Towards a Spatio-Temporal Knowledge Graph Question-Answering Benchmark for Real Estate

Luciana Tanevitch
LIFIA-CICPBA, Facultad de Informática, UNLP
La Plata, Argentina
luciana.tanevitch@lifia.info.unlp.edu.ar

Aidan Hogan
DCC, Univ. de Chile
Santiago de Chile, Chile
ahogan@dcc.uchile.cl

Diego Torres
LIFIA-CICPBA, Facultad de Informática, UNLP
La Plata, Argentina
Departamento de Ciencia y Tecnología, UNQ
Bernal, Argentina
diego.torres@lifia.info.unlp.edu.ar

***Abstract*—Spatio-temporal knowledge graphs extend traditional knowledge graphs by capturing context with respect to both location and time. The dynamic and multidimensional aspects of these knowledge graphs require specialized methods for representation, integration, and reasoning. Likewise, solving the Question Answering (QA) task over spatio-temporal knowledge graphs involves not only understanding entities and their relationships, but also reasoning over when and where events occur. The Spatio-Temporal Knowledge Graph Question Answering (STKGQA) field remains underexplored. The lack of benchmarks in this field hinders progress by limiting the community's ability to systematically evaluate and compare approaches, identify strengths and weaknesses, and measure improvements over time. In this work, we present the initial strategies for building a STKGQA benchmark in the real-estate domain to assess the feasibility of using large language models (LLMs) to support the creation of synthetic benchmarks for STKGQA. We focus on evaluating how accurately LLMs can translate natural language questions into SPARQL queries compared to manually authored gold-standard queries, and how naturally questions an LLM can generate. Our findings include common error patterns, quality evaluation criteria, and a set of lessons learned that inform future efforts toward scalable and reliable benchmark creation in this domain.**

***Index Terms*—STKGQA, Knowledge Graphs, Question Answering, Benchmark, Real estate**

I. Introduction

In recent years, Spatio-Temporal Knowledge Graphs (STKGs) have emerged as a powerful data representation paradigm for modeling entities whose attributes evolve over time and space. These graphs enable the encoding of not only static relationships but also temporal and spatial dimensions, making them suitable for applications such as transportation monitoring, environmental tracking, and real estate analysis. Structured languages like (Geo)SPARQL or Cypher enable querying STKGs but present a high barrier-to-entry for many users.

Spatio-Temporal Knowledge Graph Question Answering (STKGQA) [1] refers to the task of answering natural language questions over STKGs, thus lowering this barrier-to-entry. These questions often require complex reasoning over time-based conditions (e.g., "*during 2023*") and space-based constraints (e.g., "*in Villa Elvira*").

A temporal question includes at least one explicit or implicit time-related constraint or requires time-stamped answers. These questions often involve reasoning over dates, periods, or temporal relationships (before, after, during, etc.). For example, "*What was the median price of apartments in the 90s?*" involves finding the period referred to, which is between 1990 and 1999. After detecting the temporal aspects, the price of the properties is selected and aggregated.

A spatial question refers to a geographic location or requires reasoning over spatial relationships (within, near, intersecting, etc.). For example, "*What is the median price of apartments in La Plata?*" includes filtering real estate located in La Plata, that is, a city within the Province of Buenos Aires.

The real estate domain in the Province of Buenos Aires presents a compelling application scenario for STKGQA, as it is inherently spatio-temporal. Real estate is geolocated and tied to jurisdictional units (e.g., districts, provinces), while their attributes (such as land-use regulations, pricing, or occupancy factors) evolve over time. For instance, answering a question like "*Which private neighborhoods in La Plata saw land values rise after 2021?*" requires reasoning over both space and time dimensions. Similarly, land-use decisions often involve comparing historical and current zoning regulations, making this domain rich in complex temporal and spatial relationships. This makes real estate a natural testbed for evaluating the capabilities of STKGQA systems, and also a natural use-case for putting such systems into practice.

Despite growing interest in STKGQA, there is a lack of benchmarks for the systematic evaluation of such systems. A potential reason for this gap in benchmarks is the cost associated with generating a high-quality gold standard, which motivates the exploration of synthetic dataset generation as a potential solution. However, it remains an open question whether this approach can reliably produce high-quality examples suitable for benchmarking. A key observation is that synthetic benchmarks often fail to include the kinds of questions that humans genuinely want to ask. This shortcoming underscores the need for benchmarks that reflect real-world scenarios and user requirements, rather than relying solely on automatically generated data, which often fails to capture the kinds of questions that users genuinely want to ask.

In this work, **we conduct a preliminary evaluation to assess the viability of using large language models (LLMs) to create synthetic questions and answers** for STKGQA

benchmarks. The evaluation focuses on two main aspects: first, how well LLMs generate SPARQL queries in terms of syntactic correctness, proper use of the available vocabulary, and accurate capture of the question's intent; second, how realistic the generated natural language questions are, considering that our study is based on actual end-user needs over real-world data, rather than artificially constructed questions. Since answers in dynamic knowledge graphs change over time, we define the *answer* as the result set returned by a SPARQL query that faithfully captures the question's semantics. These gold-standard queries are manually authored for correctness.

