Application of Generative AI as an Enterprise Wikibase Knowledge Graph Q&A System

Renê de Ávila Mendes and **Dimas Jackson de Oliveira** and **Victor Hugo Fiuza Garcia**
Mackenzie Presbyterian Institute
GERTI - Department of Technology and Innovation
Rua da Consolação, 930 - São Paulo - SP - Brazil
rene.mendes@mackenzie.br dimas.oliveira@terceiros.mackenzie.br victor.garcia@mackenzie.br

Abstract

Generative AI and Large Language Models are increasingly used in business contexts. One application involves natural language conversations contextualized by company data, which can be accomplished by Enterprise Knowledge Graphs, standardized representations of data. This paper outlines an architecture for implementation of an Enterprise Knowledge Graph using open-source Wikibase software. Additionally, it is presented a Knowledge Graph Q&A System powered by Generative AI.

1 Introduction

Knowledge Graphs (KG) are semantic networks that represent information in a graph structure, with entities as nodes and relationships as edges (Heist et al., 2020), built from diverse data to integrate and organize knowledge (Paulheim, 2016). They can be applied in areas such as the labor market (Popping, 2003), education methods (cao, 2023), and medicine (Vidal Rolim et al., 2021), and are valued in Artificial Intelligence (AI) for their clarity and flexibility (Shen et al., 2022). An example is the combination of KG with AI technologies, such as Microsoft's Azure OpenAI, which further enhances their potential by facilitating the integration and analysis of large volumes of data more efficiently and accurately (Sarica et al., 2020).

In this context, the use of Wikibase to create KG offers significant advantages. Wikibase allows the integration of heterogeneous data, flexible data schema modeling, and collaborative knowledge curation, enabling the construction of comprehensive and up-to-date graphs for applications such as recommendation, analysis, and research (Sarica et al., 2020). In Brazil, KG have driven advancements in areas such as smart cities and healthcare (Vidal Rolim et al., 2021; bel, 2023). Despite the challenges, research is exploring their potential, such as the Brazilian Legislation and Brazilian History KG (de Paiva and Rademaker, 2024; Navas-Loro et al., 2022).

Large Language Models (LLMs) are reshaping the way humans interact with machines, specially through Generative AI applications. Known for their immense scale and intricate architecture, LLMs have transformed the field of natural language processing. These models undergo rigorous stages, including data gathering, preprocessing, model selection, training, and fine-tuning, all aimed at achieving peak performance (Linkon et al., 2024). Presently, experts are exploring the Gen AI capacity to redefine a company's valuation and improving its cost structure, which can fully impact several business in the future (Scappaticci, 2023). Although the progress in these models is promising, they do come with limitations. Large language models struggle to expand or modify their memory, lack transparency in their predictions, and may even generate "hallucinations." However, models that blend training data with company data (retrieval-based) can mitigate some of these challenges (Lewis et al., 2020a). This technique is called Retrieval-Augmented Generation (RAG) and allows expansion of knowledge, as well as inspection and interpretation of accessed information (Lewis et al., 2020b).

Mackenzie Presbyterian Institute (IPM) maintains one of the oldest institutions of education in Brazil, founded in 1870. Its structure encompasses a University, with campuses in 6 Brazilian cities and with about 37,000 students enrolled, a School with about 9,000 students enrolled, and two Hospitals, which together provide more than 2.3 million health care encounters and procedures in 2022 (Mackenzie Presbyterian Institute, 2022).

In a corporate context such as IPM, data integration was being addressed with the use of datawarehouses and datalakes, trying to offer a 360-degree view of the customer and allow the board to have an integrated view of the company. However, the

Proceedings of the 1st Workshop on Knowledge Graphs and Large Language Models (KaLLM 2024), pages 35–42
August 15, 2024

growing understanding of student interactions, patients and the various services offered by IPM seems to be naturally represented by a graph and well documented through business glossaries and ontologies, suggesting that Knowledge Graphs can offer a more complete and integrated experience of IPM data (Martin et al., 2021; Blumauer and Nagy, 2020).

