Managing Data Platforms for Smart Cities Using Large Language Models

Marcin Krystek
Doctoral School of PUT & Poznan Supercomputing and Networking Center
Poznan, Poland *marcin.krystek@doctorate.put.poznan.pl*

Mikołaj Basiński
Poznan Supercomputing and Networking Center
Poznan, Poland *mbasinski@man.poznan.pl*

Mikołaj Morzy
Poznan University of Technology
Poznan, Poland *mikolaj.morzy@put.poznan.pl*

Cezary Mazurek
Poznan Supercomputing and Networking Center
Poznan, Poland *cezary.mazurek@pcss.pl*

Abstract

The complexity of data in smart cities creates challenges for developers and hinders cohesive understanding of diverse datasets. These critical data sets are often underutilized due to opaque organization and accessibility. In our research, we use Large Language Models (LLMs) as "data custodians" to improve data navigability and usability in smart city platforms. We evaluated the ability of LLMs to generate accurate data descriptions, identify feature names, and discern relationships from limited raw data, demonstrating their proficiency with minimal input[1].

Keywords: data mesh, smart city, large language model, data platform

1. Introduction

Smart cities promise modern urban management by seamlessly integrating diverse data streams. This vision requires a fundamental redesign of the data platforms to ensure flexibility, openness, and utility. However, the sheer volume and diversity of the data pose significant challenges in maintaining a coherent and up-to-date perspective. The deployment of large language models offers a promising strategy to automatically discern the structure and properties of the data.

The *data mesh* architecture, emphasizing decentralized data ownership and self-service design, is ideal for smart city data platforms. Data mesh manages diverse datasets from various actors, unlike traditional centralized architectures. The data mesh improves scalability, agility, and resilience by allowing individual domains to manage their data, improving accessibility, and governance. It ensures that the data is findable, accessible, interoperable, and reusable (FAIR), facilitating seamless sharing across smart city services. A truly smart city relies on democratized access to its data infrastructure to foster agile application development that improves urban living. Ensuring easy and flexible access to datasets and their descriptions is crucial to allowing developers, researchers, and citizens to contribute to innovative applications and unlock the full potential of smart city initiatives.

[1] Due to page limits, detailed results, code listings and examples of model QAs are presented in the supplementary material available at `https://github.com/megaduks/isd24`

2. Data Platforms for Smart Cities

In the digital era, smart cities aim to use data to improve urban living and efficiency, posing the challenge of building effective data platforms to manage and analyze vast data. Traditional architectures cannot keep up with advances in technology and evolving stakeholder needs. The *data mesh* paradigm addresses these challenges by reimagining the foundations of smart city data platforms. Building *data platforms for smart cities* (DPSC) presents challenges in technology [2], usability [3], sustainability [4], and governance [5]. Technological innovation creates opportunities and complicates data storage, processing, and sharing. Usability issues arise because platforms lack flexibility to meet evolving user needs. Sustainability concerns require designing for present and long-term viability. Governance balances control and flexibility to ensure data quality and accessibility [9].

The data mesh paradigm, grounded in four basic tenets, offers a fresh perspective on the DPSC challenges [7]. It advocates a data-centric approach, emphasizing data as a product rather than focusing on technology-driven platforms. Sociologically, it ensures that data products are discoverable, understandable, addressable, interoperable, secure, trustworthy, and valuable, serving diverse smart city needs. Technologically, it leverages cloud platforms and virtualization for scalability and flexibility. From an infrastructure standpoint, it empowers data management teams with self-service capabilities, fostering agility and responsiveness. Introducing data mesh in smart cities requires considering the unique environment, governance, and organizational structures. Key best practices include forming a technology team proficient in data mesh principles, focusing on developing data products, fostering a data-sharing culture, establishing clear governance rules, and exploring data product monetization to sustain operations [6]. The data mesh paradigm offers a significant leap in the development of effective and sustainable data platforms for smart cities. By focusing on data and embracing flexibility, stakeholder collaboration, and technological agility, it overcomes traditional data architecture challenges. As smart cities evolve, the data mesh addresses current needs and supports future innovations, crucial for realizing the potential of digital transformation for a smarter and more connected urban future.

3. Challenges in DPSC Management

Building a smart cities data platform (DPSC) is challenging due to the complexity and heterogeneity of the data involved [11]. Data from multiple sources use various protocols, formats, and technologies [8]. This is further complicated by accommodating multiple tenants with different management strategies and requirements, making large-scale data management and harmonization difficult and affecting efficiency and scalability [10]. A primary issue is the lack of standardized descriptions, causing ambiguity and hindering data utilization. Inconsistent feature naming further complicates data handling and analysis, leading to inefficiencies in integration and application development.

