GeoQAMap - Geographic Question Answering with Maps Leveraging LLM and Open Knowledge Base

Yu Feng
Chair of Cartography and Visual Analytics, Technical University of Munich, Germany

Linfang Ding
Norwegian University of Science and Technology, Trondheim, Norway

Guohui Xiao
Department of Information Science and Media Studies, University of Bergen, Norway

Abstract

GeoQA (Geographic Question Answering) is an emerging research field in GIScience, aimed at answering geographic questions in natural language. However, developing systems that seamlessly integrate structured geospatial data with unstructured natural language queries remains challenging. Recent advancements in Large Language Models (LLMs) have facilitated the application of natural language processing in various tasks. To achieve this goal, this study introduces GeoQAMap, a system that first translates natural language questions into SPARQL queries, then retrieves geospatial information from Wikidata, and finally generates interactive maps as visual answers. The system exhibits great potential for integration with other geospatial data sources such as OpenStreetMap and CityGML, enabling complicated geographic question answering involving further spatial operations.

2012 ACM Subject Classification Applied computing → Cartography

Keywords and phrases Geographic Question Answering, Large Language Models, SPARQL, Knowledge Base, Wikidata

Digital Object Identifier 10.4230/LIPIcs.GIScience.2023.28

Category Short Paper

Funding Dense and Deep Geographic Virtual Knowledge Graphs for Visual Analysis (DFG Grant 500249124).

1 Motivation

The recent progress in Natural Language Processing (NLP), specifically with Large Language Models (LLMs) has demonstrated significant potential for automating a wide range of tasks. The field of GIScience is actively embracing the utilization of artificial intelligence and seeking to enhance traditional workflows through their integration. Within this context, GeoQA (Geographic Question Answering) has emerged as a prominent research area, focusing on the development of intelligent systems capable of answering questions involving geographic entities or concepts. By leveraging the power of NLP and knowledge graph, GeoQA aims to enable more efficient and effective utilization of geographic information for improved decision-making and problem-solving in various domains.

However, geospatial question answering is challenging, primarily because it involves the integration of structured geospatial data with unstructured natural language queries. Geospatial data typically has a structured format that represents spatial relationships, coordinates, and attributes of geographic entities. On the other hand, natural language queries are unstructured and require understanding and interpretation to extract the relevant geospatial information. Current models are mostly based on text or images. ChatGPT is primarily a text-based model and does not have the capability to directly generate maps. The user would only reply with guidance on how to generate a map using conventional software or

12th International Conference on Geographic Information Science (GIScience 2023).
Editors: Roger Beecham, Jed A. Long, Dianna Smith, Qunshan Zhao, and Sarah Wise; Article No. 28; pp. 28:1–28:7
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

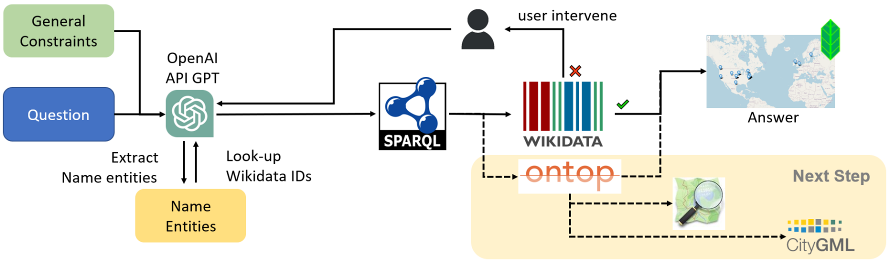


Figure 1 Overview of the proposed GeoQAMap system.