The possibility of integrating Large Language Models (LLM) with Knowledge Graphs (KG) opened a new horizon for offering data to IPM end users: the possibility of asking questions about the integrated data and being answered in natural language. But how complex would it be to implement this solution for IPM? And what would be the results of the interaction of an LLM with IPM data? These are some of the questions we need to answer.

In this article we will detail a RAG Q&A system that accesses data from an Enterprise Knowledge Graph based on Wikibase, created to integrate company data. For the experiments, the graph was loaded with synthetically generated students and patients data, thus preserving the identity and privacy of both patients and students.

This article is structured as follows: Section 2 depicts the entire solution of the Knowledge Graph, including its ontologies, its data, and its architecture; Section 3 discuss the tests and results obtained when adopting LLM to build the Q&A system; Section 4 concludes the article and presents contributions, limitations, and future improvement opportunities.

2 Knowledge Graph

The KG that is being built for our company can be defined as an Enterprise Knowledge Graph (EKG), as it is restricted to corporate use and is applied to commercial use cases. The objectives for building a KG in our context include: gain insights into students' relationships with courses, teachers, subjects and content, as well as patients' relationships with treatments, medications and medical procedures; integrate data from different sources; build the foundation of what will become a semantic data catalog, and build the foundation that supports data analysis.

An important concern for our company was to provision an EKG that could demonstrate its data integration potential in the shortest possible time and in a performant manner, or, as in the words of Blumauer and Nagy (2020), *"deliver the right data in the right format in a timely and high-performance manner"*. In this sense, Wikibase proved to be more advantageous compared to classic RDF KG solutions, as it offers out-of-the-box services (Diefenbach et al., 2021). In just a few days of work, we had a fully operational sandbox environment provisioned on Docker containers, including a SPARQL endpoint, a full-text search solution for concepts and attributes, and a graphical interface for SPARQL queries. The fact that Wikibase is a solution for Open Knowledge Graphs (OKG) is still a concern for us because Wikidata does not allow restricting data access according to user groups and, in a corporate environment, users should only access data relating to their activities. At least intuitively this concern can be addressed by using extensions (Kapica, 2023) or even developing one.

In the next sections we will discuss the KG components, including the ontologies, data and technologies adopted in its construction.

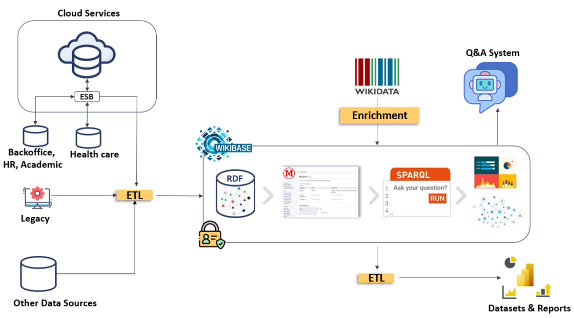

Figure 1: System architecture, highlighting the data sources, the Wikibase components, the enrichment interface, the outputs of datasets and reports and the interface with Q&A system.

2.1 Technologies

The Figure 1 shows the system architecture. Instances of concepts defined in ontologies (Subsection 2.2) are extracted from internal data sources through ETL workflows and loaded into the KG with the support of both OpenRefine[1] and QuickStaments[2] tools. For natural language processing tests synthetically generated data were loaded into KG (Subsection 2.3).

Once loaded to the KG, instances of concepts, called "items" in Wikibase, are available to be

[1]https://openrefine.org

[2]https://www.wikidata.org/wiki/Help:QuickStatements

queried by a SPARQL endpoint or by a data visualization interface called Query Service. Items can be edited or even enriched through data loads from Wikidata[3] or even through references to items defined in Wikidata. The concepts defined in the KG make up a data glossary, and instances of these concepts can be offered to data consumers through datasets or reports fed directly into Microsoft Power BI workspaces, or as a data source for the intelligent natural language processing system (Subsection 3.1).

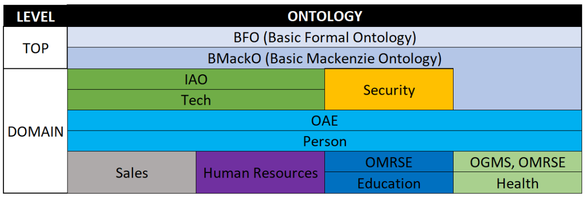

Figure 2: Ontology definition levels, organized by top and domain levels.

