Text2KGBench-LettrIA: A Refined Benchmark for Text2Graph Systems

Julien Plu[1], Oscar Moreno Escobar[1], Edouard Trouillez[1], Axelle Gapin[1,†], Pasquale Lisena[2], Thibault Ehrhart[2] and Raphaël Troncy[2]

[1]LettrIA, Paris, France
[2]EURECOM, Sophia Antipolis, France

Abstract

Recent advances in Large Language Models (LLMs) have catalyzed significant research into automated knowledge graph (KG) construction from text, a fundamental challenge at the intersection of natural language processing and semantic web technologies. However, the reliability of evaluating model performance is hindered by limitations in existing benchmarks like Text2KGBench, which exhibit shortcomings in data quality, ontological consistency, and structural design. To address these issues, this paper introduces Text2KGBench-LettrIA, a substantially revised and curated benchmark derived from the DBpedia-WebNLG portion of Text2KGBench. Our primary contributions include: (1) the systematic refinement of 19 domain ontologies to enforce hierarchical structure and formal typing; (2) a complete re-annotation of 4,860 sentences, yielding over 14,000 high-fidelity triples under a strict set of annotation guidelines; and (3) the introduction of an enriched data format with enhanced metadata to ensure reproducibility and support multifaceted evaluation. We demonstrate the utility of our benchmark by evaluating a suite of both proprietary and open-source LLMs in zero-shot and fine-tuned settings, respectively. Our results reveal a key finding: smaller, fine-tuned open-source models can achieve superior F1 accuracy compared to their larger, proprietary counterparts, underscoring the critical role of high-quality, schema-aligned training data.

1. Introduction

The recent proliferation of Large Language Models (LLMs) and foundation models has catalyzed significant advancements in Natural Language Processing (NLP). A key area of impact is the automated construction and completion of Knowledge Graphs (KGs), where the synergy between LLMs and structured knowledge is pivotal. This relationship underpins emerging applications such as the generation of explainable AI (XAI) outputs and the development of robust neuro-symbolic fact-checking systems. A significant contribution in this domain is Text2KGBench, a benchmark designed to evaluate the capacity of language models to generate KGs from text under ontological guidance [1]. The framework assesses a model's ability to extract relational triples that both conform to a predefined ontology and remain grounded in the source text. Text2KGBench is composed of two datasets: `Wikidata-TekGen`, derived from the TekGen corpus [2], containing 13,474 sentences across 10 ontologies; and `DBpedia-WebNLG`, based on the WebNLG corpus [3], with 4860 sentences across 19 ontologies. Both split in training and test set. Despite its foundational role, a detailed analysis of Text2KGBench reveals several critical limitations that hinder reliable model evaluation and impede progress. Our investigation, which focuses on the `DBpedia-WebNLG` component, identifies the following principal flaws:

- **Ontological:** The ontologies are semantically imprecise. They suffered from a flat, non-hierarchical design, contained ambiguous and out-of-domain concepts, and lacked the formal rigor needed for robust knowledge representation, making it difficult to use for precise model evaluation and knowledge extraction.

Joint proceedings of KBC-LM and LM-KBC @ ISWC 2025

† Author contributed as consulting.

julien@lettria.com (J. Plu); oscar@lettria.com (O. M. Escobar); edouard@lettria.com (E. Trouillez); pasquale.lisena@eurecom.fr (P. Lisena); thibault.ehrhart@eurecom.fr (T. Ehrhart); raphael.troncy@eurecom.fr (R. Troncy)
0000-0002-7876-3441 (J. Plu); 0000-0003-3094-5585 (P. Lisena); 0000-0003-1377-8279 (T. Ehrhart); 0000-0003-0457-1436 (R. Troncy)

- **Annotation and Data Quality:** The data annotations in the original benchmark were inconsistent and unreliable. This was caused by a lack of standardization for entity names and literal values, a failure to strictly limit annotations to textual evidence, and the presence of grammatical errors in the source sentences.
- **Structural and Technical:** From a technical perspective, the original dataset was difficult to use and lacked features essential for reproducibility. Its data structure was missing key information and contained formatting errors, while the ontologies themselves was undocumented and used an overly complicated URI scheme.

To address these shortcomings, this work makes the following primary contributions:

- We introduce **Text2KGBench-LettrIA**, a rigorously **corrected and enriched benchmark** for ontology-guided KG construction. This new version rectifies annotation errors, ensures ontological compliance, and improves overall data quality to facilitate more accurate and meaningful model evaluation. The benchmark is available upon request to the authors.
- We conduct an extensive **empirical evaluation of diverse language models**, including proprietary APIs and open-source models, on Text2KGBench-LettrIA. Our findings reveal that fine-tuned open models can consistently outperform larger, proprietary models in zero- or few-shot settings, demonstrating their effectiveness for structured information extraction.

Raphael: to check

The remainder of this paper is organized as follows. Section 2 reviews related work on KG construction from text. Section 3 details our methodology for revising the benchmark. Section 4 presents our experimental setup and comparative results. Finally, Section 5 concludes with a summary of our findings and outlines directions for future research.

2. Related Work

The task of automatically constructing Knowledge Graphs (KGs) from unstructured text, commonly known as Text-to-Knowledge-Graph (Text2KG), has become a central challenge in natural language processing and semantic web research. This process facilitates the transformation of textual information into structured, machine-readable knowledge representations. It is a composite task that typically integrates sub-problems such as Named Entity Recognition (NER), Relation Extraction (RE), and Entity Linking (EL), which are orchestrated within either pipeline or end-to-end architectures. For a comprehensive formalization of the problem and an extensive literature review, we direct the reader to the systematic survey by Regino et al. [4].

The growing interest in this field is evidenced by sustained community efforts, including the Text2KG workshop series, held annually since 2022 and approaching its fifth edition in 2025 [5], and the yearly *Knowledge Base Construction from Pre-trained Language Models* (LM-KBC) challenge [6].

These efforts are supported by the development of standardized datasets. One of the earliest and most influential is WebNLG [3], which pairs textual descriptions with RDF-style triples. WebNLG inspired subsequent work like TekGen [2], which expanded the corpus with synthetically generated data. More recently, Text2KGBench [1] established a benchmark to evaluate the generation of ontology-compliant triples grounded in source text. However, as we will detail, Text2KGBench exhibits limitations concerning data quality and ontological rigor, which directly motivates the development of our proposed benchmark.

Methodologies for relation extraction have evolved significantly. Early approaches progressed from rule-based systems to feature-engineered machine learning and subsequently to deep learning architectures. Seminal neural models introduced sequence labeling and multi-task learning frameworks [7]. More advanced architectures like Seq2RDF [8] later framed the task as a sequence-to-sequence problem to translate natural language directly into RDF triples. The advent of transformer-based encoders led to powerful models for joint entity and relation extraction [9]. A critical shortcoming of many of these

models, however, is their frequent lack of explicit integration with ontological constraints, limiting their utility for constructing semantically coherent KGs.

To address this gap, the paradigm of schema-aware extraction has emerged, where generated triples must conform to a predefined ontology. Recent studies have explored leveraging external schema constraints during training, for example through few-shot perspective transfer [10] or knowledge-driven synthetic data generation for zero-shot extraction [11]. Others have investigated the use of structured prompts or ontology-guided decoding to improve the alignment of LLM outputs with a target schema. For instance, Ding et al. [12] proposed model collaboration strategies to mitigate hallucinations and enhance recall.

