is that ChatGPT seemingly can sometimes get "stuck" at a given question. Consider the second and third questions in our student dataset: ChatGPT stated that it is not capable of answering the second question *Who won the football worldcup 2022?* due to it not having taken place by ChatGPT's knowledge. However, ChatGPT gave the same answer to the third question *Give me all Austrian female actors that are aged over 50?*. The same anomaly occurred with question 7 *"What is the most recent MineCraft Java Edition version?* and question 8 *How many people do live on earth?*. We so far did not entirely clarify, whether this behavior was due to an API issue, or due to the sequential nature of the model itself, where different answers are obviously depending on the order of interactions. Regardless of this, ChatGPT was able to generate SPARQL queries for all NLQs within the student dataset. Out of these 14 queries 13 were syntactically correct and could indeed be executed. However, 10 out of these 13 queries returned no results. The three remaining queries returning results have been the queries for question 1 "*Who is the current president of the United States?*" (Listing 1), question 2 "*Who won the football worldcup 2022?*" ((Listing 2)), and question 8 "*How many people do live on earth?*" (Listing 3).

The former, shown in listing 1 correctly returned Joe Biden, and — looking at the ORDER BY and LIMIT combination, indeed semantically attempts to retrieve the most recent president. We should note though, that question 1 was – as opposed to the other student questions – provided by the instructor upfront, as an example of a question that was correctly translated by GPT, having in mind to find a likely common example question referring to current data, but also probably available verbatim in SPARQL examples that the LLM has been trained upon.

Listing 1: ChatGPT generated query for: Who is the current president of the united states?

```
SELECT ?presidentLabel
WHERE {
    wd:Q30 p:P6 [ ps:P6 ?president ; pq:P580 ?start_time ].
    FILTER NOT EXISTS { ?president p:P582 [] }
    SERVICE wikibase:label { bd:serviceParam
                             wikibase:language "[AUTO_LANGUAGE},en" .}
} ORDER BY DESC(?start_time)
LIMIT 1
```

Listing 2: ChatGPT generated query for: Who won the football worldcup 2022?

```
SELECT ?teamLabel
WHERE {
    ?cup wdt:P31 wd:Q16510064 ;
    wdt:P585 ?date ;
    wdt:P1346 ?team .
    FILTER(YEAR(?date) = 2022)
    SERVICE wikibase:label { bd:serviceParam wikibase:language "en" }
} LIMIT 10
```

Listing 3: ChatGPT generated query for: How many people do live on earth?

```
SELECT (COUNT(?item) as ?count)
WHERE {
        ?item wdt:P31 wd:Q5.
} LIMIT 10
```

On the contrary, the query for question 2, shown in Listing 2, correctly retrieves (somewhat arguably generalizing on the question) winners of sports events in 2022, which could be correct, but among ChatGPTs chosen LIMIT of 10, only persons and no teams (and certainly not the football worldcup winning team) were retrieved. We note that question 2 has been a common example to "challenge" ChatGPT, where earlier incarnations (such as davinci) would answer "Brazil" as a statistically probable, but wrong answer and ChatGPT would, as mentioned above, refuse an answer, hinting on having only training data up to 2021.

Finally, as for question 8 shown in Listing 2, while the query returned was able to return a number, its result of 10546377 missed the target number of ~ 8 billion people by a significant amount and in fact, in this case interestingly, the resulting query simply counts all items that belong to the class human within Wikidata.

In order to test these results for robustness we executed the whole process a second time. In this second run, the before-mentioned problem of ChatGPT getting stuck at a question did not occur at all. Additionally, ChatGPT was now able to answer 11 out of 15, i.e. one additional question, question 11 regarding "*the oldest painting in the world*" now delivered a result. However, the number of correct answers only increased by one.