2.2 Ontologies

The concepts used in the KG were defined in ontologies, which are formal representations of terms in a given domain (Hogan et al., 2021). In ontologies, concepts are defined through classes, which are collections of objects, and the characteristics of concepts are represented by attributes. Interactions between classes are represented by special types of attributes: relations. Individuals in an ontology are represented as instances (Sack and Alam, 2020).

The definition of the concepts used in the KG was based on the Basic Formal Ontology (BFO) (Smith et al., 2020), an ontology that defines general terms common to all knowledge domains, that is, a top-level ontology. Under the BFO is the Basic Mackenzie Ontology (BMackO), a proprietary top-level ontology dedicated to the definition of terms and attributes common to all other ontologies adopted by IPM. The concepts relating to the domains covered by the KG were defined in either proprietary or public ontologies, the latter located using the Ontobee (Xiang et al., 2011) tool. All domain ontologies adopted in the KG (Tech, Security, Person, Sales, Human Resources, Education and Health), extend the BFO ontology. The complete hierarchy of ontologies adopted in KG is depicted in Figure 2.

The Tech and Security ontologies aim to define the concepts and properties that must be applied to all other domain ontologies. For example, in the Security ontology, the attributes "is personally identifiable information" and "is sensitive information" were defined. These two attributes are applicable to attributes of ontologies that are below the Security ontology (Figure 2). The proprietary ontologies Tech, Person, Education, and Health extend the following publicly shared ontologies: Information Artifact Ontology (IAO) (Ruttenberg et al., 2022), Ontology for General Medical Science (OGMS) (Zheng et al., 2009), Ontology for Modeling and Representation of Social Entities (OMRSE) (Brochhausen et al., 2024) and Ontology of Adverse Events (OAE) (Smith et al., 2022).

2.3 Gen AI Synthetic Data Creation

In Section 3 we will present a Knowledge Graph Q&A System that we built to allow natural language quering. Since this app is a prototype, we choose the free version of Gemini Pro 1.0 as the best option to run tests. With the aim of avoiding the leakage of sensible information, we connected the LLM only to a development knowledge base, filled with synthetic data. To generate high quality synthetic data similar to real data we used GPT-3 providing the fields and data types. An example of prompt used can be found in the Appendix B. We generated academic and customer/lead data, similar to data retrieved in our Customer Relationship Management (CRM) system.

3 Large Language Models

We will also discuss the tests and results obtained when adopting LLM to build a KG questions and answers system.

3.1 Knowledge Graph Q&A System

With the KG implementation, we looked for an intelligent natural language processing system that could understand and respond to user queries in a conversational manner. The goal was to improve the accessibility of information stored in the private Wikibase instance repository, making it easier for users to retrieve relevant data through natural language interactions even for stakeholders who do not dominate SPARQL query language.

To build the Knowledge Graph Q&A system, we started from the preliminary work on the GitHub repository Langchain Wikibase (Ziff, 2024), with proposes the use of a Langchain autonomous Reasoning-Action (or Re-Act for short) Agent to

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

retrieve Wikidata information via API and answer questions. Re-Act Agents can reason about what kind of tools must be called and how to handle the tool output. In this case, the tools provided to the Re-Act agent was python functions to get properties and items information from Wikidata API. The agent looks for properties PID and items QID from Wikidata (Wikimedia Foundation, 2009–) then convert the user input question into a SPARQL query and run it in Wikidata SPARQL endpoint. The agent also generate an human readable answer from the SPARQL query result. The original repository was a simple python script to be run via CLI passing the question as argument.