Large Language Models (LLMs) such as GPT-4 and Claude have demonstrated impressive in-context learning capabilities for information extraction. Nonetheless, their application to Text2KG is hampered by a propensity for factual hallucination and inconsistent adherence to structured output formats [13, 5]. While efforts to evaluate and mitigate these issues are ongoing, existing benchmarks often lack the ontological precision required for a fair and rigorous assessment. The benchmark introduced in this paper is specifically designed to fill this void.

3. Revision of Text2KGBench

This section details the revision and re-annotation of the Text2KGBench benchmark, undertaken to address critical limitations in its original version and enhance its utility for evaluating modern text-to-graph models. Our efforts focused on two key areas: a comprehensive revision of the underliying ontologies and a complete re-annotation of the corpus based on a new, rigorous set of guidelines. Both activities were conducted by a team of 4 experts with specializations in knowledge representation and natural language processing. The process involved an initial independent pass by each annotator, followed by a reconciliation phase to resolve disagreements.

3.1. Ontologies Refinement

The original Text2KGBench ontologies, while extensive, suffered from structural and semantic issues that limited its precision. It was organized into 19 ontologies, one for each domain, but lacked hierarchical depth and formal consistency. We conducted a thorough revision to address these limitations, focusing on improving its coherence, structural integrity, and semantic expressiveness.

Semantic Coherence and Granularity A primary objective was to ensure each domain ontology was self-contained and conceptually coherent. We systematically identified and pruned concepts and relations not directly relevant to their specified domains. For example, within the `Film` ontology, entities such as `Club` and `Station`, and relations like `spokenIn`, were removed as they are better situated in other contexts. This curation ensures that each domain ontology accurately models its core concepts, improving the benchmark's overall focus.

To reduce ambiguity and improve clarity, we harmonized property names. For instance, the property `campus` was renamed to `address` to more accurately reflect its semantic role, and `staff` was specified as `academicStaffSize` for explicitness. Similarly, the generic `location` property was refined into more specific relations such as `city` or `country`, depending of the context, thereby increasing the precision of the knowledge graph.

Structural and Formal Enhancements A significant structural enhancement was the introduction of a formal class hierarchy using `rdfs:subClassOf` relationships. In the original flat structure, `University` was an isolated class. It is now explicitly defined as a subclass of `AcademicInstitution`, which itself is a subclass of `Organization`. This hierarchical structure is not merely a formal improvement; it enables more nuanced evaluation metrics. For instance, we can now measure hierarchical

precision, rewarding a model for predicting a correct superclass (e.g. `AcademicInstitution` even if the specific subclass `University`) is missed.

Further, properties were rigorously typed as either `ObjectProperty` (linking two entities) and `DatatypeProperty` (linking an entity to a literal value), with explicit domains and ranges defined for each. Datatype ranges were specified using standard XML Schema types (e.g. `xsd:string`, `xsd:date`, or `xsd:integer`), enforcing data consistency and aiding downstream processing. To improve usability, we added `rdfs:comment` annotations for all properties and classes and simplified the URIs by removing the intermediate `relations` and `/concepts` path segments.

Finally, to support reproducibility and tracking, the new ontology includes metadata for contributors and is explicitly versioned as version 2.0 using `owl:versionIRI`. A comprehensive comparison of these changes is presented in Table 4, in the Appendix.

In the appendix, Table 2 presents an overview of the main statistics for each ontology in Text2KGBench-LettrIA and Text2KGBench. The Text2KGBench-LettrIA dataset is significantly lighter, with approximately 21.80% fewer classes and approximately 37.81% fewer properties. Additionally, datatype properties are exclusively present in Text2KGBench-LettrIA.

3.2. Re-annotations Guidelines

A robust benchmark requires annotation guidelines that are consistent, unambiguous, and computationally tractable. We established a comprehensive rulebook for the re-annotation process to ensure high-quality, reproducible data.

Normalization of Literals To ensure uniformity, we normalized literal values. Dates are standardized to the ISO 8601 format (`yyyy-mm-dd`). Ambiguous formats like `xx/xx/xxxx` are interpreted as `mm/dd/yyyy`, a common default in digital systems; if the first value exceeds 12, it is interpreted as `dd/mm/yyyy`. Partial dates (e.g. only a year, or only month plus year) associated to the `xsd:gYear` or `xsd:gYearMonth` datatypes. Durations are also standardized to the XSD notation (e.g. `20 minutes` is turned into `PT20M`).

Entity and Relation Extraction

- **Location Handling:** Our guidelines for locations proritize capturing geographical containment. When a text lists a hierarchy of locations (e.g. “Caen, Normandy, France”), we extract each as distinct entity. We then generate `isPartOf` relations to model their relationship of inclusion (e.g. `Caen` isPartOf `Normandy`, `Normandy` isPartOf `France`, and `Caen` isPartOf `France`). Even though, we take the full string `"Caen, Normandy, France"` to define a location. For example, `Antoine` livesAt `"Caen, Normandy, France"`. Finally, definite articles are omitted from place names (e.g., "the Philippines" becomes `Philippines`).
- **Strict Adherence to Textual Evidence:** Annotations are strictly confined to information explicitly present in the source text, avoiding reliance on external work knowledge. For example, in “Lettria was founded in Paris, France,” `Paris` is typed as `Place`. However, in “Lettria was founded in the city of Paris, France,”, the explicit mention allows for the more specific type `City`. This principle ensures that the benchmark evaluates a model’s ability to extract information from the provided context alone. This rule ensures that the text-2-graph task can be solved relying on the sole information in the benchmark.

Entity Scoping

- **Organization names:** Corporate suffixes (“Inc.”, “Co.”) are preserved as part of the entity name to maintain fidelity to the source text (e.g. `Caterpillar Inc.`).

- **Pronoun Resolution:** We resolve pronouns to their antecedent entity within the extracted triple. For ambiguous pronouns like "which," we employ a heuristic of selecting the immediately preceding noun phrase as the antecedent. For example, in "...beef kway teow which comes from the region of Indonesia," the pronoun "which" is resolved to `beef kway teow`.
- **Multiple Entities:** When a single statement applies to multiple entities, we create a separate triple for each. "Huseyin Butuner and Hilmi Guner designed..." yields two distinct `designer` relations, one for each person.

3.3. The Resulting Benchmark: Curation and Structural Enhancement

The culmination of the re-annotation process, guided by the revised ontology and the new annotation principles, is a benchmark of significantly higher quality and consistency. The resulting dataset comprises a total of 4860 sentences, which correspond to 14882 extracted triples.

