SHACL-Guided Small Language Models for RDF Knowledge Graph Population: Semantic Evaluation, Cross-Domain Generalisation, and Long-Tail Mitigation

Célian Ringwald[1] Fabien Gandon[1] Catherine Faron[1]
Franck Michel[1] Hanna Abi Akl[1]

[1] Université Côte d'Azur, Inria, CNRS, I3S, France.
{celian.ringwald, fabien.gandon, catherine.faron, franck.michel, hanna.abi-akl}@inria.fr

Abstract

Populating RDF knowledge graphs from natural language is challenging when both datatype and object properties must be extracted under schema constraints. We extend the Kastor framework—a SHACL-guided, pattern-based relation extraction approach—scaling it from a single `dbo:Person` shape to 16 DBpedia entity classes. We introduce four semantic evaluation metrics for object property errors: URI validity, class-level semantic similarity, relation confusion rate, and hallucination estimation. We release **KastorKG**, an open resource comprising 19 DBpedia SHACL shapes, 141,096 distilled Wikipedia abstract–graph pairs, and 16 fine-tuned small language models (SLMs, 139 M parameters). Despite having >100× fewer parameters than LLM baselines, our SLMs outperform 13 B-parameter models on 11 of 16 Text2KGBench classes with zero subject or relation hallucinations, versus 12–16% and 7–9% hallucination rates for LLMs. Cross-domain transfer analysis on WebNLG further shows that example-specific pattern overlap and register similarity are stronger predictors of transferability than property-set overlap alone. Finally, long-tail property distributions are identified as the primary performance bottleneck, and sufficient-exposure sampling—guaranteeing at least 1,000 training examples per property—is shown to substantially improve rare-property recall. All code, models, and datasets are publicly available.

Keywords: Relation Extraction; Knowledge Graph Population; SHACL; Small Language Models; RDF; Text2KGBench; Long-tail Distribution

1 Introduction

Relation Extraction (RE) is a core task in information extraction, involving the identification of entities in unstructured text and the classification of semantic relationships between them. It plays a crucial role in populating and enriching knowledge graphs (KGs) by transforming textual data into structured relational triples. Recent advances in natural language processing (NLP)—notably with pre-trained language models (LMs)—have significantly improved RE systems, offering more context-aware and adaptable extraction capabilities with minimal task-specific supervision [Ringwald et al., 2025c].

To date, the primary focus of RE research has been on frequently occurring relations and well-defined entity types. Canonical datasets [Zhang et al., 2017, Elsahar et al., 2018, Yao et al., 2019] were introduced to support this direction at scale, often employing distant supervision to align textual content with existing KGs. Such approaches have yielded effective models for extracting diverse relational facts. Nonetheless, extracting less frequent and more complex relational patterns remain challenging for LMs.

Running example. Consider the Wikipedia abstract sentence: *"Valery Kobzarenko (born 5 February 1977 in Kiev, Ukraine) is a Ukrainian former professional cyclist."* A shape-guided extractor targeting `dbo:Person` should produce the RDF triples:

```
:Valery_Kobzarenko  dbo:birthDate  "1977-02-05"^^xsd:date ;
dbo:birthPlace  dbr:Ukraine ;
dbo:nationality  dbr:Ukraine .
```

This example illustrates both the challenge of generating valid DBpedia URIs for object properties (e.g. distinguishing `dbr:Kiev` from `dbr:Ukraine`) and the importance of a SHACL shape to constrain which properties and value types are expected.

The framework [Ringwald et al., 2025a,d] [Ringwald, 2024] uses a simple SHACL shape s to specify the extraction target, constraining property usage and, optionally, the range of each property. The extractor is defined as a function $\mathcal{M}_{\mathcal{K}} : W \times S \to G;\ (t, s) \mapsto \hat{g}$, where $t \in W$ is an input text, $s \in S$ is a SHACL shape, and $\hat{g}$ is an RDF graph implied by t and valid against s. From s the set Π of all possible property combinations is derived (i.e. example-specific patterns $\pi \in \Pi$). An RDF graph g is valid against a pattern π if it contains exactly all the properties in π and no more.

Extension over the KCAP 2025 conference paper. A shorter version of this work was presented at KCAP 2025 [Ringwald et al., 2025d]. That paper covered: (1) the extension of Kastor to jointly extract datatype and object properties for the `dbo:Person` class; and (2) long-tail distribution mitigation strategies on a single shape. The present journal version adds three major contributions not included in the conference paper: (i) four semantic metrics for object property evaluation; (ii) the translation of all 19 Text2KGBench micro-ontologies into SHACL shapes and the release of KastorKG, a distilled knowledge base aligning 141,096 Wikipedia Clean and Markdown abstracts with DBpedia graphs across 16 entity classes; and (iii) the fine-tuning and evaluation of 16 shape-aware SLMs on both the KastorKG test sets and the Text2KGBench test sets, enabling comparison with LLM baselines and cross-domain transfer analysis of our models from Wikipedia to WebNLG evaluation domains.

This paper **extends the Kastor framework** to SHACL shapes encompassing both *datatype* and *object properties*. It addresses the following four research questions:

RQ1. Is it possible to perform pattern-based RE jointly for datatype *and* object properties?

RQ2. How can we evaluate the specific errors occurring in object property extraction, and do these metrics generalise across multiple SHACL shapes and evaluation domains?

RQ3a. To what extent can shape-aware SLMs distilled from Wikipedia generalise across a large collection of SHACL shapes?

RQ3b. How do models trained on Wikipedia abstracts transfer to a structurally different evaluation domain (WebNLG)?

RQ4. How do such models perform over a long-tail distribution of properties, and how can this distribution be taken into account to improve handling of under-represented properties?

The main contributions of this paper are:

1. We extend pattern-based RE to jointly extract datatype and object properties, producing RDF graphs.
2. We propose a precise error analysis and design four semantic metrics to better estimate model performance on object properties (URI validity, class similarity, relation confusion, hallucination).
3. We release **KastorKG**, a large-scale open resource comprising: 19 DBpedia SHACL Turtle shapes derived from Text2KGBench micro-ontologies, a distilled knowledge graph of 141,096 Wikipedia abstract–RDF graph pairs (Clean & Markdown) across 16 DBpedia classes, and 16 shape-aware SLMs trained on this resource and also released on HuggingFace.
4. We challenge our resulting SLMs on **Text2KGBench**, evaluating the generalisation of our method across 16 DBpedia class shapes, and perform a cross-domain transfer analysis (Wikipedia $\rightarrow$ WebNLG) to explain per-class performance variations. Compact SLMs outperform LLM baselines on most classes, with zero hallucinated subjects or relations vs. 12–16 % and 7–9 % hallucination rates for the LLM baselines.
5. We propose and benchmark a set of solutions to address the long-tail distribution of properties, revealing the necessity of ensuring a sufficient exposure threshold per property.
6. All code[1], datasets[2], models[3], and the distilled KG KastorKG[4] are open and reusable.