This work explores concrete examples to observe current LLM behavior when applied to STKGQA tasks. To support this analysis, we developed a small prototype dataset based on real-world scenarios from the Province of Buenos Aires. While not yet a full benchmark, this dataset allows us to identify the main challenges involved in aligning user questions with structured queries in spatio-temporal contexts.

The work is organized as follows: Section II overviews existing work in KGQA, focusing on temporal, spatial, and spatio-temporal QA. Section III presents the model underlying our knowledge graph. Section IV details the methodology, centered on two tasks: SPARQL generation with different prompting strategies, and natural language question generation, both analyzing LLM behavior. Section V discusses key findings on the strengths and limitations of generative models in STKGQA. Finally, Section VI summarizes lessons learned and future directions.

II. Related works

Question Answering over Knowledge Graphs (KGQA) enables users to retrieve precise information from complex knowledge bases using natural language without requiring expertise in structured query languages or data schemas [2]. Existing approaches span from subgraph-based methods that extract answers from the KG to neural translation models that generate structured queries from natural language questions.

This section reviews previous work on question answering over knowledge graphs, with a particular focus on temporal, spatial, and spatio-temporal reasoning. We organize the discussion into three subsections: (A) temporal QA over KGs, (B) spatial QA over KGs, and (C) KGQA involving both temporal and spatial dimensions.

A. Temporal Question Answering over Knowledge Graphs

To represent real-world dynamics, temporal knowledge graphs (TKGs) extend standard KGs with temporal scopes. TKGQA focuses on questions requiring reasoning over time-dependent facts. These include constraints (before/after), expressions (explicit dates or implicit references), and various granularities (year, month, etc.).

Su et al. [3] provide a taxonomy of temporal question types and classify existing methods accordingly. Their work highlights the need for benchmarks that reflect the diversity of temporal reasoning tasks. Saxena et al. [4] introduce CronQuestions, a large-scale benchmark for TKGQA based on a temporally annotated Wikidata subset. It uses templates covering a wide range of temporal reasoning patterns (e.g., before, after, first/last) and includes over 400,000 paraphrased QA pairs. Jia et al. [5] propose TIQ, a benchmark targeting questions with implicit temporal constraints, often overlooked in prior datasets. It links textual snippets with knowledge base facts, using GPT to rephrase the result into natural questions. LC-QuAD 2.0 [6] provides 30,000 questions paired with SPARQL queries over Wikidata and DBpedia. It includes a range of question types (temporal, multi-hop, boolean, count, etc.) with a semi-automatic pipeline for question generation and paraphrasing.

B. Spatial Question Answering over Knowledge Graphs

Spatial information introduces challenges in both modeling and querying. Karalis et al. [7] extend YAGO2 into YAGO2geo by incorporating geometries (lines, polygons) on top of lat/-long coordinates, enabling spatial queries using GeoSPARQL.

Kefalidis et al. [8] propose GeoQuestions1089: a benchmark of manually crafted geospatial questions and GeoSPARQL queries. It includes spatial relation questions (e.g., "*Is Liverpool east of Ireland?*") and proximity queries (e.g., "*Which churches are near castles?*").

C. Spatio-temporal Question Answering over Knowledge Graphs

Early works focused on modeling the evolution of geographic features through stRDF framework [9], [10].

KnowWhereGraph [11] is a temporally and spatially indexed KG that integrates multiple domains (e.g., disasters, climate, demographics) to support decision-making in dynamic contexts. Dai et al. [12] present STQAD, a dataset with 10,000 questions requiring combined temporal and spatial reasoning. Built from real-world spatio-temporal facts in YAGO15k, it uses templates and GPT paraphrasing to generate natural questions like "W*hich team did George Moncur play for after Stockport County F.C., located southwest of Stockholm?*". Mitsios et al. [13] propose a pipeline for generating QA datasets focused on geographic changes over time, producing GeoSPARQL queries from subgraphs and templates.

While most existing studies model temporality in relation to changing geographic features, our approach focuses on the evolution of attributes linked to real estate properties whose spatial position remains constant. These efforts reflect the growing need for benchmarks capable of testing complex spatio-temporal reasoning. While current datasets mark important progress, there remains a lack of benchmarks grounded in real-world domains, such as real estate, where rich spatio-temporal dynamics are central. Our work addresses this gap by introducing a benchmark that emphasizes realistic scenarios evolving attributes over fixed geographic regions.

III. Spatio-temporal Ontology

To support spatio-temporal question answering in the real estate domain, we designed an ontology that formalizes the core entities, relationships, and attributes involved in urban

land management across both spatial and temporal dimensions. This ontology, depicted in Figure 1, serves as the semantic backbone of the STKG, enabling structured representation, reasoning, and querying over evolving real-estate information. The ontology formalizes concepts from the real-estate domain, enabling the structured representation of properties, listings, and their associated characteristics. At its core, the ontology distinguishes between a `:RealEstateListing` (which corresponds to an advertisement published on a real estate website) and a `:RealEstate`, which is the actual asset being offered. Each real-estate instance is composed by one or more `:Space` elements, which describe the physical components of the property, such as rooms, land, or buildings. The spaces can be associated with a coordinate point to be geographically located. The ontology models real-estate characteristics using the class `:Feature`, which captures a wide range of attributes, for example, the address of the property (`:Address`) or its price within a listing (`:Price`). These features are annotated with provenance information, allowing users to trace the origin of each value (e.g., scraped from a listing, inferred through information extraction, or manually curated), the value that they take and a timestamp in which that value was obtained from the source. The full ontology is available on GitHub[1].

