Lawrence Berkeley National Laboratory
LBL Publications

Title
Query Relaxation for LLM-Generated SPARQL Queries over Building Knowledge Graphs

Permalink
https://escholarship.org/uc/item/46x480kg

Journal
ACM Sustainability Week Companion 2026 Proceedings of the 2026 ACM Sustainability Week

Authors
Paul, Lazlo
Prakash, Anand Krishnan
Pritoni, Marco

Publication Date
2026-06-22

DOI
10.1145/3765611.3815342

Copyright Information
This work is made available under the terms of a Creative Commons Attribution-NonCommercial License, available at https://creativecommons.org/licenses/by-nc/4.0/

eScholarship.org Powered by the California Digital Library
University of California

Query Relaxation for LLM-Generated SPARQL Queries over Building Knowledge Graphs

Lazlo Paul
Lawrence Berkeley National Laboratory
Berkeley, California, USA
lpaul@lbl.gov

Anand Krishnan Prakash
Lawrence Berkeley National Lab
Berkeley, California, USA
akprakash@lbl.gov

Marco Pritoni
Lawrence Berkeley National Laboratory
Berkeley, California, USA
mpritoni@lbl.gov

Abstract

When Knowledge Graph (KG) queries fail to match a pattern in a KG, they return no results. Identifying the statements causing these failures is tedious, especially for LLM-generated queries, which tend to be longer and more complex than queries written by hand. Query relaxation addresses this by systematically loosening query constraints until results are recovered. To evaluate the effectiveness of query relaxation against LLM generated queries, we propose a two-stage relaxation method combining triple deletion and path relaxation and test it against 1,823 failed queries for building KGs.

CCS Concepts

• **Information systems → Ontologies**.

Keywords

SPARQL Relaxation, Large Language Models, Knowledge Graphs

ACM Reference Format:
Lazlo Paul, Anand Krishnan Prakash, and Marco Pritoni. 2026. Query Relaxation for LLM-Generated SPARQL Queries over Building Knowledge Graphs. In *ACM Sustainability Week 2026 (ACM Sustainability Week Companion '26), June 22–25, 2026, Banff, AB, Canada.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3765611.3815342

1 Motivation

Knowledge graphs (KGs) using ontologies like Brick and ASHRAE S223 provide standardized, machine-readable representations of buildings. LLMs show promise for generating SPARQL queries for interrogating these KGs [3]. However, LLMs frequently produce queries with errors that yield empty result sets, leaving a user with no signal about how to correct the queries and obtain results.

Previous work on SPARQL query relaxation leverages ontology-driven rules and methods increasing the number of queried variables, targeting short, human-written queries with simple syntax and properly used ontology vocabulary [1, 2]. LLM-generated queries fail differently: they are larger, contain hallucinated ontology terms and property names, and use complex but incorrect property paths and filters. Moreover, many existing relaxation methods leverage OWL and RDFS ontology structures not present in all building ontologies, making it difficult to evaluate their effectiveness in this setting. It therefore remains unclear how well query relaxation can address the failures of LLM-generated queries against building KGs to allow recovery of useful results.

To answer this research question, we analyze a corpus of 1,823 failed queries produced by different agentic methods for four different building KGs in the BuildingQA benchmark [3]. We propose a two-stage relaxation approach and evaluate how well it addresses the problem of empty results from LLM generated queries.

2 Query Relaxation

Analysis of the failed query corpus reveals that 78.5% of queries contain terms absent from the target KG and ontology, while the remainder use correct vocabulary but fail for structural reasons. Failing queries had seven triple patterns on average, considerably longer than the queries targeted by prior relaxation work.

These two failure modes motivate a two-stage approach: we first identify faulty triple patterns through deletion-based search, then attempt to recover partly correct information through predicate path relaxation. These methods produce relevant results while limiting query of extraneous information

2.1 Stage 1: Search via Triple Deletion

We perform a breadth-first search over subsets of the original query's triple patterns. At each level, one additional triple is removed and the reduced query is executed. The search returns as soon as a non-empty result set is obtained. This process finds a maximal succeeding subquery [2].