Surprisingly, ChatGPT's performance wrt. query generation suffered significantly, with the chatbot on the second still being able to generate 15 queries, but out of which only 10 were syntactically correct. This time, only one of these queries (question 1) returned the desired

Listing 4: davinci generated query for: How many people do live on earth?

```
# defaultView:Map
SELECT ?country (SAMPLE(?population) AS ?totalPopulation)
WHERE {
        ?country wdt:P36 wd:Q458.
        ?country wdt:P1082 ?population.
} GROUP BY ?country
HAVING(?totalPopulation > 0)
LIMIT 10
```

results.
An important observation here is that ChatGPT was unable to generate queries for each question when asked manually through its web interface during an initial tryout of the chatbot. At this point, it must be noted that a new OpenAI account, with no prior interactions with ChatGPT, has been used to generate the respective SPARQL queries using the OpenAI API (in order to ensure no bias was added through personalized adaptation on the user account).

Interestingly, upon using the GPT 3-based model davinci (as ChatGPT's predecessor), we were able to observe structural differences between its results and the ones obtained by using ChatGPT.
First, the NLAs generated by davinci did not contain a disclaimer wrt. questions spanning outside of its training time frame. Also, it becomes obvious that ChatGPT has been trained with more recent data than davinci. While ChatGPT was able to correctly answer the question *For which team does Lionel Messi play?* with *Paris Saint-Germain (PSG)*, davinci answered this question with *Barcelona*. We note that overall davinci produced a significantly larger number of factually wrong answers than ChatGPT, which may be partially due to its *outdated* training data, and partially due to the *smaller* training date leading to more "made up" answers. We do note though, that a detailed investigation (comparing factually wrong vs. outdated answers vs. "accidentally right" guesses) in detail is yet on our agenda.

Next, davinci was unable to fully comply with our limitation of not adding any comments to the generated query, while ChatGPT consistently complied with our instructions. Additionally, 11 out of the 14 queries generated by davinci had some sort of syntactical error, making them not executable. See the query for question 8 *How many people do live on earth?* as an example for both of these phenomena:

While this query can simply be syntactically fixed — by removing the whitespace between the ? and the term population within the second line — it still diverts significantly from the original intention of the asked question.

Lastly, the problem of ChatGPT, i.e., getting stuck at a question during its first run, did not occur when using davinci. We assume that this might be related to the amount of traffic ChatGPT experienced while generating the queries, i.e., rather being related to an API problem than the model itself, since this problem does not seem to be consistent in its occurrence and did not occur when using ChatGPT via its web interface.

Overall, davinci's answers appeared, as expected, a lot more arbitrary and outdated than

Table 2
A sample of simple KGQA questions from the SimpleQuestions benchmark [42]

	Question
1	What county is port hadlock in
2	is roll over and play live a hard rock album or an electronica album
3	where does adewale ojomo get his or her nationality from
4	What position does nenad stojaković play?
5	what kinds of movie is appassionata
6	what label is jeanne cherhal signed to
7	where did alberta gay die
8	what kind of film is esterhazy
9	who was albert brooks's mother
10	is tony asher male or female
11	what is a film in the crime fiction genre.
12	What country was the underworld story filmed in
13	where is just married filmed?
14	What type of tv program is the flintstones
15	What football position does siem de jong play

those of the newer ChatGPT model in the preliminary study of our student dataset.

3.3.2. SimpleQuestions

We again generated SPARQL queries for each of the questions in the sample of the SimpleQuestions dataset, retrieved their results as well as the answers to the NLQs given by ChatGPT and davinci and evaluated them.
Table 2 shows the questions listed in this sample of the SimpleQuestions dataset.[2]

The results for this dataset mostly mirror the observations made for the students dataset. However, this time 6 out of 15 answers generated by ChatGPT are results of the chatbot being stuck at a previous question. Out of the 15 generated queries 14 queries were executable but none of them returned any result.
Again, a second run has been done which led to significantly better results. In this second run ChatGPT did not get stuck on a single question and 10 of the provided answers were correct. Similarly, all of the 15 generated queries were syntactically correct, yet still only three out of the 15 queries returned the correct results, while the query for question 14 *What type of tv program is the flintstones* returned a result completely detached from the questions (commune of Italy) which can however be interpreted as one possible answer to question 13 *where is just married filmed?*, hinting at the ChatGPT API got again stuck during the query generation this time on a previous run.
A possible reason for the comparatively higher success rate observed for this dataset (less emphasis on current or recent data) could be not only its age, with the dataset being unchanged since 2017, but also the general nature of its questions, with many questions being instructions to give an example of something or having many correct possible answers. Take for instance

[2]Please note missing question marks or different capitalization stem from a random sample of the original dataset without any modifications from our side.

