KCitychatBot: A knowledge graph based chatbot system for large-scale CityGML dataset

Shou-qi Liu[1,2], Chen Wang [1,2]

[1] Anhui Province Key Laboratory of Wetland Ecosystem Protection and Restoration, Anhui University, Hefei 230601, China - chen.wang@ahu.edu.cn;

[2] School of Resources and Environmental Engineering, Anhui University, Hefei 230601, China – chen.wang@ahu.edu.cn

Keywords: Chatbot, CityGML, Digital twin, Natural language processing, Artificial intelligence

Abstract

CityGML has been extensively studied due to its widespread use across various domains. However, its complex hierarchical structure still presents challenges for non-expert users. Recently, large language models (LLMs) have demonstrated significant capabilities in natural language processing (NLP) and chatbot systems. Nevertheless, LLMs heavily rely on pre-trained data, which can lead to hallucination issues and limitations in context length. To address these challenges, we first propose a novel automatic method for transforming CityGML data into knowledge graphs by leveraging a graph database and a transformation plugin. This approach effectively addresses the difficulties of storing and representing the complex structure of CityGML and can serve as an external knowledge base for chatbot systems. Second, we develop a collaborative multi-agent framework that enables natural language queries over CityGML data in a user-friendly manner. By integrating the constructed knowledge graphs with several knowledge augmentation strategies, the chatbot system implements a complete pipeline from natural language input to structured query generation, external knowledge retrieval, and optimized response generation. We conduct experiments on both the city knowledge graphs and the chatbot system to evaluate the accuracy of the knowledge graphs and the interpretability of the system's outputs. The experimental results demonstrate that the generated knowledge graph is accurate, and the chatbot system performs well in terms of answer accuracy, relevance, and contextual coherence. These findings highlight the potential of the proposed chatbot system to lower the barrier for non-professionals interacting with CityGML data, offering both theoretical insights and practical implications for advancing CityGML applications in the era of LLMs and promoting smart city development.

1. Introduction

CityGML (City Geography Markup Language) is one important 3D city model standard supporting the creation of comprehensive digital twin of cities. It has been extensively applied in GIS (Geographic Information System), BIM (Building Information Model) and other domains (Tan et al., 2023). However, despite its importance and utility, working with CityGML still remains a challenging task for users, especially those without a technical background (Nguyen et al., 2020). The format is complex, containing intricate hierarchical relationships between entities such as buildings, roads, and infrastructure. The size of these datasets, often spanning entire cities, further exacerbates the problem, making them difficult to query, analyse, and interpret without specialized knowledge, tools, and expertise.

Current methods for interacting with CityGML involve GIS platforms or customized software tools. These tools typically rely on relational spatial databases like PostGIS along with querying languages like SQL (Chadzynski et al., 2023). They often demand significant time and expertise to master. Consequently, users who are not GIS professionals, such as urban planners, architects, or local government officials, may find it difficult to extract meaningful insights from large-scale CityGML datasets.

To address these challenges, we propose a novel solution named KCitychatBot: a knowledge graph-based chatbot system that leverages Large Language Models (LLMs) and multi-agents to provide an intuitive, conversational natural language interface for querying and analysing large-scale CityGML data. KCitychatBot enables users to interact with complex CityGML urban models using ordinary natural languages as interface. The test demonstrated the usability of the proposed solution. This system not only simplifies querying but also encourages CityGML users without a technical background to ask questions, which can be particularly valuable in urban planning and decision-making processes with large-scale CityGML dataset.

2. Related Work

The steep learning curve of traditional GIS platforms has driven research into developing new interaction methods, such as graphic workflows (Nguyen et al., 2020) or chatbots (Saka et al., 2023), to lower the barrier for interacting with geospatial data. Among these methods, geospatial chatbots are often considered the most intuitive. However, early solutions (Tsai et al., 2019) had limited abilities to handle large, complex geospatial datasets due to their reliance on simple keyword-based queries. Recent developments of LLMs offer opportunities for advancing the capabilities of geospatial chatbots. The use of LLMs for geospatial tasks has already demonstrated promising results with 2D geospatial data (Wang et al., 2024). However, LLMs come with limitations that need to be addressed for large-scale geospatial applications. One key issue is the token input limit. LLMs like GPT-4.5 have a restricted number of tokens that can be processed in a single query. Another key issue is hallucination in LLMs, which refers to the model's tendency to generate plausible-sounding but incorrect or fabricated information.