Contributions 1 and 5 were partially presented at KCAP 2025; Contributions 2, 3, 4, and 6 are new to this journal version.

The remainder of this paper is structured as follows. Section 2 reviews related work. Section 3 introduces background and notation. Section 4 presents the extension to

[1]https://github.com/datalogism/Kastor/
[2]https://zenodo.org/records/15917325
[3]https://huggingface.co/collections/Datartisan/kastorsmodels
[4]https://zenodo.org/records/18862240

object properties on the `dbo:Person` class. Section 5 introduces semantic metrics for object property evaluation. Section 6 presents the Text2KGBench benchmark. Section 7 addresses long-tail distribution. Section 8 discusses the findings. Section 9 outlines future directions, and Section 10 concludes.

2 Related Work

2.1 From Ontology-Driven to Shape-Constrained Relation Extraction

SHACL (Shapes Constraint Language) has primarily been used to validate existing KGs against structural constraints [Mihindukulasooriya et al., 2023]. In KastorKG, SHACL shapes instead *define* the target extraction output, driving training data construction via pattern enumeration and enabling pattern-level evaluation. This contrasts with traditional ontology-driven RE methods, which use schemas for prompting or post-hoc validation but do not enforce structural validity during generation. Representative schema-driven systems include SPIRES [Caufield et al., 2024], which constructs prompts via schema-aware retrieval; GenIE [Josifoski et al., 2022], which generates entity-linked triples constrained to a predefined relation vocabulary; and REBEL [Huguet Cabot and Navigli, 2021], which produces relation-typed triples but requires post-hoc entity linking. Text2KGBench [Mihindukulasooriya et al., 2023] provides a benchmark for evaluating ontology-driven extraction across multiple DBpedia and Wikidata classes; its refinement, Text2KGBench-LettrIA [Aubert et al., 2023], re-annotated 4,860 sentences with stricter guidelines and improved ontological formalization.

2.2 Dealing with Under-Represented Knowledge

Sequence-to-sequence models fine-tuned on massive datasets, such as REBEL [Huguet Cabot and Navigli, 2021] and GenIE [Josifoski et al., 2022], generally achieve strong performance but are ill-suited to rare properties, as shown in [He et al., 2025, Josifoski et al., 2023]. SynthIE [Josifoski et al., 2023] attempts to address this issue by using an LLM as a data generator to train a specialised SLM. Despite strong performance on synthetic test data, results do not reach the same level on real-world, noisy corpora such as those covered by REBEL, and the limitation becomes particularly apparent on long-tail property distributions.

Text2KGBench [Mihindukulasooriya et al., 2023] was evaluated using two LLMs (Vicuna and Alpaca), and the results highlight the difficulty of extracting graphs described by an ontology—mainly due to the models' inability to handle under-represented properties. Similarly, Arzt et al. [2025] compares LM performance on a relation classification task, demonstrating LM limitations on under-represented relations. Allen-Zhu and Li [2025] emphasises the importance of scaling laws in learning specific facts and shows that at least 1,000 training exposures are required for stable performance—a threshold applied at the property level in this paper.

To address data imbalance, a standard solution in machine learning is to use re-weighted loss functions that inversely weight each class [Lin et al., 2017, Jiang et al., 2025, Xu et al., 2022].

2.3 Hallucination Detection and Factual Verification

Estimating whether a generated triple is supported by the source text is a central challenge in RE evaluation. Huguet Cabot et al. [2023] introduce the RED^{FM} benchmark and the associated Triplet Critic model (`mdeberta-v3-base-triplet-critic-xnli`), a cross-lingual NLI model trained to score the entailment between a text and a candidate triple. AlignScore [Zha et al., 2023] provides a unified factual consistency metric based on fine-tuned alignment functions that outperform prior NLI-based approaches on multiple benchmarks. XLM-RoBERTa NLI models offer a multilingual alternative for entailment-based scoring. Section 5 compares all three approaches to identify which best discriminates between genuine errors and unlabelled correct extractions.

3 Background and Notation

SHACL Shapes. This work focuses on simple SHACL *NodeShape* s specifying a target class and a set of property constraints, including the expected property paths, datatypes, and class ranges. Given a shape s, we denote by $\mathcal{P}(s)$ the set of all properties it constrains. A *pattern* π is a subset of $\mathcal{P}(s)$; the set of all patterns realised in a knowledge base is denoted Π. A graph g is valid against π if it contains exactly the properties in π.

Example. The following NodeShape constrains `dbo:Person` instances to carry a `dbo:birthPlace` object property whose value must be a `dbo:Place`:

```
<PersonShape> a sh:NodeShape ;
    sh:targetClass dbo:Person ;
    sh:property [ sh:path dbo:birthPlace ;
                  sh:class dbo:Place ] .
```

A valid RDF triple satisfying this constraint is:
`dbr:Albert_Einstein  dbo:birthPlace  dbr:Ulm .`

Knowledge Distillation. Let $\mathcal{W}$ be a corpus of Wikipedia abstracts and $\mathcal{G}$ a knowledge graph. The *dual base* $\mathcal{K}$ aligns each abstract w with the corresponding DBpedia graph g of the described entity:

$$\mathcal{K} := \{(w, g) \in \mathcal{W} \times \mathcal{G} \mid \exists\, e \in \mathrm{IRI} : \mathrm{desc}_{\mathcal{W}}(e) = w \wedge \mathrm{desc}_{\mathcal{G}}(e) = g\} \tag{1}$$

The distilled sub-base $\mathcal{K}_{\text{dist}}(s)$ retains only those pairs (w, g) where g is supported by w and is valid against at least one pattern of s.

Evaluation Metrics. The paper uses *micro-F1* (F_1^-), the average F1 over all graphs without weighting by property frequency, and *macro-F1* (F_1^+), the average F1 computed per property. F_1^- captures overall performance but is biased towards frequent properties; F_1^+ gives a balanced view across all properties.

False Positives and False Negatives. At the triple level, given a reference graph g and a generated graph $\hat{g}$ for the same entity, we define:

$$FP = \hat{g} \setminus g = \{(\hat{s}, \hat{p}, \hat{o}) \in \hat{g} \mid (\hat{s}, \hat{p}, \hat{o}) \notin g\} \tag{2}$$

$$FN = g \setminus \hat{g} = \{(s,p,o) \in g \mid (s,p,o) \notin \hat{g}\} \tag{3}$$

Among false positives, *substitution errors* FP_{diff} (correct predicate, wrong object) are further distinguished from *novel errors* FP_{new} (unsolicited triples not matching any reference triple with the same predicate and subject).