Table 3
A sample of simple KGQA questions from the QALD-5 benchmark [43] [3]

	Question
1	Who plays Phileas Fogg in the adaptation of Around the World in 80 Days directed by Buzz Kulik?
2	Who succeeded the pope that reigned only 33 days?
3	Which countries have more than ten caves?
4	Which other weapons did the designer of the Uzi develop?
5	A landmark of which city is the home of the Mona Lisa?
6	Was Margaret Thatcher a chemist?
7	Gaborone is the capital of which country member of the African Union?
8	In which country does the Ganges start?
9	Give me a list of all bandleaders that play trumpet.
10	How many missions does the Soyuz programme have?
11	For which movie did the daughter of Francis Ford Coppola receive an Oscar?
12	Are the Rosetta Stone and the Gayer-Andersen cat exhibited in the same museum?
13	What is the ruling party in Lisbon?
14	When were the Hells Angels founded?
15	Are the Rosetta Stone and the Gayer-Andersen cat exhibited in the same museum?

Listing 5: ChatGPT generated query for: How many missions does the Soyuz programme have?

```
SELECT (COUNT(?mission) AS ?countMissions)
WHERE {
    ?mission wdt:P31 wd:Q209343.
    ?mission wdt:P361 wd:Q127846.
} LIMIT 10
```

question 11 *what is a film in the crime fiction genre*. With numerous crime fiction films existing and their success or popularity not being a limitation stated in the question it can be assumed that ChatGPT should be able to name at least one item fitting the definition of being a film and belonging to the genre crime fiction. From this, we can expect ChatGPT to be able to answer most questions in this dataset correctly that do not ask for elements subject to change, such as the CEO of a company or which player currently plays for a certain team.

3.3.3. QALD-5

Lastly, we analogously again generated SPARQL queries for each of the questions in a sample of the QALD-5 dataset (shown in Table 3), retrieved their results, and analyzed direct answers to the NLQs given by ChatGPT and davinci.

When directly answering the questions in this subset of QALD-5, ChatGPT did not get stuck on any of the 15 questions and was able to answer *9* of the 15 questions correctly. However, while ChatGPT was able to generate syntactically correct queries for all of the 15 questions, the only query returning a result was the query for question 10 *How many missions does the Soyuz programme have?*. Yet, the only element of this query closely related to the actual question is the count function.

[3]Note: the duplicate question 12+15 were discussed in Section 3.1.

3.4. Summary of Results

Summarizing the results of our initial experiments, overall, we admittedly are only at the start of our research. Yet, we have already gained valuable insights into the potential and gaps when trying to leverage LLMs for (factual) question answering, with a focus on questions the answers of which should be retrievable from KGs.

How does LLM-based QA differ from established KGQA approaches and what are the respective strengths, weaknesses, and challenges of the two methods?

While LLMs show promising results wrt. QA it became clear that these models are limited by various factors. Most importantly, both ChatGPT's and davinci's limitations in direct question answering were mostly related to outdated training data, such that they performed particularly well on older QA benchmarks.

Unsurprisingly, the newer ChatGPT model performed significantly better on both the direct question answering tasks and also in particular in terms of the syntactical correctness of queries; we may expect further significant advances in the just released GPT4 model.

Additionally, some unexpected behaviors resulted from inexplicable effects of interacting with the OpenAI APIs', in terms of order-dependent answers that appeared to be actually "stuck" answers to prior queries. Unfortunately, we could not yet determine whether these were related to simple API bugs or due to the model; however really open LLMs would certainly allow investigating order-dependency or alike in a much more transparent manner, than OpenAI's current, closed business model that in fact may switch to a paid only approach.