In addition to the primary re-annotation, the benchmark underwent a comprehensive data curation and enhancement phase to address artifacts present in the original version and to enrich its structure for more rigorous model evaluation. These post-processing enhancements are detailed as follows:

- **Data Sanitization and Canonicalization:** A systematic normalization process was applied to entity and literal values to ensure uniformity and eliminate parsing inconsistencies. This included several key transformations:
  - **Entity Name Normalization:** Underscores used as word separators in entity names were replaced with spaces to form canonical, human-readable identifiers (e.g., `"AWH_Engineering_College"` was corrected to `"AWH Engineering College"`).
  - **Literal Value Cleaning:** Superfluous quotation marks that erroneously encapsulated object values in the original data were removed (e.g., `{"obj": "\"Kuttikkattoor\""}` was corrected to `{"obj": "Kuttikkattoor"}`).
  - **Numeric Data Typing:** String representations of numbers were parsed into their correct numeric types (e.g., `"2000"` became `2000`). Numerical Values are stripped of punctuation; for example, "18,527" is annotated as `18527`.
  - **Textual Harmonization:** Spelling inconsistencies and diacritical variations in names were corrected to ensure a true reproduction of what is in the text (e.g., `"Hüseyin Bütüner"` in the text is kept as it is and not turned into `"Huseyin Butuner"`).
- **Explicit Ontological Typing:** To improve the formal alignment between data instances and the ontology, each triple was enriched with new keys. The `subType` and `objType` fields now explicitly declare the ontological class of the subject and the datatype of the object, respectively. This structural addition is critical for enabling type-aware evaluation metrics and enforcing semantic consistency.
- **Corpus and Linguistic Refinement:** The source text corpus itself was subject to a final review. Minor grammatical and punctuation errors were corrected to improve linguistic quality.

The cumulative effect of these enhancements is illustrated in Figure 1 in the Appendix, which presents a side-by-side comparison of a data entry before and after the revision process. Table 3 in the Appendix presents a comparison between the original and new datasets. Text2KGBench-LettrIA maintains the same number of sentences as Text2KGBench, while the number of triples varies, showing both additions and reductions respect to Text2KGBench.

4. Experimental Evaluation with LLMs

Our study evaluates the performance of contemporary Large Language Models (LLMs) on the Text-to-Knowledge-Graph (Text2KG) task, which involves extracting knowledge graph triples from unstructured

text. The evaluation is conducted using the Text2KGBench-LettrIA benchmark. We assess two distinct categories of models under different conditions.

First, we assessed a comprehensive suite of proprietary models in a zero-shot setting, where models perform the task without any specific fine-tuning. The evaluated models, grouped by provider, included several from Anthropic, such as the Claude 3 family (Haiku, Sonnet, Opus) [14], the Claude 3.5 series (Haiku, Sonnet V1, Sonnet V2), the Claude 3.7 Sonnet, and the Claude 4 series (Sonnet, Opus). From Google, we evaluated the Gemini 2.0 family (Flash-Lite, Flash, Pro) and the Gemini 2.5 family (Flash-Lite, Flash, Pro) [15]. Our assessment also covered OpenAI's GPT-4.1 series (Full, Mini, Nano) [16] and GPT-4o series (Full, Mini) [17]. Finally, from Mistral AI, we included the Mistral Medium 2505 model[1].

In parallel, we fine-tuned and subsequently evaluated a selection of prominent open-source models to gauge their performance after task-specific adaptation. This set comprised Gemma 3 (4B-IT, 12B-IT, 27B-IT) [18][2], Mistral Small 3.2 (24B-Instruct)[3], Phi-4 (14B) [19], and Qwen 3 in several parameter sizes (0.6B, 1.7B, 4B, 8B, 14B, 32B) [20].

4.1. Fine-Tuning Methodology

We employed a Supervised Fine-Tuning (SFT) methodology to adapt the selected Large Language Models (LLMs) for the relation extraction task, utilizing the Unsloth framework for efficient training. The fine-tuning process involved providing each model with an input prompt containing two components: (1) a natural language sentence and (2) a compact representation of the relevant ontology. To mitigate the verbosity of the standard Turtle syntax and ensure the input fits within the models' context windows, we adopted a format inspired by Manchester syntax for representing the ontology schema. The target output for the SFT process was a JSON object containing the knowledge graph triples extracted from the sentence, mirroring the ground-truth annotations in our dataset.

To assess model performance under different data conditions, we designed and evaluated three distinct fine-tuning configurations:

Classic Models were fine-tuned on the complete, original training dataset. This configuration serves as our performance baseline.

Extended This configuration incorporates data augmentation. The original training set was supplemented with synthetic data generated by the Gemini 2.5 Pro model. The objective of this augmentation was to enrich the training data for each ontology, ensuring a number of 500 training examples per ontology, bringing the training set to 9500 examples in total.

Generalization This configuration evaluates the models' zero-shot generalization capabilities to unseen ontologies using a leave-one-out strategy. Models were trained on a dataset comprising 18 of the 19 ontologies. The held-out ontology (the *City* ontology) was then used exclusively for testing. The final test set for this scenario was composed of all examples (both original training and test splits) associated with the unseen *City* ontology.

All the fine-tuning runs[4] for each model have been conducted on a Nvidia H100 GPU.

4.2. Evaluation

To provide a multifaceted evaluation of our relation extraction approach, we introduce a suite of metrics that extends beyond the traditional F1-score. Our methodology first categorizes the components of the knowledge graph into four distinct types:

[1]Model details available at: https://mistral.ai/news/mistral-medium-3
[2]Model card: https://huggingface.co/google/gemma-3-12b-it
[3]Model card: https://huggingface.co/mistralai/Mistral-Small-3.2-24B-Instruct-2506
[4]Fine-Tuning Hyper-Parameters: **Lora Rank:** 128 **Lora Alpha:** 512 **Batch Size:** 1 **Gradient Accumulation:** 8 **Epochs:** 3 **Warmup Steps:** 5 **Learning Rate:** $2e^{-5}$ **Optimisation:** AdamW-8bit **Weight Decay:** 0.01 **Learning Rate Scheduler:** Linear

Entities (E) The classes that serve as the domain and range for object properties, or as the domain for datatype properties.

Attributes (A) The literal values that constitute the range of datatype properties.

Properties (P) The datatype properties that link entities to attributes.

Relations (R) The object properties that link entities to other entities.

Based on this categorization, we assess model performance across six key dimensions:

- **F1-Score:** The macro-averaged F1-score for the correct identification and classification of each extracted entity, attribute, property, and relation.
- **Ontological Fidelity:** A measure to quantify hallucinations, defined as the generation of types, properties, or relations that are not present in the reference ontology.
- **Domain/Range Adherence:** Assesses whether the model's outputs respect the domain and range constraints defined in the ontology for all properties (datatype properties) and relations (object properties). This metric accounts for subclass hierarchies; for instance, if an ontology specifies a domain of `Place` and the model predicts `City`, the prediction is considered valid provided `City` is a subclass of `Place`.
- **Structural Validity:** Measures whether the generated output conforms to the required JSON schema, ensuring it is well-formed and parseable.
- **Latency:** The average inference time in seconds required to generate a response, calculated across all examples in the test set.
- **Cost:** The average monetary cost per query. For proprietary models, this is the API cost. For open-source models, we estimate the cost based on the hourly price of the required hardware from a cloud provider (e.g., a OVH Cloud instance at 2.80 €/hour).

4.3. Performance and Insights

Performance was evaluated using three distinct fine-tuning configurations. The first two configurations were tested on our "full benchmark," a revised and comprehensive version of the new benchmark. The third configuration was subsequently tested on a single ontology in a "generalization" scenario. All experiments involving closed models utilized the most recent, optimized prompt from our internal text-to-graph production framework.