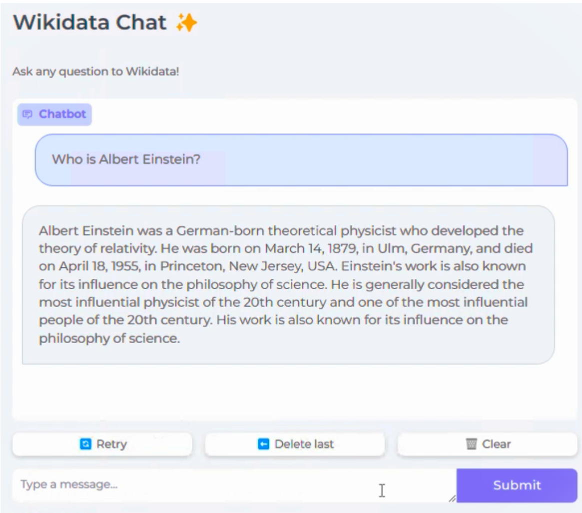

Figure 3: The user interface developed to the generative AI powered chat, built with Gradio.

We contributed to the GitHub repository Langchain Wikibase (Ziff, 2024) adding a chain tool to retrieve all properties of a given item from Wikidata rest API, enriching the Re-Act Agent app. Also, we developed a python module to wrap up the tools and customized some Lang Chain packages to use with local Wikibase instances instead of Wikidata. Furthermore, we improved the prompts and make use of the chain to properly answer descriptive questions such that “Who is Student A?” or “What is Pernambuco Federal University?” using the tool that we implemented. The application was originally designed to run over Open AI GPT models, we also enabled connection to other language models like Google Gemini 1.0 Pro and the open source model Mixtral 8×7B from Mistral AI (Jiang et al., 2024). Moreover, we developed a simple chatbot user interface by the Python library Gradio, which can be seen in Figure 3. All this development and improvements was already committed and merged to the original GitHub repository (Ziff, 2024).

The generative AI chatbot powered by Gemini 1.0 Pro was connected to the OKG database Wikidata and to our EKG database developed over a Wikibase (Wikimedia Deutschland, 2012–) instance. The same structure was used through both cases, reasoning about the question, running queries over the SPARQL endpoint and translating it into human readable answers. The chatbot was tested using three kinds of questions: type 1) descriptive questions e.g. “Who is Albert Einstein?”, “What is Google’s industry?”, “What is Google?”; type 2) questions that links a property to an item e.g. “What is the Google inception?”, “List 10 subsidiaries of Google.”,“What are the geographic coordinates of Mount Everest?”; and type 3) questions involving calculations e.g. “What is the average GDP *per capita* of the Africa continent countries?”, “What is the sum of the population of USA, Canada and Mexico?”, “What is the sum of the number of countries in South America and North America?”.

The percentage of correct answered questions for each type can be found in Figure 4. The criteria used to determine if the Re-Act Agent correctly answered a question was human evaluation, comparing the generated answer with data available in Wikidata or our EKG and the generated SPARQL queries with human written queries. To monitor the app and inspect costs we used the developer framework Langsmith, which allows an end-to-end track of the LLM-powered application lifecycle. Appendix A shows the reasoning process of the Langchain Re-Act Agent to answer a question, using Python functions as tools to interact with the KG.

4 Conclusion

Even at the prototype stage, the Knowledge Graph Q&A System demonstrated a good hit rate, specially in simple questions. As illustrated in Figure 4, the Re-Act agent performed significantly better with type 1 questions when connected to our private Knowledge Graph, yielding 31% more correct answers. This improvement can be attributed to our Enterprise Knowledge Graph (EKG) being restricted to subjects of interest for our company, as opposed to Wikidata, which covers a wide range of topics. The discrepancy shown in Figure 4 between