Graph Pattern Metrics. Let $\widehat{\mathbb{G}}_D^{\nleftrightarrow}$ denote the set of generated graphs whose property set does not exactly match the expected one, and $\widehat{\mathbb{G}}_D^{\rightarrow} \subseteq \widehat{\mathbb{G}}_D^{\nleftrightarrow}$ the subset that strictly *extends* the expected property set (i.e. it is a proper superset). The following rate is defined:

$$r_{\hat{\mathbb{G}}_D^{\leftrightarrow}} = 1 - \frac{|\widehat{\mathbb{G}}_D^{\nleftrightarrow}|}{|D|} \tag{4}$$

the rate of generated graphs that follow exactly the expected example-specific pattern. The *Pattern Extension Capacity* (PEC_D) measures, among the deviating graphs, the proportion that deviate by extension rather than reduction:

$$PEC_D = \frac{|\widehat{\mathbb{G}}_D^{\rightarrow}| - |\widehat{\mathbb{G}}_D^{\leftrightarrow}|}{|\widehat{\mathbb{G}}_D^{\nleftrightarrow}|} \tag{5}$$

$PEC_D \in [0,1]$; a value close to 1 indicates that deviations consist mainly of additional triples (over-generation), while a value close to 0 indicates under-generation.

4 Extracting Object and Datatype Properties

4.1 Knowledge Distillation

The distillation uses the 2022.09 DBpedia data dump[5] containing 6,109,994 Wikipedia abstracts and their DBpedia graphs as the dual base $\mathcal{K}$ (Eq. 1).

The experiments focus on the `dbo:Person` class. A shape that previously covered only datatype properties [Ringwald et al., 2025a,b] is extended

$$\begin{aligned}\mathcal{P}(s^*_{dt}) = \{&\texttt{rdfs:label}, \texttt{dbo:alias}, \texttt{dbo:birthName},\\ &\texttt{dbo:birthDate}, \texttt{dbo:birthYear},\\ &\texttt{dbo:deathDate}, \texttt{dbo:deathYear}\}\end{aligned}$$

to additionally consider three object properties:

$$\mathcal{P}(s^*_{op}) = \{\texttt{dbo:birthPlace}, \texttt{dbo:nationality}, \texttt{dbo:deathPlace}\}$$

The extended maximal shape covers both[6]: $\mathcal{P}(s^*) = \mathcal{P}(s^*_{dt}) \cup \mathcal{P}(s^*_{op})$.

[5] https://databus.dbpedia.org/cringwald/collections/kstor

[6] https://github.com/datalogism/KastorKG/blob/main/shapes/PersonShape_op_and_dp.ttl

Inference Rules. Two rule sets are applied. The datatype rule set $\mathcal{R}_{dt}$ infers missing temporal values:

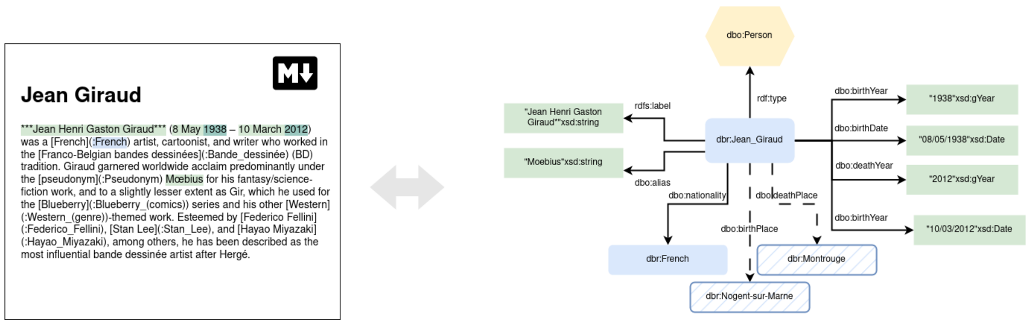
$$\mathcal{R}_{dt} : \begin{cases} \texttt{dbo:deathDate} \models \texttt{dbo:deathYear} \\ \texttt{dbo:birthDate} \models \texttt{dbo:birthYear} \end{cases} \tag{6}$$

The object-property rule set $\mathcal{R}_{op}$ infers countries from place objects:

$$\mathcal{R}_{op} : (s, p, o) \wedge (o, \texttt{dbo:country}, c) \models (s, p, c) \tag{7}$$

Applying $\mathcal{R} = \mathcal{R}_{dt} \cup \mathcal{R}_{op}$ yields an augmented base $\mathcal{K}^{\mathcal{R}}$.

Wikicheck for Object Properties. The *Wikicheck* step retains only triples evidenced in the corresponding Wikipedia abstract. For datatype properties, this is done by plain-text string or date matching. For object properties, whose values are DBpedia URIs not present in plain text, the Wikipedia page's HTML is retrieved via the REST API (https://en.wikipedia.org/api/rest_v1/) and converted to Markdown ($\mathcal{W}_{MD}$). The process verifies whether the DBpedia URIs appear in the Markdown links and checks that the object type is consistent with the range declared in the shape. This process is illustrated in Fig. 1.



Jean Giraud

Jean Henri Gaston Giraud (8 May 1938 – 10 March 2012) was a [French](:French) artist, cartoonist, and writer who worked in the [Franco-Belgian bandes dessinées](:Bande_dessinée) (BD) tradition. Giraud garnered worldwide acclaim predominantly under the [pseudonym](:Pseudonym) Mœbius for his fantasy/science-fiction work, and to a slightly lesser extent as Gir, which he used for the [Blueberry](:Blueberry_(comics)) series and his other [Western](:Western_(genre))-themed work. Esteemed by [Federico Fellini](:Federico_Fellini), [Stan Lee](:Stan_Lee), and [Hayao Miyazaki](:Hayao_Miyazaki), among others, he has been described as the most influential bande dessinée artist after Hergé.

Figure 1: Wikicheck applied to a graph containing both datatype and object properties, using the Markdown version of the Wikipedia article. URIs of object properties must appear in Markdown links.

After distillation, the base $\mathcal{K}_{\text{dist}}(s^*)$ contains 136,718 graphs:

$$\mathcal{K}_{\text{dist}}(s^*) \coloneqq \{(w, w_{MD}, g) \mid (w, g) \in \mathcal{K}^{\mathcal{R}},\ w_{MD} \models g,\ \exists \pi \in \Pi(s^*) : g \text{ valid against } \pi\} \tag{8}$$

4.2 Dataset

Dataset $D1$ is sampled from $\mathcal{K}_{\text{dist}}(s^*)$, requiring each graph to contain at least one datatype and one object property, and restricting to Wikipedia articles published before 2021 (the BART release date):

$$\begin{aligned} D1 \coloneqq \{(w, w_{MD}, g) \in \mathcal{K}_{\text{dist}}(s^*) \mid & \, g \text{ contains} \geq 1 \text{ prop. of } s^*_{op} \\ & \text{and } \geq 1 \text{ prop. of } s^*_{dt} \\ & \text{and } w \text{ predates } 2021\} \end{aligned} \tag{9}$$

Table 1 gives the number of triples and frequency per property.