4.3.1. Full Benchmark

Performance The most striking finding is the significant performance gap between the two groups. Fine-tuned models operate in a different league, with most achieving an Entity F1 score exceeding 0.80. This underscores the immense power of specialization. The top performer, `Mistral-Small-3.2 (ext.)`, achieved an outstanding Entity F1 of 0.8837, with other models from the `Qwen3` and `gemma-3` families clustering in the impressive 0.85–0.87 range. In contrast, the proprietary models, which test general-purpose reasoning without task-specific training, top out with an Entity F1 below 0.70. Within this group, a clear performance hierarchy emerges. `gemini-2.5-pro` stands out as the best all-rounder, with consistently high F1 scores across all categories (E=0.6595, A=0.8762, P=0.8627, R=0.7076). Other models act as high-performing specialists: `claude-sonnet-4` excels at understanding complex connections with the highest Relations score (R=0.7126), while `gpt-4.1-mini-2025-04-14` is best at identifying discrete items (E=0.6866). Meanwhile, models like `gemini-2.0-flash` and `claude-3-haiku` struggle with the task's complexity, proving unsuitable for this type of detailed extraction.

Run ID	F1				Hallucinations			Respect		Valid Outputs (%)	Latency (s)	Cost ($)
	Entities	Attributes	Properties	Relations	Types	Relations	Properties	Relations	Properties			
Closed - 1-shot												
claude-3.5-haiku	0.5732	0.693	0.6836	0.5649	0.0041	0.0003	0.0364	0.9637	0.9105	93.44%	14.8537	0.0038
claude-3.5-sonnet-v1	0.5804	0.8471	0.8336	0.6182	0.0052	0.0	0.0082	0.978	0.9826	97.12%	14.3797	0.0139
claude-3.5-sonnet-v2	0.6059	0.8789	0.8675	0.6697	0.0047	0.0	0.0065	0.9886	0.9935	99.30%	15.6948	0.0136
claude-3-haiku	0.5206	0.3373	0.3523	0.4389	0.0198	0.0254	0.2038	0.8429	0.6808	90.66%	8.6447	0.0014
claude-3.7-sonnet	0.6082	0.8903	0.88	0.6625	0.004	0.0	0.0053	0.9859	0.9943	99.70%	13.0294	0.0128
claude-opus-4	0.6289	0.8702	0.8532	0.6944	0.0051	0.0	0.0034	0.9977	0.9962	99.20%	37.4454	0.1682
claude-sonnet-4	0.6487	0.8657	0.8498	0.7126	0.0011	0.0	0.0065	0.9908	0.9848	99.35%	10.4307	0.0111
gemini-2.0-flash-lite	0.5276	0.6885	0.679	0.5456	0.0028	0.0014	0.0109	0.9466	0.9714	83.95%	2.3805	0.0002
gemini-2.0-flash	0.3539	0.4311	0.4195	0.3864	0.0017	0.0	0.0137	0.9626	0.9799	57.36%	1.9308	0.0004
gemini-2.5-flash-lite	0.6014	0.2542	0.4930	0.2335	0.0088	0.0221	0.3439	0.8553	0.4993	97.51%	1.9086	0.0010
gemini-2.5-flash	0.5501	0.7463	0.7339	0.6062	0.0055	0.0	0.0113	0.9736	0.9848	86.43%	2.0842	0.0013
gemini-2.5-pro	0.6595	0.8762	0.8627	0.7076	0.0014	0.0	0.0022	0.9925	0.9966	99.80%	3.9886	0.005
gpt-4.1-2025-04-14	0.6472	0.8742	0.863	0.6565	0.0014	0.0004	0.0146	0.9798	0.9843	97.27%	3.9289	0.0058
gpt-4.1-mini-2025-04-14	0.6866	0.8584	0.8471	0.6114	0.0042	0.0023	0.0324	0.9442	0.9512	98.86%	5.9905	0.0012
claude-3-opus	0.6159	0.7589	0.7492	0.6621	0.0072	0.0032	0.0753	0.9532	0.8627	97.12%	27.2936	0.068
gpt-4.1-nano-2025-04-14	0.4831	0.5148	0.4875	0.1911	0.0303	0.1238	0.4698	0.5689	0.4081	82.75%	2.5785	0.0003
gpt-4o-2024-11-20	0.6032	0.7971	0.7879	0.6021	0.0055	0.0011	0.0292	0.9527	0.9472	94.14%	3.5076	0.0086
gpt-4o-mini-2024-07-18	0.5951	0.4703	0.5082	0.3379	0.0083	0.0121	0.2745	0.8708	0.666	91.95%	9.1931	0.0005
mistral-medium-2505	0.6095	0.5524	0.564	0.6003	0.0061	0.0004	0.1141	0.9622	0.7137	99.11%	6.4915	0.0014
claude-3-sonnet	0.5869	0.7303	0.7246	0.5583	0.0088	0.0026	0.17	0.9068	0.7604	96.82%	12.6952	0.0113
Open Source (Finetuned)												
gemma-3-4b-it	0.8294	0.9080	0.8799	0.7248	0.0065	0.0195	0.0555	0.9178	0.8667	99.35%	0.0094	12.0294
gemma-3-4b-it *(ext.)*	0.8329	0.9344	0.9089	0.7647	0.0069	0.0124	0.0290	0.9438	0.9143	99.70%	0.0092	11.7466
gemma-3-12b-it	0.8606	0.9211	0.9001	0.7942	0.0066	0.0118	0.0335	0.9570	0.8904	99.95%	0.0129	16.5804
gemma-3-12b-it *(ext.)*	0.8592	0.9437	0.9302	0.8149	0.0069	0.0091	0.0155	0.9620	0.9320	99.95%	0.0130	16.6987
gemma-3-27b-it	0.8680	0.9301	0.9038	0.8027	0.0064	0.0119	0.0275	0.9533	0.9097	99.95%	0.0165	21.1579
gemma-3-27b-it *(ext.)*	0.8588	0.9439	0.9225	0.8121	0.0069	0.0099	0.0162	0.9635	0.9304	100.00%	0.0166	21.2861
Mistral-Small-3.2	0.8837	0.9497	0.9351	0.8294	0.0070	0.0106	0.0163	0.9542	0.9258	99.90%	0.0096	12.2700
Mistral-Small-3.2 *(ext.)*	0.8764	0.9474	0.9287	0.8307	0.0072	0.0089	0.0245	0.9641	0.9185	100.00%	0.0096	12.2708
phi-4	0.7420	0.8451	0.8112	0.6359	0.0161	0.0329	0.0549	0.8946	0.8548	93.69%	0.0080	10.2748
phi-4 *(ext.)*	0.7656	0.8802	0.8432	0.6810	0.0091	0.0178	0.0364	0.9300	0.8838	96.37%	0.0079	10.0703
Qwen3-0.6B	0.8272	0.8980	0.8653	0.7059	0.0118	0.0222	0.0425	0.9150	0.9100	99.65%	0.0064	8.2034
Qwen3-0.6B *(ext.)*	0.8238	0.9282	0.8947	0.7365	0.0076	0.0157	0.0268	0.9357	0.9090	100.00%	0.0063	8.0859
Qwen3-1.7B	0.8302	0.8969	0.8687	0.7193	0.0072	0.0212	0.0806	0.9180	0.8448	99.55%	0.0063	8.1014
Qwen3-1.7B *(ext.)*	0.8303	0.9264	0.8985	0.7559	0.0073	0.0110	0.0267	0.9478	0.9149	99.65%	0.0064	8.1866
Qwen3-4B	0.8482	0.9095	0.8881	0.7778	0.0051	0.0153	0.0443	0.9436	0.8800	99.40%	0.0083	10.6227
Qwen3-4B *(ext.)*	0.8447	0.9378	0.9194	0.7987	0.0067	0.0102	0.0228	0.9610	0.9230	99.75%	0.0081	10.3798
Qwen3-8B	0.8512	0.9137	0.8875	0.7758	0.0069	0.0143	0.0411	0.9512	0.8960	99.80%	0.0084	10.8296
Qwen3-8B *(ext.)*	0.8412	0.9351	0.9190	0.7949	0.0067	0.0095	0.0254	0.9583	0.9228	99.90%	0.0082	10.5739
Qwen3-14B	0.8688	0.9278	0.9014	0.8067	0.0074	0.0109	0.0339	0.9556	0.9084	99.80%	0.0094	12.0514
Qwen3-14B *(ext.)*	0.8610	0.9461	0.9227	0.8155	0.0068	0.0110	0.0198	0.9608	0.9260	100.00%	0.0092	11.7774
Qwen3-32B	0.8677	0.9288	0.9024	0.8016	0.0077	0.0115	0.0311	0.9498	0.9057	99.90%	0.0146	18.7284
Qwen3-32B *(ext.)*	0.8521	0.9358	0.9177	0.8138	0.0074	0.0112	0.0222	0.9593	0.9282	99.90%	0.0146	18.6875