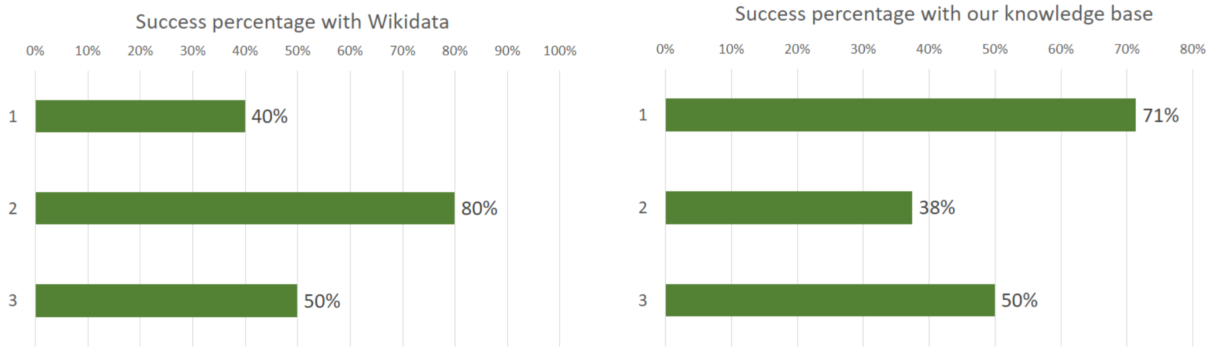

Figure 4: Comparison of KG Q&A System connected to Wikidata and our EKG. The type 1 are descriptive questions, type 2 questions that links an item to a property and type 3 questions involving calculations.

local EKG and Wikidata for type 2 questions are possibly due to training data about the Wikidata, since several PIDs and QIDs are known by Gemini 1.0 pro. Occasionally, the answers generated by the Re-Act agent connected to our private EKG contain some Wikidata properties, resulting in empty results from our SPARQL endpoint. Using a more robust LLM like Gemini 1.5 pro or GPT-4 and fine tuning the model to better generate SPARQL may fix this problem and improve the success rate. When deploying the chatbot to production, this will be our approach. Also, when using premium LLM versions, the providers guarantee that no private data are used to train models, making them suitable to a production version.

On privacy, an opportunity to improve the KG is the segregation of access to items based on the authenticated user's profile, and the reproduction of this segregation of access for the SPARQL queries submitted to the SPARQL endpoint and to the Wikibase Query Service (WDQS).

The results obtained so far have been favorable to the adoption of KGs as a solution for data integration and LLM for the construction of a search and response interface in natural language for IPM's corporate data. While adopting an Open Knowledge Graph solution like Wikibase in an enterprise environment presents the challenge of segregating data access, the gains from Wikidata Query Service's out-of-the-box data visualization options and simplicity of horizontal scaling of Wikibase's docker swarm implementation are self-demonstrable.

Acknowledgments

We would like to thank the Mackenzie Presbyterian Institute and the Department of Technology and Innovation for supporting the development of KG and the publication of this article. We would like to thank professor Donald Ziff for starting the project that originated our Q&A system.

References

2023. *Exploration on the application of knowledge graph in modern chinese teaching*. 2023 International Seminar on Computer Science and Engineering Technology (SCSET). IEEE.

2023. *Inovação de serviços em cidades inteligentes: Interação de pessoas não-especialistas com knowledge graphs*.

Andreas Blumauer and Helmut Nagy. 2020. *The knowledge graph cookbook: Recipes that work*. edition mono/monochrom.

Mathias Brochhausen, William Hogan, Amanda Hicks, Shariq Tariq, and Swetha Garimalla. 2024. Ontology for modeling and representation of social entities. Accessed: May 15, 2024.

Valeria de Paiva and Alexandre Rademaker. 2024. Towards a brazilian history knowledge graph.

Dennis Diefenbach, Max De Wilde, and Samantha Alipio. 2021. Wikibase as an infrastructure for knowledge graphs: The eu knowledge graph. In *The Semantic Web–ISWC 2021: 20th International Semantic Web Conference, ISWC 2021, Virtual Event, October 24–28, 2021, Proceedings 20*, pages 631–647. Springer.

Nicolas Heist, S Hertling, Daniel Ringler, and Heiko Paulheim. 2020. Knowledge graphs on the web - an overview. *ArXiv*, abs/2003.00719.

Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutiérrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan F. Sequeda, Steffen Staab, and Antoine Zimmermann. 2021. *Knowledge Graphs*. Number 22 in Synthesis Lectures on Data, Semantics, and Knowledge. Springer.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts. *Preprint*, arXiv:2401.04088.

Aleš Kapica. 2023. Extension access control. Accessed: May 16, 2024.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020a. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020b. Retrieval-augmented generation for knowledge-intensive NLP tasks. *CoRR*, abs/2005.11401.