Maintaining real-time data across city operations such as traffic and utilities is a challenge. The large data volume and processing power required often make conventional methods inadequate, causing potential lags that compromise decision-making and services. Reacting to dynamic attributes (e.g., traffic flow, energy consumption) demands highly adaptive algorithms and substantial computational resources, complicating platform deployment and scaling. Furthermore, producing robust, secure, and flexible universal APIs to meet diverse stakeholder needs is a significant technical hurdle, requiring handling various data types and usage scenarios without compromising platform performance. Implementing a DPSC is also challenging due to developers' struggles to understand the complex data ecosystem. Without adequate tools, they must query administrators, depending on individual availability and expertise. Maintaining coherent dataset descriptions is also difficult due to the sheer number and diversity of data sources and varying quality levels, complicating the establishment of uniformity and clarity.

4. Large Language Models in Structured Data Analysis

The generative AI revolution is transforming data analysis and content creation, with ongoing advancements from various leaders in the field. This revolution started with OpenAI's GPT in June 2018, followed by GPT-2 in February 2019, and GPT-3 in June 2020. Google released Meena in 2020 and LaMDA in May 2021, while Meta introduced OPT-175B as an open-source model in May 2022. Hugging Face's BLOOM, launched in July 2022, exemplified AI community collaboration. In 2023, major advancements included AI21 Labs' Jurassic-1 Jumbo, Google's Bard and Gemini, Anthropic's Claude, and Meta's LLaMA, all pushing the boundaries of reasoning and dialogue capabilities. Most recently, cutting-edge models like OpenAI's GPT-4, Google's DeepMind's Gemini 2, and Meta's LLaMA 3 continue to advance the field, enhancing the versatility and sophistication of AI-driven applications.

Large language models have proven highly effective in analyzing structured data, a task traditionally handled by conventional statistical and machine learning techniques. By interpreting structured data in natural language or through pre-processing, LLMs can infer relationships, trends, and insights, generate summaries, and identify anomalies within the data. LLMs excel in nonstandard tasks that require a "human" understanding of the semantics of the data set beyond statistical correlations. They identify patterns and generate predictions rooted in logical and thematic coherence. While LLMs' ability to handle human-like text is well documented, their proficiency with tabular data, common in DPSC scenarios, is less explored. This paper evaluates LLMs' understanding of tabular data structures and their potential as custodians of large datasets. We investigated their ability to provide dataset summaries, recommend intuitive feature names, and suggest relevant datasets for specific queries or business needs.

5. Experiments

5.1. Experiment 1: identify and label concepts

The initial experiment aims to assess whether LLM can accurately identify and label concepts in a database from a small sample. The LLM receives raw data without clues about attribute importance, semantics, or relationships. Pseudonymization was applied by masking attribute names with four-character base64 codes generated by the `openssl` tool. To minimize hallucinations, the general context of each dataset was provided. In the experiment, `gpt-4-turbo` [1] was given 50 random tuples from a table and a minimalist description for context. The LLM's tasks included identifying feature semantics, proposing feature names, elucidating feature relationships, and synthesizing a concise dataset description. Accessed via OpenAI's API using the `langchain` library, the experiment took place in late March 2024.

The LLM was tested on four datasets from the DPSC of the city of Poznan: *Graves* (graves located in the city cemetery), *Address Points* (address points, their locations, categories and affiliation to various functional areas of the city), *Bike Stations* (location and current occupancy status of city bike stations), and *MPK Stops* (public transport stops, their location, categories, types of vehicles, and possible transfers). For each data set, LLM was asked to provide the characteristics of the dataset, but the query explicitly forced the model to focus on feature values and their relationships, while ignoring the order of features when determining the relationship between them. Table 1 presents the summary of meta-data generation for all tables. The query provided requirements for the output format and contents.

> Analyze following JSON document where each feature represents a description of a single address point in the city of Poznań, Poland. Analyze the values and relationships between all attributes. The position of the attributes in the document does not matter. For all attributes in section, properties suggest new descriptive names. The new names must explain the meaning of the attribute value and make it easier to understand what each attribute represents. For each attribute suggest the most appropriate data type. Leave the old name if there is a lack of data or there is a very low certainty of the concept identification. As a result print triples using the following pattern: old attribute name, new attribute name in snake case, data type, attribute interpretation.

Table 1. Generation of feature descriptions using the LLM, A: no. of features, B: no. of empty features, C: no. of recognized features, D: no. of correctly recognized features, E: % of recognized features, F: % of non-empty recognized features, G: % of correctly recognized features

dataset	**A**	**B**	**C**	**D**	**E**	**F**	**G**
Graves	18	5	11	8	61,11%	84,62%	72,73%
Address Points	22	1	14	13	63,64%	66,67%	92,86%
MPK Stops	5	0	5	4	100,00%	100,00%	80,00%
Bike Stations	5	0	5	3	100,00%	100,00%	60,00%

5.2. Experiment 2: generate human like descriptions

The second experiment aimed to evaluate the LLM's ability to mimic a domain expert by generating an informative dataset description. This should include a summary of the meanings of the attributes, the domain of the dataset, and potential application areas. The LLM received a description of the structure of the dataset with attribute names, types, and brief descriptions if the attributes were not self-explanatory. The description of the data set was prepared in the form of a JSON document (see supplementary material). For the experiment, an expert created descriptions for 18 urban services datasets in Poznan, Poland (see supplementary material). Fifteen parameterized queries, grouped into four categories, were designed to produce dataset descriptions that aligned with the experiments' goals. Query responses were stored in a text file for validation and in a vector database for subsequent experiments. The queries were aimed at uncovering the data structure, generating feature descriptions, identifying the data domain, and exploring future analytic opportunities. Below is the LLM's response to a query for the *Bike Stations* dataset (for more examples see the supplementary material).