Table 1
This table compares the performance of various models on the full test set. The first section evaluates closed-source models using a 1-shot prompting strategy. The second section presents results for open-source models after two finetuning variants: "Classic" (unmarked) and "Extended" (marked with *(ext.)*).

Safety and Reliability Beyond raw performance, fine-tuning proves to be a profound method for ensuring safety and reliability. Nearly all fine-tuned models achieved over 99% validly formatted outputs—with several reaching a perfect 100%—demonstrating that specialization is an exceptionally effective way to guarantee adherence to a specific output format. Furthermore, we observed an "extended effect" in fine-tuned variants: these models often trade a slight dip in Entity F1 for improved scores in other categories and, crucially, lower hallucination rates and better adherence to the ontology. This suggests the `-extended` process prioritizes overall robustness and safety. Among the proprietary models, the top performers also demonstrate strong reliability. `gemini-2.5-pro` and `claude-opus-4` lead in producing validly formatted outputs (99.80% and 99.20%, respectively) and show superior adherence to the ontology. However, safety is not a given in this category. While models like `claude-3.7-sonnet` and `gemini-2.5-pro` boast extremely low hallucination scores, `gpt-4.1-nano` exhibits a catastrophic failure with a hallucination precision of just 0.4698, making it a high risk for generating false information.

Efficiency The efficiency profiles of the two groups present starkly different trade-offs. For the API-based proprietary models, the balance is between performance, latency, and cost-per-call. The `gemini-flash` models are the fastest, with response times around 2 seconds, while the powerful `claude-opus-4` is the slowest at a substantial 37.4 seconds. A similar trade-off exists in cost: `gemini-2.0-flash-lite` (0.0002¢) is one of the cheapest, whereas `claude-opus-4` (0.1682¢) is by

far the most expensive, illustrating the classic balance between capability and operational cost. This dynamic shifts entirely with the fine-tuned models, which run on dedicated local hardware. Latencies are astonishingly low, with all models completing the task in under 0.02 seconds—orders of magnitude faster than API calls. The trade-off here is the high, amortized cost of the fine-tuning process and hosting the model on powerful GPU infrastructure. This cost scales directly with model size, making larger models like `gemma-3-27b` and `Qwen3-32B` the most expensive to operate.

4.3.2. Generalization Benchmark

The Generalization Benchmark results are displayed in Table C in the Appendix.

Performance The most striking finding is the significant performance gap between the two groups. Fine-tuned models operate in a different league, with most achieving an Entity F1 score exceeding 0.80. This underscores the immense power of specialization. The top performer, `Mistral-Small-3.2 (ext.)`, achieved an outstanding Entity F1 of 0.8837, with other models from the `Qwen3` and `gemma-3` families clustering in the impressive 0.85–0.87 range. In contrast, the proprietary models, which test general-purpose reasoning without task-specific training, top out with an Entity F1 below 0.70. Within this group, a clear performance hierarchy emerges. `gemini-2.5-pro` stands out as the best all-rounder, with consistently high F1 scores across all categories (E=0.6595, A=0.8762, P=0.8627, R=0.7076). Other models act as high-performing specialists: `claude-sonnet-4` excels at understanding complex connections with the highest Relations score (R=0.7126), while `gpt-4.1-mini-2025-04-14` is best at identifying discrete items (E=0.6866). Meanwhile, models like `gemini-2.0-flash` and `claude-3-haiku` struggle with the task's complexity, proving unsuitable for this type of detailed extraction.

Safety and Reliability Beyond raw performance, fine-tuning proves to be a profound method for ensuring safety and reliability. Nearly all fine-tuned models achieved over 99% validly formatted outputs—with several reaching a perfect 100%—demonstrating that specialization is an exceptionally effective way to guarantee adherence to a specific output format. Furthermore, we observed an "extended effect" in fine-tuned variants: these models often trade a slight dip in Entity F1 for improved scores in other categories and, crucially, lower hallucination rates and better adherence to the ontology. This suggests the `-extended` process prioritizes overall robustness and safety. Among the proprietary models, the top performers also demonstrate strong reliability. `gemini-2.5-pro` and `claude-opus-4` lead in producing validly formatted outputs (99.80% and 99.20%, respectively) and show superior adherence to the ontology. However, safety is not a given in this category. While models like `claude-3.7-sonnet` and `gemini-2.5-pro` boast extremely low hallucination scores, `gpt-4.1-nano` exhibits a catastrophic failure with a hallucination precision of just 0.4698, making it a high risk for generating false information.

Efficiency The efficiency profiles of the two groups present starkly different trade-offs. For the API-based proprietary models, the balance is between performance, latency, and cost-per-call. The `gemini-flash` models are the fastest, with response times around 2 seconds, while the powerful `claude-opus-4` is the slowest at a substantial 37.4 seconds. A similar trade-off exists in cost: `gemini-2.0-flash-lite` (0.0002¢) is one of the cheapest, whereas `claude-opus-4` (0.1682¢) is by far the most expensive, illustrating the classic balance between capability and operational cost. This dynamic shifts entirely with the fine-tuned models, which run on dedicated local hardware. Latencies are astonishingly low, with all models completing the task in under 0.02 seconds—orders of magnitude faster than API calls. The trade-off here is the high, amortized cost of the fine-tuning process and hosting the model on powerful GPU infrastructure. This cost scales directly with model size, making larger models like `gemma-3-27b` and `Qwen3-32B` the most expensive to operate.