A summary of both ChatGPT's and davinci's results wrt. direct question answering can be found in Tables 5 and 6.

In terms of query formulation, ChatGPT produced a high share of syntactically correct queries, but very few reflecting the actual question; we do hypothesize that this is largely due to a lack of explicit entity recognition, i.e., recognizing correct relevant IRIS (i.e., in the case of Wikidata relevant Q- and P-identifiers of entities and properties. A more in depth analysis of the resulting queries in terms of *semantic distance* of the extracted identifiers, or investigating in how far LLMS can be used for supporting the entity recognition subtask in isolation as part of a KGQA pipeline is on our agenda.

Especially for the latter point, one has to assume that correct query formulations so far rather stem from verbatim SPARQL examples in the training data for common questions than from an actual understanding of the entities and query structure. We may still assume that the quality of such queries will improve in the future, even now already we encountered hardly any syntactical errors.

A summary of ChatGPT's results wrt. query generation can be found in Table 7.

Which components used in KGQA-systems could be enhanced using LLMs?

While the LLMs in questions showed mediocre results by themselves they potentially inherit the capabilities to improve already existing QA-approaches. We believe that LLMs could provide especially useful in the task of entity recognition which forms part of many existing KGQA-systems. Using LLMs to find synonyms for words occurring in the question asked, extracting the questions underlying meaning, and using them in combination with query

generation templates or by implementing extensive prompt engineering to give the LLMs hints on how to structure their queries.

What types of questions are found in existing benchmarks for KGQA approaches and in how far can these be used in benchmarking LLM-based QA approaches?

While different benchmarks use different categories to categorize their questions some studies provide a holistic categorization of the questions and queries provided in different benchmark datasets. A summary of the used (sub)datasets questions categorized in accordance to CBench's wh-questions classification [1] can be found in Table 4 while Tables 5 and 6 provide a summary of the correctly answered questions by their type.

The results of our study show that LLMs have a particularly hard time answering questions forming some type of count or, resp., asking for *all* entities of a certain category (particularly though, in terms of KGQA also because knowledge in common KGs is typically incomplete). Aside from this, LLMs struggle to answer questions including recent events due to their limited training period.

An additional analysis wrt. to the questions' categories and patterns in the LLMs' results will be conducted in future work.

How can a comprehensive benchmark for LLM-based, KGQA-based but also combined QA approaches be constructed that is challenging the current weaknesses of both approaches?

While it became obvious that a comprehensive benchmark must contain questions aimed at recent/current events, these types of questions are not only harder to fact-check but inherit additional complications due to the resulting need of constant adaption of the benchmark. Additionally, a comprehensive KGQA benchmark must include questions that require the KGQA-system to form some sort of arithmetic or logical linking, such as counting entities related to a word in the asked question, etc.

4. Conclusion

In our preliminary study, we analyzed the performance of two of OpenAI's large language models davinci (GPT 3) and ChatGPT (GPT 3.5) against a set of questions established by students and two subsets of the established benchmark datasets SimpleQuestions and QALD-5. We used both models to answer the questions in each dataset, as well as to generate SPARQL queries aimed at retrieving these answers from the knowledge base Wikidata. Our results demonstrate the limitations of large language models, which mainly lies in their training time frame as well as their stability. Additionally, we show that LLMs are in principle capable of generating functioning queries. While being able to consistently generate structurally and syntactically correct queries, they however demonstrate bad performance wrt. entity detection, resulting in the generated queries not returning the desired results. Therefore, the question remains open how large language models can be used in combination with existing question answering

systems and specifically how existing approaches can be used to substitute the LLM's deficits regarding entity detection. The presented preliminary paper comprises the first results of a recently started master thesis project. Starting from these initial insights, we look forward to discussing routes ahead at the workshop and collecting feedback for our ongoing experiments.

5. Tables

	Student	SimpleQuestions	QALD-5
What	3	9	1
When	1	0	1
Where	1	3	0
Which	2	0	6
Who	4	1	2
Whom	0	0	0
Whose	0	0	0
How	1	0	1
Yes/No	0	0	3
Requests	2	0	1
Topical	0	2	0
Sum	14	15	15

Table 4
Distribution of wh-question types.

