Protoknowledge Shapes Behaviour of LLMs in Downstream Tasks: Memorization and Generalization with Knowledge Graphs

Anonymous ACL submission

Abstract

We introduce the concept of *protoknowledge* to formalize and measure how Knowledge Graphs (KGs) are internalized during pretraining and reused at inference by Large Language Models (LLMs). LLMs are known to memorize vast amounts of token sequences and a central open question is how this memorization can serve as reusable knowledge through implicit abstraction and generalization. We categorize *protoknowledge* into *lexical*, *hierarchical*, and *topological* forms, reflecting different levels of abstraction over KGs. We measure these forms through Knowledge Activation Tasks (KATs), analyzing general properties such as semantic bias. We then examine how *protoknowledge* affects Text-to-SPARQL, a task requiring conformity to the target KG's formal structure. To this end, we adopt a novel analysis framework that assesses whether model predictions align with the successful activation of the relevant *protoknowledge* for each query.

We do not frame this phenomenon as data contamination alone: rather, *protoknowledge* provides a measurable signal of how LLMs internalize structured information during pretraining and reuse it in downstream tasks. This perspective offers a more nuanced view of semantic-level data contamination and supplies an effective strategy for interpreting the behaviour of Closed-Pretraining models.

1 Introduction

During pretraining, Large Language Models (LLMs) internalize vast amounts of factual and structured information (Carlini et al., 2023; Roberts et al., 2020), which has led to their view as Knowledge Bases (KBs) capable of abstracting learned content (Petroni et al., 2019; Moiseev et al., 2022). LLMs also show the ability to reuse memorized information across tasks, languages, and domains, suggesting emerging forms of systematic generalization (Wang et al., 2025; Fu and Frank, 2024; Yao et al., 2024).

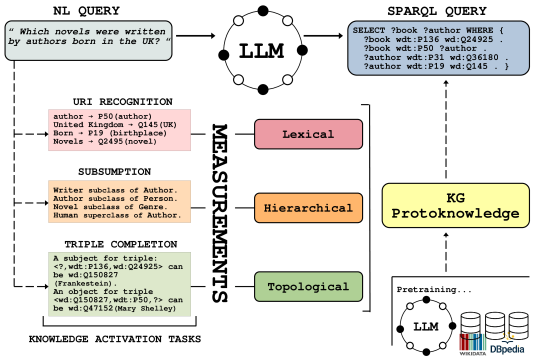

Figure 1: *Protoknowledge* Impact: LLMs acquire three *protoknowledge* forms from Knowledge Graphs, measured via Knowledge Activation Tasks (KATs). The degree of absorption of *protoknowledge* measured by KATs correlates with the Text-to-SPARQL performances of LLMs, showing that, when *protoknowledge* is acquired, it is also positively used.

A particularly demanding setting for such abilities is structured data, where generalization requires abstraction over complex relational patterns, and reliance on surface-level knowledge may lead to errors (Luo et al., 2024). Knowledge Graphs (KGs) exemplify this challenge: they encode information not only at the lexical level, but also through hierarchical relations and topological paths. Understanding how LLMs internalize such multi-layered structure during pretraining is crucial for assessing their ability to reuse it in downstream tasks.

We introduce notion of *protoknowledge*, defined as the internalization of information from pretraining that goes beyond memorization to support abstraction and reuse. We formalize and measure *KG protoknowledge* and analyze its impact on Text-to-SPARQL, a formal language generation task that requires implicit reasoning across multiple abstraction levels. Unlike generic program synthesis, which requires mapping natural language instructions to formal specifcations for coding, success in

Text-to-SPARQL also depends on respecting KG structural constraints, offering a window into how *protoknowledge* operates at inference.

By proposing *protoknowledge*, we shift the perspective on semantic-level data contamination: rather than treating it merely as a cause of overestimation, we use it as a lens to analyze how models internalize information during pretraining and reuse it in tasks requiring abstract reasoning.

Our contributions are:

- We formalize *KG protoknowledge* in *lexical*, *hierarchical* and *topological* forms to capture the different levels of abstraction of how KG information is internalized by LLMs during pretraining.
- We empirically measure *KG protoknowledge* forms by designing **Knowledge Activation Tasks** and specialized test sets uncovering its alignment with the **semantic bias** induced by pretraining data as a key property.
- We investigate the impact of *KG protoknowledge* on Text-to-SPARQL through a per-example analysis that examines how different forms correlate with query generation. This analysis provides a framework to both detect and reinterpret semantic-level data contamination, showing it not only as memorization but also as evidence of abstraction and reuse.

2 Related Works

The boundary between memorization and generalization in Large Language Models (LLMs) has become increasingly salient, particularly when pretraining corpora contain structured resources such as Knowledge Graphs (KGs). Early work focused on memorization (Carlini et al., 2023) and the risks of data contamination, showing that performance on benchmarks may be artificially inflated (Magar and Schwartz, 2022; Cheng et al., 2025; Xu et al., 2024). Related works highlight how the statistical properties of training set strongly shape models' outputs (Elazar et al., 2023), leading to phenomena like semantic-level data contamination.

Beyond memorization, research has examined how LLMs internalize KG information.While many tasks through supervised approaches (Peeters et al., 2024; Zeng et al., 2024; Zhang et al., 2021) explicitly aim to strengthen structural KG representations, recent analyses have investigated unsupervised scenarios: Bombieri et al. (2025) tested GPT models retention of biological ontologies, and Lo et al. (2023) evaluated the same on DBpedia triples completion measuring hard and soft matching on the relevance of information reported. Shifting from a task-oriented perspective Zhang et al. (2024) and Wu et al. (2023) analyze LLMs theoretical ontological knowledge acquisition from a theoretical perspective.

A tricky downstream setting is Text-to-SPARQL, which, like Text-to-SQL, stress models' using their abstraction reasoning skills in mapping natural language instructions to the query, considering KB formal constraints required for navigating them and at the same time reuse memorized information. Prior works on contamination has largely focused on performance gap using it as the only evidence, especially in Closed-Pretraining models where training data cannot be analyzed. Ranaldi et al. (2024) propose a dual analysis on GPTs: measuring pre-exposure to database schemas by masking column and table names and asking the model to reconstruct them in a name-cloze setting (Chang et al., 2023) and comparing Spider performance (Yu et al., 2018) with a structurally equivalent hand-crafted dataset designed to be contamination-free. Both analyses revealed memorization at both verbatim and semantic levels.

In this work, we shift from detecting contamination through correlated tests to analyzing inference in correlated tasks, using *KG protoknowledge* as a probing framework to assess whether LLM behaviour across these tasks reflects prior learning of structured content.

3 Protoknowledge on Knowledge Graphs

We define *protoknowledge* as an ability of Large Language Models (LLMs) to *memorize* information during pretraining and *reuse* it in downstream tasks, without task-specific supervision or fine-tuning. This ability is most evident in tasks requiring the integration of factual or structural knowledge implicitly learned from data exposure.

KG protoknowledge denotes a model's ability to internalize and exploit information about entities and properties encountered during pretraining.

While prior work show that LLMs can memorize KG content mostly at verbatim level, we examine whether they also acquire reusable abstractions that support reasoning in KG-related tasks such as Text-to-SPARQL.

3.1 Forms of Protoknowledge

KG protoknowledge can manifest in various forms depending on the downstream task, involving lexical mappings, structural reasoning, or both.

Based on the diversity of KG-related tasks, we identify three fundamental types of *protoknowledge*, each reflecting a different abstraction pattern: *lexical*, *hierarchical*, and *topological*.

Lexical Protoknowledge. Refers to a model's ability to recall entities and properties based on natural language surface forms (e.g., labels, aliases). It reflects symbolic anchoring acquired during pretraining. This form of *protoknowledge* is crucial for tasks requiring the recovery of identifiers, especially when they are are non-human-readable, as in Wikidata (e.g., mapping the label `"Moon"` to its ID `wd:Q405`).