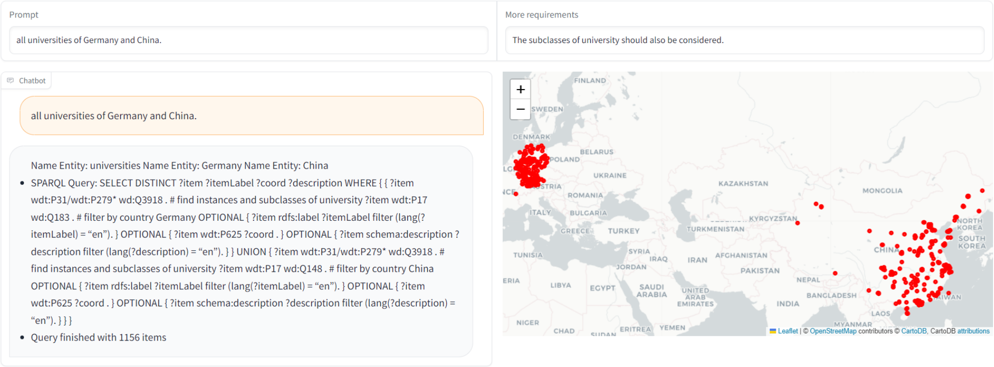
programming languages. On the other hand, there are image-based generative models, such as Midjourney or Stable Diffusion[1], that can generate images containing maps as content. However, it is important to note that these generated map images may not conform to the standard formats and conventions commonly associated with geospatial data [2].

To address this challenge, an intermediary becomes essential to bridge the gap between geospatial data and natural language queries. One potential solution is to utilize SPARQL, a query language specifically designed for querying data stored in the Resource Description Framework (RDF) format. RDF provides a standardized representation for data using subject-predicate-object triples, making it suitable for structured geospatial data. SPARQL is nowadays standard for representation and querying of linked data for semantic web. Furthermore, SPARQL's capabilities have been extended to GeoSPARQL, which incorporates spatial operations, enhancing its utility for handling and analyzing geospatial data.

However, it is worth noting that SPARQL queries often involve complex syntax and rules, making them challenging for end users to grasp and utilize effectively. The intricacies of the language can pose a barrier to entry for individuals who are not familiar with its syntax or who lack technical expertise. To address this challenge, the emerging field of LLMs has provided a promising solution. By leveraging LLMs, natural language queries can be translated into SPARQL queries that can access structured geospatial data stored in RDF format. SPARQL queries can retrieve the relevant information based on the query's spatial constraints, enabling the integration of geospatial data and natural language queries. Since the research leveraging LLM and knowledge system is a rather new research field, there were not yet many applications that demonstrate the ability answering geospatial questions with maps. Only recently, there was one work named *Autonomous GIS* presented by [3]. In their process, the steps of geo-spatial operation need to be clarified to LLM with texts. Corresponding codes in Python would help end-user to achieve their geospatial operations. One limitation of this work is that users are required to upload or download a prepared dataset, which restricts them from leveraging the vast amount of existing open geodata available on the Internet.

Therefore, in this work, we would like to present GeoQAMap, an evolving system designed to answer geospatial questions using maps. It is a further development of GPT-alike AI system. We demonstrated a preliminary example integrating the state-of-the-art LLM and the public knowledge base Wikidata. The system follows a process where questions are first translated into SPARQL queries, which are then queried in a Wikidata endpoint. The output JSON is then utilized in conjunction with the Python library to create interactive maps that provide visual answers to the geographic questions.

[1] Stable Diffusion Online. Source: `https://stablediffusionweb.com/`



Prompt

all universities of Germany and China.

More requirements

The subclasses of university should also be considered.

Chatbot

all universities of Germany and China.

Name Entity: universities Name Entity: Germany Name Entity: China

- SPARQL Query: SELECT DISTINCT ?item ?itemLabel ?coord ?description WHERE { { ?item wdt:P31/wdt:P279* wd:Q3918 . # find instances and subclasses of university ?item wdt:P17 wd:Q183 . # filter by country Germany OPTIONAL { ?item rdfs:label ?itemLabel filter (lang(?itemLabel) = “en”). } OPTIONAL { ?item wdt:P625 ?coord . } OPTIONAL { ?item schema:description ?description filter (lang(?description) = “en”). } } UNION { ?item wdt:P31/wdt:P279* wd:Q3918 . # find instances and subclasses of university ?item wdt:P17 wd:Q148 . # filter by country China OPTIONAL { ?item rdfs:label ?itemLabel filter (lang(?itemLabel) = “en”). } OPTIONAL { ?item wdt:P625 ?coord . } OPTIONAL { ?item schema:description ?description filter (lang(?description) = “en”). } } }
- Query finished with 1156 items

Figure 2 For question “all universities of China and Germany”, the interface contains: prompt input box (upper left), the extracted name entities and generated SPARQL query (bottom left), additional context information (upper right), and the output map (bottom right).