Ahmed Ali Linkon, Mujiba Shaima, Md Shohail Uddin Sarker, Norun Nabi, Md Nasir Uddin Rana, Sandip Kumar Ghosh, Mohammad Anisur Rahman, Hammed Esa, Faiaz Rahat Chowdhury, et al. 2024. Advancements and applications of generative artificial intelligence and large language models on business management: A comprehensive review. *Journal of Computer Science and Technology Studies*, 6(1):225–232.

Mackenzie Presbyterian Institute. 2022. Instituto presbiteriano mackenzie - relatório anual - institucional & sustentabilidade 2022. Accessed: May 15, 2024.

Sean Martin, Ben Szekely, and Dean Allemang. 2021. *The Rise of the Knowledge Graph: Toward Modern Data Integration and the Data Fabric Architecture*. O'Reilly Media, Incorporated.

María Navas-Loro, Carlos Badenes-Olmedo, Manolis Koubarakis, Luis Redondo, Sabrina Kirrane, Nandana Mihindukulasooriya, Ken Satoh, and Maribel Acosta, editors. 2022. *Building and analyzing the brazilian legal knowledge graph*, volume 3257. CEUR-WS.org.

Heiko Paulheim. 2016. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic Web*, 8:489–508.

Roel Popping. 2003. Knowledge graphs and network text analysis. *Social Science Information*, 42:106 – 91.

Alan Ruttenberg, Barry Smith, Bjoern Peters, Carlo Torniai, Chris Mungall, Chris Stoeckert, Holger Stenzhorn, James A. Overton, James Malone, Jennifer Fostel, Jie Zheng, Larisa Soldatova, Lawrence Hunter, Mathias Brochhausen, Melanie Courtot, Philippe Rocca-Serra, David Osumi-Sutherland, William Hogan, Adam Goldstein, Albert Goldfain, Christian A. Boelling, Darren Natale, Gwen Frishkoff, Jonathan Rees, Matt Brush, Michel Dumontier, Paolo Ciccarese, Pat Hayes, Randy Dipert, Ron Rudnicki, Satya Sahoo, Sivaram Arabandi, Werner Ceusters, William Duncan, Yongqun He, and Clint Dowland. 2022. Information artifact ontology. Accessed: May 15, 2024.

Harald Sack and Mehwish Alam. 2020. Knowledge graphs 2020.

Serhad Sarica, Jianxi Luo, and Kristin L Wood. 2020. Technet: Technology semantic network based on patent data. *Expert Systems with Applications*, 142:112995.

Chiara Silveira Scappaticci. 2023. *Artificial intelligence: how can Gen-AI tools support the current business models of the firms and add value?* Ph.D. thesis.

Tong Shen, Fu Zhang, and Jingwei Cheng. 2022. A comprehensive overview of knowledge graph completion. *Knowledge-Based Systems*, 255:109597.

Barry Smith, Alan Ruttenberg, and John Beverley. 2020. Basic formal ontology. Accessed: May 15, 2024.

Barry Smith, Jiangan Xie, Yu Lin, Abra Guo, Bingjian Yang, Desikan Jagannathan, Edison Ong, Kelly Yang, Kevin Mo, Liwei Wang, Meiu Wong, Noemi Garg, Qingping Liu, Rebecca Racz, Shelley Zhang, Sirarat Sarntivijai, Sydni Joubran, Yongqun He, Zuoshuang Xiang, Ling Wan, David Ameriguian, Jessica DeGuise, Jiangan Xie, and Qiuyue Yang. 2022. Ontology of adverse events. Accessed: May 15, 2024.

Tulio Vidal Rolim, Caio Viktor S. Avila, Narciso Arruda, José Wellington F. da Silva, José Gilvan R. Maia, Mauro Oliveira, Luiz Odorico M. Andrade, and Vânia M. P. Vidal. 2021. *Um Enfoque Incremental para Construção do Grafo de Conhecimento do SUS*, pages 327–342. Tecnologia da Informação e Comunicação: Pesquisas em Inovações Tecnológicas. Editora Científica Digital.

