Talk to Open Data: Enabling User Interaction with Open Government Data Using LLMs, RAG and Smart Agent Technologies

Mohsan Ali[1[0000-0002-3956-3543]], Maria Ioanna Maratsi[1[0000-0003-4513-6318]], Zoi Lachana[1[0000-0001-7356-8041]], Yannis Charalabidis[1[0000-0003-3760-5495]], Charalampos Alexopoulos[1[0000-0002-6610-0675]], and Euripidis Loukis[1[0000-0002-5932-4128]]

[1] University of the Aegean, ICSD, Samos 83200, Greece

Abstract. Open government data has been in the spotlight for several years, emphasising the need for publicly available datasets to foster societal innovation and enable data-driven decision-making. However, it is not sufficient for data to be available through an open data portal or repository to be usable by end users and for them to fully unveil its potential; it also needs to be accessible and intelligible. Targeting this aspect, this study builds on the synergy of open government data and emerging technologies, such as Large Language Models (LLMs) to demonstrate that technologically enabled open data portals can have enormous possibilities for end-users to understand an available dataset efficiently, and in an interactive manner. This implementation is built on the LLMs Retrieval Augmented Generation (RAG) functionality which fetches datasets after a user's choice from the Greek Open Data Portal, allowing the user to ask questions about this dataset using natural language. The Greek open data portal provides access to datasets from different domains through the Application Programming Interfaces (APIs) which makes it easier to bridge the gap between complex data and non-technical users who want to use it. This work explores the implementation potential of such a pipeline in a real-world application using the Greek Open Data portal as data source, and by utilising a conversational smart agent to interact with the open datasets available through the portal, it brings forward new capabilities in data-user interaction and efficient data exploration. Apart from the implementation, this study explores the pitfalls and shortcomings of such an endeavour (conversational agents) which requires real-time calculations and deterministic responses.

Keywords: Open Government Data, Open Data Portal, Open Data Interoperability, Large Language models, User Interaction, Retrieval Augmented Generation

1 Introduction

Open government data research has been in the spotlight for quite some years, emphasising the need for publicly available datasets to foster societal innovation and enable data-driven decision-making. However, open data needs to follow well-established

data publication standards and procedures, both in technical but also accessibility and usability terms. It is not sufficient for data to only be available through an open data portal or repository in order to be usable by end users and for them to fully unveil its potential; it also needs to be accessible and intelligible. Open government data available at data portals may still suffer practically because of several persistent challenges such as inconsistent metadata standards following the data, lack of technical interoperability, findability issues, or the difficulties which many non-technical users face during this process (also known as the digital divide), most of the time being a deterring factor for practically utilising the open datasets available. As a result, even though the volume of publicly available information has been increasing, non-specialist users or user groups lag behind in harvesting the benefits of open data compared to users with a technical background who can more easily and effectively extract value from it.

At the same time, Artificial Intelligence (AI) and Large Language Models (LLMs) bring forth new opportunities and capabilities in almost every scientific domain, thanks to their ability to understand natural language and queries and return human-like answers. Should LLMs be able to search, understand, and make data easier to interact with, a whole new world of capabilities can be unlocked for the end user. However, for this to happen in reality, technical and semantic interoperability on both dataset and data portal levels need to be achieved as basic requirements. LLMs are inherently capable of interpreting natural language and reasoning over both structured and unstructured data, this advantage being key for the purpose described. More specifically, when combined with Retrieval Augmented Generation (RAG) architectures, context-based output tailored to user queries input becomes feasible. In this context, novel types of interfaces can be designed, aiming to make open datasets easier to interact with, query, and understand at a deeper level.

This study focuses on the official Greek Open Data Portal[1] and, integrating its API infrastructure with an LLM RAG functionality, makes it feasible for users to explore and interact with chosen datasets through natural, conversational language, in a short time. The case study on the Greek OD portal demonstrates how users are able to identify certain datasets of their interest and then query the dataset content through prompts in natural language. The main purpose of this study is to contribute towards more accessible open data, not only technically but also practically, allowing users, regardless of their backgrounds, to interact with datasets in a natural, user friendly and expertise agnostic way, helping bridge the digital divide. In addition, this work aims to explore the implementation potential of such a pipeline in a real-world application using the Greek Open Data portal as a data source and utilising a conversational agent to interact with the open datasets available through the portal. Apart from the implementation, this study explores the pitfalls and shortcomings of such an endeavour (conversational agent) which requires real-time calculations and deterministic responses.

The paper's organisation is as follows. First, the research background of this work is included (Section 2), forming the study's motivation and foundational basis. Next, the

[1] https://data.gov.gr/

detailed methodological process for the experiments is presented in Section 3, while Section 4 includes the experiment results and analysis. Finally, Section 5 reflects on the results of this study, while Section 6 concludes with final remarks and future work and directions to be considered.

2 Research background

Various approaches have already been utilising LLMs and RAG to improve open data related issues, whether that pertains to open dataset's findability, accessibility, or interoperability. As far as the combination of open government data with Generative AI technologies is concerned, aspects of metadata generation and quality, and data discoverability are targeted. [1] utilised and fine-tuned the T5-small and T5-large models to improve the process of metadata category and keyword extraction on datasets available through the European Data Portal (EDP). This aimed to facilitate dataset discoverability, FAIR data compliance, and improve data portal usability. Similarly, [2]) explored the capabilities of LLMs (GPT3.5) in a case study of enabling natural language querying on the Scottish open statistics portal[2] and demonstrating how GenAI can revolutionalise the way users can interact with Open Government Data portals and the RAG potential to enhance LLM accuracy. In the same vein of interaction with open data, the report by [3] explores five use-case scenarios where open data and GenAI may form a synergy to bring forth the so-called "Fourth Wave" of open data. The report presents the scenarios, key challenges faced (e.g., data quality, bias) and requirements that emerge to create GenAI-ready data, as well as actionable recommendations to support the integration process for concerned stakeholders.