Hierarchical Protoknowledge. Refers to a model's ability to recognize and reason over taxonomic relations, such as `subclassOf` or `instanceOf` hierarchies commonly found in KGs. For example, recognizing that `Politician` is a subclass of `Person` in DBpedia Ontology. *Hierarchical protoknowledge* can be crucial in tasks like ontology alignment, where understanding the parent-child structure of concepts helps to establish semantic correspondences (e.g. in Wikidata `wd:Journalist` is a direct subclass of `wd:Writer` and in DBpedia `dbo:Journalist` is a direct subclass of `dbo:Person`).

Topological Protoknowledge. Refers to a model's ability to infer and traverse multi-hop relational paths between KG items, going beyond direct neighbours. This capability is essential when reasoning over graph structures, mostly when generating SPARQL queries that require the retrieval of RDF triples[1] by connecting entities via intermediate properties. *Topological* form implicitly requires *lexical* and *hierarchical protoknowledge*, as models must both map labels to URIs and understand their connections inside KGs.

3.2 Knowledge Activation Tasks

To explore the different forms of *KG protoknowledge*, we define **Knowledge Activation Tasks** (KATs) as formalized variants of existing evaluation settings. These tasks selectively activate specific forms of *protoknowledge* by constraining the required information and reasoning. While all rely on recalling and reusing information acquired from pretraining, they differ in the degree of abstraction expected from the model.

Tasks for *lexical protoknowledge*,like entity matching (Peeters et al., 2024), involve translating natural language labels or aliases into their corresponding symbolic identifiers.

For *hierarchical protoknowledge*, the goal is to evaluate a model's capacity for taxonomic induction (Zeng et al., 2024) by placing subclasses or superclasses within the KG, thereby revealing its understanding of relational structure beyond surface-level cues.

Finally, *topological protoknowledge* tasks assess a model's ability to perform symbolic, multi-hop (Zhang et al., 2021) reasoning over KG subgraphs by reconstructing triples.

Formalizing KATs for each form provides a scheme for measuring how much *protoknowledge* is present in pretrained models and to what extent it is influenced by the semantic bias induced by the pretraining data.

While pretraining corpora of models experimented remain inaccessible, we assume, consistent with prior observations, that their content distribution broadly reflects common web data. Based on this hypothesis, we use our experiments to explore one key limitation of *KG protoknowledge*: its tendency to be strongly influenced by the frequency distribution of KG items, which we interpret as **semantic bias**.

4 Measuring Protoknowledge

We measure *KG protoknowledge* through a set of controlled **Knowledge Activation Tasks** (KATs), each targeting a distinct form: *lexical*, *hierarchical*, *topological*. Dedicated test sets are designed to isolate each form, allowing us to evaluate how different LLMs recall and reuse KG information acquired during pretraining.

We analyze the correlation between performance and item frequency, supporting the hypothesis that *KG protoknowledge* is influenced and limited by the **semantic bias** of the pretraining data.

[1] RDF (Resource Description Framework) is a standard model for representing information as subject–predicate–object triples connecting entities.

4.1 Models and Analysis Design

We evaluate Closed-Pretraining models with undisclosed corpora, making the study of *KG protoknowledge* crucial to indirectly reveal how structured information is internalized and reused by these models. Hence, we use four LLMs[2]: GPT-4, GPT-3.5-turbo (OpenAI, 2023), Llama-3-8B-Instruct, Llama-3.1-70B and Llama-3-70B-Instruct (Grattafiori et al., 2024). All experiments adopt greedy decoding to ensure deterministic generation and consistent evaluation.

All experiments are conducted on test sets derived from portions of DBpedia and Wikidata KGs. As a recurrent statistical measure, we define **popularity** as the number of triples referencing each item, using either the mean or the median (for randomly sampled sets) as a threshold to separate frequent and infrequent items. Additionally, this allows us to analyze how *KG protoknowledge* performance varies across more and less frequent items. Assuming that it reflects the Open Web bias, we shed light on performance dependence on the distributional properties of the pretraining data. Unless otherwise specified, all analyses on *protoknowledge* forms follow the same pipeline with shared configurations and prompting strategies.

4.2 *Lexical Protoknowledge*

KATs: We design the **URI Recognition Task**, in which the model is prompted with a natural language label (e.g., `"Moon"`) and asked to predict the corresponding KG URI (Prompt 4). This task assesses the model's ability to resolve symbolic references from surface forms, a skill especially relevant in settings like Wikidata, where entity identifiers are non-human-readable.

Test Set. Three test sets of `(label, Wikidata ID)` pairs are used. The first two are biased toward high-popularity items: one includes the most common entities, the other the most common properties. Finally a third set extracted from Wikidata is considered as its distribution content is not biased by high popularity. An extensive description is reported in the Appendix D.1.

Results. For URI Recognition Task, we report accuracy as the percentage of correct predictions. Results in Table 1 show that GPT models significantly outperform Llama models in identifying the most common entities and properties. For instance, GPT-4 achieves an accuracy of 74.35% on the most frequent entities, whereas Llama-3-70B reaches 35.90%. This disparity highlights GPT models' superior capability in leveraging *lexical protoknowledge* for URI recognition task.

Index Split	Llama-3-8B	Llama-3-70B	GPT-3.5	GPT-4
Most Common Entities				
LF (0:161)	3.11% (5/9)	19.25% (31/45)	43.46% (70/94)	**48.44%** (78/107)
MF (-39:200)	10.25% (4/9)	35.90% (14/45)	61.53% (24/94)	**74.35%** (29/107)
Most Common Properties				
LF (0:77)	2.59% (2/4)	35.06% (27/38)	18.18% (14/18)	**81.81%** (63/81)
MF (-23:100)	8.69% (2/4)	47.87% (11/38)	17.40% (4/18)	**78.26%** (18/81)

Table 1: URI RECOGNITION accuracy (%) for Test Set 1 and Test Set 2. LF (Less Frequent) and MFs (More Frequent) subsets are defined based on the average popularity in the dataset. The ratio of correct prediction over the subset of LF or MF is also reported near accuracy.

A consistent trend emerges from the results: accuracy tends to be higher for entities and properties that are more frequent/popular. An exception is found in the GPTs' performance on properties, where accuracy remains comparable between more and less popular items. The same trend is also observed in the third set (an extended discussion is reported in the Appendix on Tables 9 and 10).

4.3 *Hierarchical Protoknowledge*

KATs: We focus on taxonomic inference within the DBpedia Ontology and define two subsumption-based tasks:
Direct Subsumption: given a class, the model is asked to return its direct subclasses (prompt 6).
Inverse Subsumption: given a class, the model must identify its direct superclass (prompt 5).

While both tasks target hierarchical relations, they differ in their cognitive demands. **Direct Subsumption** may be resolved through shallow pattern recall, as the co-occurrence of class pairs and the `subclassOf` relation may have been encountered during pretraining. In contrast, **Inverse Subsumption** requires a higher degree of abstraction: since the concept of Superclass is not represented explicitly for KG items, the model must infer it by generalizing over memorized structural patterns.

Test Set. For evaluating *hierarchical protoknowledge*, we focus on the top-level classes in the DBpedia Ontology, specifically those that do not have any superclasses (ten root-level classes, e.g. Person, Organization, Place ...). For each of them, we extract their direct subclasses. In the Direct Subsumption task, the test set consists of pairs formed by a

[2]Model versions in Appendix B

parent class and the full list of its direct subclasses. In the Inverse Subsumption task, the test set is composed of all subclasses of the root classes, each paired with its corresponding unique superclass. Other details are reported in Appendix Table 13.

Results. We report accuracy as the proportion of correct predictions for both the *Subsumption Tasks*. Table 2 on `Direct Subsumption` shows that GPT-4 consistently outperforms all models, achieving the highest accuracy across both more and less popular classes. GPT-3.5 Turbo follows with strong results on *Person* (80.56%) and *Organisation* (76.92%). Llama-3-70B lags behind, prevailing mostly for popular classes, and Llama-3-8B performs poorly on both sets.