2 Methodology

The entire workflow of the proposed GeoQAMap system is illustrated in Figure 1. There are in general three steps: (1) prompt optimization, (2) prompt interpretation and knowledge base query, and (3) map visualization. Figure 2 shows how this system interact with users.

2.1 Prompt optimization

Formulating the prompt sentence is essential as it directly influences the response generated by the LLM. Additionally, it is crucial to specify the desired output format. The majority of existing LLMs are not readily openly accessible to developers, limiting their ability to retrain or fine-tune the provided models together with the computational resource constraints. Therefore, in this work, frequently happened issues are recorded, and we improve the system performance by giving additional constraints in the prompt sentence.

Considering a user-submitted geospatial question in natural language, we have established several constraints for the output:

(1) Specifically, we require the output to be a pure SPARQL sentence, devoid of any headers or explanatory text that could potentially create issues when interacting with the SPARQL endpoint.

(2) According to our observations, the LLM system would often make mistakes on finding correct Wikidata ID for the corresponding name entities. Therefore, we ask the LLM first to extract name entities from users' prompts, and then look up the corresponding Wikidata ID using the Wikidata API via HTTP requests. (3) Additionally, users are provided with an extra textbox to expand the prompt whenever they encounter incomplete results, allowing them to provide further context to refine their query. As in Figure 2, the extra context input make the LLM to consider the sub-classes (P279) of university in addition to instances (P31) when generating the query.

2.2 Prompt interpretation and query knowledge base

The GeoQAMap utilizes *GPT-3.5* as its underlying LLM, providing access to a range of natural language processing capabilities, including Name Entity Recognition (NER). To interact with *GPT-3.5*, OpenAI API[2] was used. The generated SPARQL query would then be directly given to Wikidata Query Service, where the query is sent to the endpoint server[3] via Python package sparqlwrapper.

Of course, there may be instances where the SPARQL query is not executable on the Wikidata Query Service. In such cases, users may need to manually intervene, for example, by utilizing the Wikidata Query Service web interface to verify the validity of the query sentence. This could involve checking for potential issues such as mismatched entity IDs, mismatched SPARQL syntax or other related issues. Even though this process may occasionally require user interaction, it still significantly reduces the effort compared to constructing complex SPARQL queries from scratch every time.

2.3 Visualization

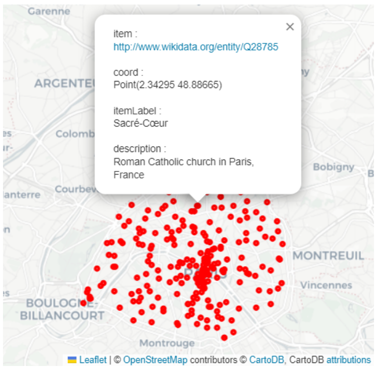
The Wikidata endpoint responses the SPARQL query with data in JSON format. The output text strings are then parsed and visualized with Python package *folium*. Within the standard Jupyter Notebook or Google Colab implemented with *gradio* interface, users can easily interact with this map and explore more details of the results.

3 Results and discussion

In this section, we present three case studies, which demonstrate the questions that GeoQAMap system can already answer with maps.

3.1 Questions for geo-entities of affiliation relationship

The most common type of questions is to search for specific geo-entities that are located within a certain administrative region. This type of query helps in finding relevant information about the relationship between geographical entities and the administrative regions they are associated with. With the first example, we present GeoQAMap's answer to the question "churches in Paris and all districts of Paris". The name entities of this question were first extracted with `church`, `Paris`, and `district`, where the corresponding Wikidata IDs are also given to the prompt sentence. The LLM-generated SPARQL query is in Figure 3.