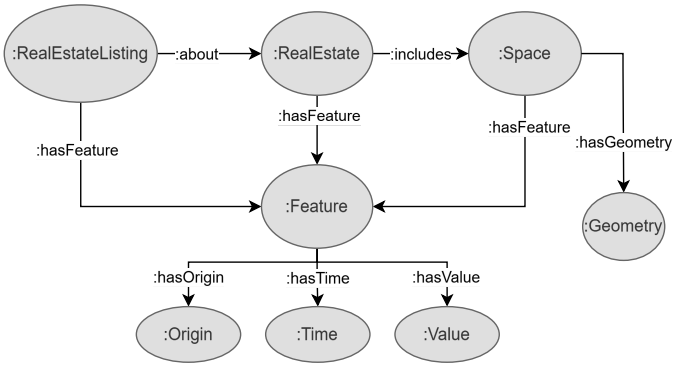

Fig. 1. Main concepts of the real estate ontology

IV. Evaluation

To evaluate the capabilities and limitations of generative models in the context of STKGQA, we consider the following research questions:

1) How accurately do generative models translate natural language questions into SPARQL queries compared to human-authored gold-standard queries?
2) What is the impact of different prompt engineering strategies on the quality of the generated SPARQL queries?
3) Which types of questions (e.g., temporal, spatial, or spatio-temporal) are more challenging for the models to handle correctly?
4) What are the common types of errors made by generative models when generating queries?

[1]https://github.com/tanevitch/QA-prompts

The evaluation is based on a set of ten questions provided by domain experts. These questions reflect real informational needs regarding real estate values, often involving temporal and spatial constraints. Since the questions are intended to be in Spanish, an equivalent translation is provided for understanding purposes.

1) ¿Cuánto vale una casa en Berisso? *(How much does a house cost in Berisso?)*
2) ¿Cuál es el precio por metro cuadrado de los terrenos en La Plata? *(What is the price per square meter of land plots in La Plata?)*
3) ¿Qué diferencia de precio por m² existe entre los terrenos de Villa Elvira (La Plata) y City Bell (La Plata)? *(What is the price difference per square meter between the land plots in Villa Elvira (La Plata) and City Bell (La Plata)?)*
4) ¿Cuál es el precio promedio de los departamentos que están a menos de 5 km de la Universidad Nacional de La Plata? *(What is the average price of apartments located within 5 km of the National University of La Plata?)*
5) ¿Aumentó la oferta de inmuebles en la zona de la Facultad de Humanidades desde su construcción? *(Has the supply of properties increased in the area around the Faculty of Humanities since its construction?)*
6) ¿Cómo evolucionó el stock de oferta de inmuebles en los últimos tres meses? *(How has the stock of property listings evolved over the last three months?)*
7) ¿Cuánto sale una casa cerca de la Playa Varese? *(How much does a house near Playa Varese cost?)*
8) ¿Cómo evolucionó el precio de los departamentos en Mar del Plata en los últimos 3 meses? *(How have apartment prices in Mar del Plata evolved over the last three months?)*
9) ¿En qué mes del año los inmuebles estuvieron más baratos en La Plata? *(In which month of the year were properties cheapest in La Plata?)*
10) ¿Son más baratos los departamentos en Berisso que en La Plata? *(Are apartments in Berisso cheaper than in La Plata?)*

For each question, we manually authored a corresponding SPARQL query that correctly captures its intent using the ontology defined in Section III. These gold-standard queries serves as a reference point for evaluation.

During the manual construction of the gold-standard SPARQL queries, we observed recurring structural patterns across different questions. Many queries shared similar logic, especially in how they computed prices (e.g., average price vs. price per square meter), applied geographic filters (e.g., by city URI, point location, or polygon), and handled temporal constraints (typically through the date associated with a specific feature of the real estate). Based on these recurring elements, we defined a set of SPARQL templates that capture the core semantic structures underlying different types of questions. A similar idea of leveraging templates has been used in related work, such as [12], [14], where templates were applied to

generate natural language questions, while in our case they serve to formalize the structure of SPARQL queries, as in [15], [16].

We identified four main template categories: (1) queries about prices in or near a specific location, (2) price comparisons across locations, (3) evolution of listing counts over time, and (4) evolution of prices over time. Each template has a base structure and optional blocks that vary depending on question interpretation, covering price calculation modes, spatial filters (e.g., point, polygon), and temporal constraints.

To serve as an example, Template T1 (Figure 2) captures questions that request the price of real estate, either as the average or the average per square meter. These questions are common in real-estate analysis and require conditional aggregation logic based on whether surface information is mentioned or implied. The template includes two alternative aggregation modes: `price_per_m2`, used when expressions like "per square meter" appear, requiring surface data and `total_price`, applied when surface is not mentioned, computing a simple average. The core of the template is a `base_query` that comprises the common structure shared by the questions of this type, including the retrieval of listings, their associated real estate units, price features, and optional surface values. The template is modular: optional blocks can be added based on question semantics, including filters by property type, location (e.g., city or neighborhood), spatial constraints (e.g., polygons, proximity), and temporal intervals.