The resulting dataset $D1$ contains **1,200** Wikipedia abstract–graph pairs.

Table 1: Frequency and number of triples per property in the distilled base $\mathcal{K}_{\text{dist}}(s^*)$ and in sample $D1$.

	$\mathcal{K}_{\text{dist}}(s^*)$		$D1$	
Property	Count	Freq.	Count	Freq.
birthYear	105,818	0.77	1,010	0.84
birthPlace	15,279	0.11	996	0.83
birthDate	97,039	0.71	927	0.77
label	92,691	0.68	870	0.73
deathYear	37,633	0.28	457	0.38
deathDate	32,743	0.24	395	0.33
deathPlace	4,859	0.04	291	0.24
nationality	2,211	0.02	162	0.14
birthName	12,265	0.09	130	0.11
alias	1,398	0.01	14	0.01

4.3 REBEL Baseline

Results are compared against REBEL-large (406 M parameters). Since REBEL [Huguet Cabot and Navigli, 2021] outputs linearised sequences of entity surface forms and relation labels without producing IRIs, URIs are constructed from the extracted labels following the Wikipedia convention (variant *REBEL*) or reconciled via DBpediaLookup (variant *REBEL*$\times$*DBpediaLookup*). Table 2 shows the property alignment used.

Table 2: REBEL / DBpedia relation alignment.

REBEL label	DBpedia property
place of birth	`dbo:birthPlace`
place of death	`dbo:deathPlace`
country of citizenship	`dbo:nationality`
date of birth	`dbo:birthDate`
date of death	`dbo:deathDate`

4.4 Training Details

All models use BART-base (139 M parameters). Graphs are linearised in the *Turtle-Light one-line factorised* format, which achieved the best performance in prior work [?Ringwald et al., 2024]. Models are fine-tuned on a single NVIDIA Tesla V100 GPU using an inverse square root scheduler with an initial learning rate of 5×10^{-5}, 1,000 warmup steps, and early stopping with patience 5. The prompt is: `$entity_URI : $Abstract`. Each model uses 10-fold cross-validation.

A model is trained for each combination of input type $W_{\text{type}} \in \{\text{MD}, \text{PLAIN}\}$ and shape $s \in \{s^*, s^*_{op}, s^*_{dt}\}$, yielding models denoted $\mathcal{M}D1(W_{\text{type}}, s)$.

4.5 Results: Input Type and Property Type

Table 3 summarises performance. Overall extraction results are strong; however, object properties (s^*_{op}) yield lower F_1 scores and higher standard deviation than datatype properties (s^*_{dt}), confirming that extracting object properties is less stable and more error-prone. Markdown input does not consistently outperform plain text.

Table 3: Performance of $\mathcal{M}D1$ models with different shapes and input types.

Model	Recall	Prec.	F_1^-	F_1^+
$\mathcal{M}D1(\text{MD}, s^*_{dt})$	98 ± 2	89 ± 3	97 ± 3	89 ± 11
$\mathcal{M}D1(\text{MD}, s^*_{op})$	91 ± 13	88 ± 13	90 ± 13	83 ± 22
$\mathcal{M}D1(\text{MD}, s^*)$	95 ± 5	92 ± 6	94 ± 5	86 ± 14
$\mathcal{M}D1(\text{PLAIN}, s^*_{dt})$	98 ± 2	96 ± 3	98 ± 3	91 ± 10
$\mathcal{M}D1(\text{PLAIN}, s^*_{op})$	90 ± 14	87 ± 15	89 ± 14	83 ± 22
$\mathcal{M}D1(\text{PLAIN}, s^*)$	96 ± 4	94 ± 6	95 ± 5	88 ± 14

4.6 Comparison with the REBEL Baseline

The Kastor models produce 100 % of triples in well-formed RDF with correct subject URIs, whereas only 65 % of REBEL's subjects point to valid DBpedia URIs (improved to 72 % with DBpediaLookup). Table 4 shows that REBEL achieves reasonable precision but fails to extract many facts (low recall) and exhibits a large gap between micro and macro F_1, indicating that it can reliably extract only a small number of properties.

Table 4: Comparison of $\mathcal{M}D1$ with REBEL.

Model	Recall	Prec.	F_1^-	F_1^+
$\mathcal{M}D1(\text{PLAIN}, s^*_{rebel})$	96 ± 5	93 ± 5	94 ± 5	89 ± 10
REBEL	54 ± 1	83 ± 1	65 ± 1	41 ± 1
REBEL×DBpediaLookup	59 ± 0.3	83 ± 1	69 ± 0.3	44 ± 0.6

4.7 Role of Property Representation

To investigate the impact of property frequency on performance, the property set $\mathcal{P}(s^*)$ is split into two sub-shapes around the mean property frequency $\mu_p = 0.295$ in $D1$:

- s^*_+ (frequent): `rdfs:label`, `dbo:birthDate`, `dbo:birthPlace`, `dbo:birthYear`, `dbo:deathDate`, `dbo:deathYear`
- s^*_- (rare): `dbo:birthName`, `dbo:nationality`, `dbo:deathPlace`, `dbo:alias`

Table 5 confirms that property frequency in the training set directly affects performance, and that specialising a model on rare properties alone is insufficient.

A per-property breakdown of F_1^+ is provided in Section 7 (Fig. 4), which also covers the `dbo:Person` shape in the context of the long-tail analysis.

Table 5: Performance of $\mathcal{M}D1$ models on frequency-based subshapes.

Model	F_1^-	F_1^+
$\mathcal{M}D1(\text{PLAIN}, s^*)$	95 ± 5	88 ± 15
$\mathcal{M}D1(\text{PLAIN}, s^*_+)$	$\mathbf{96 \pm 4}$	$\mathbf{96 \pm 5}$
$\mathcal{M}D1(\text{PLAIN}, s^*_-)$	70 ± 21	64 ± 26

5 Semantic Metrics for Object Property Evaluation

5.1 Error Taxonomy

All errors produced during the 10-fold validation of $\mathcal{M}D1(\text{PLAIN}, s^*)$ and $\mathcal{M}D1(\text{MD}, s^*)$ on object properties were annotated according to the following eight categories:

E1 — More precise: Generated URI exists but refers to a more specific class than the ground truth.
Abstract: “Valery Kobzarenko (born 5 February 1977 in Kiev, Ukraine) is a Ukrainian former professional cyclist.”
Generated: `:Valery_Kobzarenko  dbo:birthPlace  :Kiev`
Expected: `:Valery_Kobzarenko  dbo:birthPlace  :Ukraine`

E2 — Less precise: Generated URI exists but is less specific than expected (e.g., `:Jianyang` instead of `:Jianyang%2C_Sichuan`).