GenAI and LLM applications on interoperability also include semantic interoperability and enhancing semantic search capabilities on data portals and knowledge repositories. [4] proposed a methodology which includes ontological foundation and LLMs aiming to improve the hierarchical structure on scientific methods, tools, and techniques representation on Wikidata in order to enable interdisciplinary (or domain-agnostic) search capabilities. Viewing Wikidata as a knowledge graph, the method utilises knowledge graph metrics as well, to alleviate identified challenges on this repository, such as the inability to retrieve all (or most) relevant data using the currently offered SPARQL endpoint. Pertaining to dataset discoverability, in open data portals, [5] developed a semantic similarity recommender to assist users in discovering "dark data", so data that remains unused due to the difficulty in discovering it. The recommendation system by [5] makes use of Natural Language Processing (NLP) to capture semantic proximity among dataset's metadata and the metadata is made compatible with several OGD portals. Similarly, [6] analysed semantic similarity, interoperability and availability among datasets available through the Gulf Cooperation Council (GCC) countries' OGD portals (organised in how they relate to Sustainable Development Goals - SDGs) and proposed standardisation and region-specific strategies in this regard (to enhance OGD quality and semantic alignment with SDGs). Mapping SDG indicators and open

[2] statistics.gov.scot

data facilitated through LLMs was proposed by [7], who addressed the challenge of computing SDG indicators from heterogeneous open data sources. Their proposed framework (hybrid LLM- and rule-based approach) utilised LLMs and KGs in this context, demonstrating effective results through an applied case study with improved precision, recall and F1-score. LLMs aspire to be leveraged to facilitate the process of cross-domain ontology semantic alignment (or mapping). As proposed by [8] the initial mapping of Cultural Heritage domain ontologies is performed following a manual mapping process, the results are assessed using OpenAI's GPT zero- and few-shot learning from cultural heritage but also using mapping examples from other domains (e.g., legal domain) to move towards cross-domain reuse.

On the applications of LLMs and RAG discourse for policy and organisational aspects, [9] analysed public submissions to an Australian Federal Government inquiry in the education sector (inquiry on how GenAI impacts education), and, through their methodological approach for policy analysis, highlighted emerging concerns about academic integrity and a need for assessment reform. Lee-Geiller & Ali, (2025) used NLP to analyse Open Government Partnership (OGP) policy documents from 75 countries, examining their alignment with democratic values. Similarly, as [6], Lee-Geiller & Ali, (2025) also applied similarity calculation (e.g., cosine similarity) in order to identify semantically proximal content. In an effort to create or curate language-specific data for training, LLMs may facilitate the collection, annotation, and storage of data. [11] outlined a system to enable this process for the Greek language, on a Greek Data Pile.

Following the literature and traction of LLM applications for the facilitation of knowledge representation, discoverability, but also to improve open data portal facilitations, this research aims to contribute to the body of scientific knowledge by introducing the ability to bridge the gap between non-technical data user groups and data portals and allowing them to more easily interact with datasets on the portal. This is achieved by leveraging a proposed LLM and RAG architecture towards making data more discoverable and accessible to the general public, irrespective of their technical literacy and experience with data.

3 Experimental Methodology

This study follows the proposed methodology outlined in the steps in Figure 1. There are four major components, as follows.

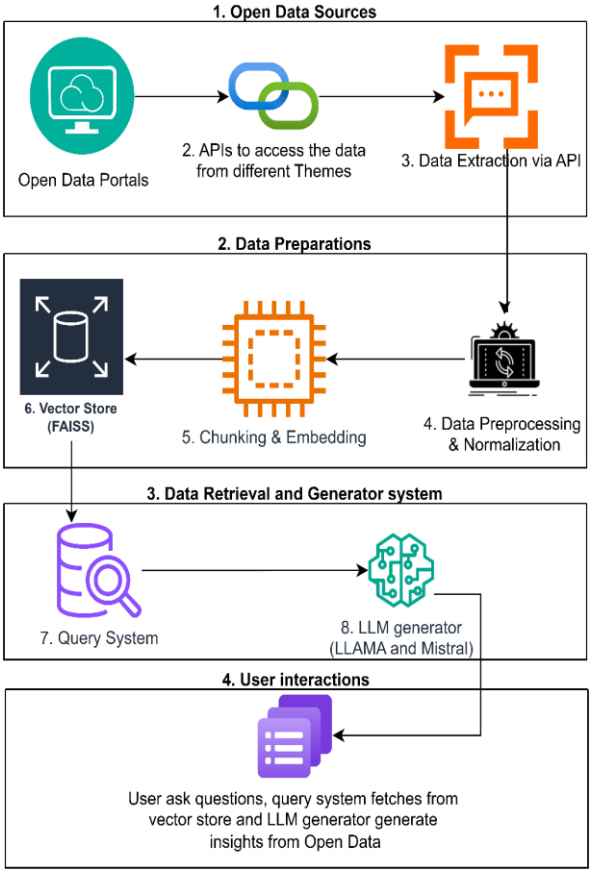

Fig. 1: Proposed methodology for "Talk to open data": Data is fetched from open data portals, it is prepared, data retrieval and generation are performed, and user interactions are handled, process can be seen from top to bottom.