By combining the base structure with relevant optional blocks, the template adapts to a wide range of semantically distinct but structurally related questions. The model is expected to choose the correct aggregation mode and blocks based on question interpretation. For example, the question *"What is the average price per square meter of apartments in La Plata?"* would use the `price_per_m2` aggregation mode along with the location URI filter for La Plata city. *"What is the average price of the houses near Playa Varese?"* would use the `total_price` aggregation mode combined with the distance filter centered on Playa Varese and a default or specified radius. *"What is the price per square meter of the land plots in the northern zone?"* would use `price_per_m2` and activate the polygon filter representing the northern zone polygon. This flexible combination of aggregation modes and modular filters enables broad coverage of diverse question types. The templates serve as robust, adaptable patterns for mapping natural language to SPARQL in STKGQA. The full list is available on GitHub.

With these elements (natural language questions, gold SPARQL queries, and template-based queries) we now describe the evaluation methodology, focused on two components: SPARQL generation and question generation.

A. SPARQL generation

In this phase, we assess whether a generative model can generate an appropriate SPARQL query to retrieve the corresponding answer from the knowledge graph.

To this end, we compare three prompting strategies:

```
aggregation_mode:
    - id: price_per_m2
      condition: The question includes expressions like per
           square meter, m², or similar.
      select_clause: SELECT ?currency (SUM(?price) / SUM(?
          surface) AS ?avgPricePerM2)
      requires_surface: true
    - id: total_price
      condition: The question refers to average price but
          does not mention surface area.
      select_clause: SELECT ?currency (AVG(?price) AS ?
          avgPrice)
      requires_surface: false

base_query: |
  {{SELECT_CLAUSE}}
  WHERE {
    ?listing a pronto:RealEstateListing ;
             sioc:about ?realEstate .
    ?realEstate a ?propertyType ;
                rec:includes ?space .
    ?listing io:hasFeature ?priceFeature .
    ?priceFeature a io:Price ;
                  io:hasValue ?priceSpec .
    ?priceSpec a gr:UnitPriceSpecification ;
               gr:hasCurrency ?currency ;
               gr:priceType 'BASE' ;
               gr:hasCurrencyValue ?price .
    {{#if requires_surface}}
    ?space io:hasFeature ?surfaceFeature .
    ?surfaceFeature a io:Surface ;
                    io:hasValue ?surfaceSpec .
    ?surfaceSpec a pronto:SizeSpecification ;
                 gr:hasValue ?surface .
    {{/if}}
    ?realEstate io:hasFeature ?addressFeature .
    ?addressFeature a io:Address ;
                    io:hasValue ?addressSpec .
    ?addressSpec a io:PostalAddress .
    {{OPTIONAL_BLOCKS}}
  }
  GROUP BY ?currency

optional_blocks:
  - id: property_type_filter
    condition: The question specifies the type of real
        estate
    sparql: |
      FILTER(?propertyType = {rdfs:subClassOf rec:
          RealEstate})
  - id: location_filter
    condition: The question mentions a place
    sparql: |
      ?addressSpec ?propertyLocation ?locationUri .
      FILTER (?propertyLocation = {property io:city or io:
          neighborhood}) .
      FILTER(?locationUri = {individual of io:City or io:
          Neighborhood}) .
  - id: spatial_filter_within
    condition: The question asks for a zone or polygon
    sparql: |
      ?space geo:hasGeometry ?geom .
      ?geom geo:asWKT ?wkt .
      FILTER(geof:sfWithin(?wkt, 'POLYGON({POLYGON})'^^geo:
          wktLiteral)) .
  - id: spatial_filter_distance
    condition: The question asks for a closeness to a place
        . If no distance provided, use 5000
    sparql: |
      ?space geo:hasGeometry ?geom .
      ?geom geo:asWKT ?wkt .
      FILTER(geof:distance(?wkt, 'POINT({POINT})'^^geo:
          wktLiteral, uom:metre) < {DISTANCE}) .
  - id: time_filter
    condition: The question includes a temporal constraint
    sparql: |
      ?priceFeature time:hasTime ?date .
      FILTER(?date >= xsd:dateTime({DATE_FROM}) && ?date <
          xsd:dateTime({DATE_TO})) .
```

Fig. 2. SPARQL query template for average price questions (Template T1)

- **Zero-shot prompting**: The model receives the ontology and is asked to produce a SPARQL query for an input question, without being shown examples.
- **Few-shot prompting**: The model receives the ontology and a few annotated examples pairing natural language questions with their SPARQL queries, and is tasked with generating a SPARQL query for a new question. The few-shot examples illustrate different reasoning patterns, such as computing average price by location, retrieving price per square meter over a time interval, or combining both dimensions.
- **Template-based**: The model is given a predefined SPARQL template relevant to the question type, including placeholders and conditional blocks, which guide the generation of the query. An example of the instantiation of the template for answering a question is also provided.

