

Exploring the Potential of Artificial Intelligence based Chatbots for Generating Federated SPARQL Queries over Bioinformatics Knowledge Graphs^{*}

Sourav Maiti^{1,*†}, Qurratal Ain Fatimah², Syeda Mah-e-Fatima² and Ali Hasnain^{1,*†}

¹*School of Pharmacy and Biomedical Sciences, Royal College of Surgeons in Ireland, Dublin*

²*University Hospital Galway, Ireland*

Abstract

In this paper we investigate the efficacy of five AI bots - ChatGPT, Gemini, Copilot, Chatsonic, and YouChat - in formulating simple and complex federated SPARQL queries across Dbpedia, DrugBank, and KEGG databases. Through comparison with manually created queries, we unveil the bots' capabilities and limitations. Our findings highlight the potential of AI in data science and healthcare research, offering insights into cross-domain query generation and its implications for interdisciplinary collaboration.

Keywords

AI Chatbots, Federated queries, SPARQL, Healthcare and Life Science Datasets

1. Introduction

The deluge of data in biological databases offers a diverse range of information in the healthcare and life sciences domain. These databases provide opportunities for researchers, scientists and working professionals to accelerate discoveries, develop new hypotheses and identify novel patterns[1]. On the other hand, these databases need implementation of sophisticated storage and retrieval systems to retrieve information from these large databases. This becomes a challenge for researchers and scientists[2]. Most biological databases published as RDF Knowledge Graphs rely on complex query languages like SPARQL (SPARQL Protocol and RDF Query Language)[3] to retrieve information from databases. With no or limited technical knowledge, researchers and domain users are unable to write accurate and reliable SPARQL queries, which could become a bottleneck to exploit the full potential of these databases[3][1]. SPARQL is a query language which enables users and provides a standardised way to query information from databases[4][3]. Many biological databases leverage the RDF (Resource Description Framework) data model, where RDF represents the information as interconnected triples (subject, predicate, object) suitable for complex biological relationships like protein functions, gene interactions [2][4]. The RDF data is made available via SPARQL endpoints and SPARQL query language was specifically designed to query RDF data, allowing for efficient

SeWebMeDA-2024: 7th International Workshop on Semantic Web Solutions for Large-scale Biomedical Data Analytics, May 26, 2024, Hersonissos, Greece

*Corresponding author.

†These authors contributed equally.

✉ souravmaiti@rcsi.ie (S. Maiti); alihasnain@rcsi.com (A. Hasnain)

🌐 <https://www.rcsi.com/people/profile/alihasnain> (A. Hasnain)

🆔 0000-0003-4014-4394 (A. Hasnain)

© 2024 Copyright © 2024 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

retrieval of information[4]. Since many databases use the RDF standard, SPARQL queries can be used to access and query different databases helping researchers to integrate data from various databases [3][2]. Similar to other querying languages, SPARQL also has a learning curve, and, for non-technical biological researchers and scientists with limited experience, it could pose a limitation to access data for their research while generating complex queries.

As aforementioned, the Healthcare and Life Sciences Knowledge Graphs are generally available across multiple locations and also sometimes in different formats. In order to search, query and integrate the data coming from different sources of different formats, it sometimes become challenging for working biologists without any technical knowledge or experience. Moreover, creating federated queries that can retrieve information from multiple databases and combine the information needs technical knowledge about RDF and SPARQL[2].

Furthermore, after the data has been extracted it requires further processing and without coding skills it is a challenge for researchers to interpret and explain the results accurately and consistently, potentially leading to misinterpretations[5]. Many of these databases put more focus on developing new functionalities and increasing the data volume rather than improving UI (User Interface). This limits the user base for these databases and are only usable and interpret-able by researcher with understanding of RDF and SPARQL [6].

Significant work has already been done to facilitate domain users and working biologist to either formulate complex queries using user interfaces [7, 8] or with the help of visualisations [9, 10]. Similarly, Hasnain et, al [11, 12] presented a service named SPORTAL, which is a system that collects meta-data about the content of endpoints and collects them into a central catalogue over which clients can search. This service focuses on the problem of helping clients to find relevant SPARQL endpoints over the Web.

Although these man-made services designed to help researcher to access, search and query relevant RDF database but with the advent of generative AI based tools and chatbots, there is a surge of accessing these services for nearly every walk of life. These chatbots provides an opportunity to ask question in natural language, although this current generation of these bots have limitations that could generate errors and bias in their results, researchers and working professionals have started using these for different purposes. More recently Ana et al, [2] explored the potential of Artificial Intelligence Chatbots (e.g., ChatGPT) for data exploration of federated bioinformatics knowledge graphs. We believe that the performance of other chatbots could also be explored for generating SPARQL queries to access data from SPARQL endpoints.

1.1. Related Work

Generative AI tools e.g, Open AI's ChatGPT, Google Gemini use NLP (Natural Language Processing)[13] and GPT (Generative Pre-trained Transformer)[14] which form the core architecture of these tools. Generative AI tools appear to be in use by the researchers and scientists to apply their biological knowledge combined with the power of AI and available biological databases available to access and retrieve information from multiple databases without or with limited knowledge of SPARQL querying or RDF structure.