	CLASS	Llama-3-8B	Llama-3-70B	GPT-3.5	GPT-4	SUPPORT
MOST	Organisation	7.69	30.77	69.23	**69.23**	13
	Place	18.18	9.10	9.09	**36.36**	11
	Work	0.00	16.67	25.00	**58.33**	12
	Person	19.44	38.89	16.66	**52.67**	36
	Species	0.00	0.00	0.00	0.00	3
LEAST	SportFacility	25.00	0.00	0.00	**75.00**	4
	UnitOfWork	0.00	0.00	0.00	0.00	2
	CelestialBody	40.00	**80.00**	**80.00**	**80.00**	5
	MeanOfTransportation	0.00	0.00	42.86	**85.71**	7
	ArchitecturalStructure	28.57	**50.00**	25.00	**50.00**	4
	Device	25.00	**75.00**	0.00	25.00	4

Table 2: DIRECT SUBSUMPTION performance on *Most* and *Least* Frequent Classes.

On `Inverse Subsumption` results confirm trends about popularity (see Table 3) that are similar to those of *lexical protoknowledge*. GPT-4 leads, exceeding 90% on four out of five classes and reaching 100% on Person. GPT-3.5 Turbo maintains good performance for the most popular classes but struggles on the least. Llama-3-70B behaves similarly, predicting the Superclass better for the most popular. Again Llama-3-8B struggles, particularly on rare classes. Notably, GPT-4 maintains high accuracy even for rare categories (e.g., 85.72% on *MeanOfTransportation*), suggesting its generalization capabilities are less related to KG items' popularity.

	CLASS	Llama-3-8B	Llama-3-70B	GPT-3.5	GPT-4	SUPPORT
MOST	Organisation	7.7	76.9	76.9	**92.3**	13
	Place	18.2	36.4	63.6	**90.1**	11
	Work	0.0	33.3	41.7	**100.0**	12
	Person	19.4	61.1	80.6	**94.4**	36
	Species	0.0	0.0	0.0	**33.3**	3
LEAST	SportFacility	0.0	0.0	0.0	**50.0**	4
	UnitOfWork	0.00	**40.0**	0.0	0.0	2
	CelestialBody	0.0	40.0	20.0	**60.0**	5
	MeanOfTransportation	0.0	42.9	0.0	**85.7**	7
	ArchitecturalStructure	0.0	25.0	25.0	**75.00**	4
	Device	0.0	**75.0**	50.0	70.0	4

Table 3: INVERSE SUBSUMPTION performance on *Most* and *Least* Popular Classes

4.4 *Topological Protoknowledge*

KATs: We design Knowledge Activation Tasks (KATs) to assess *topological protoknowledge* via *Subject-Verb-Object* SVO triple completion:

SV?: Given S and V, predict O (Prompt 7).

?VO: Given V and O, predict S (Prompt 8).

Speculative Protoknowledge for SPARQL. We argue that Text-to-SPARQL strongly activates this *protoknowledge* type, since generating correct queries requires retrieving and combining triples. Following Moiseev et al. (2022), we hypothesize that *protoknowledge* on triples acquired during pretraining enhances this task. We therefore define and measure *topological protoknowledge* speculatively on Text-to-SPARQL benchmarks to assess its impact. However, *topological protoknowledge* can also be studied in any task where triple-centric reasoning is essential. To study it in Text-to-SPARQL, we introduce a framework that first constructs a dedicated test set and then evaluates the model's ability to tackle it. We define the ***Speculative Protoknowledge for SPARQL*** (SPS) as a score:

$$\text{SPS} = \frac{|T_{\text{predicted}} \cap T_{\mathcal{Q}}|}{|T_{\mathcal{Q}}|} \quad (1)$$

where $T_{\mathcal{Q}}$ is the set of entity-property pairs extracted from $\mathcal{Q}_{\text{gold}}$ and $T_{\text{predicted}}$ is the set of pairs for which the model correctly inferred a valid triple. Triple validity is verified using ASK query[3].

Test Set. We evaluate SPS on the DBpedia and Wikidata versions of the Text-to-SPARQL Dataset QALD-9plus, extracting all entity-property pairs relevant to $\mathcal{Q}_{\text{gold}}$.

Metrics. On generated triples, we distinguish between two levels of correctness: Perfect match, where the predicted URI exactly corresponds to the gold triple, and Soft match, where the predicted URI is related to the correct entity but through a different property. We considered Soft matches as they reflect a form of partially relevant knowledge activation.

Results *Topological protoknowledge* is generally stronger on DBpedia than Wikidata, likely due to the human-readable nature of URIs. In Figure 2, the GPT models consistently exhibit higher SPS scores than the Llama models in both datasets.

[3]ASK is a type of SPARQL query used for checking the existence of a triple inside the KG returning True or False.

Among models, GPT-4 achieves the highest Perfect SPS in all configurations except for the ?VO task on DBpedia, where GPT-3.5 Turbo slightly outperforms it. Comparing these two, GPT-4 shows lower Soft SPS compared to GPT-3.5 Turbo in almost all settings, indicating a more precise activation of *topological protoknowledge*. Within the Llama family, we observe a consistent increase in both Soft and Perfect SPS scores when moving from 8B to the larger 70B, confirming a size-dependent improvement in *topological protoknowledge*. Results on smaller models are displayed in Appendix 8.

An additional analysis based on popularity (see Table 12) shows that correctly predicted triples are typically associated with higher popularity of the involved KG items. GPT-4 completed 40 triples (on both tasks) with joint perfect accuracy, of which 27 (67.5%) exceeded the median popularity; similar trends were observed for GPT-3.5 (19/28), Llama-3-70B (8/12), and Llama-3.1-8B (7/11).

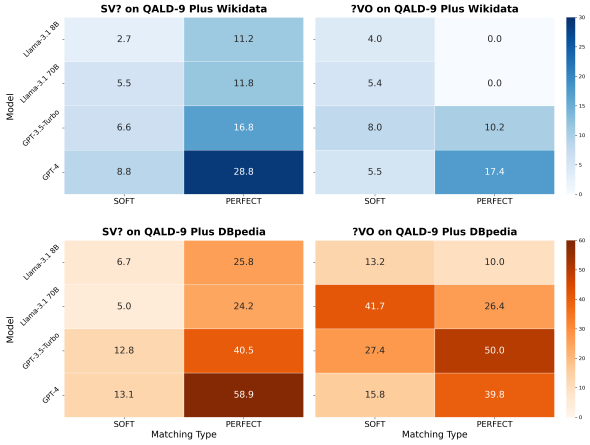

Figure 2: SPS scores for Wikidata and DBpedia.

4.5 General Comment: Semantic Bias and Overconfidence

Popular KG entities and properties show higher accuracy across measurements of *lexical*, *hierarchical*, and *topological protoknowledge*. This consistent pattern across models and tasks indicates that *protoknowledge* on measured KGs are someway aligned with Web content distribution.

Assuming that pretraining data align with the distributional properties of general-domain KGs such as Wikidata and DBpedia, we interpret this as evidence of a persistent **semantic bias**: models internalize KG information more effectively for more popular items. Consistent with broader findings (Elazar et al., 2023; Manvi et al., 2024), this leads to an **overconfidence effect** where performance rises when test data reflect dominant semantic patterns.

These results stress the need to account for semantic distribution in evaluation benchmarks and to interpret performance not only by task competence but also by the latent frequency priors embedded in pretraining.

5 Protoknowledge shapes Text-to-SPARQL

We analyze how different forms of *KG protoknowledge* influence model performance in Text-to-SPARQL, where the core challenge lies in semantically accurate selection and positioning of KG entities and properties. Using the KGQA framework, we test models under three prompting setups with increasing contextual support, each requiring different levels of implicit reasoning and knowledge activation.