	Student	SimpleQuestions	QALD-5
What	1	5	1
When	0	0	1
Where	0	2	0
Which	1	0	4
Who	2	1	1
Whom	0	0	0
Whose	0	0	0
How	0	0	0
Yes/No	0	0	2
Requests	0	0	0
Topical	0	2	0
Sum	4	10	9

Table 5
Correctly answered questions GPT 3.5.

	Student	SimpleQuestions	QALD-5
What	0	4	0
When	0	0	1
Where	0	1	0
Which	1	0	4
Who	1	1	0
Whom	0	0	0
Whose	0	0	0
How	0	0	0
Yes/No	0	0	2
Requests	0	0	0
Topical	0	0	0
Sum	2	6	7

Table 6
Correctly answered questions GPT 3.

	Student	SimpleQuestions	QALD-5
Generated queries	14	15	15
Syntactically correct	10	15	15
Syntactically incorrect	4	0	0
Correct	1	3	0
Incorrect	1	1	1
No answer	8	11	14

Table 7
Results for generated queries GPT 3.5.

References

[1] A. Orogat, I. Liu, A. El-Roby, Cbench: Towards better evaluation of question answering over knowledge graphs, 2021. URL: https://arxiv.org/abs/2105.00811. doi:10.48550/ARXIV.2105.00811.

[2] D. Diefenbach, A. Both, K. Singh, P. Maret, Towards a question answering system over the semantic web, 2018. URL: https://arxiv.org/abs/1803.00832. doi:10.48550/ARXIV.1803.00832.

[3] A. Dhandapani, V. Vadivel, Question answering system over semantic web, IEEE Access 9 (2021) 46900–46910. URL: https://doi.org/10.1109/access.2021.3067942. doi:10.1109/access.2021.3067942.

[4] S. Vakulenko, J. D. F. Garcia, A. Polleres, M. de Rijke, M. Cochez, Message passing for complex question answering over knowledge graphs, in: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, ACM, 2019. URL: https://doi.org/10.1145/3357384.3358026. doi:10.1145/3357384.3358026.

[5] S. Liang, K. Stockinger, T. M. de Farias, M. Anisimova, M. Gil, Querying knowledge graphs in natural language, Journal of Big Data 8 (2021). URL: https://doi.org/10.1186/s40537-020-00383-w. doi:10.1186/s40537-020-00383-w.

[6] H. Cui, T. Peng, L. Feng, T. Bao, L. Liu, Simple question answering over knowledge graph enhanced by question pattern classification, Knowledge and Information Systems 63 (2021) 2741–2761. URL: https://doi.org/10.1007/s10115-021-01609-w. doi:10.1007/s10115-021-01609-w.

[7] M. Yani, A. A. Krisnadhi, I. Budi, A better entity detection of question for knowledge graph question answering through extracting position-based patterns, Journal of Big Data 9 (2022). URL: https://doi.org/10.1186/s40537-022-00631-1. doi:10.1186/s40537-022-00631-1.

[8] M. Zaib, W. E. Zhang, Q. Z. Sheng, A. Mahmood, Y. Zhang, Conversational question answering: a survey, Knowledge and Information Systems 64 (2022) 3151–3195. URL: https://doi.org/10.1007/s10115-022-01744-y. doi:10.1007/s10115-022-01744-y.

[9] C. Mercer, The rise of chat gpt: The future of conversational ai, https://medium.com/@conan.mercer/the-rise-of-chat-gpt-the-future-of-conversational-ai-91622b9db303, 2023. Accessed on March 06, 2023.

[10] S. Garg, ChatGPT alternatives that will blow your mind in 2023 — writesonic.com, https://writesonic.com/blog/chatgpt-alternatives/, 2023. Accessed on March 06, 2023.

[11] L. Jiang, R. Usbeck, Knowledge graph question answering datasets and their generalizability, in: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2022. URL: https://doi.org/10.1145/3477495.3531751. doi:10.1145/3477495.3531751.