The complexity, syntax and steep learning curve of SPARQL poses a major problem, but with the help of NLP[13], these problems can be partly addressed as it allows to write queries in natural language and eliminates the need for having deep knowledge of querying language[15]. When creating complex queries where data needs to be gathered from multiple databases, NLP

could also help in breaking down these complex biological questions into smaller and simpler steps[15][13].

GPT leverages the powerful Transformer architecture[16] to analyse and process entire sentences simultaneously, unlike its predecessor Recurrent Neural Networks (RNN), which processes information sequentially, so it could forget and subsequently ignore information learnt in the past to interpret the present information[14]. Transformers parallel processing capabilities allow GPT to learn and understand complex patterns and relationships within the sentence or paragraph[16] [5][17]. GPT(s) can help researchers with limited technical knowledge for creating SPARQL queries, generating explanations on them and any questions related to SPARQL or databases etc [18]. In this case GPT can act like a bridge between natural language[15] and the complexities of querying databases. GPT can translate any natural language query [14][18] into their corresponding SPARQL query, eliminating the need for researchers to learn SPARQL complex syntaxes and the complete RDF structure[19]. Similarly the results obtained from SPARQL queries can be complex and sometimes overwhelming for non-technical users where GPT-based systems can also assist by analysing the results and explaining complex biological information in simple language, highlighting key findings, presenting them in a user-friendly format[5][18]. In short, NLP and GPT combined together can complement non-technical researchers to write SPARQL queries and accessing vast amounts of biological databases[15].

Since the introduction of Transformer AI model [16] by Google in 2017, the core architecture behind ChatGPT and other GPTs[20][19], there has been a significant rise in the release of Generative and Conversational AI tools[17], some of the most popular ones being ChatGPT[21], Google Gemini[22], Microsoft Copilot, Chatsonic, Youchat. This paper provides a performance review of the different GPTs (Generative pre-trained Transformer)[16] for generating simple SPARQL queries to access data from single biological database as well as generating complex federated SPARQL queries [2] to access, combine data from multiple biological databases by providing prompts written in natural language "English".

In this paper, we provide our findings while generating SPARQL queries through AI chatbots to access healthcare and life science databases. For our experiments we considered five chatbots namely ChatGPT, Gemini, Microsoft Copilot, Chatsonic and Youchat. For this study we considered three databases namely DBpedia (covers cross domain including healthcare), DrugBank and KEGG (Kyoto Encyclopedia of Genes and Genomes) which covers data on drugs and genes respectively. Using aforementioned chatbots, we tried to generate both single source queries (non-federated) as well as simple and complex federated queries presented earlier in [1]. These queries were humanly generated from [citesaleem2018largerdfbench](https://github.com/citesaleem2018/largerdfbench) and the rationale behind selecting these five queries is that they are reflective of the overall simple and complex SPARQL queries set. These queries cover the wide range of SPARQL constructs (Select, Optional, Filter etc), hence became the reason to be selected. Out of all the queries presented in [1], we selected S8, S11, S14 and C1 queries, whereas the complete list of queries is also available at: <https://shorturl.at/cfoAE>. In essence we asked following four questions from respective five chatbots while considering three databases and five SPARQL queries:

1. What is (Dbpedia, Drugbank, KEGG) database? Limit the answer to 60 words.
2. Does database (Dbpedia, Drugbank, KEGG) has a SPARQL endpoint, provide its URL?

3. Build the query by providing specific prompts
4. What is the result of this query (created in previous step)?

In this paper we are focusing to check the correctness of syntax of SPARQL query generated through these bots and, it is out of scope to evaluate the results, correctness, completeness, retrieval time of results or how the complexities of the queries can be simplified.

Remaining of this paper is organized as follows: Section 2 presents the description generated regarding different AI chatbots, prompts, queries and the improved queries provided to AI chatbots. Section 3 provides simple and federated SPARQL queries generated by the AI chatbots whereas Section 4 provides the results of the AI chatbots when asked to get the results of the query. Section 5 provides a detailed discussion of the results generated by AI chatbots about database descriptions, endpoints and SPARQL queries, and section 6 addresses the limitations in using AI chatbots to generate SPARQL queries. Lastly, Section 7 provides the conclusion.

2. Use Cases

The sensitive nature of healthcare data creates fragmented, siloed and mostly private repositories of data by design. Some of those repositories are publicly available as RDF graphs and can be queried via SPARQL endpoints. This section provides information about the Chatbots used, databases considered, SPARQL endpoints(if available) and original SPARQL queries as well as the "Improved Prompts for Chatbots"

2.1. Chatbot Used:

The table 2.1 presents the list of chatbots, date accessed, version, developer and the year developed along with the coverage statement of those bots.

Chatbots Date Access	Version	Year	Developer	Coverage
ChatGPT 11-03-2024	GPT-3.5	2020	OpenAI	Covers a range of topics, including but not limited to general knowledge, science, technology, literature, history etc. It is designed to engage in natural language conversations and assist users with various inquiries and tasks.
Gemini 11-03-2024	1.0 Pro	N/A	Google AI	Capabilities are constantly evolving, but it can access and process information from the real world through Google Search and keep the response consistent with results.
Copilot 14-03-2024	GPT-4 model	2023	Microsoft	It is an advanced chatbot based on the powerful GPT-4 model and offers a wide range of capabilities, operates in multiple languages, and is available both as a free version and a paid Copilot Pro version.
Chatsonic 12-03-2024	Advanced AI	2021	Writesonic	It can assist with a range of topics including but not limited to general knowledge, writing assistance, research etc.
Youchat 12-03-2024	YouChat2.0	2023	You.com	YouChat is an AI language model designed to provide assistance, information, and insights on a wide range of topics. It is capable of engaging in detailed discussions and answering factual questions based on available information

2.2. Database Descriptions

This section presents the description about the databases involved for our experiments. We asked different chatbots to explain the database by themselves.