Although SPARQL generation can be evaluated in different ways, our evaluation focuses on three key aspects:

1) Syntactic validity: Syntactic validity assesses whether a generated SPARQL query is well-formed and can be successfully parsed by a SPARQL engine. This includes correct syntax, balanced parentheses, and proper use of keywords and punctuation (dots, semicolons). Each query is manually inspected to verify correctness. Syntactic validity is computed as shown in Equation 1.

$$\text{Syntactic Validity} = \frac{\text{Valid queries}}{\text{Total queries}} \tag{1}$$

2) Vocabulary accuracy: Vocabulary accuracy measures whether the URIs used in a generated query belong to the ontology vocabulary, reflecting the model's correct use of domain terms. For each query, URIs were extracted and compared against the ontology. The ratio of correct URIs to total URIs was computed as shown in Equation 2.

$$\text{Vocabulary Accuracy} = \frac{\text{URIs belonging to ontology}}{\text{Total URIs in query}} \tag{2}$$

3) Intent Correctness: Intent correctness evaluates whether the generated SPARQL query accurately captures the intended meaning of the inputed natural language question. This includes correct variable selection, application of appropriate filters, and retrieval of relevant values. We considered four dimensions for each query:

1) **Geographic constraints applied correctly**: Evaluates whether the query correctly incorporates the expected geographic filter based on the question intent (e.g., filtering by a specific area using a URI, a polygon, or a distance-based constraint). It checks both the type of spatial constraint used and its proper implementation according to the ontology and data structure.
2) **Temporal constraints applied correctly**: Assesses whether the query accurately reflects the temporal conditions stated or implied in the question (e.g., filtering by date, time range, or temporal ordering). It considers both the presence of temporal filters and their structural and semantic adequacy.
3) **Relevant elements selected to answer the question**: Evaluates whether the query retrieves results suitable for answering the question (e.g., returning a number when the question asks for a count, rather than URIs or unrelated data).
4) **Logical correctness of the query structure**: Assesses whether the query is structurally well-formed and logically consistent. Errors include use of undefined functions, incorrect placement or omission of filter blocks, or other structural flaws.

Each dimension was scored as: 1 (correct), 0.5 (fixable by minor corrections like URI adjustments or literal values changes), or 0 (incorrect). The intent correctness score for each question is the average of the four dimension scores, and the overall intent correctness of the model is computed as the mean across all the evaluated questions

$$\text{Overall Intent Correctness} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{4}\sum_{j=1}^{4} s_{i,j}\right) \tag{3}$$

where N is the total number of evaluated queries and $s_{i,j}$ is the score (0, 0.5, or 1) assigned to dimension j for question i (Equation 3).

4) Answer correctness: This criterion evaluates whether the output of an automatically generated query is equivalent to the gold-SPARQL result or could do so after minor adjustments, such as date formatting or URI correction. Queries are scored as follows:

1) **Correct**: the result matches the expected output (1 point is given).
2) **Adjustable**: the result would be correct after minimal changes (0.5 point is given).
3) **Incorrect**: the query cannot be easily fixed; (0 points are given).

The overall score is calculated as the average across the ten evaluated queries, using the Equation 4:

$$\text{Answer Correctness} = \frac{\sum_{i=1}^{N} \text{query\_score}_i}{N} \tag{4}$$

where each query_score is 1, 0.5, or 0, and N is the total number of queries evaluated.

The prompts used for this evaluation are available on GitHub (files starting with `text-to-sparql`). Each question was inputed one by one into the GPT-4.1[2] model to generate SPARQL queries.

Table I summarizes the results of the experiment. The syntactic validity of the generated queries is consistently high across all methods. The model reliably produces queries that compile without syntax errors. Regarding vocabulary accuracy, the zero-shot approach fails to use the ontology vocabulary appropriately in any case. The few-shot method shows significant improvement, correctly using vocabulary terms but struggling with generalizing individuals and handling polygon-related spatial data, occasionally hallucinating vocabulary outside the ontology. Template-based generation exhibits very

[2]https://platform.openai.com/docs/models/gpt-4.1

TABLE I
EVALUATION OF SPARQL AUTOMATIC GENERATION

Criteria	Zero-shot	Few-shot	Template-based
Syntactic validity	1.00	1.00	1.00
Vocabulary accuracy	0.00	0.85	0.98
Intent correctness	0.00	0.56	0.83
Answer correctness	0.00	0.00	0.60

TABLE II
EVALUATION OF AUTOMATIC QUESTIONS GENERATION

Criteria	Score
Alignment with template	0.90
Plausibility	0.87
Spatio-temporal validity	1.00

few vocabulary errors, mostly limited to minor URI mistakes, and otherwise adheres strictly to the ontology vocabulary. For intent correctness, evaluation of zero-shot queries is not meaningful since the queries do not use appropriate vocabulary or structure, resulting in scores of zero. Few-shot queries show logical errors such as omission of necessary GROUP BY clauses, use of unsupported functions, and incorrect modeling of geographic dimensions due to improper URIs or polygon representations. Template-based queries demonstrate the highest intent correctness, with remaining errors mostly attributable to incomplete parameter values that, once provided, correct and functional queries. For answer correctness, the zero-shot strategy is excluded from evaluation, as none of its generated queries could be executed or repaired to yield meaningful results. The few-shot approach produced more structurally coherent queries, but all of them contained non-trivial errors such as missing GROUP BY clauses when using aggregation, use of unsupported functions, or incorrect geospatial modeling, which prevent execution or lead to incorrect outputs. These are not considered minor issues, and all few-shot queries are thus labeled as incorrect. In contrast, the template-based strategy produced three fully correct queries. Additionally, six queries presented only minimal errors (such as incorrect URIs, missing coordinates, or missformatted dates) that could be easily adjusted to obtain the correct answer. Therefore, under the answer correctness criterion, nine out of ten template-based queries reached the correct answer.