[12] F. A. Acheampong, H. Nunoo-Mensah, W. Chen, Transformer models for text-based emotion detection: a review of BERT-based approaches, Artificial Intelligence Review 54 (2021) 5789–5829. URL: https://doi.org/10.1007/s10462-021-09958-2. doi:10.1007/s10462-021-09958-2.

[13] Z. Abbasiantaeb, S. Momtazi, Entity-aware answer sentence selection for question answering with transformer-based language models, Journal of Intelligent Information Systems 59 (2022) 755–777. URL: https://doi.org/10.1007/s10844-022-00724-6. doi:10.1007/

s10844-022-00724-6.

[14] G. Mai, K. Janowicz, L. Cai, R. Zhu, B. Regalia, B. Yan, M. Shi, N. Lao, iSE-KGE/i : A location-aware knowledge graph embedding model for geographic question answering and spatial semantic lifting, Transactions in GIS 24 (2020) 623–655. URL: https://doi.org/10.1111/tgis.12629. doi:10.1111/tgis.12629.

[15] S. Shin, X. Jin, J. Jung, K.-H. Lee, Predicate constraints based question answering over knowledge graph, Information Processing &amp Management 56 (2019) 445–462. URL: https://doi.org/10.1016/j.ipm.2018.12.003. doi:10.1016/j.ipm.2018.12.003.

[16] J. Gomes, R. C. de Mello, V. Ströele, J. F. de Souza, A study of approaches to answering complex questions over knowledge bases, Knowledge and Information Systems 64 (2022) 2849–2881. URL: https://doi.org/10.1007/s10115-022-01737-x. doi:10.1007/s10115-022-01737-x.

[17] P. Do, T. H. V. Phan, Developing a BERT based triple classification model using knowledge graph embedding for question answering system, Applied Intelligence 52 (2021) 636–651. URL: https://doi.org/10.1007/s10489-021-02460-w. doi:10.1007/s10489-021-02460-w.

[18] W. Jin, B. Zhao, H. Yu, X. Tao, R. Yin, G. Liu, Improving embedded knowledge graph multi-hop question answering by introducing relational chain reasoning, Data Mining and Knowledge Discovery 37 (2022) 255–288. URL: https://doi.org/10.1007/s10618-022-00891-8. doi:10.1007/s10618-022-00891-8.

[19] R. Wang, M. Wang, J. Liu, M. Cochez, S. Decker, Structured query construction via knowledge graph embedding, Knowledge and Information Systems 62 (2019) 1819–1846. URL: https://doi.org/10.1007/s10115-019-01401-x. doi:10.1007/s10115-019-01401-x.

[20] U. Sawant, S. Garg, S. Chakrabarti, G. Ramakrishnan, Neural architecture for question answering using a knowledge graph and web corpus, Information Retrieval Journal 22 (2019) 324–349. URL: https://doi.org/10.1007/s10791-018-9348-8. doi:10.1007/s10791-018-9348-8.

[21] H. Jung, W. Kim, Automated conversion from natural language query to SPARQL query, Journal of Intelligent Information Systems 55 (2020) 501–520. URL: https://doi.org/10.1007/s10844-019-00589-2. doi:10.1007/s10844-019-00589-2.

[22] D. Diefenbach, V. Lopez, K. Singh, P. Maret, Core techniques of question answering systems over knowledge bases: a survey, Knowledge and Information Systems 55 (2017) 529–569. URL: https://doi.org/10.1007/s10115-017-1100-y. doi:10.1007/s10115-017-1100-y.

[23] S. Kafle, N. de Silva, D. Dou, An overview of utilizing knowledge bases in neural networks for question answering, Information Systems Frontiers 22 (2020) 1095–1111. URL: https://doi.org/10.1007/s10796-020-10035-2. doi:10.1007/s10796-020-10035-2.

[24] Y.-M. Kim, T.-H. Lee, S.-O. Na, Constructing novel datasets for intent detection and ner in a korean healthcare advice system: guidelines and empirical results, Applied Intelligence 53 (2022) 941–961. URL: https://doi.org/10.1007/s10489-022-03400-y. doi:10.1007/s10489-022-03400-y.