3.1 Open government data sources

To enable user interaction in open government data, we have selected the Greek and the European open data portals. The Greek open data portal supports API-based access to open data, making it easy to integrate into an application for development purposes. We focused on three thematic areas in the Greek open data portal: Health, Education, and Travel. In these categories, we listed APIs from each of these sources. This information is shown in Table 1.

We also considered one dataset[3] from the European open data portal[4] which redirected to the Greek open data portal. This API call returns the metadata of the dataset. From

[3] https://data.europa.eu/api/hub/search/datasets/https-data-gov-gr-internet_traffic~~1

[4] https://data.europa.eu/en

this metadata, the original landing page of the dataset was extracted, which can help in data retrieval and further usage for generating open data insights.

Table 1. Open data sources: Greek open data portal[5] and European open data portal

Category	Dataset Name	API Endpoint
Education	List of Schools	/api/v1/query/minedu_schools
	University Teaching Staff	/api/v1/query/minedu_dep
	Students by School	/api/v1/query/minedu_students_school
	Atlas Internship System Statistics	/api/v1/query/grnet_atlas
	Eudoxos Requests & Deliveries	/api/v1/query/grnet_eudoxus
Health	COVID-19 vaccination statistics	/api/v1/query/mdg_emvolio
	COVID-19 vaccination statistics (weekly)	/api/v1/query/mdg_emvolio_weekly
	Drug Exports	/api/v1/query/drug-exports
	Drug Sales	/api/v1/query/drug-sales
	Inspections & Violations	/api/v1/query/efet_inspections
	Number of Pharmacists	/api/v1/query/minhealth_pharmacists
	Pharmacies	/api/v1/query/minhealth_pharmacies
	Number of Dentists	/api/v1/query/minhealth_dentists
	Number of Doctors	/api/v1/query/minhealth_doctors
Travel	OASA Daily Ridership	/api/v1/query/oasa_ridership
	Commercial Sailing Traffic and Routes	/api/v1/query/sailing_traffic
	Road Traffic for the Attica region	/api/v1/query/road_traffic_attica
Technology	Internet Traffic in Greece	/api/v1/query/internet_traffic
	Use of digital technologies for the development of products and/or business processes	/api/v1/query/ekt-tech-growth-assessment
	Use of digital technologies for the development of new or improved products and/or business processes	/api/v1/query/ekt-digital-tech-use
	Sectors of interest for future use of digital technologies	/api/v1/query/ekt-future-interest-sectors
	Adoption factors of digital technologies	/api/v1/query/ekt-adoption-factors

3.2 Data preparation

Most of the time, the data fetched from the open data portal was in tabular format. We needed to perform data preprocessing, conversion, and then store the data features in a vector store database. There are many data stores available; however, in this context, FAISS was used. After fetching data through the API, we converted it to JSON and then to a Pandas DataFrame (tabular structure). However, for the data to be understandable by the LLMs, it needed to be in textual format. For this purpose, the DataFrame was converted to textual format. The data was then ready to be fed into the tokeniser and to generate embeddings. “SentenceTransformer('all-MiniLM-L6-v2')” from

[5] https://data.gov.gr

HuggingFace was used to perform this step. The embeddings were stored in FAISS (vector store), allowing them to be fetched and then matched to relevant data based on user queries. This step requires a lot of attention as data may be in different formats (CSV, XML, or JSON). Thus, effort in the transformation was required for chunking and for generating embeddings for proper use in the next steps. If the embeddings are not correctly formed, models may have difficulty in retrieving the right data, while the generator may produce incorrect text.

3.3 Data retrieval and generator system

Data retrieval and generator functions will have access to the data stored previously in the FAISS store and the capabilities of the generator, (such as the large language model), to utilise the open data and then generate a response for the user. The data retrieval system only fetches data based on the user query, for instance, if the user wants to fetch the whole dataset or some of the samples, and what kind of insights does the user require? What are the top values based on the user query? These tasks are handled by the query module within the retrievals and generation system. We expanded this stage of the experiment based on the initial results generated using the pipeline, such as RAG, Vector store, and LLMs, but this step was not capable of generating accurate results, as it requires some agent to perform real time calculations on the data. We proposed and implemented a Langchain agent "**create_pandas_dataframe_agent**" **and** tested it within the experiment setup on the same data space which produced more accurate results. These results were validated by running the code as well.

3.4 User interaction

In the experimental initial stages, the user is asked to select the data theme by writing among the sources, and then to choose the specific datasets within that theme. Based on the user's choice, data will be fetched from the respective open data portal through the API and will be prepared to be stored in the vector store after chunking and embedding. Following that, the query system will understand the query and pass it to the vector store, and the vector store will return data based on the query. The data fetched from the vector store will be provided to the data generator (LLM), so it will provide insight and useful information to the end-user. In this manner, the whole communication will take place by bridging the technical requirements of open data required by the user and exposing the open data to LLMs to generate real-time insights from the data. Figure 2 explains all the functions that a user while interacting with "Talk to open data" agent can perform and also it explains that how retrieval and generator system with the help of LLMs, RAG, and smart agent can perform functions based on thee user interactions. **System requirements:** To perform this experiment, we utilised Kaggle IDE, which provides access to Jupyter notebooks. Kaggle has GPUs T400, along with memory, and CPU, which were enough to run this experiment. We have used Python in this experiment. Furthermore, we have used the HuggingFace transformer library, FAISS vector store, Mistral (mistralai/Mistral-7B-Instruct-v0.1), and LLAMA (meta-llama/Llama-3.1-8B & LLAMA -llama3-8b-8192) for experimentation of prompt. The Client.http

library was used to fetch data from the Greek open data portal. Furthermore, after knowing the shortfalls of the above models, we implemented an LLM agent using Langchain "**create_pandas_dataframe_agent,**". On the backend, we used a better (considering performance) LLM model (LLAMA -llama3-8b-8192) provided through API from Groq[6], which allowed us to run small experiments with less resources. The implementation stages have been explained in a stepwise format in Figure 3.