4.3.3. Lessons Learned

Our experimental results, detailed in Table 4.3.1 and in Table C, evaluate two distinct categories of models: proprietary, closed-source models used in a 1-shot context, and open-source models that have been specifically fine-tuned for the task. The analysis reveals a fundamental trade-off between the out-of-the-box flexibility of generalist models and the specialized power of fine-tuned experts.

Ultimately, the choice of model depends entirely on the use case. The results clearly show that for a well-defined, repetitive task, a specialized, fine-tuned model will almost always outperform a general-purpose one. For general-purpose tasks, rapid development, or applications where latency is not a primary concern, a closed-source API model like `gemini-2.5-pro` is ideal, offering the best blend of high performance, excellent reliability, and reasonable efficiency; for tasks heavily dependent on complex connections, `claude-sonnet-4` is an equally compelling alternative. Conversely, for a specific, high-value, high-volume task where performance and speed are paramount, investing in fine-tuning an open-source model is the clear path forward. Self-hosted, fine-tuned models deliver astonishingly low latencies of under 20 milliseconds—orders of magnitude faster—but require significant amortized costs for fine-tuning and GPU infrastructure, with expenses scaling directly with model size. In this scenario, `Mistral-Small-3.2 (ext.)` stands out as the overall performance leader, while a model like `Qwen3-14B (ext.)` offers a superb balance of top-tier F1 scores, enhanced safety, and a more moderate infrastructure cost.

5. Conclusion and Future Work

In this paper, we introduced Text2KGBench-LettrIA, a rigorously revised benchmark for evaluating ontology-guided Text-to-Knowledge-Graph systems. By systematically overhauling the DBpedia-WebNLG portion of Text2KGBench, we addressed critical limitations in its ontological design, annotation quality, and structural consistency. The resulting benchmark features 19 refined ontologies with enforced hierarchical relationships and strict typing, alongside over 14,000 high-fidelity triples re-annotated under stringent guidelines to ensure textual grounding and reproducibility. This work provides the community with a resource that enables a more precise and nuanced evaluation of model capabilities in structured knowledge extraction.

Our experiments yield a significant finding: smaller, open-source language models, when properly fine-tuned on our high-quality benchmark, can outperform larger, proprietary models in terms of F1-score for triple extraction. This result underscores the pivotal role that task-specific data quality and model adaptation play in achieving state-of-the-art performance. Nevertheless, our analysis also highlights a persistent challenge: even high-performing models exhibit a tendency to hallucinate or deviate from ontological constraints, indicating that high accuracy on individual components does not guarantee perfect schema adherence.

Building on this work, we identify several key directions for future research.

- **Post-Hoc Alignment:** The prevalence of schema violations and hallucinations, even after supervised fine-tuning (SFT), suggests the need for a subsequent alignment phase. Investigating reinforcement learning-based techniques such as Proximal Policy Optimization (PPO) or Direct Preference Optimization (DPO) could further refine model outputs to improve ontological fidelity.
- **Explainability and Reasoning:** Future work could focus on developing a reasoning layer atop the extraction models. Such a component would not only extract triples but also generate explanations for its predictions, thereby increasing the transparency and trustworthiness of the KG construction process.
- **Context Window Extension:** A current limitation of many open-source models is their relatively small context window compared to proprietary counterparts. Future experiments should explore methods to extend the effective context size of fine-tuned models, enabling them to process larger and more complex documents and ontologies.

- **Ontology:** The ontologies have only binary relations (they cannot describe complex entities such as event), an improvement would be to create n-ary relations with reification, in order to have more realistic ontologies, and see if the LLMs, even fine-tuned, can properly handle such complex ontologies.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT and LeChat by MistralAI in order to: Grammar and spelling check; Paraphrase and reword. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

Acknowledgments

This work was supported by the French Public Investment Bank (Bpifrance) i-Demo program within the LettRAGraph project (Grant ID DOS0256163/00).

References

[1] N. Mihindukulasooriya, S. Tiwari, C. F. Enguix, K. Lata, Text2KGBench: A Benchmark for Ontology-Driven Knowledge Graph Generation from Text, in: The Semantic Web – ISWC 2023: 22nd International Semantic Web Conference, Proceedings, Part II, Springer-Verlag, Berlin, Heidelberg, 2023, p. 247–265. doi:10.1007/978-3-031-47243-5_14.

[2] O. Agarwal, H. Ge, S. Shakeri, R. Al-Rfou, Knowledge Graph Based Synthetic Corpus Generation for Knowledge-Enhanced Language Model Pre-training, in: K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tur, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, Y. Zhou (Eds.), Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Online, 2021, pp. 3554–3565. doi:10.18653/v1/2021.naacl-main.278.

[3] C. Gardent, A. Shimorina, S. Narayan, L. Perez-Beltrachini, Creating Training Corpora for NLG Micro-Planners, in: R. Barzilay, M.-Y. Kan (Eds.), Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 179–188. doi:10.18653/v1/P17-1017.

[4] A. G. Regino, A. Rossanez, R. da Silva Torres, J. C. dos Reis, A Systematic Literature Review on RDF Triple Generation from Natural Language Text, Semantic Web (2025).

[5] S. Tiwari, N. Mihindukulasooriya, F. Osborne, D. Kontokostas, J. D'Souza, M. Kejriwal, et al., Preface for the Third International Workshop on Knowledge Graph Generation from Text, in: 3rd International workshop one knowledge graph generation from text. Data Quality meets Machine Learning and Knowledge Graphs 2024, volume 3747, CEUR-WS, 2024, pp. 1–4.

[6] J. Kalo, T. Nguyen, S. Razniewski, B. Zhang, Preface: Lm-kbc challenge 2024, in: KBC-LM-LM-KBC 2024 Joint proceedings of the KBC-LM workshop and the LM-KBC challenge 2024, CEUR-WS.org, 2024, pp. 1–5.

[7] D. Zeng, K. Liu, S. Lai, G. Zhou, J. Zhao, Relation Classification via Convolutional Deep Neural Network, in: J. Tsujii, J. Hajic (Eds.), Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, Dublin City University and Association for Computational Linguistics, Dublin, Ireland, 2014, pp. 2335–2344. URL: https://aclanthology.org/C14-1220/.

[8] Y. Liu, T. Zhang, Z. Liang, H. Ji, D. L. McGuinness, Seq2rdf: An end-to-end application for deriving triples from natural language text, in: Proceedings of the ISWC 2018 Posters & Demonstrations, Industry and Blue Sky Ideas Tracks co-located with 17th International Semantic Web Conference (ISWC 2018), 2018.

[9] J. Wang, W. Lu, Two are Better than One: Joint Entity and Relation Extraction with Table-Sequence Encoders, in: B. Webber, T. Cohn, Y. He, Y. Liu (Eds.), Proceedings of the 2020 Conference on

Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 1706–1721. URL: https://aclanthology.org/2020.emnlp-main.133/. doi:10.18653/v1/2020.emnlp-main.133.

[10] J. Fei, W. Zeng, X. Zhao, X. Li, W. Xiao, Few-Shot Relational Triple Extraction with Perspective Transfer Network, in: Proceedings of the 31st ACM International Conference on Information & Knowledge Management, CIKM '22, Association for Computing Machinery, New York, NY, USA, 2022, p. 488–498. doi:10.1145/3511808.3557323.