Experimental Setting. The baseline prompt (referred to as `Original`) relies on zero-shot from D'Abramo et al. (2025), where for each question, all the entities and relations URIs needed to build the query (and their label) are given as input context. In `No Label`, URIs are not accompanied by their corresponding labels, thereby requiring the model to recognize them based on its acquired information. In `No URI` instead, no additional information is given to support the model. On KGs like DBpedia, where labels are incorporated in the URI, we limit the experiments to `Original` and `No URI` variants. Prompts are summarized in Table 15. For more details, see the prompts in Appendix F.

Datasets. We considered QALD-9 (Unger et al., 2012), referred to as **QALD-9 DB**, a KGQA dataset with queries based on DBpedia, and a parallel version based on Wikidata, QALD-9 Plus (Perevalov et al., 2022) referred to as **QALD-9 WD**.

Metrics. Performance is evaluated using the F1 score, measuring overlap between answers from the target and predicted SPARQL queries. An F1 score of 1 is assigned when both queries return an empty set, and the final score is obtained by averaging all the examples.

Results. Table 4 , shows a consistent performance degradation across all models from `Orginal` to `No URI` according to the richness of contextual information.

On QALD-9 Wikidata, GPT-4 outperforms all other models across all settings and exhibits the

smallest drop in `No Label`, suggesting the influence of *lexical protoknowledge*. GPT-3.5 Turbo, while weaker overall, shows a similarly small drop from `Original` to `No Label`. Llama models have a similar trend, with Llama-3.1-70B performing better in `No URI`.

On QALD-9 DBpedia, GPT-4 again leads in `No URI`, while Llamas slightly outperform it in the `Original` setting, with Llama-3.1-70B remaining competitive across prompts. GPT-3.5 Turbo performs poorly across all configurations. Full results for smaller models are reported in Table 16.

As later shown in the *KG protoknowledge* impact analysis, errors in Text-to-SPARQL arise from semantics (wrong URI choice) rather than syntax (correct query form). To confirm that the main challenge is not simple program synthesis (producing syntactically valid queries) but also relying on KG formal rules, we run an additional experiment (discussed in G.3) with **Rephrased**, varying in linguistic form. Results in Table 18 indicate that stronger models are increasingly robust to such prompt variations.

Model	Approach	QALD-9 WD	QALD-9 DB
Llama-3.1 70B	`Original`	56.34	61.45
	`No Label`	47.7	-
	`No URI`	13.41	13.94
Llama-3 70B	`Original`	57.88	62.79
	`No Label`	48.1	-
	`No URI`	4.20	24.87
GPT-4	`Original`	62.74	59.32
	`No Label`	58.0	-
	`No URI`	25.14	29.09
GPT-3.5-Turbo	`Original`	23.79	47.21
	`No Label`	20.7	-
	`No URI`	3.67	9.29

Table 4: Text-to-SPARQL Performance.

5.1 Impact of *Protoknowledge*

Framework To study the influence of *KG protoknowledge* on Text-to-SPARQL performance, we assess for each query whether the relevant *protoknowledge* type on its content is correctly activated. The general assumption is that a correct SPARQL query implies successful activation of *protoknowledge*. The reverse is not always true: for *lexical* and *hierarchical protoknowledge*, correct *protoknowledge* does not guarantee full query success. Conversely, for *topological protoknowledge*, being more exhaustive and measured speculatively for the task, we hypotesize also the reverse. Hence, we measure on all forms **Positive Agreement** (PA) as the cases where correct SPARQL generation is accompanied by correct *protoknowledge* activation on its content. Additionally, for *topological* form we measure **Agreement** the cases where SPARQL query generation and *protoknowledge* activation are both correct or incorrect (complementary cases are measured with **Disagreement**). A brief description is reported in Appendix Sec. H.1.

Topological As observed in the results (Table 4), all LLMs struggle when no additional KG information is provided. Performance in `No URI` strongly correlates with SPS scores: models tend to fail in Text-to-SPARQL when they also fail the corresponding triple prediction. Thus, without URI cues, successful query generation relies heavily on the model's *topological protoknowledge* of the entities and properties involved.

To capture this, we compare query-by-query the model's SPS scores (Figure 2) and Text-to-SPARQL performance in both `Original` and `No URI` conditions. Examples of Agreement and Disagreement are reported in Appendix Sec. H.2.

Figure 3 shows that in `No URI`, Agreement consistently exceeds Disagreement, confirming high reliance on *protoknowledge* when context is absent. Conversely, in `Original`, Disagreement dominates slightly, indicating a shift towards contextual information. Nonetheless, *topological protoknowledge* remains relevant even with full input. Notably, GPT-3.5 achieves 100% PA in `No URI`, fully relying on *protoknowledge* for correct queries without additional context.

We also examine whether something analogous holds across KGs. Comparing Positive Agreement between Wikidata and DBpedia (Figure 15 and Figure 16), we observe lower values on Wikidata in most models. However, in the `No URI` setting, the difference in PA between the two KGs is minimal, with GPT-3.5 even achieving 100% on both. The exception is Llama-3-70B, where PA is based on only three examples (this ratio decreases further in the `Original` approach).

In `Original`, the gap between DBpedia and Wikidata widens again, with DBpedia consistently higher. GPT-4, however, maintains high PA on Wikidata, reflecting its strong performance on this KG.

Lexical We analyze the impact of *lexical protoknowledge* in the `No Label` setting, where non-human-readable URIs are provided in the input

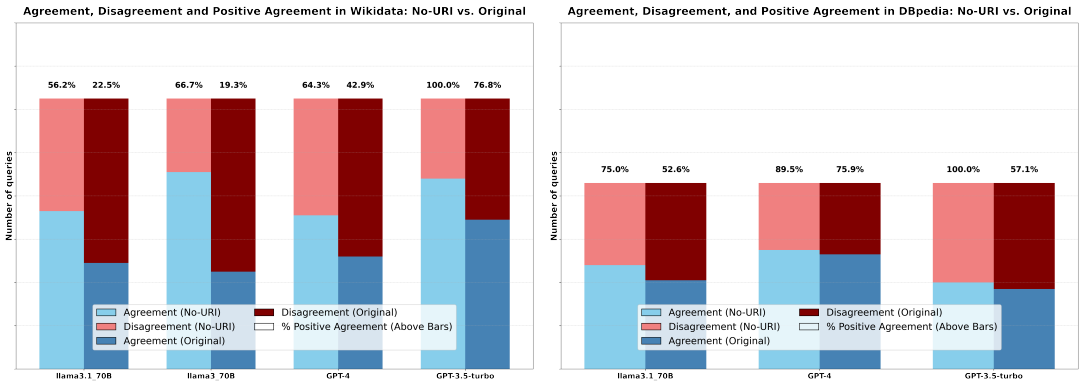

Figure 3: NO-URI vs Original Agreement and Disagreement. Above bar is reported Positive Agreement Ratio %.

prompt (considering QALD-9 WD experiments), and no explicit mapping to label is given. In this scenario, the model must implicitly solve the link between the natural language query containing surface forms or approximate label mentions and the correct KG identifiers. A visual description is reported in Appendix 12.

We report for a subset of models (Tab 5) the Positive Agreement (PA) as the ratio of correctly generated queries for which *lexical protoknowledge* was also measured successfully (see example in Fig. 12). The analysis excludes items below the 50th percentile of popularity (see Tab 14 in Appendix). It's clearly observable that in most cases, the PA ratio is very high, achieving 100% for GPT-3.5 Turbo.

Model	Llama-3-70B	Llama-3.1-70B	GPT-3.5 Turbo	GPT-4
PA ratio	28/40	15/27	13/13	41/50

Table 5: PA with *lexical protoknowledge* on QALD-9 WD

Model	Llama-3-70B	Llama-3.1-70B	GPT-3.5 Turbo	GPT-4
PA ratio	32/40	35/40	22/31	40/43

Table 6: PA with *hierarchical protoknowledge* on QALD-9 DB

Hierarchical We analyze *hierarchical protoknowledge* on the QALD-9 DB, extending the evaluation beyond DBpedia Ontology classes to include other *hierarchical* relations such as `rdf:type` and `rdf:subpropertyOf`. Here we analyze the Positive Agreement (PA) as the ratio of correctly generated queries for which *hierarchical protoknowledge* was successful.