While this approach successfully identifies a viable subquery, removing triple patterns can loosen constraints, which can cause an explosion in the number of returned rows and increased query runtime, or eliminate variables that were intended to appear in the output. To mitigate runtime concerns, we cap the number of returned results at a limit M. To recover lost constraints and restore queried variables where possible, we relax predicate paths.

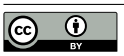
2.2 Stage 2: Predicate Path Relaxation

The second stage searches over predicate path replacements for faulty triples, using chains of paths and inverse paths of arbitrary length n:

$$(\hat{}p_1|p_1)/(\hat{}p_2|p_2)/.../(\hat{}p_n|p_n) \quad (1)$$

where each p_i is drawn from predicates in the target KG, $\hat{}$ denotes an inverse path, | the "OR" operator, and / the "Followed By" operator.

The two stages compose: for a given query, some triples may be deleted while others are relaxed. This model is expressive enough to capture unforeseen errors and relaxation patterns from prior work. For example, prior work repairs invalid class declarations by substituting with a superclass determined via the ontology [1]; our framework would achieves this through a path like *a/rdfs:subClassOf*. Our



This work is licensed under a Creative Commons Attribution 4.0 International License.
ACM Sustainability Week Companion '26, Banff, AB, Canada
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2199-1/26/06
https://doi.org/10.1145/3765611.3815342

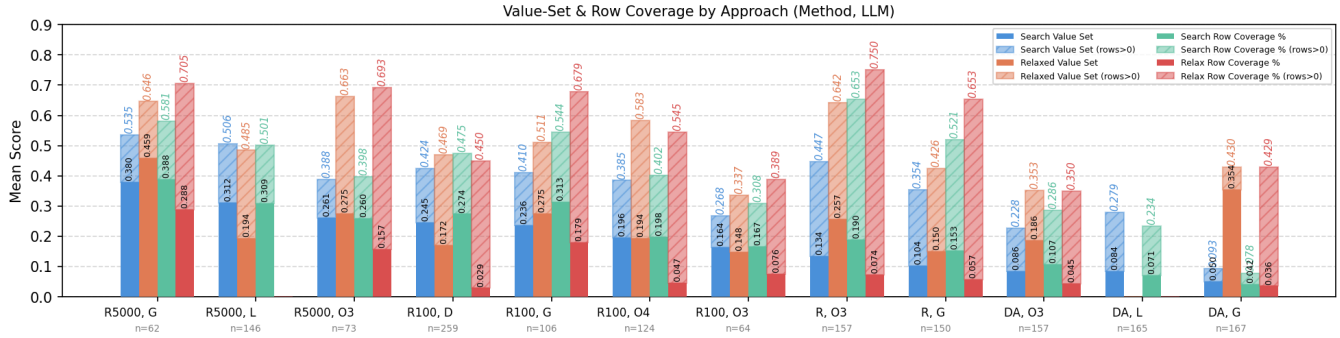


Figure 1: Performance metrics across twelve LLM configurations. Solid bars show performance considering all failed queries; hatched bars show conditional scores where relaxation successfuly returned results (rows>0).
LLM configurations consist of a method and model from [3]: ReAct5000 (R5000), ReAct100 (R100), ReAct with no KG (R), and Domain-Adapted KGQA (DA), run with Gemini Flash 2.5 (G), O3-mini (O3), O4 (O4), Deepseek R1 (D), and LLama 4 (L).

method of path search is less efficient than ontology-based rules, but it allows us to evaluate relaxation performance over an ontology-agnostic set of potential rules without excessively altering queries to retrieve additional variables that may confound evaluation.