Aiming at these limitations, some knowledge augmentation strategies in geospatial chatbots have shown promising results but also highlight the complexity of synchronizing LLMs with large-scale geospatial datasets through RAG (Yu et al., 2025) and knowledge graph (Dang et al., 2025). In this context, many researchers have emphasized the critical role of knowledge graphs in enhancing the interpretability and factual accuracy of chatbot systems (Luo et al., 2022). Knowledge graphs could not only enhance the chatbot's ability to interpret and process

complex queries but also allow for more advanced reasoning capabilities (Agrawal et al., 2023). They may help mitigate the LLM chatbot hallucination issue by providing a structured and accurate reference for the model. Recently, the integration of multi-agent systems with knowledge augmentation methods (Guo et al., 2024; Xu et al., 2024) has emerged as a promising research trend, it has the potential to enhance the chatbot system robustness in handling complex situations and offer new opportunities for processing and interacting with CityGML data.

CityGML incorporates hybrid semantic and spatial information, facilitating a comprehensive representation of city features. However, this multi-dimensional data has not yet been effectively integrated with powerful LLMs in chatbot systems for data exploration and decision making. To bridge the research gap and considering the possibilities provided by knowledge augmented multi-agent system, this paper proposes a knowledge graph-based chatbot system, hoping to provide a practical CityGML chatbot and acquiring theoretical insights for CityGML applications in the era of LLMs.

3. Methodology

This section elaborates the overall construction and testing workflow of the proposed KCitychatBot. As shown in Figure 1. We divide all the methodology into two modules: (1) the construction of knowledge graphs and (2) the design of collaborative multi-agent chatbot. Both modules adopt the 'design and evaluation' methodology to ensure systematic development and rigorous validation.

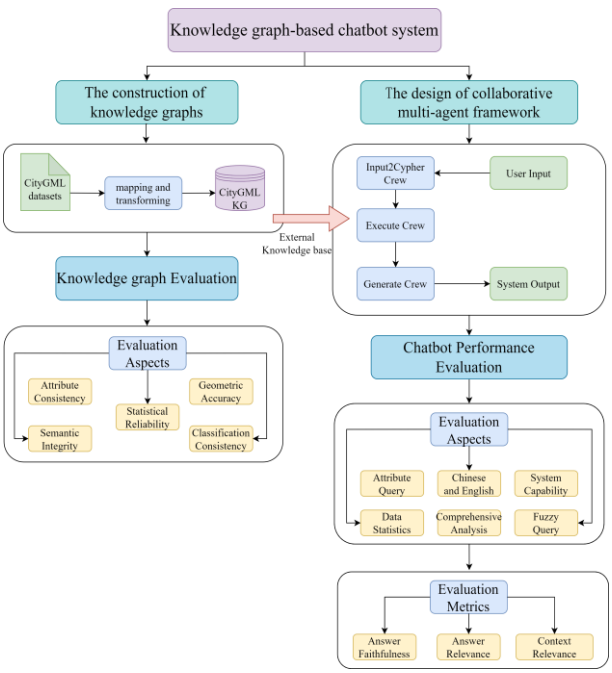

Figure 1. The construction and testing workflow of the proposed knowledge graph-based chatbot system

3.1 City Knowledge Graph Construction

3.1.1 Structure of CityGML: As an extension of XML (Extensible Markup Language), CityGML inherits its hierarchical labels, which represents data in a tree structure. However, CityGML also uses identifiers (such as XLinks) to build cross-references between objects**,** resulting in an overall cyclic graph data structure (Nguyen et al., 2020). Figure 2 illustrates the 'bldg' theme's hierarchical structure, expanding from building objects through LOD 0 and LOD 1 levels to geometric and material attributes, forming a cyclic graph with over six hierarchical levels. This cyclic graph structure is incompatible with the tabular model of traditional relational databases, which requires complex table decomposition and relationship design during storage. In contrast, the native graph structure of graph databases not only align closely with the CityGML but also provide a more efficient retrieval method.