item :
http://www.wikidata.org/entity/Q28785

coord :
Point(2.34295 48.88665)

itemLabel :
Sacré-Cœur

description :
Roman Catholic church in Paris, France

Leaflet | © OpenStreetMap contributors © CartoDB, CartoDB attributions

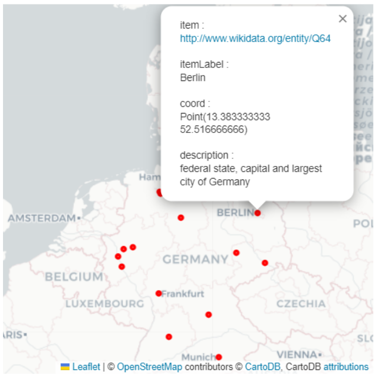
```
SELECT DISTINCT ?item ?itemLabel ?coord ?description
WHERE {
 { ?item wdt:P31 wd:Q16970 ;
        wdt:P131* wd:Q90 ;
        wdt:P625 ?coord . }
 UNION
 { ?item wdt:P31/wdt:P279* wd:Q16970 ;
        wdt:P131 wd:Q90 ;
        wdt:P625 ?coord . }
 ?item rdfs:label ?itemLabel .
 ?item schema:description ?description .
 FILTER(LANG(?itemLabel) = "en") .
 FILTER(LANG(?description) = "en") . }
```

Figure 3 Answering "churches in Paris and all districts of Paris".

[2] GPT - OpenAI API. Source: `https://platform.openai.com/docs/guides/gpt`
[3] SPARQL endpoint server of Wikidata. Source: `https://query.wikidata.org/sparql`

3.2 Questions for geo-entities of attribute conditional filtering

A second frequent type of question is to select geo-entities with respect to certain criteria. With the second example, we would like to present GeoQAMap's answer to the question "cities of Germany with a population more than 500,000". Since many cities are only associated with the subclasses of "city (Q515)", such as "big city (Q1549591)", "Hanseatic city (Q707813)". Therefore, the user would need to declare that "the subclasses of city should also be considered". With the SPARQL generated as following, the answers as map in Figure 4 can be generated.


item :
http://www.wikidata.org/entity/Q64

itemLabel :
Berlin

coord :
Point(13.383333333 52.516666666)

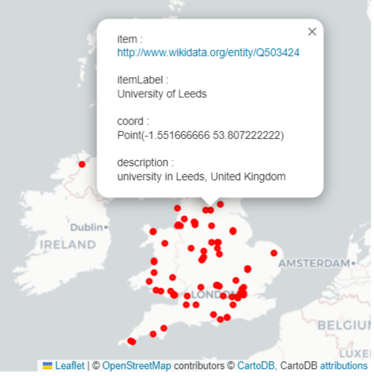
description :
federal state, capital and largest city of Germany

```
SELECT DISTINCT ?item ?itemLabel ?coord ?description
WHERE {
  ?item wdt:P31/wdt:P279* wd:Q515 .
  ?item wdt:P625 ?coord .
  ?item wdt:P17 wd:Q183 .
  ?item wdt:P1082 ?population .
  ?item rdfs:label ?itemLabel .
  ?item schema:description ?description .
  FILTER(LANG(?itemLabel) = "en" &&
   LANG(?description) = "en")
  FILTER(?population > 500000)
} ORDER BY ?population
```

Figure 4 Answering "cities of Germany with a population more than 500,000".

3.3 Questions for geo-entities that need further calculation

Moreover, some questions may need further calculation since the answers are not directly given in the Wikidata knowledge base. For example, a user queries "universities of the United Kingdom established more than 100 years". Only the established time was recorded for universities in the Wikidata under the field of "inception (P571)". However, this does not directly answer the user's question. The LLM-generated SPARQL as in Figure 5 can perform the process of calculation properly. However, similar to the example in Figure 2, it would need to consider the subclasses of university. Therefore, in certain cases, a user may need to intervene and identify the errors to help the LLM to generate correct queries.


item :
http://www.wikidata.org/entity/Q503424

itemLabel :
University of Leeds

coord :
Point(-1.551666666 53.807222222)

description :
university in Leeds, United Kingdom

Dublin
IRELAND
AMSTERDAM
LONDON
BELGIUM
LUXE
Leaflet | © OpenStreetMap contributors © CartoDB, CartoDB attributions

```
SELECT DISTINCT ?item ?itemLabel ?coord ?description
WHERE {
 ?item wdt:P31/wdt:P279* wd:Q3918;
       wdt:P625 ?coord;
       wdt:P17 wd:Q145.
 ?item wdt:P571 ?inception.
 FILTER(YEAR(NOW()) - YEAR(?inception) > 100)
 OPTIONAL {
  ?item schema:description ?description.
  FILTER(LANG(?description) = "en") }
 SERVICE wikibase:label { bd:serviceParam
  wikibase:language "[AUTO_LANGUAGE],en". }
} ORDER BY ?itemLabel
```

Figure 5 Answering "universities of the United Kingdom established more than 100 years".