E3 — URI redirect OK: Generated URI redirects to the correct one when dereferenced.

E4 — Encoding error: Generated URI differs only in character encoding.

E5 — Wrong predicate: Correct object URI but associated with a wrong property.

E6 — Wrong URI: URI exists but corresponds to a different entity mentioned in the abstract.

E7 — Both false: Both the generated and ground-truth triples are incorrect.

E8 — Hallucination: Generated triple is not expressed anywhere in the input text.
Abstract: “Yousef Haikal (1907–1989) was a Jordanian Ambassador and the Mayor of Jaffa between 1945 and 1948.”
Generated: `:Yousef_Haikal  dbo:deathPlace  :Jaffa`
Expected: $\emptyset$ (deathPlace not mentioned in the text)

Hallucination (E8) is the most frequent error type in both input formats, followed by precision errors (Fig. 2). Markdown abstracts produce more precision and encoding errors, whereas plain abstracts produce more predicate-selection errors.

5.2 Proposed Metrics

The eight error categories motivate four complementary metrics. Table 6 shows which metric addresses which error type.

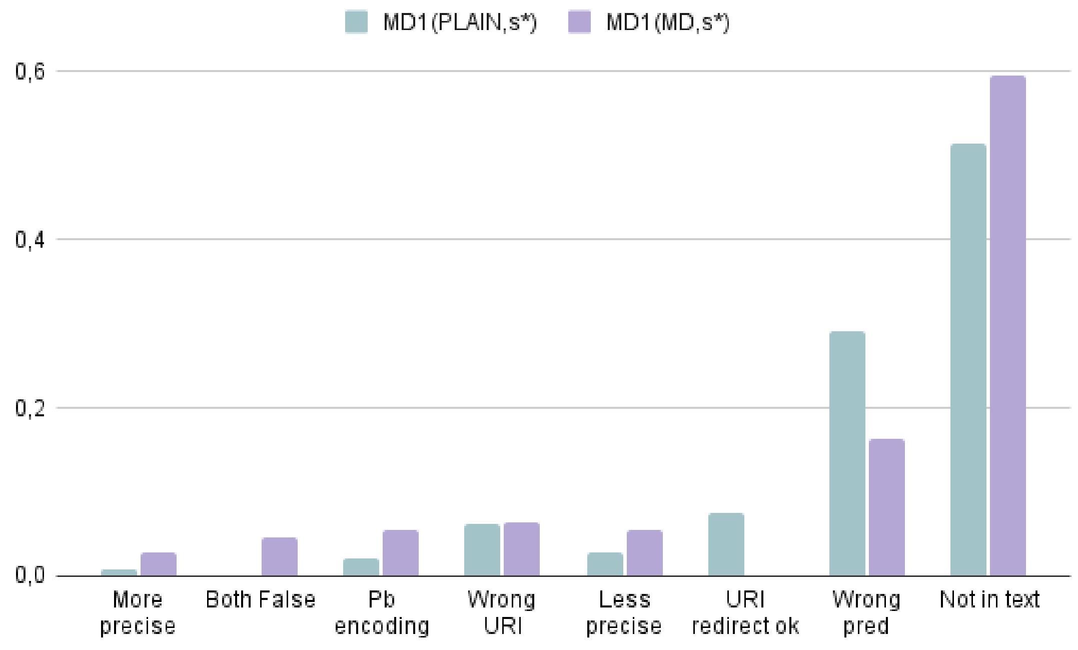


Figure 2: Distribution of error categories by model and input type.

Table 6: Mapping from error categories (E1–E8) to proposed metrics.

Error categories	**Metric**
E1 (more precise), E2 (less precise)	Sem_{sim} (class-level semantic similarity)
E3 (URI redirect), E4 (encoding error)	$r_{\widehat{\mathcal{O}}^+}$ (valid URI rate; E3/E4 penalised)
E5 (wrong predicate)	WRC (wrong relation choice rate)
E6 (wrong URI, valid entity)	$r_{\widehat{\mathcal{O}}^+}$ (URI exists but wrong entity)
E8 (hallucination)	AS (AlignScore hallucination estimator)

FP (Eq. 2) is partitioned into two subsets. FP_{diff} refers to cases where the correct predicate is used but with a different object than the ground truth. FP_{new} covers triples whose predicate is absent from the ground truth entirely:

$$FP_{diff} = \{(\hat{s}, \hat{p}, \hat{o}) \mid (s, p, o) \in g,\ (\hat{s}, \hat{p}, \hat{o}) \in \hat{g},\ \hat{s} = s,\ \hat{p} = p,\ \hat{o} \neq o\} \quad (10)$$

$$FP_{new} = FP - FP_{diff} \quad (11)$$

5.2.1 Rate of Valid Object URIs

$r_{\widehat{\mathcal{O}}^+}(FP)$ measures, among the object-property false positives, the proportion whose generated URI exists in the KG $\mathcal{K}$ even though it differs from the ground truth:

$$\widehat{\mathcal{O}}^+(FP) = \{\hat{o} \mid (\hat{s}, \hat{p}, \hat{o}) \in FP,\ \hat{o} \in \mathcal{K},\ \hat{o} \in E\} \quad (12)$$

$$\widehat{\mathcal{O}}^-(FP) = \{\hat{o} \mid (\hat{s}, \hat{p}, \hat{o}) \in FP,\ \hat{o} \notin \mathcal{K},\ \hat{o} \in E\} \quad (13)$$

$$r_{\widehat{\mathcal{O}}^+}(FP) = \frac{|\widehat{\mathcal{O}}^+(FP)|}{|\widehat{\mathcal{O}}^+(FP) \cup \widehat{\mathcal{O}}^-(FP)|} \quad (14)$$

5.2.2 Class-Level Semantic Similarity

For false-positive object URIs that exist in the KG, their type is retrieved and the Wu–Palmer taxonomic similarity [Wu and Palmer, 1994] computed—the same formulation used by the *Sem@k* metric of Hubert et al. [2023]:

$$\text{Sem}_{sim}(c_1, c_2) = \frac{2 \times \delta(c_3, \rho)}{\delta(c_1, c_3) + \delta(c_2, c_3) + 2 \times \delta(c_3, \rho)} \tag{15}$$

where c_3 is the most specific common superclass of c_1 and c_2, ρ is the taxonomy root, and δ is path length. This is applied to compare $c_{\hat{o}}$ (type of generated object) with either the ground-truth object type c_o (for FP_{diff}) or the expected range in s^* (for FP_{new}).