Dbpedia: When asking different chatbots "What is DBpedia database? Limit the answer to 60 words", we have the response presented in 2.2.

Chatbot	Database Descriptions
ChatGPT	DBpedia is a structured database extracted from Wikipedia, containing structured information about topics covered in the encyclopedia. It transforms unstructured Wikipedia content into a machine-readable format, enabling easier access and utilisation of Wikipedia's knowledge. DBpedia organises information into RDF triples, making it useful for semantic web applications and knowledge graphs.
Gemini	DBpedia is essentially a giant knowledge base built from Wikipedia. It extracts information from Wikipedia articles and organises it in a structured format, making it easily searchable and usable by machines. Imagine turning the vast knowledge in Wikipedia into a database that computers can understand. This allows you to ask complex questions about things mentioned on Wikipedia, like finding cities in a specific state or artists from a particular era.
Copilot	DBpedia is a community-driven project that extracts structured content from various Wikipedia projects, including Wikipedia. It creates an open knowledge graph (OKG) available on the web, allowing users to semantically query relationships and properties of Wikipedia resources. DBpedia is a valuable resource for researchers, industry professionals, and anyone interested in linked open data
Chat-sonic	DBpedia extracts structured content from Wikipedia and makes it available on the web. It provides a large, multilingual knowledge base with information about notable people, places, organisations, and things. DBpedia allows users to query this information and explore the relationships between different entities. It's a valuable resource for data integration, knowledge exploration, and semantic querying.
Youchat	DBpedia is a large-scale, multilingual knowledge base extracted from Wikipedia, aiming to provide structured content from the information created in the Wikipedia project. It allows users to find answers to questions where the information is spread across multiple Wikipedia articles. The structured information is made available on the web as Linked Open Data, adhering to Linked Data principles.

DrugBank: When asking different chatbots "What is Drugbank database? Limit the answer to 60 words", we have the response presented in 2.2.

Chatbot	Database Description
ChatGPT	DrugBank is a comprehensive, freely accessible database containing information on drugs, targets, and interactions. It provides chemical structures, pharmacological actions, and mechanisms of action for drugs. Researchers and healthcare professionals use DrugBank for drug discovery, pharmacology research and clinical applications as it provides drug properties and interactions.
Gemini	Drugbank is a free online library related to drugs and their targets. It is a giant encyclopedia specifically for drugs that provides detailed information on chemical structure, how drugs interact with the body, and even links to ongoing research. This comprehensive resource is used by researchers, doctors, and even the general public to understand how medications work and their potential effects.
Copilot	DrugBank is a biomedical knowledgebase that provides structured drug data for research, discovery, and software. It contains information on over 500,000 drugs and drug products, making it a valuable for researchers and industry professionals interested in linked open data. DrugBank's technology seeks out relevant evidence, builds deep connections, and delivers usable, trustworthy drug knowledge
Chat-sonic	DrugBank is a comprehensive resource that combines drug data with target information and contains extensive information about drugs, their mechanisms, interactions and targets. It is widely used by the pharmaceutical industry, medicinal chemists and researchers for drug discovery and development. It is a valuable tool for understanding the properties and effects of various pharmaceutical compounds.
Youchat	DrugBank is a comprehensive, freely accessible online database containing information on drugs and targets. It combines detailed drug data, including chemical, pharmacological, and pharmaceutical information, with target information, such as sequence, structure, and pathway details. DrugBank is widely used by the drug industry, medicinal chemists, pharmacists, physicians, students and the public

Kegg: When asking different chatbots "What is KEGG database? Limit the answer to 60 words", we have the response presented in 2.2.

Chatbot	Database Description
ChatGPT	The KEGG database is a resource for understanding biological pathways, genomes, diseases, drugs and chemical substances. It offers curated data on molecular interactions and functions, aiding research in bioinformatics, systems biology and drug discovery. KEGG's pathway maps and analysis tools facilitate the exploration of complex biological processes and their relationships.
Gemini	KEGG, short for Kyoto Encyclopedia of Genes and Genomes, is a database powerhouse for understanding biological systems. Launched in 1995, it offers a wealth of information on genes, proteins, pathways, and diseases. Imagine KEGG as a central hub connecting the dots between genes, their functions, and how they work together in cellular processes. This knowledge base is a valuable tool for researchers in areas like genomics, drug development, and understanding the intricate workings of living organisms.
Copilot	KEGG is a comprehensive biological knowledgebase that provides valuable insights into the molecular interactions, pathways, diseases, drugs, and chemical substances. Researchers and educators utilise KEGG for bioinformatics studies, including genomics, metagenomics, and metabolomics. It encompasses data analysis, modeling, and simulation across various omics fields
Chat-sonic	KEGG is a comprehensive database that integrates genomic, chemical, and systemic functional information. It offers a wide array of data related to biological pathways, diseases, drugs, and chemical substances. KEGG is widely utilised in bioinformatics and computational biology for studying molecular interactions and cellular processes. It provides valuable insights into the relationships between genes, proteins, and biochemical pathways.
Youchat	KEGG is a database resource for understanding high-level functions and utilities of the biological system, such as the cell, the organism, and the biosphere, from genomic and molecular-level information. It provides a computer model of the biological system, integrating molecular building blocks of genes and proteins, chemical substances, and molecular wiring diagrams of interaction and reaction networks. KEGG also offers a REST-style Application Programming Interface for accessing its database resources.