[11] L. He, H. Zhang, J. Liu, K. Sun, Q. Zhang, Zero-Shot Relation Triplet Extraction via Knowledge-Driven LLM Synthetic Data Generation, in: D.-S. Huang, Z. Si, C. Zhang (Eds.), Advanced Intelligent Computing Technology and Applications, Springer Nature, Singapore, 2024, pp. 329–340.

[12] Z. Ding, W. Huang, J. Liang, Y. Xiao, D. Yang, Improving Recall of Large Language Models: A Model Collaboration Approach for Relational Triple Extraction, in: N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, N. Xue (Eds.), Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), ELRA and ICCL, Turin, Italy, 2024, pp. 8890–8901. URL: https://aclanthology.org/2024.lrec-main.778/.

[13] A. Ananya, S. Tiwari, N. Mihindukulasooriya, T. Soru, Z. Xu, D. Moussallem, Towards Harnessing Large Language Models as Autonomous Agents for Semantic Triple Extraction from Unstructured Text, in: TEXT2KG 2024: Third International Workshop on Knowledge Graph Generation from Text, Hersonissos, Greece, 2024.

[14] Anthropic, The Claude 3 Model Family: Opus, Sonnet, Haiku, 2024. URL: https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf.

[15] G. Comanici, et al., Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multimodality, Long Context, and Next Generation Agentic Capabilities, 2025. URL: https://arxiv.org/abs/2507.06261. arXiv:2507.06261.

[16] OpenAI, et al., GPT-4 Technical Report, 2024. URL: https://arxiv.org/abs/2303.08774. arXiv:2303.08774.

[17] OpenAI, et al., GPT-4o System Card, 2024. URL: https://arxiv.org/abs/2410.21276. arXiv:2410.21276.

[18] Gemma Team, et al., Gemma 3 Technical Report, 2025. URL: https://arxiv.org/abs/2503.19786. arXiv:2503.19786.

[19] M. Abdin, J. Aneja, H. Behl, S. Bubeck, R. Eldan, S. Gunasekar, M. Harrison, R. J. Hewett, M. Javaheripi, P. Kauffmann, J. R. Lee, Y. T. Lee, Y. Li, W. Liu, C. C. T. Mendes, A. Nguyen, E. Price, G. de Rosa, O. Saarikivi, A. Salim, S. Shah, X. Wang, R. Ward, Y. Wu, D. Yu, C. Zhang, Y. Zhang, Phi-4 Technical Report, 2024. URL: https://arxiv.org/abs/2412.08905. arXiv:2412.08905.

[20] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, C. Zheng, D. Liu, F. Zhou, F. Huang, F. Hu, H. Ge, H. Wei, H. Lin, J. Tang, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Zhou, J. Lin, K. Dang, K. Bao, K. Yang, L. Yu, L. Deng, M. Li, M. Xue, M. Li, P. Zhang, P. Wang, Q. Zhu, R. Men, R. Gao, S. Liu, S. Luo, T. Li, T. Tang, W. Yin, X. Ren, X. Wang, X. Zhang, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Zhang, Y. Wan, Y. Liu, Z. Wang, Z. Cui, Z. Zhang, Z. Zhou, Z. Qiu, Qwen3 Technical Report, 2025. URL: https://arxiv.org/abs/2505.09388. arXiv:2505.09388.

A. Dataset Statistics

Ontology Name	**Text2KGBench-LettrIA**			**Text2KGBench**	
	Classes	Object Prop.	Datatype Prop.	Classes	Object Prop.
airport	11	15	5	13	39
artist	12	16	7	19	39
astronaut	18	10	11	15	38
athlete	10	18	9	14	37
building	9	11	12	14	38
celestialbody	5	1	17	8	27
city	13	10	10	11	23
comicscharacter	8	8	4	10	18
company	9	13	6	10	28
film	5	10	5	18	44
food	12	13	2	12	24
meanoftransportation	12	20	28	20	68
monument	10	10	4	14	26
musicalwork	15	22	3	15	35
politician	17	25	9	19	40
scientist	12	15	5	15	47
sportsteam	9	12	3	14	24
university	11	16	11	15	46
writtenwork	10	17	13	10	44
TOTAL	208	262	164	266	685

Table 2
Comparison of ontology statistics for Text2KGBench-LettrIA and Text2KGBench.

Ontology	Text2KGBench-LettrIA						Text2KGBench			
	Sentences			Triples			Sentences		Triples	
airport	79 /	227 /	273	260 /	702 /	989	79 /	227	237 /	714
artist	84 /	302 /	198	256 /	896 /	638	84 /	302	252 /	896
astronaut	68 /	86 /	414	266 /	264 /	985	68 /	86	279 /	241
athlete	107 /	186 /	314	304 /	568 /	811	107 /	186	299 /	575
building	103 /	172 /	328	276 /	593 /	956	103 /	172	309 /	588
celestialbody	72 /	122 /	378	203 /	329 /	885	72 /	122	223 /	373
city	217 /	131 /	369	1289 /	479 /	1038	217 /	131	651 /	398
comics character	36 /	66 /	434	92 /	165 /	934	36 /	66	107 /	215
company	56 /	97 /	403	174 /	314 /	928	56 /	97	157 /	300
film	127 /	137 /	363	368 /	369 /	622	127 /	137	378 /	398
food	153 /	245 /	255	473 /	683 /	681	153 /	245	532 /	734
mean of transportation	92 /	222 /	278	271 /	646 /	745	92 /	222	276 /	647
monument	19 /	73 /	427	64 /	343 /	1365	19 /	73	55 /	293
musicalwork	209 /	81 /	419	842 /	285 /	912	209 /	81	604 /	221
politician	135 /	184 /	316	415 /	688 /	1089	135 /	184	424 /	550
scientist	149 /	110 /	390	387 /	259 /	559	149 /	110	411 /	300
sportsteam	110 /	125 /	375	375 /	369 /	1294	110 /	125	401 /	375
university	71 /	85 /	415	337 /	228 /	749	71 /	85	283 /	248
writtenwork	127 /	195 /	305	267 /	628 /	861	127 /	195	381 /	557
TOTAL	**2014** /	**2846** /	**6654**	**6919** /	**8808** /	**17101**	**2014** /	**2846**	**6259** /	**8623**

Table 3
Number of sentences and triple per dataset version: test / train / train ext for T2KB-LettrIA and test / train for T2KGBench

B. Ontology and Annotation Comparison


```
{
 "id": "ont_1_university_train_37",
 "sent": "The University of Burgundy employs 2900
     staff members with 1299 doctoral students",
 "triples": [{
    "sub": "University_of_Burgundy",
    "rel": "staff",
    "obj": "2900"
  },{
    "sub": "University_of_Burgundy",
    "rel": "numberOfDoctoralStudents",
    "obj": "1299"
  }]
}
```

```
{
 "id": "ont_1_university_train_37",
 "sent": "The University of Burgundy employs 2900
     staff members with 1299 doctoral students",
 "triples": [{
    "sub": "University of Burgundy",
    "subType": "University",
    "rel": "academicStaffSize",
    "obj": "2900",
    "objType": "integer"
  }, {
    "sub": "University of Burgundy",
    "subType": "University",
    "rel": "numberOfDoctoralStudents",
    "obj": "1299",
    "objType": "integer"
  }]
}
```


Figure 1: Comparison of the same dataset entry in Text2KGBench (left) and Text2KGBench-LettrIA (right).