Q: *Identify all features available in the Bike Stations dataset. Use feature titles to understand their meaning and discover possible categories to which each feature could be assigned. The example categories could be location address or geospatial characteristics. List the categories together with the city bike station dataset features assigned to each category.*

A: *Features categories: (1) Geospatial characteristics: geometry (coordinates) (2) Bike station details: bike_racks, bikes, free_racks, label (3) Time-related information: updated*

5.3. Experiment 3: support in decision making

Experiment 2 generated detailed descriptions of 18 datasets, including attributes and characteristics, providing domain context, and suggesting future applications. Experiment 3 aimed to assess whether the LLM could help analysts select suitable data sets for specific research scenarios. Assuming to assist professional analysts or statisticians, the experiment involved nine queries, each representing a research issue. Each query had three parts: defining the user's role, detailing the analytical problem and objective, and providing model guidelines for the source information and response format. The questions ensured that the analytical task's content did not directly reference dataset names. Due to the lack of space, examples of queries and model's responses are presented in the supplementary material.

6. Discussion

Our observations show that when prompts are clear, the LLM can accurately contextualize and interpret the data. For example, the model correctly identified birth, death and burial dates in the *Graves* dataset. However, initial tests revealed a flaw: the model misnamed attributes, aligning them incorrectly due to their original sequence labels (*g1*, *g2*, *g3*). We solved this by assigning unique random strings to each attribute and decoupling them from their sequence. This eliminated naming ambiguities, indicating that the model's accuracy improves when freed from biases introduced by conventional labeling. The model's responses are heavily influenced by the source data's structure, but this can be mitigated by modifying the instructions to discourage reliance on data structure or by altering the input data structure. This is crucial when aiming to shift the model's focus from structural aspects to attribute-value-driven responses. It underscores the importance of strategically designing both the model's instructions and data representation, especially when the focus is on intrinsic attributes over organizational structure.

A single value attribute in a data set does not inherently hinder interpretability if the contextual significance is clear. In contrast, completely unique attributes, such as consecutive natural numbers, do not necessarily enhance interpretability. This shows that uniqueness alone is not sufficient for clarity. Instead, meaningful and contextually relevant information is crucial. This challenges the assumption that higher variability or uniqueness is directly correlated with interpretability. Our observations show variability in system responses, especially with interpretative queries. This randomness was notable with the variable *graveyard_id*, where decisions were justifiable but inconsistent. The queries about dates and names were more repeatable due to their clear nature. Generally, the model excels at recognizing and classifying categorical values compared to numerical ones. Despite limitations with isolated numerical data, the LLM effectively interprets numerical columns near categorical attributes when the context clarifies their meaning, as seen in its accurate identification of house numbers within address data.

The main goal of our research was to reduce entry barriers for DPSC application development. Initial findings suggest that LLMs can serve as accessible, cost-effective developer tools, simplifying interactions with the data platform's complexity. Our research aims to speed up urban data application development, enhancing smart city life. Next, we plan to move from proof-of-concept to practical application by developing a conversational interface for DPSC, focusing on testing its usability and efficiency with end-users in real-world scenarios.

References

1. Achiam, J., et al.: GPT-4 technical report. arXiv preprint arXiv:2303.08774 (2023)
2. Mazurek, C., Stroinski, M.: Technology pillars for digital transformation of cities based on open software architecture for end2end data streaming. (2022)
3. Sarker, I.H.: Smart City Data Science: Towards data-driven smart cities with open research issues. Internet of Things 19, 100528 (2022)
4. Lavalle, A., et al.: Improving sustainability of smart cities through visualization techniques for big data from IoT devices. Sustainability 12(14), 5595 (2020)
5. Pereira, G.V., et al.: Smart governance in the context of smart cities: A literature review. Information Polity 23(2), 143-162 (2018)
6. Moustaka, V., Vakali, A., Anthopoulos, L.G.: A systematic review for smart city data analytics. ACM Computing Surveys 51(5), 1-41 (2018)
7. Krystek, M., et al.: Introducing Data Mesh Paradigm for Smart City Platforms Design. (2023)
8. Silva, B.N., Khan, M., Han, K.: Towards sustainable smart cities: A review of trends, architectures, components, and open challenges in smart cities. Sustainable cities and society 38, 697-713 (2018)
9. Micheli, M., et al.: Emerging models of data governance in the age of datafication. Big Data & Society 7(2), 2053951720948087 (2020)
10. Lai, C.S., et al.: A review of technical standards for smart cities. Clean Technologies 2(3), 290-310, (2020)
11. Barns, S.: Smart cities and urban data platforms: Designing interfaces for smart governance. City, culture and society 12, 5-12 (2018)