Results in Tab 6 show the presence of *hierarchical protoknowledge* in successful completions. GPT-4 achieves the highest alignment with a score of 40/43 correct queries, followed by Llama-3.1-70B (35/40), Llama-3-70B (32/39) and GPT-3.5 Turbo (22/31). These findings suggest that correct query generation strongly correlate with models' ability to internalize and reuse hierarchical relations learned from pretraining.

We conduct an additional experiment extending the analysis to an Ontology Alignment (OA) task, presented in the Appendix (H.4). This setting illustrates how the approach can also measure the impact of *KG protoknowledge* to other tasks. Results (see Fig 20) obtained reveal a consistent correlation, as we observe a comparable trend when computing the PA ratio between OA performance and the relevant *hierarchical protoknowledge*.

6 Conclusions

We formalize three core forms of *protoknowledge* within the context of Knowledge Graphs (KGs): *lexical*, *hierarchical*, and *topological*. By designing targeted Knowledge Activation Tasks and constructing dedicated test sets, we show that *KG protoknowledge* strongly aligns with web-scale semantic bias, which we hypothesize to be reflected in the pretraining of LLMs. We further propose a novel framework that correlates Text-to-SPARQL performance with specific forms of *KG protoknowledge*. Our findings indicate that *protoknowledge* strongly influences models' behaviour in this task. Particularly, *topological* proves the most predictive of Text-to-SPARQL success, given the nature of the structured triple information it captures. Yet, the other forms of *protoknowledge* remain valuable. Indeed, *lexical protoknowledge*, is helpful in tackling the `No Label` setting, confirming its relevance in tasks requiring URI recognition.

We show that structured knowledge internalization is central to LLM behaviour and provide a systematic framework to characterize models' abstraction capacities, examining how memorized information is reused and how different abstraction levels shape their outputs.

Limitations

Our analysis of *KG protoknowledge* is restricted to DBpedia and Wikidata, limiting insights into other Knowledge Graphs (e.g., Freebase, YAGO). Additionally, a deeper comparison between DBpedia and Wikidata would enhance understanding of how *KG protoknowledge* varies across different KG structures.

Expanding to more Text-to-SPARQL benchmarks beyond QALD-9 Plus (e.g., LC-QuAD, GrailQA) would improve the generalizability of our research.

Topological protoknowledge is measured by applying triple completion tasks (SV? and ?VO) lacking the S?O task. This omission was intentional, as properties like `wd:P31` ("instance of" in Wikidata) are frequently used in KGQA benchmkarks, making the evaluation less effective and unfairly favouring this task over others in the SPS metric.

Lexical Protoknowledge was specifically analyzed in Wikidata to highlight the challenge of memorizing non-human-readable URIs. *Hierachical Protoknowledge* was examined systematically within the DBpedia Ontology subgraph which features a well-defined *hierarchical* structure and in a small way in other KG-portions extracted from QALD-9 DB. Both analyses deserve expansion to more samples and other KGs based on specific domains, in order to see if *KG protoknowledge* is still present.

Finally, for computational reasons only GPT-4, GPT-3.5 Turbo, and Llama-3.x were tested. To maintain a fair comparison in this study, we had to exclude open models that, under unsupervised pretraining settings and poor-information prompt settings, did not reach performance levels high enough to allow for meaningful analysis. Anyway, evaluating newer LLMs and retrieval-augmented models would refine our understanding of *KG protoknowledge* in evolving architectures.

References

Marco Bombieri, Paolo Fiorini, Simone Paolo Ponzetto, and Marco Rospocher. 2025. Do llms dream of ontologies? *Preprint*, arXiv:2401.14931.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2023. Quantifying memorization across neural language models. *Preprint*, arXiv:2202.07646.

Kent K. Chang, Mackenzie Cramer, Sandeep Soni, and David Bamman. 2023. Speak, memory: An archaeology of books known to chatgpt/gpt-4. *Preprint*, arXiv:2305.00118.

Yuxing Cheng, Yi Chang, and Yuan Wu. 2025. A survey on data contamination for large language models. *Preprint*, arXiv:2502.14425.

Jacopo D'Abramo, Andrea Zugarini, and Paolo Torroni. 2025. Investigating large language models for text-to-SPARQL generation. In *Proceedings of the 4th International Workshop on Knowledge-Augmented Methods for Natural Language Processing*, pages 66–80, Albuquerque, New Mexico, USA. Association for Computational Linguistics.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Amir Feder, Abhilasha Ravichander, Marius Mosbach, Yonatan Belinkov, Hinrich Schütze, and Yoav Goldberg. 2023. Measuring causal effects of data statistics on language model's 'factual' predictions. *Preprint*, arXiv:2207.14251.

Xiyan Fu and Anette Frank. 2024. The mystery of compositional generalization in graph-based generative commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 8376–8394, Miami, Florida, USA. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Pei-Chi Lo, Yi-Hang Tsai, Ee-Peng Lim, and San-Yih Hwang. 2023. On exploring the reasoning capability of large language models with knowledge graphs. *Preprint*, arXiv:2312.00353.

Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. 2024. Reasoning on graphs: Faithful and interpretable large language model reasoning. *Preprint*, arXiv:2310.01061.

Giulio Macilenti, Armando Stellato, and Manuel Fiorelli. 2024. Prompting is not all you need evaluating gpt-4 performance on a real-world ontology alignment use case. *Procedia Computer Science*, 246:1289–1298. 28th International Conference on Knowledge Based and Intelligent information and Engineering Systems (KES 2024).

Inbal Magar and Roy Schwartz. 2022. Data contamination: From memorization to exploitation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 157–165, Dublin, Ireland. Association for Computational Linguistics.

Rohin Manvi, Samar Khanna, Marshall Burke, David Lobell, and Stefano Ermon. 2024. Large language models are geographically biased. *Preprint*, arXiv:2402.02680.

Fedor Moiseev, Zhe Dong, Enrique Alfonseca, and Martin Jaggi. 2022. SKILL: Structured knowledge infusion for large language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1581–1588, Seattle, United States. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Ralph Peeters, Aaron Steiner, and Christian Bizer. 2024. Entity matching using large language models. *Preprint*, arXiv:2310.11244.

Aleksandr Perevalov, Dennis Diefenbach, Ricardo Usbeck, and Andreas Both. 2022. Qald-9-plus: A multilingual dataset for question answering over dbpedia and wikidata translated by native speakers. *Preprint*, arXiv:2202.00120.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Federico Ranaldi, Elena Sofia Ruzzetti, Dario Onorati, Leonardo Ranaldi, Cristina Giannone, Andrea Favalli, Raniero Romagnoli, and Fabio Massimo Zanzotto. 2024. Investigating the impact of data contamination of large language models in text-to-SQL translation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13909–13920, Bangkok, Thailand. Association for Computational Linguistics.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2024. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. *Preprint*, arXiv:2310.11324.

Christina Unger, Lorenz Bühmann, Jens Lehmann, Axel-Cyrille Ngonga Ngomo, Daniel Gerber, and Philipp Cimiano. 2012. Template-based question answering over rdf data. *WWW'12 - Proceedings of the 21st Annual Conference on World Wide Web*.

Xinyi Wang, Antonis Antoniades, Yanai Elazar, Alfonso Amayuelas, Alon Albalak, Kexun Zhang, and William Yang Wang. 2025. Generalization v.s. memorization: Tracing language models' capabilities back to pretraining data. *Preprint*, arXiv:2407.14985.

Weiqi Wu, Chengyue Jiang, Yong Jiang, Pengjun Xie, and Kewei Tu. 2023. Do PLMs know and understand ontological knowledge? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3080–3101, Toronto, Canada. Association for Computational Linguistics.

Cheng Xu, Shuhao Guan, Derek Greene, and M-Tahar Kechadi. 2024. Benchmark data contamination of large language models: A survey. *Preprint*, arXiv:2406.04244.