3.4 Discussion

The system demonstrated in this work can already answer many geographic questions, especially questions such as the locations of geo-entities. Still, complicated questions that require geospatial operations, such as applying a buffer, are not yet achieved.

However, despite the advancements in automatically generating SPARQL queries, there are still several failure cases that often require manual intervention for correction. These failures can be broadly categorized into the following three types, as far as we observed:

(1) Mismatch of named entities and relationships with incorrect Wikidata IDs: Although most cases can be resolved through extracting the named entities and look-up their code in the Wikidata server. It is reasonable to expect that the vast number of concepts and relationships in Wikidata may not be fully covered and learned by the language model.

(2) Syntax errors: The LLM system may occasionally produce syntax errors, such as generating invalid syntax resulting in query inconsistencies. To address these issues, regular expression rules can be established to identify and rectify such syntax errors.

(3) Inconsistencies in Wikidata knowledge base: Within Wikidata, geo-entities can be linked to different geographical entities, including nation names, city names, or city district names. While the general affiliation may be clear, there are instances where the associations are not accurately recorded in the Wikidata knowledge base. This discrepancy can lead to incorrect or incomplete results when querying geospatial information.

To handle these failure cases, manual intervention becomes necessary to identify errors in the generated SPARQL queries and communicate with the LLM to make it remember. Despite these challenges, the automated generation of SPARQL queries by LLMs still greatly reduces the overall effort required to construct complex queries from scratch. It serves as a valuable starting point, with manual correction acting as a backup step to refine and ensure the accuracy of the queries.

4 Conclusions and outlook

In this work, we presented our early implementation of GeoQAMap, a system that has been built on the current state-of-the-art LLM and open knowledge base to answer geospatial questions using maps. Many geospatial questions can be answered with an interactive map visualization and it allows users to explore details of individual geo-entities.

There are several aspects that require further exploration. Firstly, since there are still many cases that the LLM would generate incorrect SPARQL queries, it is important to comprehensively evaluate the performance of the current LLM models for this certain task and design proper strategies to ensure the correctness of the generated queries. Secondly, the implementation would benefit from the inclusion of a filter mechanism that determines which questions specifically require answers using maps and which associated geo-entities are in need of visualization for the user. Lastly, at present, the system's capabilities are limited to query-based questions, and the depth and breadth of its ability to answer complicated questions would require significant enhancements.

Furthermore, as illustrated in Figure 1 and highlighted in orange, we aim to leverage the capabilities of Virtual Knowledge Graph (VKG) technology to include more geospatial data into the process, e.g., OpenStreetMap and CityGML, in order to achieve geo-analytical question answering [4]. By combining Ontop[4] and GeoSPARQL, Ding et al. (2021) [1]

[4] Ontop - A Virtual Knowledge Graph System. Source: `https://ontop-vkg.org/`

demonstrated the ability to answer questions involving geospatial operations, such as buffering. The LLM can therefore act as a crucial entry point, allowing users to pose complex geospatial questions using natural language.

References

1 Linfang Ding, Guohui Xiao, Albulen Pano, Claus Stadler, and Diego Calvanese. Towards the next generation of the linkedgeodata project using virtual knowledge graphs. *Journal of Web Semantics*, 71:100662, 2021.

2 Yuhao Kang, Qianheng Zhang, and Robert Roth. The ethics of ai-generated maps: A study of dalle 2 and implications for cartography. *arXiv preprint arXiv:2304.10743*, 2023.

3 Zhenlong Li and Huan Ning. Autonomous gis: the next-generation ai-powered gis. *arXiv preprint arXiv:2305.06453*, 2023.

4 Simon Scheider, Enkhbold Nyamsuren, Han Kruiger, and Haiqi Xu. Geo-analytical question-answering with gis. *International Journal of Digital Earth*, 14(1):1–14, 2021.