[25] E. Erdem, M. Kuyu, S. Yagcioglu, A. Frank, L. Parcalabescu, B. Plank, A. Babii, O. Turuta, A. Erdem, I. Calixto, E. Lloret, E.-S. Apostol, C.-O. Truică, B. Šandrih, S. Martinčić-Ipšić, G. Berend, A. Gatt, G. Korvel, Neural natural language generation: A survey on multilinguality, multimodality, controllability and learning, Journal of Artificial Intelligence Research 73 (2022) 1131–1207. URL: https://doi.org/10.1613/jair.1.12918.

doi:10.1613/jair.1.12918.
[26] T. Adewumi, F. Liwicki, M. Liwicki, State-of-the-art in open-domain conversational AI: A survey, Information 13 (2022) 298. URL: https://doi.org/10.3390/info13060298. doi:10.3390/info13060298.
[27] D. M. Korngiebel, S. D. Mooney, Considering the possibilities and pitfalls of generative pre-trained transformer 3 (GPT-3) in healthcare delivery, npj Digital Medicine 4 (2021). URL: https://doi.org/10.1038/s41746-021-00464-x. doi:10.1038/s41746-021-00464-x.
[28] Y. Matveev, O. Makhnytkina, P. Posokhov, A. Matveev, S. Skrylnikov, Personalizing hybrid-based dialogue agents, Mathematics 10 (2022) 4657. URL: https://doi.org/10.3390/math10244657. doi:10.3390/math10244657.
[29] Y. Yang, J. Cao, Y. Wen, P. Zhang, Multiturn dialogue generation by modeling sentence-level and discourse-level contexts, Scientific Reports 12 (2022). URL: https://doi.org/10.1038/s41598-022-24787-1. doi:10.1038/s41598-022-24787-1.
[30] Z. Ahmad, A. Ekbal, S. Sengupta, P. Bhattacharyya, Neural response generation for task completion using conversational knowledge graph, PLOS ONE 18 (2023) e0269856. URL: https://doi.org/10.1371/journal.pone.0269856. doi:10.1371/journal.pone.0269856.
[31] D. Jannach, L. Chen, Conversational recommendation: A grand AI challenge, AI Magazine 43 (2022) 151–163. URL: https://doi.org/10.1002/aaai.12059. doi:10.1002/aaai.12059.
[32] S. Huh, Are ChatGPT's knowledge and interpretation ability comparable to those of medical students in korea for taking a parasitology examination?: a descriptive study, Journal of Educational Evaluation for Health Professions 20 (2023) 1. URL: https://doi.org/10.3352/jeehp.2023.20.01. doi:10.3352/jeehp.2023.20.01.
[33] G. Caldarini, S. Jaf, K. McGarry, A literature survey of recent advances in chatbots, Information 13 (2022) 41. URL: https://doi.org/10.3390/info13010041. doi:10.3390/info13010041.
[34] V. Shankar, S. Parsana, An overview and empirical comparison of natural language processing (NLP) models and an introduction to and empirical application of autoencoder models in marketing, Journal of the Academy of Marketing Science 50 (2022) 1324–1350. URL: https://doi.org/10.1007/s11747-022-00840-3. doi:10.1007/s11747-022-00840-3.
[35] N. Alswaidan, M. E. B. Menai, A survey of state-of-the-art approaches for emotion recognition in text, Knowledge and Information Systems 62 (2020) 2937–2987. URL: https://doi.org/10.1007/s10115-020-01449-0. doi:10.1007/s10115-020-01449-0.
[36] S.-E. Kim, Y.-S. Lim, S.-B. Park, Strong influence of responses in training dialogue response generator, Applied Sciences 11 (2021) 7415. URL: https://doi.org/10.3390/app11167415. doi:10.3390/app11167415.
[37] N. Tsinganos, P. Fouliras, I. Mavridis, Applying BERT for early-stage recognition of persistence in chat-based social engineering attacks, Applied Sciences 12 (2022) 12353. URL: https://doi.org/10.3390/app122312353. doi:10.3390/app122312353.
[38] H. Snyder, Literature review as a research methodology: An overview and guidelines, Journal of Business Research 104 (2019) 333–339. URL: https://doi.org/10.1016/j.jbusres.2019.07.039. doi:10.1016/j.jbusres.2019.07.039.
[39] R. Usbeck, X. Yan, A. Perevalov, L. Jiang, J. Schulz, A. Kraft, C. Moeller, J. Huang, J. Reineke, A.-C. N. Ngomo, M. Saleem, A. Both, Semantic Web Journal under submission (2023). URL: https://semantic-web-journal.net/content/qald-10-%E2%80%