2.3. SPARQL Endpoints

Some databases provides endpoints to query their data. It is therefore important to ask directly the respective chatbot: If the considered database has endpoint available or not. We asked the following question trough different chatbots.

DBpedia endpoint: When asked "Does DBpedia database has a SPARQL endpoint? What is its URL?" following results were generated

1. **ChatGPT:** Yes, DBpedia provides a SPARQL endpoint for querying its database. The URL for the DBpedia SPARQL endpoint is: <http://dbpedia.org/sparql>.
2. **Gemini:** Yes, DBpedia offers a public SPARQL endpoint for querying its data. You can access it directly through the following URL: <https://dbpedia.org/snorql/>
3. **Copilot:** DBpedia provides a public endpoint: <http://dbpedia.org/sparql>
4. **Chatsonic:** Yes, DBpedia has a SPARQL endpoint: <https://dbpedia.org/sparql>.
5. **YouChat:** Yes, the DBpedia has a SPARQL endpoint: <http://dbpedia.org/sparql>.

Drugbank endpoint: When asked "Does Drugbank database has a SPARQL endpoint? What is its URL?" following results were generated

1. **ChatGPT:** As of my last update in January 2022, DrugBank does not have a publicly available SPARQL endpoint.

2. **Gemini:** DrugBank itself doesn't offer a public endpoint. However, you can access DrugBank data through Third-party endpoints: Platforms like Bio2RDF <https://old.datahub.io/-dataset/bio2rdf-drugbank>).
3. **Copilot:** Certainly! DrugBank provides a public SPARQL endpoint that allows users to query its structured data: <https://go.drugbank.com/>
4. **Chatsonic:** DrugBank does have a SPARQL endpoint, but the specific URL for it is not provided in the search results. Therefore, I am unable to provide the URL for DrugBank's SPARQL endpoint based on the information available.
5. **YouChat:** Yes, the DrugBank has an endpoint: <http://s4.semanticscience.org:16006/sparql>.

KEGG endpoint: When asked "Does KEGG database has a SPARQL endpoint? What is its URL?" following results were generated

1. **ChatGPT:** As of my last update in January 2022, KEGG does not provide a SPARQL endpoint for querying its database
2. **Gemini:** KEGG doesn't have its own SPARQL endpoint. For SPARQL access, consider: Downloading KEGG data: Various formats exist (<https://www.kegg.jp/kegg/download/>). Third-party endpoints: Bio2RDF offers SPARQL access to KEGG data.
3. **Copilot:** KEGG provides a public SPARQL endpoint. Remember to review the documentation and usage notes before running any queries.
4. **Chatsonic:** The KEGG database offers a SPARQL endpoint, and its URL is <https://www.genome.jp/sparql/linkdb>. This endpoint allows users to execute SPARQL queries to retrieve specific information from the LinkDB RDF, facilitating data access and exploration within the KEGG database.
5. **YouChat:** The URL for the KEGG SPARQL endpoint is not provided in the search results.

2.4. SPARQL Queries

We considered five different man-made SPARQL Queries with the following scheme: one non-federated SPARQL query, three simple federated SPARQL queries and single complex federated queries provided by Saleem et al[1]. We take the Human English language description of the queries and tried generating the queries on chatbots. Section 2.5 provides the English description of those original queries whereas section 2.6 provides the "Improved Prompts written for Chatbots" to see the results and performance of these chatbots.

2.5. Original Queries

As mentioned before, out of all the queries presented in [1], for this experiment we selected S8, S11, S14 and C1 queries, whereas the complete list of queries is also available at: <https://shorturl.at/cfoAE>.

1. **Q1: Find all drugs from Drugbank** (single source query)
2. **Q2: Find all drugs from Drugbank and DBpedia with their melting points** (simple federated query)
3. **Q3: Find all the equations of reactions related to drugs from category Cathartics and their drug description** (simple federated query)

4. **Q4: Find drugs that affect humans & mammals for those having a description of their biotransformation, also return this description** (simple federated query)
5. **Q5: Find the equations of chemical reactions and reaction title related to drugs with drug description and drug type 'smallMolecule'. Show only those whose molecular weight average larger then 114** (complex federated query)

2.6. Improved Prompts for Chatbots

We rewrote the original English descriptions of these queries (provided in section 2.5) as Improved Prompts for Chatbots in order to exploit the full potential of these chatbot systems.

We run the improved prompts single time in this work, the prompts were not revised or improved after that, and the results were analysed based on the syntax of the query generated in their first run. It was out of the scope to see the incremental improvement of prompts, checking the correctness of results or how the complexities of the queries can be simplified.