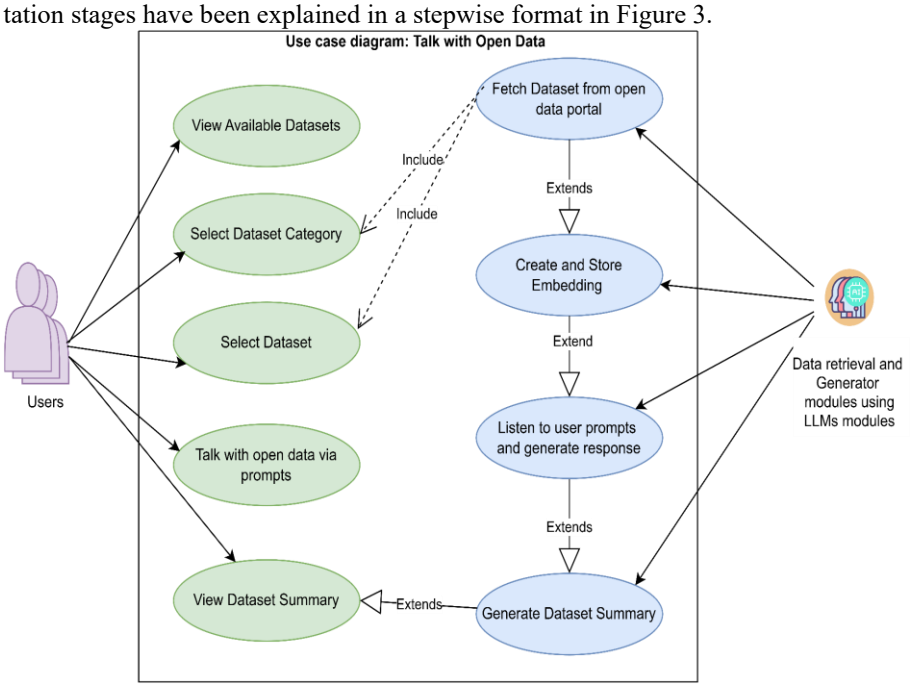

Fig. 2. Use case diagram explaining all the functionalities of the "Enabling User Interaction in Open Government Data" system.

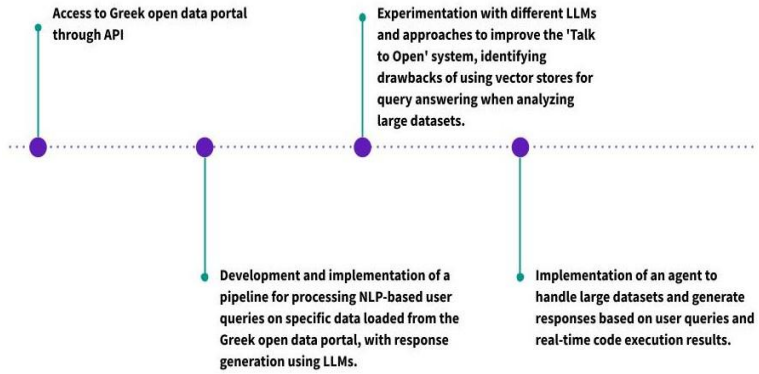

Fig. 3. Implementation stages based on the developed methodology

[6] https://groq.com

4 Results and analysis

The methodology described in the previous section, helped us to execute the experiment on the open-source tool "Kaggle," which provides access to hardware for limited time. First of all, we collected all the API endpoints from the Greek open data portal and added them to a dictionary along with their titles and themes. In this way, users can easily select a dataset theme and title, and our data fetching module will fetch the data and parse it as a DataFrame. This process is shown in Figure 4.

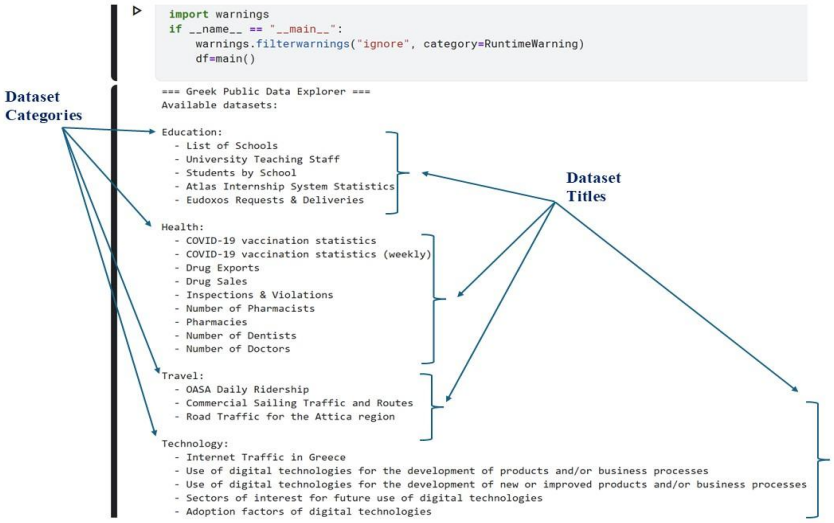

Fig. 4. List of available datasets that can be fetched within the experimental environment (other datasets can also be fetched, but the API endpoints would need to be listed/brought into this environment)