Overall, these results suggest that providing example queries (few-shot) improves the model's ability to generate syntactically valid queries and to better use domain vocabulary compared to zero-shot generation. However, few-shot generation does not guarantee an accurate recognition of the user's intent, as logical and structural errors remain frequent. Template-based generation, by explicitly guiding query construction and enforcing ontology compliance, appears to be more effective in aligning generated queries with the intended semantics, making it a promising approach for controlled SPARQL query generation with large language models.

B. Question generation

The experiment also aims to assess whether the model can generate natural and semantically faithful questions from predefined SPARQL templates. This objective helps evaluate whether LLMs can be used to generate questions that resemble how people naturally express their information needs. Success would indicate that LLMs can help construct larger, realistic QA benchmarks, while deviations would highlight the need for filtering or manual refinement. The quality of generated questions is evaluated manually based on three key criteria.

1) Query alignment with template: This criterion assesses whether the generated question can be effectively answered by instantiating the corresponding SPARQL template.

2) Plausibility: This criterion evaluates the grammatical correctness, naturality, and plausibility of the generated question. High scores indicate the question reads naturally, is easy to understand, and resembles the questions that people would realistically ask. Lower scores reflect awkward phrasing or questions that seem artificially constructed or unlikely to be posed by a human.

3) Spatio-temporal validity: This measures whether the spatial and temporal references in the question are correctly defined and consistent. For example, mentioning "near the beach" in a locality without beaches would be considered invalid. On the other hand, an invalid temporal indicator refers to any date or time period that is non-existent, contradictory, or inconsistent with the domain data such as impossible calendar dates, or conflicting time intervals that cannot be logically interpreted within the given context.

Each criterion is rated for each question with 1 if it satisfies the criterion and 0 if not. The final quality score for each paraphrase is calculated as the average of the three criteria scores. By averaging across multiple paraphrases, we obtain a estimation of the model's capability to generate high-quality, semantically consistent, and linguistically fluent paraphrases aligned to the templates.

For this experiment, we used the ontology defined in Section III and the prompt defined on GitHub (file `text-to-sparql`) to generate ten new questions in natural language for each template.

Table II summarizes the results. Not all generated questions can be answered by just instantiating the available templates, especially when spatial references are ambiguous or require more complex representations. For example, answering a question about "*the nearness to outskirts*" may involve defining a template with the capacity of filtering by distance to a polygon, in particular a complex one because it could be represented as a ring-shaped area, while concepts such as "*residential neighborhoods*" or "*rural zones*" require access to external classifications or derived spatial features that go beyond simply specifying a polygon. This highlights a limitation of the templates when dealing with semantically rich or imprecise spatial concepts. In relation with the plausibility criteria, the model straights some times in its naturality. For example, the question "*How did the prices per square meter compare for Bahía Blanca and Necochea in the year 2023?*") was associated

with Template T2, which handles price comparisons. However, this formulation suggests a focus on the features influencing price differences rather than the prices itself. An observation based on the generated dataset is that the model did not generate questions referring to specific temporal events or milestones. For example, it did not formulate questions such as whether there were more properties available before the gas network was extended in a particular district, which suggests that certain common user intentions are not spontaneously inferred by the model. It was observed that the model seems to be effective at generating temporal expressions, both explicit (such as specific years or defined periods) and relative (e.g., "*last quarter*" or "*the summer*"). A similar pattern emerged with spatial references: the model generated questions involving areas like "*residential neighborhoods*", "*the outskirts of a city*", or "*near the river San Fernando*", reflecting a realistic understanding of everyday language.

V. Discussion

This study highlights several challenges inherent to modeling and querying complex real estate information within a spatio-temporal knowledge graph. First, linguistic expressions such as "*a house near the beach*" are inherently ambiguous: what precisely does "*near*" mean in this context? Does it refer to a fixed distance radius such as 500 meters from any coastal location? And if so, which exact beach? Similarly, terms like "*Conurbano Bonaerense*" represent socio-geographical entities whose boundaries may not be explicitly modeled within the ontology or knowledge graph, raising the question of whether spatial queries relying on GIS polygons can be effectively implemented in SPARQL. Secondly, when it comes to temporal characteristics, there are also inherent ambiguities that arise in the real estate context and are worth analyzing. For example, when asking "*how much does a house cost in Berisso?*", a temporal window is implicitly assumed, since not all the properties in the graph are relevant for the calculation. Only a subset is considered, but which one? It could refer to properties listed in the last three months, but that time frame may vary depending on external factors or the perspective of the person asking.

The current knowledge graph includes only a portion of the spatial information relevant for reasoning, which is not necessarily a limitation of its design but rather a reflection of how real-world data is often distributed across multiple sources. Entities such as cities and neighborhoods are represented as URIs, but their spatial geometries, including boundaries or area definitions, are typically not encoded directly in the graph. Instead, they can be retrieved from external GIS layers when needed. This is particularly relevant for queries that depend on spatial inclusion or proximity, such as estimating house prices near a specific location like "*Playa Varese*", where the definition of the area requires access to polygon data. Similarly, features such as avenues, which follow a linear path, or beaches, which occupy continuous surface areas, may require more detailed spatial representations than simple points. External datasets containing urban zoning boundaries, land-use classifications, or environmentally-risky areas can provide the necessary spatial context to enhance reasoning over the graph, especially for real estate-related tasks where geographic context significantly influences property value.