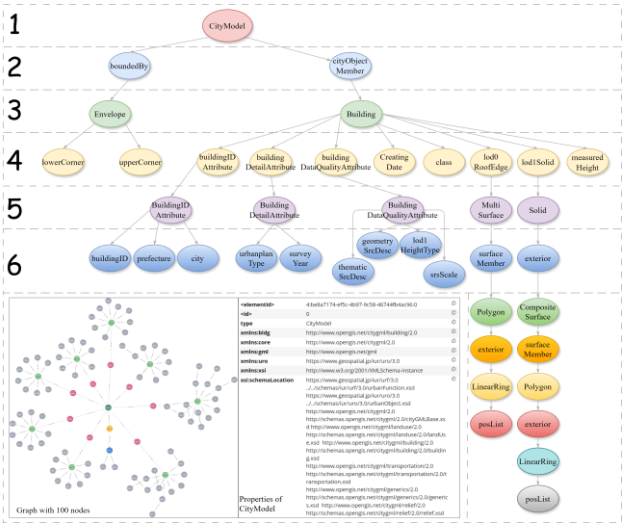

Figure 2. An example of CityGML and its corresponding graph

3.1.2 City Knowledge Graph Construction Method: Based on the structural characteristics of cyclic graphs in CityGML (Nguyen et al., 2020), we propose a hierarchical knowledge graph construction method tailored for CityGML, as illustrated in Figure 3. The proposed method is implemented with a graph database plugin that can convert a given input CityGML dataset into a hierarchical JSON object, where all entities and attributes are encapsulated within a container. Specifically, each level's label is mapped to the '_type' attribute, and corresponding attribute values are stored in the '_text' attribute, while nested child structures are embedded within the '_children' attribute. For the entities and attributes have transformed, this method performs the following three steps: (1) it separates the container into two parts: basic entities, which only contain '_type' and '_text' attributes, and subset containers, including the nested structures. (2) the method instantiates basic entities as graph nodes, with their '_type' and '_text' mapped to node labels and properties, respectively. For subset containers, we recursively processes their '_children' attribute until all leaf nodes are reached. This process gradually unfolds the nested structure and generates new basic entities and subset containers along the way. (3) Depending on the labels of each child node, the method automatically establishes the relationships between parent and child nodes. At each recursive step, all nodes at the same hierarchical level are instantiated and linked to their respective parent nodes. Once all leaf nodes have been processed and instantiated, the city knowledge graph is complete.

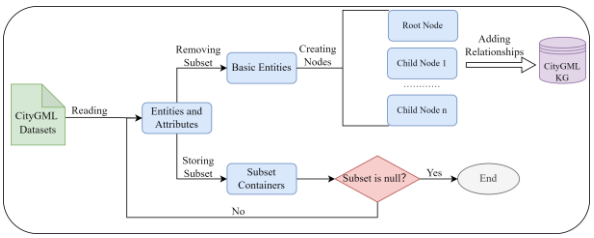

Figure 3. City knowledge graph creation workflow

3.2 Multi-agent Chatbot System

The design of a multi-agent chatbot system faces two key challenges: (1) how to adapt pre-trained LLM without fine-tuning to CityGML Knowledge base and CityGML-related natural language queries, and (2) how to organize multiple LLM agents, each with unique roles, to function collaboratively within the chatbot system. Due to the heterogeneity of different CityGML datasets, the constructed knowledge graphs have unique labels, properties and relationships. However, LLMs without fine-tuning often lack sufficient understanding of the knowledge graph schema, resulting in unreliable response. To address this issue and improve query precision, we incorporate knowledge augmentation methods, saying embedding model and few-shot learning into the design. These methods not only improve the accuracy of generated queries but also offer acceptable output for subsequent tasks.

3.2.1 Embedding Model: Embedding models are widely used in deep learning and question answering systems. They can also be applied to knowledge graphs for link prediction and other downstream tasks (Ge et al., 2024). We employ embedding models to extract user intents, labels, and properties as shown in Figure 4. Firstly, we collect some external knowledge from the CityGML datasets, they may originate from various sources such as CSV files, JSON documents, plain text, or other structured and unstructured formats. Combined with user input (typically in text form), all content is segmented into smaller chunks for further processing. Secondly, each chunk is mapped into a high-dimensional vector through neural network models depending on data types. Finally, all the vectors are compared using cosine similarity to identify the most relevant match. Based on the most similar vector, the system retrieves relevant information from the external knowledge, thereby enabling a better understanding of user input.