Based on the user's inputs (dataset category and dataset name/title), the module retrieved the proper dataset from the Greek open data portal, converting it to tabular format. On the back end, we managed the collected data, as it requires transformation from API response to DataFrame/JSON and then further usage in the experiment. After fetching the dataset, we generated the embeddings of the tokenised corpus using freely available embeddings from HuggingFace 'all-MiniLM-L6-v2' via the transformer package. We also displayed a few instances of the fetched data. Some of the datasets are large while some others are very long. For the sake of simplicity, we fetched a dataset that only has five records and explains how different factors affect the adoption of digital technologies in Greece. The whole process can be seen in Figure 5. The blue labels explain the inputs/outputs. These embeddings were stored in the FAISS vector store so that we could fetch them later based on the user queries.

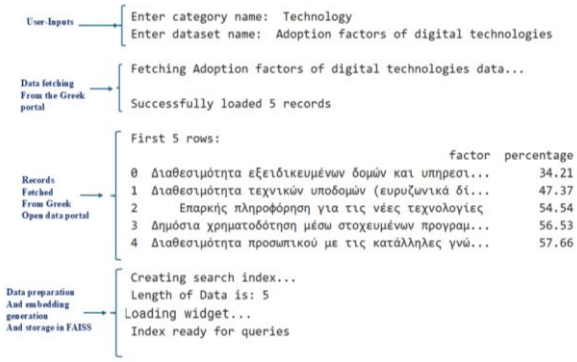

Fig. 5. The user selected the dataset theme (category) and then entered the same name of the data they wanted to learn about.

After dataset embeddings are stored, LLMs (in our case Mistral and LLAMA 3.1) were ready to be used. We used Mistral, but it did not generate the desired results, so we soon moved to LLAMA 3.1 with an 8B parameters setup, and a more deterministic approach by setting the model temperature to 0.0 to get more accurate results based on our dataset. Users asked questions/prompts, as shown in Figure 6. When the user asked a question, the model responded using the whole dataset. First, the model showed a data summary, and then an explanation. The response can vary every time based on the prompt. In this example, the model used a total of 5 records from the dataset along with its schema and summary. In total, 7 records were stored in the FAISS database.

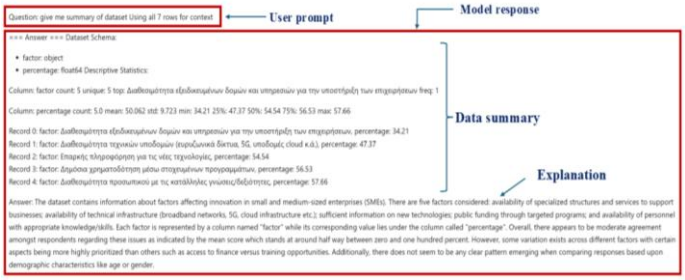

Fig. 6. Users asked a question, and output was generated using the dataset information that the user requested in the previous stage (dataset category: technology, dataset name: Adoption factors of digital technologies). Five actual data rows were used, and to guide the model, we also fetched the data schema and data summary using programming language and stored them in FAISS to guide the explanation generation.

Some datasets have records in the thousands, which understandably require plenty of resources for embedding generation, searching from the vector store, large text generation models and processing power along with RAM. We applied some filtering to utilise our available resources well, such as data chunking. This allows users to fetch the whole dataset or a chunk of it based on the date filtering functionalities supported by the Greek open data portal. This requires date filters—our code, while fetching the data from the Greek open data portal, checks if the data requires date filtering. This process can be seen on Figure 7. The next steps remain the same for the prompts, data summary, and explanation based on user prompts. However, we have noticed that in

our experimental setup, LLAMA 3.1 with fewer parameters, was not able to perform authenticated data aggregation as shown in Figure 8 (e.g., there were not 16 columns in the dataset). For accurate data aggregations and complex questions, it requires an AI agent to perform real-time experiments/coding and execute commands based on user prompts to generate and verify the results' consistency before presenting them to the end user.

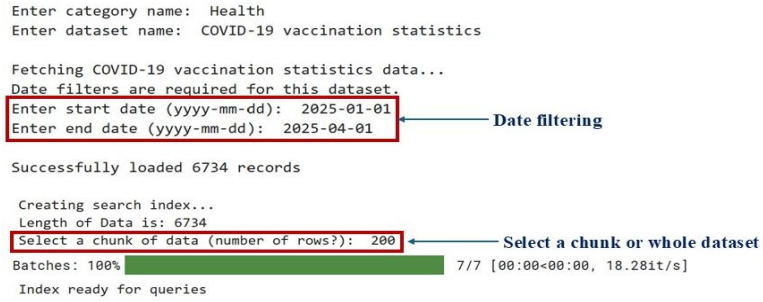

Figure 7: A more complex scenario with date filtering and a larger dataset. The model was not able to preserve the Context information while generating response. See Figure 8.