Although the SPARQL templates devised in this work appear effective within their domain, scaling them to accommodate more diverse or complex questions remains a challenge. Expanding templates to cover a broader question space may quickly become unwieldy, whereas few-shot prompting with sufficiently representative examples might offer a more flexible alternative, albeit with less explicit semantic control.

Finally, this study raises a fundamental question regarding query validity: should a valid SPARQL query return the exact final answer, or is it sufficient that it retrieves the necessary components to compute that answer? This distinction is particularly relevant in complex domains like real-estate, where post-processing or reasoning beyond SPARQL might be needed.

VI. Conclusions and future works

This work explored the use of large language models (LLMs) for automatically generating SPARQL queries and natural language questions in the context of spatio-temporal knowledge graph question answering (STKGQA). The evaluation was guided by four research questions that we addressed in turn. A central aspect of our approach is the focus on realistic use cases: the questions were designed to reflect genuine information needs, rather than being artificially constructed or based on synthetic data with limited relevance to actual users. As a result of this evaluation, we produced a dataset containing the questions, SPARQL queries, and their corresponding annotations, which is publicly available on GitHub.

From this evaluation process, several insights emerged regarding the capabilities and limitations of current LLMs in the STKGQA setting. First, regarding the accuracy of LLM-generated SPARQL queries compared to manually authored gold-standard queries, our findings indicate that while the models often grasp the intent behind the question, the generated queries frequently diverge from the gold standard in terms of structure and correctness. In many cases, the output query resembled a plausible attempt but failed to use the correct vocabulary, suggesting that the main bottleneck lies not in understanding the question, but in accurately expressing it using the available ontology and URIs.

Second, we observed that prompt engineering has a notable impact on query quality. Providing examples that reflect the structure of the vocabulary, such as URIs that follow a consistent pattern (e.g., `district_name_province_name` for cities), helps the model generalize better, even without access to the full graph. Similarly, exposing the model to question templates improves its ability to align new questions with existing query structures and apply the appropriate constraints.

Third, we found that temporal questions are generally easier for the models to handle when time constraints are explicitly stated (e.g., specific years or dates). However, when

time expressions are implicit or relative (e.g., "*a year ago*"), performance drops due to the model's limited capacity to resolve temporal references to concrete values. Spatial questions, in contrast, posed a greater challenge overall. The model often struggled to determine whether it should use a URI, a geographic point, a polygon, or a distance filter, especially in scenarios involving vague or ambiguous locations. For example, in questions like "*near the Faculty of Humanities*", the model sometimes invented a URI instead of applying a distance-based filter to a known coordinate.

Finally, common errors across generated queries included: hallucination of predicates and URIs, failure to apply necessary filters (especially temporal ones), incorrect or missing aggregation operations, and improper handling of spatial data structures. These errors highlight the model's difficulties in conforming to the strict syntactic and semantic requirements of SPARQL, particularly when operating under a restricted vocabulary.

Since current evaluation is based on a limited set of queries, future work will focus on developing a complete dataset to serve as a benchmark for evaluating STKGQA systems, with particular emphasis on capturing the kinds of questions that real users genuinely want to ask. This includes introducing more challenging and nuanced queries that combine temporal constraints, spatial conditions, aggregations, and comparisons, in order to reflect the complexity of real-world information needs, particularly in domains such as real estate.

Additionally, while template-based approaches have shown promise for structuring query generation, they may not scale well as new and diverse question types emerge. Future research should therefore investigate whether this strategy remains optimal at scale, or if more flexible, hybrid approaches are needed to accommodate broader linguistic and reasoning variability.

REFERENCES

[1] W. Chen, H. Wan, S. Guo, H. Huang, S. Zheng, J. Li, S. Lin, and Y. Lin, "Building and exploiting spatial–temporal knowledge graph for next poi recommendation," *Knowledge-Based Systems*, vol. 258, p. 109951, 2022.

[2] X. Huang, J. Zhang, D. Li, and P. Li, "Knowledge graph embedding based question answering," in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, ser. WSDM '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 105–113. [Online]. Available: https://doi.org/10.1145/3289600.3290956

[3] M. Su, Z. Li, Z. Chen, L. Bai, X. Jin, and J. Guo, "Temporal knowledge graph question answering: A survey," *CoRR*, vol. abs/2406.14191, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2406.14191

[4] A. Saxena, S. Chakrabarti, and P. P. Talukdar, "Question answering over temporal knowledge graphs," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds. Association for Computational Linguistics, 2021, pp. 6663–6676. [Online]. Available: https://doi.org/10.18653/v1/2021.acl-long.520

[5] Z. Jia, P. Christmann, and G. Weikum, "TIQ: A benchmark for temporal question answering with implicit time constraints," in *Companion Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, Singapore, May 13-17, 2024*, T. Chua, C. Ngo, R. K. Lee, R. Kumar, and H. W. Lauw, Eds. ACM, 2024, pp. 1394–1399. [Online]. Available: https://doi.org/10.1145/3589335.3651895