3 Evaluation

We evaluate query relaxation with n=3 and a limit of M = 50,000 on four building KGs from BuildingQA [3]: `bldg11`, `TUC_building`, `dflexlibs_multizone`, and `b59`. BuildingQA pairs natural language questions with ground truth SPARQL queries and LLM-generated queries via various agentic methods. We apply relaxation to all failed LLM queries and compare the deletion search-only phase (Search) against the full two-phase approach (Relax).

Table 1 summarizes results: Search returned results for 32–61% of failed queries by building, and Relax augmented those results an additional 15–28% of the time. Success differs by building because of variation in the original quality of queries produced by the KGQA methods. Successful relaxations were completed quickly, taking 0.8–1.4 seconds on average.

Since column names between ground truth and relaxed queries are not aligned, we evaluate on returned values. The *Value Set* metric measures what fraction of ground truth result values appear in the relaxed output. *Row Coverage* evaluates relationship preservation: it matches ground truth rows to relaxed rows where the relaxed row is entirely contained within a ground truth row, reporting the percentage of ground truth rows represented. *Row Excess* shows the fraction of rows returned by relaxed queries that are not represented by rows in the ground truth results.

Figure 1 shows results across methods from [3]. Stronger base methods yielded more useful relaxed results. The best performing configuration in BuildingQA [3] (R5000, G) recovers 46% of desired values across all failed queries using the Relax method. Looking only at successful relaxations, the Value Set increased to 65% and row coverage achieved 70%, meaning that a significant amount of the desired data is returned and retains useful relationship information.

Table 2 summarizes results for the Search and Relax phases for queries that were successfully relaxed. The Value-Set and Row Coverage metrics indicate that about half of the desired results for a natural language request can be recovered through query relaxation, but the Row Excess metric indicates that half of the returned results may be less relevant to the original request. Results including the Relax phase are better than just the Search phase, returning more of the desired information and less excess information.

Table 1: Relaxation results across four buildings.

	Count			Avg. Seconds	
Building	Total	Search	Relax	Search	Relax
bldg11	665	403	112	0.5	0.8
TUC_building	281	109	15	0.1	0.7
dflexlibs_multizone	336	118	22	0.1	0.8
b59	541	171	25	1.3	1.4

Table 2: Row and column percentage averages by phase.

Method	Value Set	Row Coverage	Row Excess
Search	0.36 ± 0.34	0.40 ± 0.48	0.74 ± 0.41
Relax	0.53 ± 0.32	0.58 ± 0.49	0.52 ± 0.48

4 Conclusion

We present a SPARQL query relaxation method targeting the failure modes of LLM-generated queries over building KGs. Our approach identifies the triple patterns leading to failure and recovers useful results. Tested against 1,836 faulty queries, we are able to recover relevant results from 44% of them. Practically, relaxed queries and their partial results can simplify handoff of LLM generated queries for human refinement, or may serve as structured feedback within agentic LLM loops for iterative query correction.

Acknowledgments

The work is supported by the Assistant Secretary for Critical Minerals and Energy Innovation, Building Technologies Office, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231

References

[1] Imane Lahmam Bennani, Anand Krishnan Prakash, Marina Zafiris, Lazlo Paul, Carlos Duarte Roa, Paul Raftery, Marco Pritoni, and Gabe Fierro. 2021. Query relaxation for portable brick-based applications. In *Proceedings of the 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation.* ACM, Coimbra Portugal, 150–159. https://doi.org/10.1145/3486611.3486671

[2] Ginwa Fakih, Patricia Serrano-Alvarado, and Matthieu Perrin. 2025. SHARP: A Hybrid Approach for SPARQL Query Relaxation. In *21st International Conference on Semantic Systems (SEMANTiCS)*, Vol. 62. 194–216.

[3] Ozan Baris Mulayim, Avia Anwar, Umut Mete Saka, Lazlo Paul, Anand Krishnan Prakash, Gabe Fierro, Marco Pritoni, and Mario Bergés. 2025. BuildingQA: A benchmark for natural language question answering over building knowledge graphs. In *Proceedings of the 12th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation.* ACM, New York, NY, USA, 65–75.