5.2.3 Wrong Relation Choice Rate

WRC estimates how often a model generates the correct object URI but assigns it to the wrong predicate. Formally, FP_{rel-} is the set of novel false positives whose object also appears in the false negatives under a different predicate:

$$FP_{rel-} = \{(\hat{s}, \hat{p}, \hat{o}) \mid (s, p, o) \in FN,\ (\hat{s}, \hat{p}, \hat{o}) \in FP_{new},\ \hat{s} = s,\ \hat{p} \neq p,\ \hat{o} = o\} \tag{16}$$

$$WRC(FP_{new}) = \frac{|obj(FP_{rel-})|}{|obj(FP_{new}) \cup obj(FN)|} \tag{17}$$

where $obj(\cdot)$ retrieves the set of objects from a set of triples. This metric is not a perfect estimator—it does not account for FP_{diff} or the complete ground truth—and may be refined in future work.

5.2.4 Hallucination Estimation

Three entailment-based systems are compared: (i) AlignScore [Zha et al., 2023] (AS), (ii) Triplet Critic (TC), the `mdeberta-v3-base-triplet-critic-xnli` model released by Babelscape alongside the RED^{FM} dataset [Huguet Cabot et al., 2023], and (iii) an XLM-RoBERTa NLI model (NLI). AlignScore best discriminates between false positives that are semantically supported by the text (FP^+, i.e. unlabelled correct extractions) and those that are genuine errors (FP^-) across both property types (Table 7), where FP^+ and FP^- denote the annotation-based split of false positives from the 10-fold validation set of $\mathcal{M}D1$.

Table 7: Entailment system scores by property type and error class.

Prop. type	Class	$AS(w, FP)$	$TC(w, FP)$	$NLI(w, FP)$
dt	FP+	**0.74**	0.58	<u>0.57</u>
	FP-	**0.29**	<u>0.70</u>	0.26
obj	FP+	0.80	**0.90**	0.48
	FP-	**0.11**	<u>0.51</u>	0.29
dt∪obj	FP+	**0.76**	0.69	<u>0.54</u>
	FP-	**0.12**	<u>0.52</u>	0.28

5.3 Results on $\mathcal{M}D1$

Table 8 summarises the proposed metrics applied to both models. Markdown input yields a higher rate of valid generated URIs ($r_{\widehat{\mathcal{O}}^+} = 0.93$ vs. 0.76 for plain) and higher semantic similarity to the expected range. Plain input, however, produces FP_{diff} objects that are ontologically closer to the ground truth. For false negatives, plain abstracts yield more genuine KG discoveries, whilst Markdown false negatives tend to reflect KG noise.

Table 8: Proposed metrics applied to errors of $\mathcal{M}D1(\text{PLAIN}, s^*)$ and $\mathcal{M}D1(\text{MD}, s^*)$.

Focus	Metric	PLAIN	MD
Pattern	$r_{\hat{\mathbb{G}}^{\leftrightarrow}_{D}}$	0.78	0.73
	PEC	0.34	**0.38**
Object prop. errors	$AS(w, FP)$	0.57	0.56
	$r_{\widehat{\mathcal{O}}^+(FP_{new})}$	0.76	**0.93**
	$WRC(FP_{new})$	9.36%	**8.57%**
	$\text{Sem}_{sim}(\hat{g}, s^*)$	0.80	**0.98**
	$r_{\widehat{\mathcal{O}}^+(FP_{diff})}$	**0.49**	0.29
	$\text{Sem}_{sim}(\hat{g}, g)$	**0.60**	0.37
	$AS(w, FN)$	0.42	**0.30**

6 Challenging Text2KGBench

6.1 From Ontology-Driven to Shape-Driven RE

Text2KGBench[7] originally benchmarks LLM ontology comprehension on 19 DBpedia classes (WebNLG subset) and 10 Wikidata classes (TekGen). Each micro-ontology is translated into a simple SHACL shape and entity-centred graphs, with the corresponding data expressed as RDF triples. Each shape lists the properties and the range constraints they must satisfy.

Fig. 3 shows the `dbo:City` shape as an example. After verifying which properties are actually instantiated in $\mathcal{K}$, all shapes are simplified by removing properties never observed in the distilled KG, reducing the number of property constraints by roughly half. The resulting set of shapes is denoted $\mathbb{S}$. All shapes are available at https://github.com/datalogism/KastorKG/tree/main/shapes/txt2kg.

6.2 KastorKG-Distillation

The knowledge distillation process for Text2KGBench proceeds as follows. Instance coverage was first analysed for each shape $s \in \mathbb{S}$. Three classes (`dbo:Astronaut`, `dbo:ComicsCharacter`, and `dbo:Monument`) were excluded owing to insufficient DBpedia instances for model fine-tuning.

Following Eq. 8, the initial knowledge base $\mathcal{K}$ was distilled for each shape s, with the resulting triples stored in dedicated RDF named graphs. For each shape, at