Feng Yao, Yufan Zhuang, Zihao Sun, Sunan Xu, Animesh Kumar, and Jingbo Shang. 2024. Data contamination can cross language barriers. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17864–17875, Miami, Florida, USA. Association for Computational Linguistics.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3911–3921, Brussels, Belgium. Association for Computational Linguistics.

Qingkai Zeng, Yuyang Bai, Zhaoxuan Tan, Shangbin Feng, Zhenwen Liang, Zhihan Zhang, and Meng Jiang. 2024. Chain-of-layer: Iteratively prompting large language models for taxonomy induction from limited examples. *Preprint*, arXiv:2402.07386.

Yao Zhang, Hongru Liang, Adam Jatowt, Wenqiang Lei, Xin Wei, Ning Jiang, and Zhenglu Yang. 2021. GMH: A general multi-hop reasoning model for KG completion. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3437–3446, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yizhuo Zhang, Heng Wang, Shangbin Feng, Zhaoxuan Tan, Xiaochuang Han, Tianxing He, and Yulia Tsvetkov. 2024. Can LLM graph reasoning generalize beyond pattern memorization? In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 2289–2305, Miami, Florida, USA. Association for Computational Linguistics.

A Knowledge Activation Tasks Prompts

Task: Guess the URI by Label

Return the label of the entity with the following Wikidata ID: `uc[0]`.
Important: Answer only a string according to your knowledge of Wikidata.
Important: The string must satisfy the triple: `<http://www.wikidata.org/*/uc[0], rdfs:label, [YOUR ANSWER STRING]>`.
Important: Do not answer other stuff apart from the English label.

Figure 4: URI Recognition Task

Task: Inverse Subsumption

Using your knowledge of DBpedia ontology, provide the URI of the superclass for the resource `[SUBCLASS]`.
Important: Your response must contain only the URI.
Important: The URI must satisfy the triple: `<[SUBCLASS], rdfs:subClassOf, [YOUR ANSWER URI]>`.

Figure 5: Inverse Subsumption Task

Task: Direct Subsumption

Using your knowledge of DBpedia ontology, return only a list of URIs that are direct subclasses of `CLASS`.
Important: URIs must be connected to `CLASS` by the property rdfs:subClassOf.
Important: URIs must satisfy the triple: `<[URI]>,rdfs:subClassOf,CLASS>`.

Figure 6: Direct Subsumption Task

Task: SV? Guess the OBJECT

Considering your knowledge of DBpedia triples, can you fill the masked `[MASKED_OBJECT]` with an existing URI inside DBpedia?
TRIPLE: `{{'S','V',[MASKED_OBJECT]}}`
Important: Answer only the URI! Do not invent URIs.

Figure 7: SV? Task

Task: ?VO Guess the SUBJECT

Considering your knowledge of DBpedia triples, can you fill the masked `[MASKED_SUBJECT]` with an existing URI inside DBpedia?
TRIPLE: `{{[MASKED_SUBJECT],'V','O'}}`
Important: Answer only the URI! Do not invent URIs.

Figure 8: ?VO Task

B Models and Hyperparameters

To get a comprehensive evaluation, we use four different LLMs: GPT-4, GPT-3.5 (OpenAI, 2023), Llama-3.1-8B, Llama-3-8B, Llama3-70b-instruct (Grattafiori et al., 2024). We use greedy decoding in all experiments to ensure a more deterministic generation process. We set the temperature to 0 and the maximum generation length to 2048. We observed that these settings deliver better and deterministic performance.

Model	Version
GPT-4	OpenAI API (gpt-4-o)
GPT-3.5-Turbo	OpenAI API (gpt-3.5-turbo)
Llama-3-8B	meta-llama/Meta-Llama-3-8B-Instruct
Llama-3-70B	meta-llama/Meta-Llama-3-70B-Instruct
Llama-3.1-8B	meta-llama/Llama-3.1-8B-Instruct
Llama-3.1-70B	meta-llama/Llama-3.1-70B-Instruct

Table 7: Models (huggingface.co). We used the configurations described in Section 4.1 in the repositories for each model *(access verified on 18 May 2025).

C Topological protoknowledge: SPS score on parallel Text-to-SPARQL benchmarks

Model	Dataset	KG	S V ?		? V O		Perfect	Loose
			S	P	S	P	SV+VO	SV+VO
Llama-3 8B	QALD-9 Plus	Wikidata	1.00	0.76	4.11	0.3	2.37	1.5
Llama-3 70B	QALD-9 Plus	Wikidata	5.71	7.81	2.11	8.62	9.45	8.27
Llama-3.1 8B	QALD-9 Plus	Wikidata	2.71	11.25	4.00	0.00	7.66	3.75
Llama-3.1 70B	QALD-9 Plus	Wikidata	5.55	11.77	5.37	0.00	14.52	13.53
GPT-3.5-Turbo	QALD-9 Plus	Wikidata	6.56	16.77	**8.01**	10.19	19.90	21.05
GPT-4	QALD-9 Plus	Wikidata	**8.76**	**28.77**	5.47	**17.37**	**27.29**	**30.07**
Llama-3 8B	QALD-9 Plus	DBpedia	**20.00**	12.50	30.90	10.00	34.41	39.39
Llama-3 70B	QALD-9 Plus	DBpedia	13.33	39.16	**43.48**	25.45	58.50	62.62
Llama-3.1 8B	QALD-9 Plus	DBpedia	6.67	25.83	13.18	10.00	25.84	29.29
Llama-3.1 70B	QALD-9 Plus	DBpedia	5.00	24.17	41.66	26.36	47.79	51.52
GPT-3.5-Turbo	QALD-9 Plus	DBpedia	12.77	40.55	27.42	**50.00**	**62.54**	**66.67**
GPT-4	QALD-9 Plus	DBpedia	13.05	**58.88**	15.75	39.84	58.96	63.15

Table 8: Results overview on triple completion tasks performed on Llamas and GPTs. The tasks are conducted on two versions of QALD-9 Plus, based on Wikidata and DBpedia. P represents "Perfect" satisfaction of the triple, while S represents "Soft" satisfaction of the triple. "Perfect SV+VO" reports the joint evaluation of the triples contingent to a predicted query that are satisfied in a "Perfect" manner. "Loose SV+VO" reports the joint evaluation of the triples contingent to a predicted query that are satisfied either "Perfectly" or in a "Soft" manner.

D Semantic Bias Analysis

D.1 URI recognition task evaluated on unbiased random set

The third set, was built from the Freebase-Wikidata-Mapping dataset (1M+ triples linking Freebase IDs, Wikidata IDs, and `rdfs:label`), is unsorted by popularity and thus includes both popular or not entities. We randomly sample 1K (`Wikidata ID, label`) pairs to reduce inductive bias and test *lexical protoknowledge* over a broader distribution. The first two test sets are inherently biased: they contain the most popular Wikidata items, resulting in very high average frequencies (800K for entities, 2M for properties). In contrast, the third test, containing 1k samples extracted randomly, presents a much lower popularity (50K average against the 800k of the first set). For computational reasons, this set was evaluated only on Llama models (Results reported on 9). An additional experiment including also GPTs is conducted on 100 random samples from the third set whose popularity mean is now around 2k (Results reported on 10). In D.2 is reported also an error analysis on this last subset. On both tables 9 and 10 we can observe that even if accuracies scaled negatively with respect to first and second set, the trend persists: models perform better on more frequent entities.

Criterion	Subset	Llama-3-8B	Llama-3-70B
Med.-based	0:500	0.0% (0/3)	0.00% (0/24)
Med.-based	500:1000	0.6% (3/3)	4.8% (24/24)
Mean-based	0:912	0.21%(1/3)	1.35% (11/24)
Mean-based	912:1000	2.29% (2/3)	14.94% (13/24)

Table 9: URI Recognition Task Accuracy. Two splitting criteria were considered: median-based and mean based. Although the second method of splitting leads to two subsets of incomparable size, both models tend to retrieve the IDs of entities whose popularity is above the avg. The ratio of correct prediction over the subset of LF or MF is also reported near accuracy.