To improve the original English descriptions, we added more detail to the original prompts by explicitly mentioning the i) query language to be used, ii) task the chatbot needs to perform, iii) which databases it needs to get data from and iv) what specific data we are looking for. For federated queries, we explicitly provide details on which features to collect from each data source and how to combine the data from these data sources. Further detailed methodology is available in section 5 of this paper. Prompt improvement or alternatively the prompt Engineering [23] is an emerging field and there is a lot of ongoing work already in place. In summary- better the prompt- better the response coming from these chatbots. P1 is the prompt query of Q1 and similarly the other respective Qs i.e **P <==> Q**.

1. **P1:** Write a single SPARQL query to retrieve the complete list of drugs from the Drugbank database. Use DrugBank endpoint URL (<http://drugbank.bio2rdf.org/sparql>) to retrieve a list of all distinct registered drugs in the Drugbank database.
2. **P2:** Write a single federated SPARQL query to retrieve all drugs from two biological databases: DBpedia and Drugbank, along with their corresponding melting points. The result from the query must be combined results from both databases.
3. **P3:** Generate a single federated SPARQL query to obtain drugs data from the category "Cathartics", their corresponding enzymes, chemical reactions, from DrugBank and KEGG databases. Use the DrugBank database, to obtain drugs in the "Cathartics" category. For each Cathartics drug identified, extract its description and KEGG compound ID. Use the retrieved KEGG compound IDs from Drugbank to find corresponding enzyme entries in the KEGG database and obtain the associated equations of reactions within KEGG.
4. **P4:** Generate a single federated SPARQL query to retrieve all drugs from DrugBank biological database that affect "humans and mammals". For each drug found, get its CAS registry number. Using the CAS number from DrugBank, find the same drug in KEGG biological database and obtain the mass information for each drug. Also, retrieve the drug label, description of their bio-transformation from DrugBank, if they are available.
5. **P5:** Generate a single federated SPARQL query to retrieve distinct drugs and get information like drug description, molecular weight average, compound, reaction title, and chemical equation. Use the DrugBank and KEGG databases. From DrugBank, select drugs with the drug type "small molecule" and retrieve their descriptions and KEGG compound

IDs. Then, from KEGG, obtain information about enzymes, chemical reactions, reaction titles and chemical equations associated with the selected compounds. Finally, include an optional clause to filter drugs with a molecular weight average greater than 114.

3. Query Formulation through Chatbots

It is worth noting that when asking Q2- Q5 the queries formulated by the chatbots were either incomplete, inconsistent or simply wrong. This was the reason, more elaborative, descriptive and direct prompt were generated (section 2.6). Detailed queries formulated asking different prompts is presented in this section. In this paper we only presented the queries formulated using three prompts out of five (section 2.6): P1 (single source query), P4 (simple federated query) and P5 (complex federated query). These three selected queries are reflective of the overall simple and complex queries, and constitute a good sample out of all the available queries that also cover different SPARQL constructs.

The queries(P1, P4, P5) generated by the AI chatbots are available at this Github link: <https://github.com/Sourav-rcsi/Artificial-Intelligence-based-Chatbots-for-generating-Federated-SPARQL-queries.git>

4. Query Results

The table4 shows the summary regarding the outcome of the AI chatbots for providing the correct database descriptions, SPARQL endpoint URL and correct/ incorrect syntax of P1, P4 and P5 queries.

Chatbots	Database Description	SPARQL Endpoints	P1	P4	P5
ChatGPT	✓	×	✓	✓	×
Gemini	✓	✓	×	×	×
Copilot	✓	✓	✓	✓	×
Chatsonic	✓	×	×	✓	×
Youchat	✓	×	✓	✓	×

When asked "What is the result of this query?"

1. **ChatGPT:** I'm unable to directly execute federated SPARQL queries. You can run the provided SPARQL query on your local SPARQL endpoint or an online SPARQL query service that supports federated queries to obtain the results.
2. **Gemini:** I can't directly execute the SPARQL query to retrieve the results from the Drug-bank database. However, I can explain what the query retrieves and how to potentially interpret the results.
3. **Copilot:** I apologise, but I am unable to execute the SPARQL query directly. However, you can run this query using a SPARQL endpoint that supports federated queries. The result would be a list of distinct drugs meeting the specified criteria, along with their descriptions, molecular weights, compounds, reaction titles, and chemical equations. If you have access to a SPARQL endpoint, you can input the query there to retrieve the actual results.

4. **Chatsonic:** I'm unable to directly execute the query against the databases, but when you run the provided federated SPARQL query, the result will be a table and will include distinct drugs from DrugBank with the drug type "small molecule", along with their descriptions and KEGG compound IDs. Additionally, it will provide information about enzymes, chemical reactions, reaction titles, and chemical equations associated with the selected compounds from the KEGG database.
5. **YouChat:** I'm sorry, but I couldn't find any search results for the specific federated SPARQL query you requested. If you have any other questions or need further assistance, feel free to ask!

It is important to note that the results of the queries could not be generated by the chatbots as currently they are not capable of running SPARQL queries at the endpoints. However, in some cases they can provide some hint(s) to inform how will an example result of the query look like.

5. Discussion

5.1. Database and SPARQL endpoint

DBpedia: When chatbots were asked "*what is the DBpedia database*", all generated the correct and relevant description for DBpedia database, with good structure, in simple, easy-to-understand language except Copilot which has a slightly complex description. The bots were also asked to limit the number of words to 60 in these questions and all the bots limited their words to less than 60 except Gemini which generated more words. However, the description generated by Gemini is much simpler to understand among others and goes beyond by giving example how the database can be used. When asked if "*the DBpedia database has a SPARQL endpoint*", all the bots unanimously said yes and generated the correct URL to connect to the SPARQL endpoint.