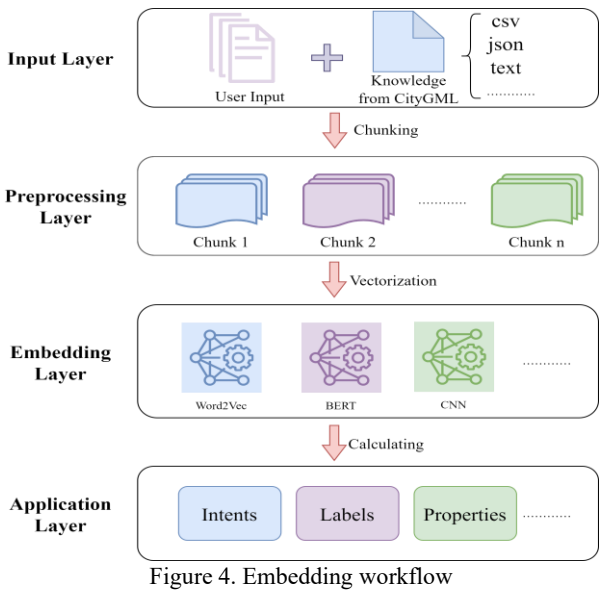

Figure 4. Embedding workflow

3.2.2 Few-shot Learning: Few-shot learning strategy is a commonly adopted training paradigm when large-scale training data is unavailable (Brown et al., 2020). It enables the model to infer class membership based on only a few examples, without requiring comprehensive prior knowledge of the target classes. We utilize this strategy to convertnatural language input intostructured graph database queries. Figure 5 shown the conversion of natural language input to a structured Cypher query (a query language for graph databases). Guided by the **extract prompt**, the user input is first converted into a standardized schema that captures the user's intent, entity labels, and attribute-value pairs. Then this schema serves as the input for the next step, with the query prompt generating an appropriate query.

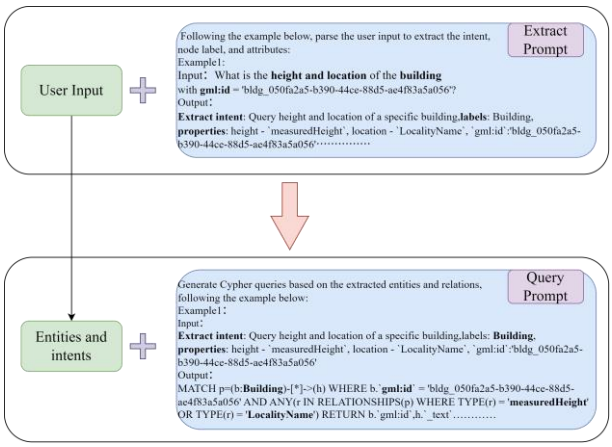

Figure 5. An example of few-shot learning for graph query

3.2.3 Multi-agent Framework: We design a three-crew framework for the CityGML chatbot. The functionalities of each intelligent agent are as follows.

Input2Cypher Crew: This agent handles conversations with users and automatically constructs Cypher queries based on the dialogue. It manages two sub-tasks: Extract_Task and Query_Task. The Extract_Task receives domain knowledge from the CityGML knowledge graph and analyses user input, processing it through an embedding model to extract user intent, relevantlabels, and attribute information. Based on the results of Extract_Task, the Query_Task uses multiple preset template cases to automatically construct appropriate Cypher queries tailored to the city knowledge graph.

Execute Crew: This agent executes the Cypher queries generated by the Input2Cypher Crew and validates the query results from the knowledge graph. The system incorporates a three-iteration mechanism to mitigate the inherent randomness of LLMs. If the query result is empty or contains a syntax error, the process returns to the Input2Cypher Crew for further refinement.

Generate Crew: This agent processes the query results provided by the Execute Crew and converts them into natural language expressions aligned with human communication conventions.

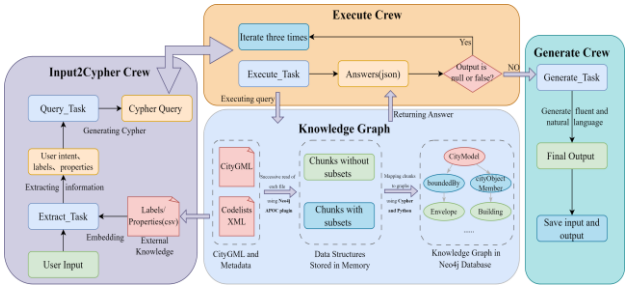

Figure 6. The structure of the multi-agent framework

When users interact with the system, the Input2Cypher Crew transforms the conversation input into a structured Cypher query, which is passed to the Execute Crew. The Execute Crew retrieves results from the knowledge graph. If valid results are returned, they are forwarded to the Generate Crew to produce natural language responses; otherwise, the results are sent back to the Input2Cypher Crew to iterate and refine the query generation process. Additionally, if users engage in multi-round conversations, the Generate Crew concatenates user inputs and system outputs into a contextual list to maintain coherence for subsequent interactions.

4. Implementation

4.1 City Knowledge Graph

We utilize the PLATEAU dataset from Japan (Seto et al., 2023) to construct city knowledge graphs. Considering the limited computing resources, we arbitrarily select 14 regions within Tokyo, covering 12 different CityGML themes (such as “bldg”, “brid”, and “dem”, etc). The total size of the dataset is approximately 124 GB, and additional information about the constructed knowledge graph can be found in Table 1.