Crit.	Subset	Llama-3-8B	Llama-3-70B	GPT-3.5Turbo	GPT-4
Med.	0:50	0% (0/1)	4% (2/6)	4% (2/15)	4% (2/20)
Med.	50:100	2% (1/1)	8% (4/6)	26% (13/15)	36% (18/20)
Mean	0:93	1.1% (1/1)	4.3% (4/6)	10.8% (10/15)	15.1% (14/20)
Mean	93:100	0% (0/1)	33.3% (2/6)	83.3% (5/15)	100% (6/20)

Table 10: URI Recognition Task Accuracy on 100 random entities to extend the analysis also to GPTs. Popularity mean of this set is around 2k. Two splitting criteria were considered: median-based and mean-based. Although the second method of splitting leads to two subsets of incomparable size, both models tend to retrieve the IDs of entities whose popularity is above the average. The ratio of correct prediction over the subset of LF or MF is also reported near accuracy.

D.2 Error Analysis on URI Recognition Task

We conduct an error analysis on the URI recognition task by splitting the errors between **Non-Existent** IDs generated, Existent but **Unrelated** and **Related**. Under **Related** we consider all those generations whose IDs are connected with the gold ID by another property. The detection method for **Related** IDs is analagous to Soft-Accuracy (see 4.4). **Related** IDs are like generating the ID of "Germany" instead of "Berlin". **Unrelated** are generated IDs without direct connections with the gold ID. Results reported on 11 shows that most proficient LLMs tends to generate **Unrelated** IDs with GPT reporting also a non-negligible number of **Related**.

Model	Related	Unrelated	Non-Existent
GPT-4	18/80	62/80	0/80
GPT-3.5 Turbo	8/88	6/88	74/88
LLaMA-3 70B	9/94	10/12	6/12
LLaMA-3 8B	3/99	60/11	20/11

Table 11: Error analysis for URI RECOGNITION TASKS on the subset of 100 random samples from the third set (the unbiased). Errors are splitted in Related, Unrelated and Non-Existent basing on the ID generated.

Model	Above Median
GPT-4	27/40
GPT-3.5 Turbo	19/28
LLaMA-3 70B	8/12
LLaMA-3.1 70B	7/11

Table 12: Percentage of correctly predicted triples on both triple completion tasks (Perfect Accuracy on Joint > 0.5) whose average popularity exceeds the dataset mean (12.8M) and median (1.9M). Under Above Median is reported the number of triples with an high average popularity (computed on its items) over all correctly completed triples.

E Statistics on KG items

Class	Example Subclasses	Occurrences
Person	Astronaut, Politician, ...	3,415,598
Species	Archaea, Eukaryote, ...	3,989,728
Place	WineRegion,Park ...	1,526,870
Work	Database, MusicalWork, ...	1,439,666
Organisation	NonProfitOrganisation,, ...	869,554
Device	Battery, Engine, ...	72,356
MeanOfTransportation	SpaceShuttle, Train, ...	34,768
CelestialBody	Constellation, Planet, ...	30,130
Arch. Structure	Tunnel, Tower, ...	13,636
UnitOfWork	Project, Case, ...	7,100
SportFacility	CricketGround, RaceTrack, ...	5,792

Table 13: Popularity of subclasses for top-level DBpedia classes. DBpedia Ontology was choosen for its well-formed hierarchical structure lacking inconsistencies that are common in other KGs.

ID	Occurrence	Popularity	Label
Entities			
Q55	2	1,058,631	Netherlands
Q82955	1	1,748,496	politician
Q34	1	1,777,306	Sweden
Q148	2	2,340,375	China
Q145	1	3,181,387	UK
Q183	5	3,556,517	Germany
Q6581072	3	5,379,193	female
Q5	2	5,586,977	human
Q30	2	7,026,527	USA
Q2	1	11,862,402	Earth
Properties			
P27	3	5,575,274	citizenship
P569	2	7,105,365	birthdate
P21	3	10,044,816	sex or gender
P106	12	11,964,330	occupation
P131	4	13,853,240	located in
P17	17	18,663,328	country
P50	4	34,011,532	author
P1433	1	45,835,521	published in
P577	2	49,376,272	publication date
P31	40	118,388,701	instance of

Table 14: Top 10 entities and properties above the 50th percentile in QALD-9 WD with their dataset frequency, popularity, and label. We observe that the 50% of the 50th percentile corresponds to the global top 100 properties and the 25% of the 50th percentile corresponds to the global top 200 entities.

G Text-to-SPARQL Experiments

G.1 Text-to-SPARQL Prompt Configurations

Approach	Description
`Original`	explicits URI-label associations
`No Label`	URIs w/o label associations
`No URI`	No any additional informations.

Table 15: Text-to-SPARQL Prompt Configurations.

G.2 Extensive Text-to-SPARQL results

Model	Approach	QALD-9 WD	QALD-9 DB
Llama-3.1 70B	`Original`	56.34	61.45
	`No Label`	47.7	-
	`No URI`	13.41	13.94
Llama-3 70B	`Original`	57.88	62.79
	`No Label`	48.1	-
	`No URI`	4.20	24.87
Llama-3.1 8B	`Original`	43.60	21.24
	`No Label`	35.3	-
	`No URI`	0.00	3.37
Llama-3 8B	`Original`	49.69	29.34
	`No Label`	40.29	-
	`No URI`	0.00	5.93
GPT-4	`Original`	62.74	59.32
	`No Label`	58.0	-
	`No URI`	25.14	29.09
GPT-3.5-Turbo	`Original`	23.79	47.21
	`No Label`	20.7	-
	`No URI`	3.67	9.29

Table 16: Performance of Llamas and GPTs on QALD-9 -Plus in Wikidata and DBpedia version on the Original and No URI approaches.

G.3 Prompt Variation Experiments

To emphasise that a key difficulty of the task lies not only in mapping natural language instructions to Text-to-SPARQL but also in handling the formal ontology constraints that require KG-protoknowledge, we conduct a short prompt perturbation experiment. We reformulate the natural language queries from a direct, common form into a more indirect **Rephrased** version. As illustrated in Table 17, these indirect variants were generated with Llama-3.1 405B. The results in Table 18 show only minor performance variations for Llama-3 70B and Llama-3.1 70B. These results are in line with many related works on the effect of nl prompt variations on proficient models performances (Sclar et al., 2024).

Original	Rephrased
What is the time zone of Salt Lake City?	I would like to know the time zone of Salt Lake City.
How many companies were founded by the founder of Facebook?	I would like to know the number of companies founded by the founder of Facebook.
When did Paraguay proclaim its independence?	I would like to know when Paraguay proclaimed its independence.

Table 17: Examples of original and rephrased questions used in the prompting approach (paraphrasing with Llama-3.1 405B).

Model	Approach	QALD-9 WD	QALD-9 DB
Llama-3 70B	`Original`	57.88	62.79
	`Rephrased`	56.43	62.04
Llama-3.1 70B	`Original`	56.34	59.82
	`Rephrased`	54.96	59.97

Table 18: Performance of Llama-3 and Llama-3.1 (70B) on QALD-9 (Wikidata and DBpedia) for the Original and Rephrased approaches.

F Text2SPARQL Prompts

Approach: Original

You are an expert in SPARQL and `{KG_name}`.
Your task is to translate natural language questions into precise SPARQL queries that retrieve the desired information from `{KG_name}`.
Guidelines:

1. Understand the input: Analyze the question and use the provided Entities and Relations to construct the query.
2. Construct a valid SPARQL query: Use proper syntax and ensure the query retrieves accurate results from `{KG_name}`.
3. Format the output: Enclose the SPARQL query within <SPARQL></SPARQL> tags. Do not output anything else.