DrugBank: When chatbots were asked "*what is the DrugBank database*", all gave the correct and relevant information about the database, with good but varied structure, in simple, easy-to-understand language. All bots except Gemini limited their answers to less than 60 words. Gemini, along with ChatGPT and Copilot generated their description with examples on who and how these databases can be used, while Youchat provided information which industry this database is used. Interestingly, the structure for all the answers is slightly different. They all provide the description first; ChatGPT, Copilot, Chatsonic then provides the applications next and a general statement to finish off, while Gemini and Youchat provide the applications in the last sentence. When bots were asked if the "*DrugBank database has a SPARQL endpoint and to provide its URL*", ChatGPT could not provide the relevant information as its cut-off for information is until January 2022. Gemini pointed to the correct resource from where you can find the DrugBank SPARQL endpoint, while, Youchat correctly mentions an endpoint exists, it provides a non-existing URL. Microsoft Copilot pointed to the official DrugBank website and provided the website URL, while, Chatsonic could not provide the URL to the endpoint or the website, but mentions to refer the official DrugBank resource to find this information.

KEGG: When chatbots were asked "*what is the KEGG database*", yet again all gave the correct and relevant information about the database, with good but varied structure, in human

understandable language. For this prompt only ChatGPT and Gemini limited their answers to 60 words while Copilot, Chatsonic and Youchat generated more than 60 words. Gemini, yet again generated a similar kind of descriptive answer with examples. Interestingly, again the structure for all the answers is slightly different but similar for each bots. ChatGPT, Copilot, Chatsonic provide the description first then the applications next a general statement to finish off, while Gemini provides the applications in the last sentence. Youchat did not provide any application in this case, it could be because the words were limited to 60. When chatbots were asked if *"the KEGG database has a SPARQL endpoint and to provide its URL"*, ChatGPT generated the same result as above stating it does not have up-to-date information. Gemini incorrectly states that there is no SPARQL endpoint for the database and provides two URL's, among which the second is a non-existing URL, while the first URL brings you to download KEGG data. Copilot correctly identifies a SPARQL link to the KEGG database exists, however provides link only to the database homepage. Chatsonic correctly identifies a SPARQL endpoint does exist, and provides a link to it, while Youchat cannot provide any URL.

5.2. Queries

In this paper, for our experiments we presents the results of only three prompts namely **P1** (single source query), **P4** (simple federated query) and **P5** (complex federated query). This section provides the comparison between the Human-made query and the chatbot generated query regarding the correctness of syntax of the queries. It is indeed a limited comparison which is entirely based on the findings on these five queries. The complexity of the queries can further be increased and the results generated could be different that still remains an open question to be investigated.

P1 query: To *"Find all drugs from DrugBank database"*, we created a prompt describing in detail each step the SPARQL query needs to perform. We start by asking it to write a single query as we noticed bots would generate several separate SPARQL queries to address one question containing several steps. Next, we explicitly explain the that query needs to generated and also provided the exact DrugBank SPARQL endpoint URL. We explain our query in more detail to find distinct drugs from the DrugBank database. Here, we ask it to generate only the query (however, the explanations can also be limited to 50 words or 5 sentences or 1 paragraph, if needed). The simple SPARQL query generated by ChatGPT, Microsoft Copilot and Youchat is correct and directly executable at the DrugBank SPARQL endpoint. The SPARQL endpoints provided by Chatsonic and Gemini are incorrect, they will not be directly executable as they point to a non-existing URL. Interestingly, they were provided and asked to use the DrugBank SPARQL endpoint URL in the prompt and both Chatsonic and Gemini have access to up-to-date information unlike ChatGPT (access to data up to January 2022), so an accurate SPARQL endpoint result was expected from them. This was a relatively simple query and all the GPTs generated correct SPARQL syntax to find all drugs from the Drugbank database.

P4 query: To *"find all drugs that affect humans and mammals and getting their description of biotransformation if available"*, the prompt was created to describe in detail each step the SPARQL query needs to perform. We start by specifying the type of SPARQL query and specific database that must be queried (which are federated query and DrugBank respectively in this case). For federated queries, we found it important to mention the word "single", as the bots would generate one or more queries against single question. We also mentioned to retrieve

all the drugs from this database that affects "human and mammals", which were in quotations in order to emphasise the significance of those words. Next, we tell it to get the CAS registry number for each of the drug found from the DrugBank database and find the same drug in KEGG database using the CAS number to obtain its mass. Additionally, we ask it to collect the drug label, description of their biotransformation from DrugBank database only if that information is available. In the prompt, it was asked to generate the query only, not the descriptions/explanations, as explanation of the generated queries can be quite lengthy and varied in details. The query generated by ChatGPT uses meaningfully named variables to represent features, use a simpler structure with prefixes and UNION clause to combine results from both the databases. Gemini created a very detailed query with comments to understand the query but used useless constructs like BIND for this query. We can clearly see that this is not the correct query and would not bring the correct results. Copilot, Chatsonic and YouChat created nearly correct query with some limitations e.g, in case of Copilot additional constructs like FILTER that makes queries more complex and expensive to run.