Name	CityGML dataset
Dataset size	124GB
Region	[“Toshima City”, “Minato City”, “Tokyo Metropolis”, “Chiyoda City”, “Chuo City”, “Bunkyo City”, “Taito City”, “Sumida City”, “Koto City”, “Shinagawa City”, “Meguro City”, “Ota City”, “Setagaya City”, “Shibuya City”]
Theme	[“bldg”, “brid”, “dem”, “fld”, “frn”, “htd”, “lsld”, “luse”, “tran”, “ubld”, “urf”, “veg”]
Knowledge graph size	281.75GB
Nodes count	1,420,375,552
Construction time	10,915.218s

Table 1. Research data and knowledge graph information

Based on the Neo4j graph database plugin (APOC) and Python language, we develop two automatic knowledge graph construction modes: (1) Single-file parsing (incremental update) and (2) Batch-processing (full construction). In the single-file mode, the system first evaluates the file size; if exceeding a pre-defined threshold (300 MB under a 32 GB memory condition), the file is partitioned according to a splitting criterion set to one-tenth of the number of file elements, and then the segmented files are read sequentially. For the folder batch-processing mode, the system obtains the absolute paths of all GML files within the current folder, stores them in a list, and processes them sequentially using the single-file mode. To test scalability, we conducted knowledge graph construction experiments under consistent conditions (32 GB memory, 24-core processor). Table 2 presents average results from five repeated experiments, indicating that our hierarchical method effectively handles datasets of varying scales with acceptable construction time and storage consumption.

Dataset size (MB)	Knowledge graph size (MB)	Nodes count	Construction time (s)
0.42	2.12	4,343	0.731
5.71	17.08	65,080	3.219
51.2	154.98	601,170	9.321
291	972.88	4,440,942	43.776
1,689.6	5,120	19,947,887	196.124
126,976	288,512	1,420,375,552	10,915.218

Table 2. Knowledge graph construction experiment results

4.2 Chatbot Prototype System

This paper leverages the CrewAI framework, Deepseek API, Neo4j graph database to develop a collaborative multi-agent chatbot prototype system. Specifically, we employ the Deepseek-R1 model within the Input2Cypher Crew to construct structured queries, leveraging its strong reasoning capabilities. Meanwhile, the Generate Crew utilizes the Deepseek-V3 model, which demonstrates stronger capabilities in handling daily conversations. The backend is primarily developed in Python, while the frontend is implemented using JavaScript. Besides, FastAPI is employed as the interface framework to connect the frontend and backend, enabling efficient information exchange between the two components. Figure 7 illustrates the layout of chatbot system, it mainly consists of two buttons and a dialogue box. The ‘Connection’ button enables this system to establish a link with an external graph database. Users are required to provide the URL (Uniform Resource Locator), username, database name, and password. The ‘Loading’ button allows users to import CityGML data into the graph database and construct city knowledge graphs. Users can enter questions into the input box, and the system’s response will be promptly displayed in the dialog box.

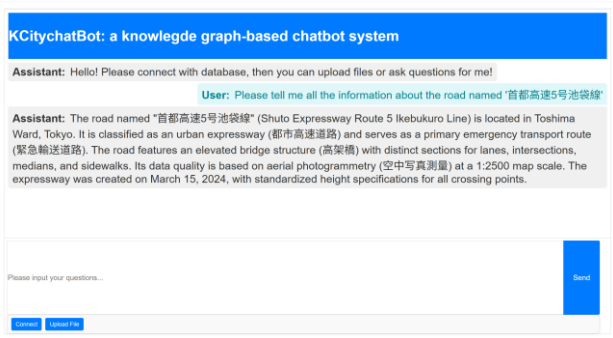

Figure 7. The layout of the proposed chatbot system

As shown in Figure 8, a user can directly request all available information regarding the road ‘首都高速 5 号池袋線’ (Metropolitan Expressway Route 5 Ikebukuro Line). The process begins with the Extract_Task, which identifies the user’s intent, extracts the ‘Road’ label, and infers that the road name may be stored under the ‘_text’ property of the ‘Road’ label. Under few-shot prompting, the Query_Task generates a Cypher query by first locating the root node labelled as ‘Road’, then traversing its child nodes, and finally returning both the root and selected child nodes that contain relevant road information. Next, the

Execute_Task receives the Cypher query from the Query_Task, executes it in the graph database, and returns the results in JSON format. Subsequently, the Generate_Task interprets the returned data and formulates a response in fluent natural language. The system promptly identifies the road's location as 'Toshima City', classifies its usage as a 'Primary Emergency Transport Road', and provides additional contextual information.