[7] https://github.com/cenguix/Text2KGBench

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix sh:  <http://www.w3.org/ns/shacl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<http://shaclshapes.org/CityShape> a sh:NodeShape ;
    sh:targetClass <http://dbpedia.org/ontology/City> ;
    sh:property
        [ sh:datatype xsd:double ;
          sh:path <http://dbpedia.org/ontology/populationDensity> ],
        [ sh:path <http://dbpedia.org/ontology/timeZone> ],
        [ sh:class <http://dbpedia.org/ontology/PopulatedPlace> ;
          sh:path <http://dbpedia.org/ontology/state> ],
        [ sh:datatype xsd:string ;
          sh:path <http://dbpedia.org/ontology/postalCode> ] .
        % ... (full shape at the GitHub repository)
```

Figure 3: Excerpt of the `dbo:City` SHACL shape derived from Text2KGBench.

Table 9: Statistics of the knowledge distillation process across 19 entity classes. The table compares the original SHACL shape specification with the coverage in the full DBpedia knowledge base ($\mathcal{K}$) and the final distilled, wiki-validated knowledge base ($\mathcal{K}_{\text{dist}}$). Bold values indicate maximum; underlined values indicate minimum.

Entity Class	**SHACL Shape**		**Full DBpedia ($\mathcal{K}$)**		**Wikipedia MD**	**Distilled KB ($\mathcal{K}_{\text{dist}}$)**	
	Properties	Max. Patterns	Properties	Entities	Abstracts (%)	Patterns	Properties
Airport	39	2^{39}	20	10,000	69	14	4
Artist	39	2^{39}	18	103,603	41	173	12
Astronaut	38	2^{38}	26	668	100	—	—
Athlete	37	2^{37}	13	**260,000**	5	121	15
Building	38	2^{38}	20	60,000	22	37	7
CelestialBody	27	2^{27}	19	10,000	100	4	3
City	23	2^{23}	16	10,691	100	9	4
ComicsCharacter	18	2^{18}	9	3,520	100	2	2
Company	28	2^{28}	20	65,851	27	44	8
Film	44	2^{44}	18	73,837	47	180	10
Food	18	2^{18}	13	7,331	100	6	3
MeansOfTransport	**68**	2^{68}	**49**	36,954	30	4	18
Monument	26	2^{26}	12	1,707	100	2	2
MusicalWork	35	2^{35}	17	11,175	17	126	9
Politician	40	2^{40}	29	100,000	13	175	17
Scientist	47	2^{47}	13	43,595	48	**199**	11
SportsTeam	24	2^{24}	12	11,338	57	19	6
University	46	2^{46}	27	13,389	91	55	10
WrittenWork	44	2^{44}	26	64,650	52	73	12

least 2,000 Wikipedia Markdown abstracts (W_{MD}) were collected and the Wikicheck verification process applied to validate and enrich graph properties.

Table 9 shows that the number of distinct property patterns (π) extracted from each shape remains manageable without combinatorial explosion. Three groups emerge from these statistics:

- **Well-described classes** ($|\pi(s)| > 100$): `dbo:Artist`, `dbo:Athlete`, `dbo:Film`, `dbo:MusicalWork`, `dbo:Politician`, and `dbo:Scientist`, which have rich representation in both knowledge graphs and text.
- **Poorly described classes**: `dbo:CelestialBody`, `dbo:Food`, and `dbo:MeansOfTransport`, with limited coverage.
- **Moderately described classes**: remaining shapes with intermediate representation.

The resulting **KastorKG**[8] is a distilled knowledge graph produced by the Kastor Framework [Ringwald et al., 2025a]. It is constructed from DBpedia by (i) filtering triples through SHACL shape constraints and (ii) verifying that each triple can be grounded in Wikipedia Markdown abstracts (*WikiCheck*). The KB covers **19 `dbo:` entity classes** and approximately **141,000 wiki-validated entities**, released as an N-Quads dump. It is structured using RDF named graphs[9] as described in Table 10, and can be loaded in Apache Jena TDB and queried through a local Corese SPARQL endpoint.

Table 10: Named graph organisation under `http://ns.inria.fr/kstor/`. The default graph (`urn:x-arq:DefaultGraph`) holds the full DBpedia source dump.

Graph suffix	**Content**
`shapes/{shape}`	SHACL shape definition
`class_randoms_id/{class}`	Random UUID per entity (reproducible sampling)
`inferences/{shape}`	Triples inferred by Corese rules (e.g. `dbo:birthDate` ⇒ `dbo:birthYear`)
`wiki_md/{class}`	Wikipedia Markdown abstracts (`rdfs:comment`)
`wikichecked/{shape}/abstract_md`	**Main distilled KB** — wiki-validated triples
`wikinew_202004/{shape}`	Temporal split — articles published after April 2020

6.3 KastorKG-Text2KGBench comparison

Since both datasets derive from WebNLG [Gardent et al., 2017], they are first compared using the lexical and syntactic metrics proposed in [Gardent et al., 2017], as shown in Table 20. Compared to WebNLG and Text2KGBench, KastorKG is a large-scale resource with 141,096 data–text pairs and high lexical richness (CTTR = 58.35; 308,109 word types). Texts are substantially longer and more complex: 59.2% contain three or more sentences, with an average of 98.80 words per abstract compared to 22.69 (WebNLG) and 20.15 (Text2KGBench), reflecting the encyclopaedic nature of Wikipedia abstracts. Moreover, the graphs aligned in KastorKG are notably larger than those in the other two datasets. Comparing KastorKG and the Text2KGBench DBpedia subset across 20 entity categories (Table 21), KastorKG contains substantially more entities overall (116,097 vs. 721 in Text2KGBench), reflecting the large-scale distillation from the full DBpedia dump. The two resources embody different philosophies of knowledge representation. Text2KGBench's text-extraction approach captures a broad but sparse relation vocabulary (663 unique relations, averaging 1.43 per entity), whereas KastorKG maintains a curated, denser structure (152 relations, 2.49 per entity) constrained by SHACL shapes. Per-class property breakdowns, detailed in Appendix C, further illustrate this contrast: KastorKG consistently favours standardised temporal and spatial attributes (e.g., 81.3% of artists have `dbo:birthDate` in KastorKG vs. only 7.8% in Text2KGBench), whilst Text2KGBench exhibits more varied relation distributions. This density is both

[8] `https://zenodo.org/records/18862240`
[9] as detailed in our documentation

an asset—providing richer training signals—and a challenge: models must learn to identify the small subset of sentences relevant to each SHACL property among many that are not. Both datasets must also contend with long-tail property distributions.

6.4 Kastor SLMs evaluation

This section reports the empirical evaluation of the 16 shape-aware Kastor SLMs, addressing **RQ3a** (generalisation across SHACL shapes within the Wikipedia training domain) and **RQ3b** (cross-domain transfer to the Text2KGBench/WebNLG evaluation domain).

6.4.1 Fine-tuning details

For each shape $s \in \mathbb{S}$, a random subset D_s is sampled as training data.

$$\begin{aligned} &\forall s \in \mathbb{S}, \\ D_s := \{(w, w_{MD}, g) &\in \mathcal{K}_{dist}(s)\} \end{aligned} \tag{18}$$