P5 query: In order to *"find the equations of chemical reactions and reaction title related to drugs with drug description and drug type "small molecule" with molecular weight average larger than 114"*, we create a highly-detailed prompt to retrieve this information by querying DrugBank and KEGG databases. We start by giving a high-level overview of our question to generate a single federated SPARQL query that will get the unique drugs and specify the exact information we are looking for. For federated queries, we found it important to mention the word "single", as the bots would generate one or more queries for a single question. Next, the two databases which needs to queried are specified and only the drugs with drug type "small molecule" must be retrieved along with their descriptions and KEGG compound ID from the DrugBank database. From the KEGG database, information about enzymes, chemical reactions, reaction titles and chemical equations associated with the selected compounds from DrugBank must be gathered. Finally, these drugs must be filtered to only include drugs which have a molecular weight average greater than 114. Here, we ask it to generate only the query rather than the description about the generated query, as explanation of these queries can be varied in detail and length. Clearly all the generated complex queries were not correct in the first iteration and in this research we are not interested to improve the prompt for multiple iterations in order to get the correct SPARQL query. Hence we stopped our experiment at this stage.

6. Limitations

Generative AI has recently found its use in variety of applications[24], including medical examinations, education, bioinformatics, plastic surgery[25]. It has outperformed several previous architecture of AI in performance and achieved good results.

In this article the comparison of generative AI bots is entirely based on the findings on these five queries. The complexity of the queries can further be increased and the results generated could be different that still remains an open question to be investigated.

Since Generative AI is a new technology, it is prone to some limitations as follows:

1. **Understanding SPARQL Basics:** While generative AI has the capability to generate SPARQL queries by converting natural language(English), a basic understanding of SPARQL core concepts, keywords, syntax, will be beneficial for researchers to interpret

and understand the results generated by these SPARQL queries and to make modifications to the queries generated.

2. **Background knowledge:** To generate the improved prompts it is essential to have: i) some background knowledge of biomedical databases, ii) information regarding the concepts need to be extracted and iii) knowledge regarding how can the results be combined from federated queries. Moreover it is important to provide clear and concise prompts to the AI chatbots. More detailed and concise the prompts are, higher the probability of getting a comprehensive response.
3. **Multiple Databases:** Not all databases have the same underlying data model[2]. Federated SPARQL queries require retrieving and combining information from multiple databases. It can be challenging to interpret those results by non-technical users if the databases have different data models. This can also have an effect on the results and potential errors in the query generated. Basic knowledge on the structure of databases is required to provide relevant information to AI chatbots on how to combine the results from multiple databases.
4. **Accuracy of Prompts:** The results generated by Generative AI are directly dependent on the prompts/questions asked by the user[26]. The prompts must be clear, concise and finely tuned for the research problem in question. They must also be highly detailed, explaining what it must do for each step. Accuracy of the generated results by Generative AI can be highly improved by improving the prompt/question.
5. **Bias and Interpreting results:** For complex federated queries, the data generated from various databases can be at scale[2]. Having understanding of bias, background knowledge on the problem and basic SPARQL skills or consultation along with domain experts can enhance interpretation of results. Generative AI tools should be used iteratively by starting with simpler queries/questions to understand the type and accuracy of result it generates before moving to complex federated queries[26].
6. **Generative AI tool:** Depending on the question, the right Generative AI tool or a combination of them should be used. If an up-to-date information is needed, Google, Microsoft AI or ChatGPT-4 should be considered over ChatGPT-3.5 as it has information until January 2022. For example, ChatGPT-3.5 will not have the current updates and information on new databases. Each tool has its own pros and cons, a study of different tools would be beneficial for researchers to select one that suits specific problem.

7. Conclusion

The future of data access is likely to involve federated open research data, driven by the growing number of datasets and databases. Technological advancements are necessary to bring federated data closer to users, particularly through improved user-facing services. Our research demonstrates the competence of AI bots (ChatGPT, Gemini, Copilot, Chatsonic, and YouChat) for generating syntax for SPARQL queries across diverse databases. By analysing their outputs with human-authored queries, it is evident that these Chatbots still poses significant limitations and we subsequently identify areas of strength and opportunities for improvement. This study fosters synergy between data science and healthcare, facilitating more efficient query formulation and advancing interdisciplinary research initiatives for the working biologists

who are unable to query biomedical databases due to technical nature of constructing SPARQL queries to access RDF biological databases. In summary we have the following reflections:

- Publicly available chatbots have a potential in Find-ability and Reuse of databases, aiding researchers in discovering relevant information about the databases.
- Conversational AI chatbots like ChatGPT, Gemini, and Copilot etc offer high-level database summaries, improving researchers' understanding of database contents.
- Domain experts can utilise these chatbots for explaining SPARQL queries, while also contributing to model improvement through feedback.
- AI chatbots can be trained and fine-tuned to generate better results suited for specific needs.
- Caution is advised when using AI chatbots for data access due to potential hallucinations. Filtering methods based on confidence levels in language models may be explored, with user validation remaining crucial for the results generated by these chatbots in their current forms.