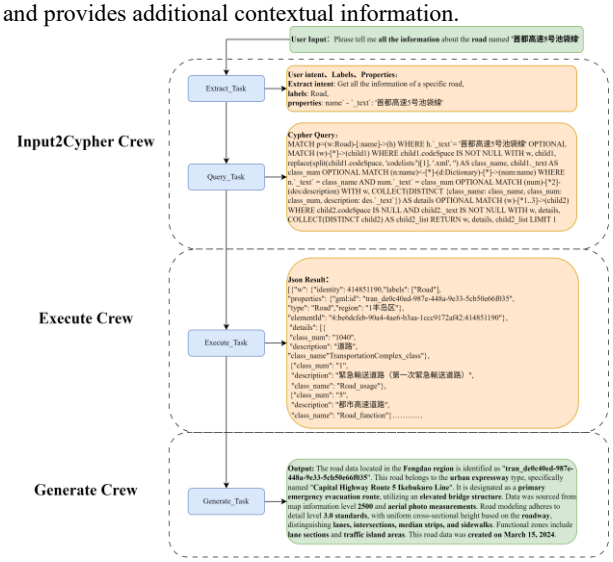

Figure 8. A dialogue example

5. Evaluation

5.1 Knowledge Graph Evaluation

To assess the accuracy of the knowledge graph, we compare query results obtained from both the 3D CityDB (SQL) and Neo4j (Cypher) that correspond to the same testqueries. Inspired by the comparison between SQL and NoSQL databases (Khan et al., 2019), we categorize the evaluation into five dimensions, with their names and descriptions provided in Table 3. The comparison demonstrates identical outcomes from both querying approaches, thereby confirming the validity of the constructed city knowledge graph and indirectly validating the effectiveness of our graph construction method.

Name	Description
Attribute Consistency	Consistency of attribute values across different databases, reflects potential data loss during converting
Semantic Integrity	Preservation of semantic information during converting, indicates structural and meaning completeness
Statistical Reliability	Accuracy of aggregate query results, reflects stability and trustworthiness in statistical tasks
Geometric Accuracy	Precision of geometric objects, such as coordinates, area, and topological relationships
Classification Consistency	Correct retention and identification of feature categories and codes after converting

Table 3. Knowledge graph evaluation criteria and descriptions

Figure 9 gives an example to present the comparison before and after knowledge graph construction. The structure of CityGML is fully preserved during the conversion process, and hierarchical labels and intra-label attributes accurately recorded as node labels and properties.

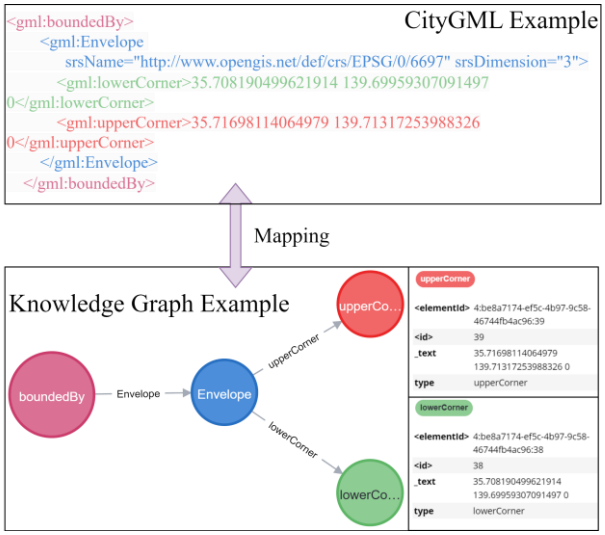

Figure 9. Comparison before and after CityGML to knowledge graph conversion

5.2 Chatbot Performance Evaluation

To evaluate chatbot performance, we built a test set with 180 cases covering diverse question types, and utilized the Ragas evaluation tool (Es et al., 2024). This tool proposes three metrics to evaluate the performance of RAG-based systems. First is the Answer Faithfulness (AF), which measures the consistency between the generated answer and the retrieved context. Specifically, the generated answer is segmented into short sentences, $|S|$ denote the total number of sentences, $|V|$ represent the number of sentences that are relevant to the retrieved context. The value of AF is calculated as the ratio of relevant sentences to the total number of sentences, as defined in Equation (1):

$$AF = \frac{|V|}{|S|} \tag{1}$$

Second is the Answer Relevance (AR), evaluating the degree to which the generated answer is directly related to the original question. A large language model is used to generate n potential questions based on the answer, and the cosine similarity between each generated question and the original question is computed using an embedding model. The AR score is obtained by averaging these cosine similarities, as defined in Equation (2):

$$AR = \frac{1}{n}\sum_{i=1}^{n} sim(q, q_i) \tag{2}$$

Last is the Context Relevance (CR), which assesses the alignment between the retrieved context and the original question. It is calculated as the ratio between the number of extracted sentences relevant to the answer and the total number of sentences in the context, as defined in Equation (3):

$$CR = \frac{\text{number of extracted sentences}}{\text{total number of sentences in } c(q)} \tag{3}$$