Question: Give me a statistical summary of the dataset
Using all 202 rows for context
=== Answer ===
The dataset contains 200 records, each record has 16 columns. The dataset has 3 unique values for the column'referencedate', 74 unique values for the column 'area', and 1304 unique values for the column 'areaid'. The column 'daytotal' has a mean of 2.005 and a standard deviation of 7.502426743072923. The column 'daydiff' has a mean of -0.185 and a standard deviation of 10.544034251246385. The column 'totalvaccinations' has a mean of 312846.73 and a standard deviation of 504030.015072672. The column 'totaldistinctpersons' has a mean of 110132.695 and a standard deviation of 172283.73600759776. The column 'dailydose1' has a mean of 0.025 and a standard deviation of 0.18587022002856562. The column 'dailydose2' has a mean of 0.0 and a standard deviation of 0.0. The column 'dailydose3' has a mean of 1.98 and a standard deviation of 7.461095073680041.

Fig. 8. Some erroneous values were returned because Model generated the results based on the embeddings which are not deterministic most of the time for the tabular data, so it indicated clues to move further with the agentic executor chain. However, the model tried to answer in natural language

Experiment with Agentic approach: The embeddings approach is not optimal in some scenarios when data size increases drastically. It leads to inaccurate results (as shown in Figure 7) because the model is not capable of capturing the whole dataset and analyzing it well. However, in this case, LLMs give semantically understandable but not deterministic (correct) answers, since for tabular data, accuracy matters more than the meaning of the individual features. In this way, we expanded the experiment with the help of a Pandas agent from Langchain called "**create_pandas_data-frame_agent**[7]". This agent uses an LLM as a model and then generates and executes code based on the user query, which returns more reliable and accurate results. To

[7]https://python.langchain.com/api_reference/experimental/agents/langchain_experimental.agents.agent_toolkits.pandas.base.create_pandas_dataframe_agent.html

implement the agentic based user interaction with open data functionality, we utilised the Langchain agent, LLAMA -llama3-8b-8192; LLM integrated via the Groq API, and a code execution environment. The example is given in Figure 9.

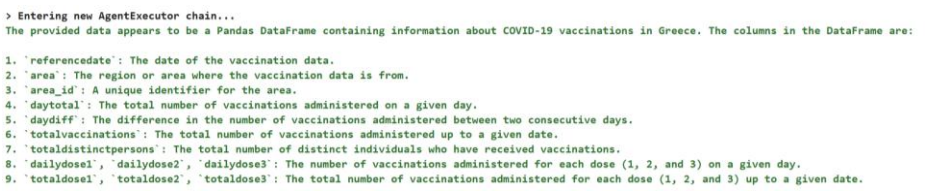

Fig. 9. Query: “Give me a description of the data” and response from the LLM based on code execution or model self-understanding.

In Figure 9, the user asked in natural language, "Give me a description of the dataset." The LLM generated a response by analyzing the dataset under consideration and produced the explanation on its own. It is worth noting that the original dataset did not contain this explanation when it was fed to the model. The model provided the response in a very simple way, explaining each data attribute. This is beneficial for the end-user to understand the data and gain insights quickly from real data.

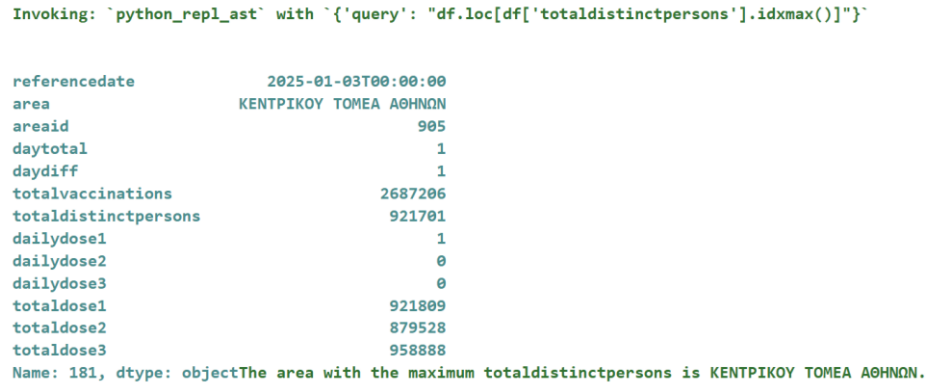

Fig. 10. Query: “How can I find an area with the maximum total distinct persons?” and response generated using LLM’s generated code and executed in the backend.

For proof of concept and validation purposes, we ran the query as demonstrated in Figure 10. We found that it returned the same result as the one produced by the AgentExecutor chain. This result is not for the whole dataset, but it applies just for the chunk that we had loaded into the environment; incl. 200 instances from the health category and the dataset name “Covid-19 vaccination statistics”. At this point, we obtained the results based on actual code generated by the LLMs and executed specifically for the dataset, which returned more accurate results than the semantic embeddings. This offers a benefit in terms of both exploring the dataset and providing an explanation of the output. As depicted in Figure 8, it takes queries in natural language, processes data using coding generated by LLMs and then provides an output of the code and an LLM-based summary. With this approach, end-users can easily understand data within the

open data portal environment, which is subject to integration within open data portals. There are more than one kind of data source and formats published in the open data portals, a diverse interaction to open data platforms will benefit the end-users to find their motivation within open data. For instance, open data use in the education (school level) requires this kind of tool to ask NLP queries within the open data portal to develop interesting open data usage paradigms. Another interesting use could be journalism based on open data; the journalists can utilise this tool to explore the dataset in the easiest way possible.