94-10th-challengequestion-answering-over-linked-data.
[40] M. Dubey, D. Banerjee, A. Abdelkawi, J. Lehmann, Lc-quad 2.0: A large dataset for complex question answering over wikidata and dbpedia, in: Proceedings of the 18th International Semantic Web Conference (ISWC), Springer, 2019.
[41] P. Trivedi, G. Maheshwari, M. Dubey, J. Lehmann, Lc-quad: A corpus for complex question answering over knowledge graphs, in: Proceedings of the 16th International Semantic Web Conference (ISWC), Springer, 2017, pp. 210–218.
[42] D. Diefenbach, T. P. Tanon, K. D. Singh, P. Maret, Question answering benchmarks for wikidata, in: Proceedings of the ISWC 2017 Posters & Demonstrations and Industry Tracks co-located with 16th International Semantic Web Conference (ISWC 2017), Vienna, Austria, October 23rd - to - 25th, 2017., 2017. URL: http://ceur-ws.org/Vol-1963/paper555.pdf.
[43] C. Unger, C. Forescu, V. Lopez, A.-C. Ngonga Ngomo, E. Cabrio, P. Cimiano, S. Walter, Question answering over linked data (qald-5), 2015.
[44] D. Vrandečić, M. Krötzsch, Wikidata: A free collaborative knowledgebase, Commun. ACM 57 (2014) 78–85. URL: https://doi.org/10.1145/2629489. doi:10.1145/2629489.
[45] V. Lopez, C. Unger, P. Cimiano, E. Motta, Evaluating question answering over linked data, Web Semantics Science Services And Agents On The World Wide Web 21 (2013) 3–13. doi:10.1016/j.websem.2013.05.006.
[46] A. Bordes, N. Usunier, S. Chopra, J. Weston, Large-scale simple question answering with memory networks, 2015. URL: https://arxiv.org/abs/1506.02075. doi:10.48550/ARXIV.1506.02075.
[47] R. F. Jonathan Berant, Andrew Chou, P. Liang, Semantic parsing on freebase from question-answer pairs, Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (2013) 1533–1544.
[48] D063520, Github - wdaquacore0questions, 2017. URL: https://github.com/WDAqua/WDAquaCore0Questions, accessed on February 28, 2023.
[49] S. de Rooij, W. Beek, P. Bloem, F. van Harmelen, S. Schlobach, Are names meaningful? quantifying social meaning on the semantic web, in: P. Groth, E. Simperl, A. J. G. Gray, M. Sabou, M. Krötzsch, F. Lécué, F. Flöck, Y. Gil (Eds.), The Semantic Web - ISWC 2016 - 15th International Semantic Web Conference, Kobe, Japan, October 17-21, 2016, Proceedings, Part I, volume 9981 of *Lecture Notes in Computer Science*, 2016, pp. 184–199. URL: https://doi.org/10.1007/978-3-319-46523-4_12. doi:10.1007/978-3-319-46523-4_12.
[50] R Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, 2022. URL: https://www.R-project.org/.
[51] I. Rudnytskyi, openai: R Wrapper for OpenAI API, 2023. URL: https://CRAN.R-project.org/package=openai, r package version 0.4.0.
[52] M. Popov, WikidataQueryServiceR: API Client Library for 'Wikidata Query Service', 2020. URL: https://CRAN.R-project.org/package=WikidataQueryServiceR, r package version 1.0.0.