Adopting the classification from RAGEval (Zhu et al., 2024), we divide the questions into six types, with their names and descriptions shown in Table 4. For each question type, we develop 30 questions, each accompanied by a chatbot response

and corresponding context (i.e., the Execute Crew output, which also serves as the retrieved answer from the city knowledge graph). It is worth noting that current chatbot design and evaluation have not cover spatial analysis, which requires more than dataset understanding and chatbot conversation. Finally, we construct a test set comprising 180 cases, covering a diverse range of question types.

Name	Description
Attribute Query	Questions focus on semantic attributes stored in the knowledge graph and do not involve spatial analysis
Data Statistics	Questions involve statistical aggregation, such as counting or averaging objects or attributes
Multilingual	User inputs presented in English or Chinese, used to evaluate multilingual capabilities
Comprehensive Analysis	Questions require either detailed analysis of a single object or a combination of multiple query types
System Capability	Questions related to the chatbot's underlying capacity, including data coverage, scalability, and response range
Fuzzy Query	Questions contain vague or imprecise expressions, excludes highly subjective or uncertain queries

Table 4. Chatbot performance evaluation criteria and descriptions

Metric	Attribute Query	Data Statistics	Multilingual	Compre-hensive Analysis
Answer Faithfulness	0.7320	0.9802	0.8532	0.9167
Answer Relevance	0.8060	0.8923	0.9726	0.7778
Context Relevance	0.8230	0.9538	0.8964	0.8622

Table 5. Chatbot performance evaluation results

With three evaluation metrics and a multi-dimensional test set, we successfully conduct the automatic evaluation experiments using the "text-embedding-3-small" embedding model (from OpenAI), and some results provided in Table 5. Across the four question types: Attribute Query, Data Statistics, Multilingual, and Comprehensive Analysis, the system attained strong performance on all three evaluation metrics (each above 0.7), demonstrating its robustness and adaptability to a wide range of dialogue scenarios. Specifically, in the case of Data Statistics, the structured query statements exhibit a significant advantage, with each metric approaching or exceeding 0.9. Moreover, we design two types of language prompts: (1) Chinese, the primary language spoken at our research institution, and (2) English, the predominant language in academic communication. Experimental results show that when the system prompt and user input are in the same language (e.g., English), the system achieves the highest score in Answer Relevance. This demonstrates that our chatbot system can effectively interact in multiple languages, provided that appropriate prompts in the corresponding language are supplied in advance.

5.3 Discussion

The developed KCitychatBot system integrates knowledge graphs with a multi-agent framework to interpret CityGML data and respond to user queries through natural language conversations. Evaluation experiments demonstrate that the system provides effective support for tasks such as data statistics and comprehensive analysis. In addition, it helps mitigate large language model (LLM) hallucinations and address limitations related to dialogue context.

Although the system has shown success in specific scenarios, the evaluation results indicate suboptimal performance in areas such as System Capability and Fuzzy Query, as well as other scenarios not explicitly covered during development. These shortcomings are primarily due to the limitations of structured queries. The chatbot's interoperability decreases significantly when dealing with complex questions involving diverse qualifiers. Furthermore, we observed inconsistencies in responses to identical user queries—a common issue in chatbot systems—which may cause confusion.

To enhance the system's robustness, we plan to introduce a parallel processing mechanism, such as GraphRAG (Han et al., 2025). Additionally, the current design and implementation lack spatial analysis capabilities, which limits the system's broader applicability. We aim to address this limitation by integrating knowledge graphs, LLMs, and existing geospatial processing tools such as the GDAL library and PostGIS database. Despite its current limitations, we believe this work represents a valuable step toward intelligent, user-friendly interaction with large-scale CityGML datasets.

6. Conclusion

This manuscript presents a novel chatbot system using multi-LLM agents and a knowledge graph for large-scale CityGML datasets, aiming to lower the interaction barrier for ordinary users. The main innovations are a hierarchical knowledge graph construction method tailored for CityGML and a knowledge graph -based multi-agent framework for the chatbot. Extensive evaluation with the PLATEAU dataset affirms the usability of the proposed method and its corresponding prototype system. This work is one of the latest attempts in knowledge graph-supported city digital twin applications and establishes a new methodological contribution in bridging CityGML, knowledge graphs, and LLMs. The resulting chatbot system can significantly alleviate the burden for ordinary users' querying information from large-scale CityGML datasets, thus bearing broad practical implications. It also has the potential to be embedded into other spatial natural language-driven multi-agent systems. In the future, we plan to strengthen the system's capability for complex spatial analysis and robustness in handling user input. It would also be valuable to explicitly define the value gained by using a city knowledge graph in the proposed chatbot system and its capability boundaries.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grants 41901410. Natural Science Research Project of Anhui Educational Committee under Grants 2023AH050103.