References

- [1] M. Saleem, A. Hasnain, A.-C. N. Ngomo, Largerdfbench: A billion triples benchmark for sparql endpoint federation, *Journal of Web Semantics* 48 (2018) 85–125.
- [2] A.-C. Sima, T. M. de Farias, On the potential of artificial intelligence chatbots for data exploration of federated bioinformatics knowledge graphs, *arXiv preprint arXiv:2304.10427* (2023).
- [3] M. Saleem, Y. Khan, A. Hasnain, I. Ermilov, A.-C. Ngonga Ngomo, A fine-grained evaluation of sparql endpoint federation systems, *Semantic Web* 7 (2016) 493–518.
- [4] W. Ali, M. Saleem, B. Yao, A. Hogan, A.-C. N. Ngomo, A survey of rdf stores & sparql engines for querying knowledge graphs, *The VLDB Journal* (2022) 1–26.
- [5] Y. Shokrollahi, S. Yarmohammadtoosky, M. M. Nikahd, P. Dong, X. Li, L. Gu, A comprehensive review of generative ai in healthcare, *arXiv preprint arXiv:2310.00795* (2023).
- [6] A. Hasnain, M. R. Kamdar, P. Hasapis, D. Zeginis, C. N. Warren, H. F. Deus, D. Ntalaperas, K. Tarabanis, M. Mehdi, S. Decker, Linked biomedical dataspace: lessons learned integrating data for drug discovery, in: *The Semantic Web–ISWC 2014: 13th International Semantic Web Conference, Riva del Garda, Italy, October 19–23, 2014. Proceedings, Part I* 13, Springer, 2014, pp. 114–130.
- [7] A. Hasnain, Q. Mehmood, S. Sana e Zainab, M. Saleem, C. Warren, D. Zehra, S. Decker, D. Rebholz-Schuhmann, Biofed: federated query processing over life sciences linked open data, *Journal of biomedical semantics* 8 (2017) 1–19.
- [8] A. Hasnain, S. S. e Zainab, D. Zehra, Q. Mehmood, M. Saleem, D. Rebholz-Schuhmann, Federated query formulation and processing through biofed., in: *SeWeBMeDA@ ESWC, 2017*, pp. 16–19.
- [9] S. S. e Zainab, M. Saleem, Q. Mehmood, D. Zehra, S. Decker, A. Hasnain, Fedviz: A visual interface for sparql queries formulation and execution., *VOILA@ ISWC 1456* (2015) 49.

- [10] M. R. Kamdar, D. Zeginis, A. Hasnain, S. Decker, H. F. Deus, RevealD: A user-driven domain-specific interactive search platform for biomedical research, *Journal of biomedical informatics* 47 (2014) 112–130.
- [11] A. Hasnain, Q. Mehmood, S. S. e Zainab, A. Hogan, Sportal: Searching for public sparql endpoints., in: *ISWC (Posters & Demos)*, 2016.
- [12] A. Hasnain, Q. Mehmood, S. S. e Zainab, A. Hogan, Sportal: profiling the content of public sparql endpoints, in: *Information Retrieval and Management: Concepts, Methodologies, Tools, and Applications*, IGI Global, 2018, pp. 368–401.
- [13] K. Chowdhary, K. Chowdhary, *Natural language processing, Fundamentals of artificial intelligence* (2020) 603–649.
- [14] C. Wang, M. Li, A. J. Smola, Language models with transformers, *arXiv preprint arXiv:1904.09408* (2019).
- [15] P. Zhang, M. N. Kamel Boulos, Generative ai in medicine and healthcare: Promises, opportunities and challenges, *Future Internet* 15 (2023) 286.
- [16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, *Advances in neural information processing systems* 30 (2017).
- [17] B. Meskó, E. J. Topol, The imperative for regulatory oversight of large language models (or generative ai) in healthcare, *NPJ digital medicine* 6 (2023) 120.
- [18] K. Nova, Generative ai in healthcare: advancements in electronic health records, facilitating medical languages, and personalized patient care, *Journal of Advanced Analytics in Healthcare Management* 7 (2023) 115–131.
- [19] J. Varghese, J. Chapiro, Chatgpt: The transformative influence of generative ai on science and healthcare, *Journal of Hepatology* (2023).
- [20] S. S. Biswas, Role of chat gpt in public health, *Annals of biomedical engineering* 51 (2023) 868–869.
- [21] P. P. Ray, Chatgpt: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope, *Internet of Things and Cyber-Physical Systems* (2023).
- [22] G. AI, An overview of bard: an early experiment with generative ai, <https://ai.google/static/documents/google-about-bard.pdf>, 2023.
- [23] B. Meskó, Prompt engineering as an important emerging skill for medical professionals: tutorial, *Journal of Medical Internet Research* 25 (2023) e50638.
- [24] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang, et al., A survey on evaluation of large language models, *ACM Transactions on Intelligent Systems and Technology* (2023).
- [25] J. Abi-Rafteh, H. H. Xu, R. Kazan, R. Tevlin, H. Furnas, Large language models and artificial intelligence: a primer for plastic surgeons on the demonstrated and potential applications, promises, and limitations of chatgpt, *Aesthetic Surgery Journal* 44 (2024) 329–343.
- [26] S. Sai, A. Gaur, R. Sai, V. Chamola, M. Guizani, J. J. Rodrigues, Generative ai for transformative healthcare: A comprehensive study of emerging models, applications, case studies and limitations, *IEEE Access* (2024).