5 Discussion

This research highlights the opportunities along with the constraints in using LLMs in open data environments. A constraint concerns the technical and semantic interoperability of open data as inconsistencies in formats, schemas, definitions and metadata quality along portals may occur, making it difficult to integrate datasets or reuse code [12–15]. These issues are exacerbated further when aiming for cross domain ontology alignment or even cross-portal access [16]. Another complexity occurs from the processing limitations of current LLMs, when operating on large scale tabular data. Models like LLaMA 3.1 8B have shown promising results when structured data are used with many capabilities such as suggesting analyses, guiding users through datasets and data reasoning support at a basic level [17]. However, they are less capable in terms of handling large aggregations, precise calculations or tasks that require deterministic logic. This limitation becomes even more evident in high volume datasets, where token limits and context compression affect the reliability of the results [18]. In such cases, the solution is not to rely on the LLM's generation capacity but to combine it with code executing agents or intelligent databases [19]. For example, when user queries require statistical summaries or aggregations over thousands of records an architecture may involve (1) an LLM to generate the code, (2) an agent or a database to execute the code, and (3) an LLM to present the results in natural language.

The issue of scalability [20] is also worth noticing, since, while current setups run effectively in limited computer environments, scaling this architecture for production within national open data portals will require optimisation of preprocessing pipelines and model hosting strategies. Hybrid systems that combine retrieval components and external computational tools for heavy lifting may offer useful capabilities. A key takeaway is how open government data APIs can serve as both data sources and interfaces for real time experimentation. Bonding the LLMs and coding agents with open data APIs brings forward a new mode of interacting with public databases, one that feels intuitive and conversational. Rather than sticking to a usual download and analyse circle, users have the opportunity to interact with live data, posing questions with common phrases and receiving personalized answers. This transition opens new possibilities for designing open data platforms that are more focused on the experience of its users. An assistant that is built into the government data portal could help users explore trends, generate charts, or get summaries without necessarily needing to write code or use external tools. For users without a technical background or for users that are not open data

experts, this makes open data more approachable while, for others, it creates a faster, more intuitive path to insight. Such interaction also allows a scope for real-time feedback as responses could be rated or corrected by users, incrementally enhancing the system with the help of reinforcement learning.
In the future this approach may see the development of language models that specialise in national datasets. These models could serve as a means for stakeholders, (e.g., citizens, journalists, researchers, etc.) to access public information quickly in ways that support transparency, inclusion and decision making. They could be trained or fine-tuned on local terminology, administrative data, and domain specific language, to be even more useful and understandable by the stakeholders. Open data portals already offer basic tools for visualisation and download, and this could act as a real-time playground encouraging curiosity, insights generation and more inclusive participation in public data and information ecosystems.

6 Conclusions and future work

In this research, we looked into how LLMs could be combined with open government data systems to support stakeholders in exploring and understanding public open datasets. Through a setup that utilises a RAG pipeline, simple coding executing agents and publicly available LLMs, we built a system where users can ask questions in plain language and receive meaningful insights, without the need of a technical background. Thus, this solution lowered the barrier to entry for non-expert users and offered a new way to engage with open government data, shifting from static downloads to interactive, real-time exploration.

While models like LLaMA 3.1 (8B) performed well in terms of conversational utility and guidance, they lacked the capability to handle more demanding numerical or aggregational tasks. Incorporating code generation agents improved reliability and precision but challenges remain, especially regarding the interoperability of open data schemas, preprocessing efforts, and scalability of the system.

Looking ahead, an important direction would be the development of a dedicated open data LLM model tailored to the Greek open data portal. To move towards this direction, it would be important to take into account how the existing datasets are structured, what kind of information they include, and how much of it can realistically be used by a language model. Using a more powerful model, something beyond 7B or 8B parameters, with a larger context window, could lead to more accurate and consistent results. It would be interesting also to add multimodal capabilities, like generating real-time visualisations or charts that could make the data more accessible to a wider audience.

Future directions of this research may include dedicated open data LLM development for the Greek open data portal, keeping in view the paradigm of LLM model working and available datasets, their quality, and the level of information that can be fed into the model. It would also be beneficial to use a much bigger model (bigger than 7B or 8B parameters) to get more accurate results with a much larger context window size. One other possibility would be to enhance it through multimodal output

generation, like visualisations and charts, to engage a wider audience in the open data (e.g., journalism). In the present study, we only used the LLM's RAG functionality, but it would be a futuristic enhancement of this project to fine-tune a dedicated model on the Greek open data portal and enhance it to much larger datasets, such as datasets from the European data portal.

Acknowledgments. The research presented in this paper has been conducted as part of a project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 955569. The opinions expressed in this document reflect only the author's view and in no way reflect the European Commission's opinions. The European Commission is not responsible for any use that may be made of the information it contains.

Disclosure of Interests. There is no conflict of interest at any level of this research study among authors.