References

Agrawal G, Kumarage T, Alghamdi Z., 2023. Can knowledge graphs reduce hallucinations in llms?: A survey. arXiv preprint arXiv:2311.07914. doi.org/10.48550/arXiv.2311.07914.

Brown T, Mann B, Ryder N., 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877-1901.

Chadzynski A, Li S, Grišiūtė A., 2023. Semantic 3D city interfaces—Intelligent interactions on dynamic geospatial knowledge graphs. *Data-Centric Engineering*, 4: 20.

Dang P, Zhu J, Dang C., 2025. Semantic-driven parametric 3D geographic scene modeling: Integrating knowledge graphs and large language models. *Environmental Modelling & Software*, 188: 106399.

Es, S., James, J., Anke, L.E., and Schockaert S. 2024: Ragas: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations.* 150-158.

Ge X, Wang Y C, Wang B., 2024. Knowledge graph embedding: An overview. A*PSIPA Transactions on Signal and Information Processing*, 13(1).

Guo T, Chen X, Wang Y., 2024. Large language model based multi-agents: A survey of progress and challenges. arXiv preprint arXiv:2402.01680. doi.org/10.48550/arXiv.2402.01680.

Han H, Shomer H, Wang Y., 2025. Rag vs. graphrag: A systematic evaluation and key insights. arXiv preprint arXiv:2502.11371. doi.org/10.48550/arXiv.2502.11371.

Khan W, Ahmad W, Luo B., 2019. SQL Database with physical database tuning technique and NoSQL graph database comparisons. 2019. In *IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference*: 110-116.

Luo, B., Lau, R.Y.K., Li, C., 2022: A critical review of state-of-the-art chatbot designs and applications. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 12(1): 1434. doi.org/10.1002/widm.1434.

Nguyen S H, Kolbe T H., 2020. A multi-perspective approach to interpreting spatio-semantic changes of large 3D city models in CityGML using a graph database. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 6: 143-150.

Peng B, Quesnelle J, Fan H., 2023. Yarn: Efficient context window extension of large language models. arXiv preprint arXiv:2309.00071. doi.org/10.48550/arXiv.2309.00071.

Saka A B, Oyedele L O, Akanbi L A., 2023. Conversational artificial intelligence in the AEC industry: A review of present status, challenges and opportunities. *Advanced Engineering Informatics*, 55: 101869.

Seto, T., Furuhashi, T., Uchiyama, Y., 2023: Role of 3d city model data as open digital commons: a case study of openness in japan's digital twin" project plateau". *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 48, 201-208. doi.org/10.5194/isprs-archives-XLVIII-4-W7-2023-201-2023.

Tan Y, Liang Y, Zhu J., 2023. CityGML in the Integration of BIM and the GIS: Challenges and Opportunities. Buildings, 13(7): 1758.

Tsai, M.H., Chen, J.Y., Kang, S.C., 2019: Ask Diana: A keyword-based chatbot system for water-related disaster management. Water, 11(2), 234. doi.org/10.3390/w11020234.

Wang, S., Hu, T., Xiao, H., 2024: GPT, large language models (LLMs) and generative artificial intelligence (GAI) models in ge ospatial science: a systematic review. *International Journal of D igital Earth*, 17(1): 2353122. doi.org/10.1080/17538947.2024.2 353122.

Xu H, Yuan J, Zhou A., 2024. Genai-powered multi-agent paradigm for smart urban mobility: Opportunities and challenges for integrating large language models (llms) and retrieval-augmented generation (rag) with intelligent transportation systems. arXiv preprint arXiv:2409.00494, 2024. doi.org/10.48550/arXiv.2409.00494.

Yu, D., Bao, R., Mai, G., 2025. Spatial-rag: Spatial retrieval aug mented generation for real-world spatial reasoning questions. ar Xiv preprint arXiv:2502.18470. doi.org/10.48550/arXiv.2502.1 8470.

Zhu K, Luo Y, Xu D., 2024. RAGEval: Scenario specific rag evaluation dataset generation framework. arXiv preprint arXiv:2408.01262. doi.org/10.48550/arXiv.2408.01262.