Question: `{question}`
Entities: `{URI} ({label}), ...`
Relations: `{URI} ({label}), ...`
Query:

Figure 9: Original Approach

Approach: No Label

You are an expert in SPARQL and `{KG_name}`.
Your task is to translate natural language questions into precise SPARQL queries that retrieve the desired information from `{KG_name}`.
Guidelines:

1. Understand the input: Analyze the question and use the provided Entities and Relations to construct the query.
2. Construct a valid SPARQL query: Use proper syntax and ensure the query retrieves accurate results from `{KG_name}`.
3. Format the output: Enclose the SPARQL query within <SPARQL></SPARQL> tags. Do not output anything else.

Question: `{question}`
Entities: `{URI}`, ...
Relations: `{URI}`, ...
Query:

Figure 10: No Label Approach

Approach: No URI

You are an expert in SPARQL and `{KG_name}`.
Your task is to translate natural language questions into precise SPARQL queries that retrieve the desired information from `{KG_name}`.
Guidelines:

1. Understand the input: Analyze the question and use the provided Entities and Relations to construct the query.
2. Construct a valid SPARQL query: Use proper syntax and ensure the query retrieves accurate results from `{KG_name}`.
3. Format the output: Enclose the SPARQL query within <SPARQL></SPARQL> tags. Do not output anything else.

Question: `{question}`
Query:

Figure 11: No URI approach

H Protoknowledge Analysis Impact

H.1 Protoknowledge Analysis Framework in brief

Given a form of *KG Protoknowledge* and a Text-to-SPARQL query instance, we perform the following steps:

1. Extract a mini test set from the query, containing relevant KG elements.
2. Evaluate *protoknowledge* on this mini set using Knowledge Activation Tasks (KATs).
3. If both the *protoknowledge* evaluation and the SPARQL generation are correct, the instance is marked as a **Positive Agreement**.

H.2 Examples of Framework Application on *protoknowledge* forms

We report some examples of framework application for correlating *KG protoknowledge* forms and Text-to-SPARQL results for a given query.

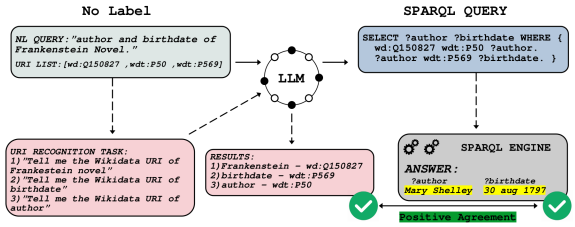

Figure 12: *Lexical protoknowledge* impact on Text-to-SPARQL in `No Label` approach.

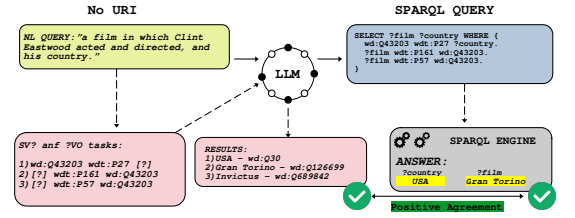

Figure 13: *Topological protoknowledge* impact on Text-to-SPARQL in `No URI` approach.

Fig. 14 reports an example of Disagreement. While SPARQL query is correct syntactically and semantically, the *topological protoknowledge* is not correct with a case in which the model is answering the wrong type of item (P31 is the ID of a possible property that cannot be connected to another property). This may suggest the worst effect of contamination. Model may have memorized the benchmark without acquiring any form of abstract knowledge over the information.

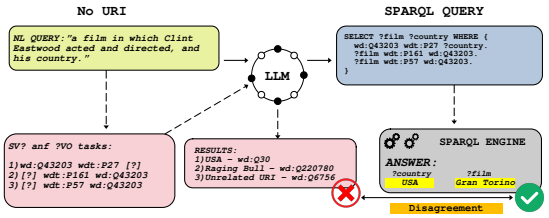

Figure 14: *Topological protoknowledge* disagreement examples on Text-to-SPARQL in `No URI` approach.

H.3 *Topological protoknowledge* analysis: Wikidata vs DBpedia

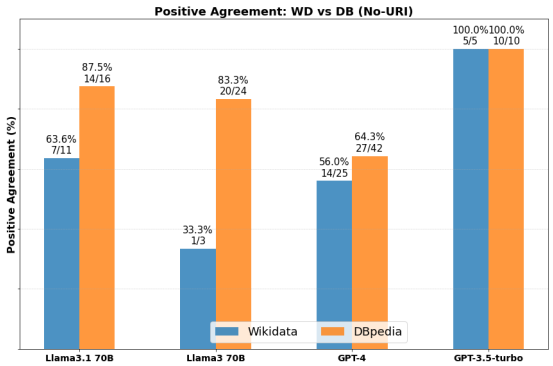

Figure 15: Wikidata vs DBpedia Positive Agreement in `No URI` setting.

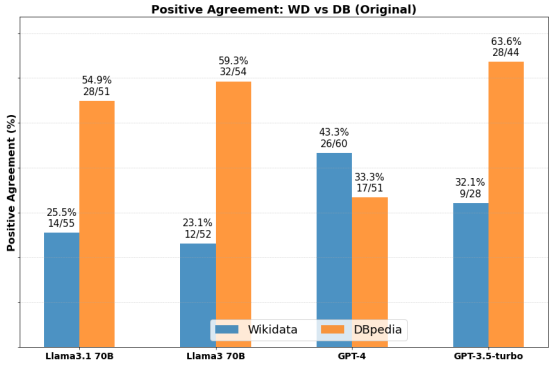

Figure 16: Wikidata vs DBpedia Positive Agreement in `Original` setting.

H.4 *Hierarchical protoknowledge* shapes Ontology Alignment: a short demonstration

We compute the Positive Agreement between the *hierarchical protoknowledge* measured through Subsumption Tasks on the DBpedia Ontology and an Ontology Alignment task with Wikidata. In this setting, the model is prompted to return the Wikidata entity equivalent to a given DBpedia class from the Test Set. The Test Set comprises 60 DBpedia classes, and the gold Wikidata mappings are obtained via the property `owl:equivalentClass`.

DBpedia Class	Wikidata Entity
SportsTeamSeason	Q1539532 (season)
WrittenWork	Q7725634 (literary text)
PersonalEvent	Q1190554 (occurrence)
AmericanFootballCoach	Q41583 (coach)
ArchitecturalStructure	Q811979 (human-designed structure)
Athlete	Q2066131 (sportsperson)
BusCompany	Q10438042 (bus operator)
ComedyGroup	Q18510489 (comedy troupe)
HistoricPlace	Q1081138 (historic location)
TelevisionEpisode	Q21191270 (television show episode)
WineRegion	Q2140699 (wine subregion)

Table 19: Examples of equivalence mappings between DBpedia Ontology classes and Wikidata entities (with the corresponding English labels). The complete Test Set contains 60 pairs.

Given the complexity of the task and the *hierarchical protoknowledge* values reported most successful in GPT-3.5 and GPT-4, we restrict this experiment to these two models, adopting a basic prompt inspired by the ontology alignment setting of (Macilenti et al., 2024).

Ontology Mapping

Using your knowledge of DBpedia and Wikidata, return a Wikidata URI equivalent to the DBpedia Class `CLASS`.
Important: Wikidata URIs must be connected to `CLASS` by the property `owl:equivalentClass`.
Important: You must satisfy the triple:
`< [CLASS] owl:equivalentClass [URI] >`.
Important: Answer only the Wikidata `URI`.

Figure 17: Ontology Alignment prompt.

The results in Table 20 show that Positive Agreement provides a clear signal of the relationship between hierarchical *protoknowledge* and task performance. Despite the difference in absolute accuracy between GPT-3.5 and GPT-4, in both cases we observe a strong alignment between successful ontology mappings and the measurements of *protoknowledge*.

Model	Accuracy	Positive Agreement
GPT-3.5	16/60	12/16
GPT-4	25/60	18/25

Table 20: Ontology Alignment performance of GPT models: Accuracy and Positive Agreement over a Test Set of 60 DBpedia classes.