References

1. Ahmed, U., & Polini, A. (2025). Enhancing Open Data Findability: Fine-Tuning LLMs(T5) for Metadata Generation. *Conference on Digital Government Research*, *1*. https://doi.org/10.59490/dgo.2025.941
2. Mamalis, M. E., Kalampokis, E., Karamanou, A., Brimos, P., & Tarabanis, K. (2023). Can Large Language Models Revolutionalize Open Government Data Portals? A Case of Using ChatGPT in statistics.gov.scot. In *Proceedings of the 27th Pan-Hellenic Conference on Progress in Computing and Informatics* (pp. 53–59). Presented at the PCI 2023: 27th Pan-Hellenic Conference on Progress in Computing and Informatics, Lamia Greece: ACM. https://doi.org/10.1145/3635059.3635068
3. Chafetz, H., Saxena, S., & Verhulst, S. G. (2024). A Fourth Wave of Open Data? Exploring the Spectrum of Scenarios for Open Data and Generative AI. arXiv. https://doi.org/10.48550/ARXIV.2405.04333
4. Maratsi, M. I., Gialoussi, N., Alexopoulos, C., & Charalabidis, Y. (2025). A Proposed Methodology for Sub-Ontology Development in Comprehensive Scientific Investigation Methods and Tooling. In M. Sfakakis, E. Garoufallou, M. Damigos, A. Salaba, & C. Papatheodorou (Eds.), *Metadata and Semantic Research* (Vol. 2331, pp. 28–43). Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-81974-2_3
5. Huseynov, R., Nikiforova, A., Symeonidis, D., & Dueñas-Cid, D. (2025). May the Data Be with You: Towards an AI-Powered Semantic Recommender for Unlocking Dark Data.
6. Alexopoulos, C., Ali, M., Maratsi, M. I., Rizun, N., Charalabidis, Y., Loukis, E., & Saxena, S. (2025). Assessing the availability and interoperability of open government data (OGD) supporting sustainable development goals (SDGs) and value creation in the gulf cooperation council (GCC). *Quality & Quantity*, *59*(2), 1573–1593. https://doi.org/10.1007/s11135-024-02025-2
7. Benjira, W., Atigui, F., Bucher, B., Grim-Yefsah, M., & Travers, N. (2025). Automated mapping between SDG indicators and open data: An LLM-augmented knowledge graph approach. *Data and Knowledge Engineering*, *156*. https://doi.org/10.1016/j.datak.2024.102405
8. Maratsi, M. I., Ahmed, U., Alexopoulos, C., Charalabidis, Y., & Polini, A. (2024). Towards Cross-Domain Linking of Data: A Semantic Mapping of Cultural Heritage Ontologies. In

Proceedings of the 25th Annual International Conference on Digital Government Research (pp. 165–176). New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3657054.3657077

9. Knight, S., Dickson-Deane, C., Heggart, K., Kitto, K., Çetindamar Kozanoğlu, D., Maher, D., … Zarrabi, F. (2023). Generative AI in the Australian education system: An open data set of stakeholder recommendations and emerging analysis from a public inquiry. *Australasian Journal of Educational Technology*, *39*(5), 101–124. https://doi.org/10.14742/ajet.8922
10. Lee-Geiller, S., & Ali, M. (2025). Decoding open government policy: Recurrent or innovative liberalism? *JeDEM - eJournal of eDemocracy and Open Government*, *17*(1), 1–32. https://doi.org/10.29379/jedem.v17i1.960
11. Ali, M., Giallousi, N., Melidis, A., Alexopoulos, C., & Charalabidis, Y. (2024). GlossAPI: Architecturing the Greek Data Pile for LLM development. In *Proceedings of the 28th Pan-Hellenic Conference on Progress in Computing and Informatics* (pp. 16–25). Presented at the PCI 2024: 28th Pan-Hellenic Conference on Progress in Computing and Informatics, Athens Greece: ACM. https://doi.org/10.1145/3716554.3716557
12. Cigliano, A., & Fallucchi, F. (2025). The Convergence of Open Data, Linked Data, Ontologies, and Large Language Models: Enabling Next-Generation Knowledge Systems. In M. Sfakakis, E. Garoufallou, M. Damigos, A. Salaba, & C. Papatheodorou (Eds.), *Metadata and Semantic Research* (Vol. 2331, pp. 197–213). Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-81974-2_17
13. Hees, L. V. (2024). *Exploring the utility of large language models for achieving ...* Utrecht University Student Theses Repository, Netherlands.
14. Lehmann, R. (2024). Towards Interoperability of APIs - an LLM-based approach (pp. 29–30). Presented at the MIDDLEWARE 2024 - Proceedings of the 25th ACM International Middleware Conference Demos, Posters and Doctoral Symposium, Part of: MIDDLEWARE 2024. https://doi.org/10.1145/3704440.3704788
15. Xia, Y., Xiao, Z., Jazdi, N., & Weyrich, M. (2024). Generation of Asset Administration Shell With Large Language Model Agents: Toward Semantic Interoperability in Digital Twins in the Context of Industry 4.0. *IEEE Access*, *12*, 84863–84877. https://doi.org/10.1109/ACCESS.2024.3415470
16. Thalhath, N. (2024, September). *A Metadata Application Profile Framework for Actionable Profiling and Semantic Data Publishing*. Graduate School of Library, Information and Media Studies University of Tsukuba.
17. Fang, X., Xu, W., Tan, F. A., Zhang, J., Hu, Z., Qi, Y., … Faloutsos, C. (2024). Large Language Models(LLMs) on Tabular Data: Prediction, Generation, and Understanding -- A Survey. https://doi.org/10.48550/ARXIV.2402.17944
18. Jiang, H., Wu, Q., Luo, X., Li, D., Lin, C.-Y., Yang, Y., & Qiu, L. (2023). LongLLMLingua: Accelerating and Enhancing LLMs in Long Context Scenarios via Prompt Compression. arXiv. https://doi.org/10.48550/ARXIV.2310.06839
19. Li, J., Hui, B., Qu, G., Yang, J., Li, B., Li, B., … Li, Y. (2023). Can LLM Already Serve as A Database Interface? A BIg Bench for Large-Scale Database Grounded Text-to-SQLs. arXiv. https://doi.org/10.48550/ARXIV.2305.03111
20. Villalobos, P., Ho, A., Sevilla, J., Besiroglu, T., Heim, L., & Hobbhahn, M. (2022). Will we run out of data? Limits of LLM scaling based on human-generated data. arXiv. https://doi.org/10.48550/ARXIV.2211.04325