Wikimedia Deutschland. 2012–. Wikibase. Accessed: May 15, 2024.

Wikimedia Foundation. 2009–. Wikidata. Accessed: May 15, 2024.

Zuoshuang Xiang, Chris Mungall, Alan Ruttenberg, and Yongqun He. 2011. Ontobee: A linked data server and browser for ontology terms. In *ICBO*.

Jie Zheng, Lindsay Cowell, Barry Smith, Alan Ruttenberg, Albert Goldfain, Werner Ceusters, Anand Kumar, Bill Hogan, Cornelius Rosse, Daniel Merico, Richard Scheuermann, Sivaram Arabandi, and Brian Aevermann. 2009. Ontology for general medical science. Accessed: May 15, 2024.

Donald Ziff. 2024. GitHub - donaldziff/langchain-wikibase — github.com. https://github.com/donaldziff/langchain-wikibase. [Accessed 14-05-2024].

A Langchain Agent-Chain Reasoning and Answering

Example of the reasoning process of the Knowledge Graph Q&A System to consult graph, generate SPARQL and answer a question:

Question: What is the population of the city of São Paulo?

Answer:

1. Find the Q item for "city of São Paulo" using the getQItem tool
Action: getQItem
Action Input: city of São Paulo
Q174

2. Find P properties for "population" using the getProperty tool
Action: getProperty
Action Input: population
P1082

3. Generate the simplest possible SPARQL query and check its validity with the checkSparql tool
Action: checkSparql
Action Input:

```
SELECT ?population WHERE {
   wd:Q174 wdt:P1082 ?population.
}
```

Query is valid

4. Run the SPARQL query using the runSparql tool and retrieve the result
Action: runSparql
Action Input:

```
SELECT ?population WHERE {
   wd:Q174 wdt:P1082 ?population.
}
```

result:


```
{
    'head': {
        'vars': ['population']
    },
    'results': {
        'bindings': [{
            'population': {
                'datatype':
'http://www.w3.org/2001/XMLSchema#decimal',
                'type': 'literal',
                'value': '11451245'
            }
        }]
    }
}
```


5. Summarize JSON results in natural language to generate a human-readable response

Final Answer:
The population of the city of São Paulo is approximately 11,451,245 inhabitants.

B Generating Synthetic Data through LLM

This is an example of prompt used to generate student synthetic data, similar to data from the academic Data Warehouse:

```
Create a table of synthetic student
data, with all fields filled in as
per the instructions below. The table
columns must be all of the options below
in </columns>:

<columns>
registration
status
isActive
person.code
person.name
person.socialName
person.contact.telephone
person.contact.branchLine
person.contact.email
person.contact.businessEmail
person.contact.businessTelephone
person.contact.businessBranchLine
entity.code
entity.name
subsidiary.code
subsidiary.name
educationLevel.code
educationLevel.name
school.code
school.name
courses.code
```

course.name
sourceSystem
</columns>

registration is a 7-digit identifier for each student and the person code is a 5-digit string. The status must be one of the options below between </status>:

<status>
'Inactive'
'Active'
'Canceled'
</status>
The isActive field can be true or false The entity.code must be '1' for all lines The entity.name must be 'Entity A' for all rows The subsidiary.code and subsidiary.name must be one of the options below between </sub>

<sub>
1,Subsidiary A
2,Subsidiary B
3,Subsidiary C
</sub>
The educationLevel.code and educationLevel.name must be one of the options below between </edu>:

<edu>
10,High School
11,Undergraduate
12,Post Graduation
</edu>
The school.code and school.name must be one of the options below between </sch>:

<sch>
20,School A
21,School B
22,School C
</sch>
course.code and course.name must be one of the options below between </course>

<course>
A0010,Program A
A0020,Program B
A0030,Program C
</course>

sourceSystem must be 'System A' on all lines. In each example, the code and name are separated by a comma, always in the same order. Adjust the choice of courses, school and teaching level according to similarity. Generate the table in CSV format. Don't generate code, write the table rows. The table must have 200 rows, do not stop until you complete the table. Do not repeat people's names. Fill in all fields and lines with values as per the instructions above. Follow all the rules.