[6] M. Dubey, D. Banerjee, A. Abdelkawi, and J. Lehmann, "LC-QuAD 2.0: A Large Dataset for Complex Question Answering over Wikidata and DBpedia," in *The Semantic Web – ISWC 2019*, C. Ghidini, O. Hartig, M. Maleshkova, V. Svátek, I. Cruz, A. Hogan, J. Song, M. Lefrançois, and F. Gandon, Eds. Cham: Springer International Publishing, 2019, vol. 11779, pp. 69–78, series Title: Lecture Notes in Computer Science. [Online]. Available: https://link.springer.com/10.1007/978-3-030-30796-7_5

[7] N. Karalis, G. M. Mandilaras, and M. Koubarakis, "Extending the YAGO2 knowledge graph with precise geospatial knowledge," in *The Semantic Web - ISWC 2019 - 18th International Semantic Web Conference, Auckland, New Zealand, October 26-30, 2019, Proceedings, Part II*, ser. Lecture Notes in Computer Science, C. Ghidini, O. Hartig, M. Maleshkova, V. Svátek, I. F. Cruz, A. Hogan, J. Song, M. Lefrançois, and F. Gandon, Eds., vol. 11779. Springer, 2019, pp. 181–197. [Online]. Available: https://doi.org/10.1007/978-3-030-30796-7_12

[8] S. Kefalidis, D. Punjani, E. Tsalapati, K. Plas, M. Pollali, M. Mitsios, M. Tsokanaridou, M. Koubarakis, and P. Maret, "Benchmarking geospatial question answering engines using the dataset geoquestions1089," in *The Semantic Web - ISWC 2023 - 22nd International Semantic Web Conference, Athens, Greece, November 6-10, 2023, Proceedings, Part II*, ser. Lecture Notes in Computer Science, T. R. Payne, V. Presutti, G. Qi, M. Poveda-Villalón, G. Stoilos, L. Hollink, Z. Kaoudi, G. Cheng, and J. Li, Eds., vol. 14266. Springer, 2023, pp. 266–284. [Online]. Available: https://doi.org/10.1007/978-3-031-47243-5_15

[9] K. Bereta, P. Smeros, and M. Koubarakis, "Representation and querying of valid time of triples in linked geospatial data," in *The Semantic Web: Semantics and Big Data, 10th International Conference, ESWC 2013, Montpellier, France, May 26-30, 2013. Proceedings*, ser. Lecture Notes in Computer Science, P. Cimiano, Ó. Corcho, V. Presutti, L. Hollink, and S. Rudolph, Eds., vol. 7882. Springer, 2013, pp. 259–274. [Online]. Available: https://doi.org/10.1007/978-3-642-38288-8_18

[10] G. Garbis, K. Kyzirakos, and M. Koubarakis, "Geographica: A benchmark for geospatial RDF stores (long version)," in *The Semantic Web - ISWC 2013 - 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part II*, ser. Lecture Notes in Computer Science, H. Alani, L. Kagal, A. Fokoue, P. Groth, C. Biemann, J. X. Parreira, L. Aroyo, N. F. Noy, C. Welty, and K. Janowicz, Eds., vol. 8219. Springer, 2013, pp. 343–359. [Online]. Available: https://doi.org/10.1007/978-3-642-41338-4_22

[11] K. Janowicz, P. Hitzler, W. Li, D. Rehberger, M. Schildhauer, R. Zhu, C. Shimizu, C. Fisher, L. Cai, G. Mai *et al.*, "Know, know where, knowwheregraph: A densely connected, cross-domain knowledge graph and geo-enrichment service stack for applications in environmental intelligence," *AI Magazine*, vol. 43, no. 1, pp. 30–39, 2022.

[12] X. Dai, H. Li, and G. Qi, "Question answering over spatio-temporal knowledge graph," *CoRR*, vol. abs/2402.11542, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2402.11542

[13] M. Mitsios, D. Punjani, S. Abdollahi, S. Gottschalk, E. Tsalapati, E. Demidova, and M. Koubarakis, "Generating a question answering dataset about geographic changes in a knowledge graph," in *International Conference on Knowledge Engineering and Knowledge Management*. Springer, 2024, pp. 471–489.

[14] Z. Chen, J. Liao, and X. Zhao, "Multi-granularity Temporal Question Answering over Knowledge Graphs," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds. Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 11 378–11 392. [Online]. Available: https://aclanthology.org/2023.acl-long.637/

[15] L. Jiang, J. Huang, C. Möller, and R. Usbeck, "Ontology-guided, hybrid prompt learning for generalization in knowledge graph question answering," 2025. [Online]. Available: https://arxiv.org/abs/2502.03992

[16] M. Dubey, D. Banerjee, A. Abdelkawi, and J. Lehmann, "Lc-quad 2.0: A large dataset for complex question answering over wikidata and dbpedia," in *The Semantic Web - ISWC 2019 - 18th International Semantic Web Conference, Auckland, New Zealand, October 26-30, 2019, Proceedings, Part II*, ser. Lecture Notes in Computer Science, C. Ghidini, O. Hartig, M. Maleshkova, V. Svátek, I. F. Cruz, A. Hogan, J. Song, M. Lefrançois, and F. Gandon, Eds., vol. 11779. Springer, 2019, pp. 69–78. [Online]. Available: https://doi.org/10.1007/978-3-030-30796-7_5