Aspect	**Text2KGBench**	**Text2KGBench-LettrIA**
Domain Coherence	Included out-of-domain concepts	Strictly domain-specific concepts.
Property Semantics	Ambiguous or overly generic properties	Properties renamed and specified for clarity.
Class Structure	Flat, non-hierarchical	Hierarchical using `subClassOf`
Property Types	All properties treated as `ObjectProperty`	Strict distinction between `ObjectProperty` and `DatatypeProperty` with specified domains and ranges.
URI format	`.../<domain>/<type>#` where `<type>` is *relations* or *concepts*	Simplified to `.../<domain>#`.
Documentation	Absent	`rdfs:comment` for all classes and properties.
Metadata	Absent	Contributor list and `owl:versionIRI`

Example	`onto:University a owl:Class ;`
	`  rdfs:label "University" .`
		`onto:University a owl:Class ;`
		`  rdfs:subClassOf`
		`  onto:AcademicInstitution ;`
		`  rdfs:label "University" ;`
		`  rdfs:comment "A higher education..." .`
		
		`onto:AcademicInstitution a owl:Class ;`
		`  rdfs:subClassOf onto:Organization ;`
		`  rdfs:label "AcademicInstitution" ;`
		`  rdfs:comment "An institution for..." .`
		
		`onto:Organization a owl:Class ;`
		`  rdfs:label "Organization" ;`
		`  rdfs:comment "A formal structure..." .`

Table 4
Comparison of the original and revised ontologies, highlighting key structural and semantic enhancements.

C. Generationzation Results

Run ID	F1				Hallucinations			Respect		Valid Outputs (%)	Latency (s)	Cost ($)
	Entities	Attributes	Properties	Relations	Types	Relations	Properties	Relations	Properties			
Closed - 1-shot												
mistral-medium-2505	0.7661	0.7875	0.796	0.6444	0.0	0.0	0.0	0.9681	0.7836	99.58	3.2267	0.0033
claude-sonnet-4	0.7829	0.9509	0.9283	0.7179	0.0	0.0	0.0	0.9967	0.9878	99.44	7.5647	0.0297
claude-3-opus	0.7825	0.9405	0.9102	0.7199	0.0	0.0	0.0	0.9836	0.958	99.44	27.9306	0.1635
claude-3.5-sonnet-v2	0.7823	0.9581	0.9333	0.7089	0.0	0.0	0.0	0.9916	0.9818	100.00	14.7003	0.0319
claude-3-sonnet	0.7777	0.876	0.8384	0.6338	0.0	0.0	0.0	0.944	0.8365	99.30	9.9851	0.0296
claude-3.7-sonnet	0.7775	0.9471	0.9278	0.7146	0.0	0.0	0.0	0.9902	0.9854	99.72	11.0482	0.0311
gpt-4.1-mini-2025-04-14	0.7764	0.8906	0.8607	0.6766	0.0	0.0	0.0	0.9839	0.9764	99.86	3.2743	0.0031
gemini-2.5-pro	0.7748	0.958	0.9368	0.7242	0.0	0.0	0.0	0.9913	0.9881	99.86	3.4366	0.0117
gpt-4.1-2025-04-14	0.7731	0.9193	0.9013	0.6773	0.0	0.0	0.0	0.9866	0.9933	99.30	4.351	0.0154
gemini-2.5-flash-lite	0.771	0.2881	0.2503	0.6158	0.0	0.0	0.0	0.9197	0.4166	98.33	2.0679	0.0011
mistral-medium-2505	0.7661	0.7875	0.796	0.6444	0.0	0.0	0.0	0.9681	0.7836	99.58	3.2267	0.0033
claude-3.5-sonnet-v1	0.7539	0.9277	0.91	0.6842	0.0	0.0	0.0	0.987	0.9815	97.77	15.9461	0.0331
claude-3.5-haiku	0.7499	0.8811	0.8489	0.651	0.0	0.0	0.0	0.9729	0.9201	97.77	11.8934	0.0088
gpt-4o-2024-11-20	0.7489	0.88	0.8584	0.6539	0.0	0.0	0.0	0.9766	0.9754	95.96	6.3608	0.0203
gemini-2.0-flash-lite	0.6968	0.7877	0.7572	0.5783	0.0	0.0	0.0	0.9105	0.9536	89.54	2.4511	0.0006
claude-3-haiku	0.6898	0.4962	0.4943	0.5257	0.0	0.0	0.0	0.8894	0.758	88.56	9.7084	0.0029
gpt-4o-mini-2024-07-18	0.68	0.6632	0.6585	0.438	0.0	0.0	0.0	0.9448	0.7539	88.42	7.3224	0.0012
gpt-4.1-nano-2025-04-14	0.6756	0.6431	0.6315	0.2831	0.0	0.0	0.0	0.7622	0.5021	91.07	3.7023	0.0008
gemini-2.5-flash	0.5572	0.7033	0.676	0.5447	0.0	0.0	0.0	0.9921	0.9824	74.34	1.5099	0.0029
gemini-2.0-flash	0.4522	0.6017	0.5789	0.449	0.0	0.0	0.0	0.9843	0.9837	64.99	1.9098	0.0012
Open Source (Finetuned)												
Mistral-Small-3.2	0.8014	0.9368	0.9105	0.7221	0.0005	0.0019	0.0152	0.9288	0.9704	99.86	10.8891	0.0085
gemma-3-12b-it	0.8376	0.9279	0.8919	0.7219	0.0	0.0	0.0019	0.9468	0.9601	99.30	14.911	0.0116
gemma-3-27b-it	0.8372	0.9315	0.8901	0.7061	0.0	0.0014	0.0051	0.9325	0.9322	100.00	18.9624	0.0148
Qwen3-8B	0.8198	0.9197	0.8904	0.7139	0.003	0.005	0.0025	0.9629	0.9789	100.00	9.4616	0.0074
Qwen3-14B	0.7943	0.936	0.8979	0.7024	0.0011	0.0	0.0243	0.9699	0.9109	99.72	10.8778	0.0085
Qwen3-1.7B	0.7827	0.8951	0.8552	0.592	0.0026	0.017	0.0397	0.8767	0.8937	99.02	7.3623	0.0057
Qwen3-4B	0.7767	0.9147	0.8811	0.6516	0.0015	0.006	0.0228	0.9054	0.9221	98.61	9.599	0.0075
Qwen3-32B	0.7748	0.9203	0.8982	0.7087	0.0	0.0007	0.0	0.9512	0.9272	99.72	16.8119	0.0131
phi-4	0.7727	0.8783	0.8565	0.6375	0.0012	0.0025	0.0194	0.9635	0.9121	97.77	9.0272	0.007
Qwen3-0.6B	0.7659	0.8768	0.8329	0.5129	0.0044	0.0372	0.0601	0.7991	0.8892	99.72	7.2455	0.0057
gemma-3-4b-it	0.7382	0.8757	0.8443	0.6361	0.0	0.0004	0.0244	0.9059	0.9087	99.30	10.6816	0.0083

Table 5
This table compares the performance of various models on the generalization test set. The first section evaluates closed-source models using a 1-shot prompting strategy. The second section presents results for open-source fine-tuned models.