Each dataset D_s comprises 1,000 examples used with 10-fold cross-validation to fine-tune a BART model, yielding a specialised Kastor extractor $\mathcal{M}_s$. The 16 trained model checkpoints are publicly available on HuggingFace[10] together with a usage tutorial[11].

6.4.2 Metrics

Performance is evaluated using both classical and pattern-based indicators. Classical metrics include the rate of successfully parsed graphs (r_{ttl}), micro-level precision, recall, and F_1^-, as well as macro-level F_1^+. The pattern metrics defined in Section 3 are also reported: $r_{\hat{\mathbb{G}}_D^{\leftrightarrow}}$ (Eq. 4), the rate of graphs exactly matching the expected property pattern; and PEC_D (Eq. 5), measuring the tendency of deviating graphs to over-generate rather than under-generate triples.

6.4.3 Evaluation on the KastorKG test sets

Global results. Table 11 summarises the 10-fold averaged results. All models generated syntactically valid graphs ($r_{ttl} = 1.00$), with the sole exception of $\mathcal{M}D_{\text{MusicalWork}}$ ($r_{ttl} = 0.98$), whose rare failures stem from complex URL-encoding issues. Every model achieves a robust average F_1^-; per-fold variance can be substantial, but standard deviations remain below ten points for most shapes. The pattern-conformance rate ($r_{\hat{\mathbb{G}}^{\leftrightarrow}}$) confirms that models adhere closely to the predefined SHACL structures, and when deviations occur the Pattern Extension Capacity (PEC) shows that extractors tend to under-generate rather than hallucinate superfluous triples.

The highest F_1^+ belongs to $\mathcal{M}D_{\text{CelestialBody}}$, whose target shape has only four distinct example-specific patterns ($\pi_{\text{CelestialBody}}$)—the smallest inventory of any shape. Conversely, $\mathcal{M}D_{\text{Film}}$ achieves the lowest F_1^+, associated with 188 distinct patterns in $\mathcal{K}_{\text{dist}}$. This inverse relationship confirms that structural variability is the primary driver of

[10] https://huggingface.co/Datartisan
[11] Kastor HuggingFace Models Tutorial

macro-F_1 variance. $\mathcal{MD}_{\text{City}}$ stands out with near-perfect scores across all metrics; eight further models ($\mathcal{MD}_{\text{Artist}}$, $\mathcal{MD}_{\text{Athlete}}$, $\mathcal{MD}_{\text{CelestialBody}}$, $\mathcal{MD}_{\text{Politician}}$, $\mathcal{MD}_{\text{Scientist}}$, $\mathcal{MD}_{\text{SportsTeam}}$, $\mathcal{MD}_{\text{University}}$, and $\mathcal{MD}_{\text{WrittenWork}}$) achieve $F_1^- > 0.90$. The consistent gap between F_1^+ and F_1^- indicates that performance is concentrated on frequent properties, motivating the long-tail analysis in Section 7.

Table 11: Performance of the 16 shape-based extractors (10-fold average). r_{ttl}: rate of syntactically valid Turtle graphs. Bold = best; underline = worst.

Model	r_{ttl}	F_1^+	F_1^-	Recall	Prec.	$r_{\hat{\mathbb{G}}\leftrightarrow}$	PEC
$\mathcal{MD}_{\text{Airport}}$	1.00	81 ± 25	92 ± 10	95.71	95.26	0.91	0.42
$\mathcal{MD}_{\text{Artist}}$	1.00	74 ± 21	91 ± 6	93.25	94.00	0.84	0.36
$\mathcal{MD}_{\text{Athlete}}$	1.00	54 ± 25	90 ± 4	87.36	95.14	0.84	0.16
$\mathcal{MD}_{\text{Building}}$	1.00	75 ± 21	87 ± 11	93.66	89.92	0.89	0.27
$\mathcal{MD}_{\text{CelestialBody}}$	1.00	$\mathbf{95 \pm 3}$	97 ± 1	98.21	97.00	0.96	0.37
$\mathcal{MD}_{\text{City}}$	1.00	80 ± 22	$\mathbf{99 \pm 0}$	**99.61**	**99.90**	**1.00**	$\underline{0.00}$
$\mathcal{MD}_{\text{Company}}$	1.00	76 ± 26	82 ± 18	87.68	87.28	0.85	0.15
$\mathcal{MD}_{\text{Film}}$	1.00	$\underline{51 \pm 19}$	$\underline{76 \pm 9}$	$\underline{76.21}$	$\underline{82.11}$	$\underline{0.57}$	0.30
$\mathcal{MD}_{\text{Food}}$	1.00	77 ± 15	82 ± 13	87.24	85.47	0.81	0.25
$\mathcal{MD}_{\text{MeansOfTransport}}$	1.00	72 ± 15	87 ± 12	93.24	87.71	0.85	**0.46**
$\mathcal{MD}_{\text{MusicalWork}}$	$\underline{0.98}$	74 ± 18	85 ± 11	92.10	86.81	0.73	0.50
$\mathcal{MD}_{\text{Politician}}$	1.00	74 ± 18	91 ± 7	93.31	93.34	0.84	0.40
$\mathcal{MD}_{\text{Scientist}}$	1.00	64 ± 26	90 ± 6	91.64	91.74	0.78	0.41
$\mathcal{MD}_{\text{SportsTeam}}$	1.00	84 ± 17	92 ± 7	96.54	93.23	0.92	0.46
$\mathcal{MD}_{\text{University}}$	1.00	75 ± 25	85 ± 14	93.22	91.19	0.84	0.41
$\mathcal{MD}_{\text{WrittenWork}}$	1.00	82 ± 20	92 ± 8	94.00	94.38	0.91	0.28

Object property analysis. Table 12 reports the object property metrics (Section 5) for each extractor on its KastorKG test set.

The $\mathcal{MD}_{\text{City}}$ row is almost entirely zero: there are no false positives and the AlignScore for false negatives is low, confirming that omitted triples are rarely supported by the text. False positives from $\mathcal{MD}_{\text{Building}}$ have notably high AlignScores, suggesting entailment by the input; most other models cluster near 0.5. For FP_{new} (predicted properties absent from the ground truth), $\mathcal{MD}_{\text{Food}}$ achieves the highest URI-validity rate, and $\mathcal{MD}_{\text{Artist}}$ and $\mathcal{MD}_{\text{University}}$ generate objects with strong semantic alignment (Sem_{sim}). Regarding the wrong-relation rate (WRC), $\mathcal{MD}_{\text{Company}}$, $\mathcal{MD}_{\text{Film}}$, $\mathcal{MD}_{\text{Food}}$, $\mathcal{MD}_{\text{Politician}}$, and $\mathcal{MD}_{\text{SportsTeam}}$ show the most predicate confusion; the low Sem_{sim} of $\mathcal{MD}_{\text{MeansOfTransport}}$ may reflect imprecise range definitions in its SHACL shape. For FP_{diff} cases (predicted objects differing from the ground truth), $\mathcal{MD}_{\text{Building}}$ and $\mathcal{MD}_{\text{SportsTeam}}$ stand out by generating existing KG URIs that are semantically close to the expected object.

6.4.4 Evaluation on the Text2KGBench Test Sets

Domain-shift characterisation. The Text2KGBench evaluation uses WebNLG-derived texts, which differ structurally from the Wikipedia abstracts used to train the Kastor models. To characterise this shift, three comparative metrics are introduced

Table 12: Object property metrics (KastorKG test sets, 10-fold average). Bold = best; underline = worst.

Model	$AS(w, FP)$	$r_{\hat{\mathcal{O}}^+(FP_{new})}$	$Sem_{sim}(\hat{g}, s^*)$	$r_{\hat{\mathcal{O}}^+(FP_{diff})}$	$Sem_{sim}(\hat{g}, g)$	$AS(w, FN)$	$WRC(FP_{new})$
$\mathcal{MD}_{\text{Airport}}$	0.48	0.45	0.69	0.38	0.40	0.19	0.53%
$\mathcal{MD}_{\text{Artist}}$	0.58	0.71	0.90	0.41	0.60	0.27	0.00%
$\mathcal{MD}_{\text{Athlete}}$	0.33	0.32	0.60	0.40	0.40	0.43	0.36%
$\mathcal{MD}_{\text{Building}}$	**0.79**	0.41	0.90	**0.83**	**0.90**	$\underline{0.59}$	0.44%
$\mathcal{MD}_{\text{CelestialBody}}$	0.15	0.25	0.30	$\underline{0.00}$	$\underline{0.00}$	0.04	**0.00%**
$\mathcal{MD}_{\text{City}}$	$\underline{0.00}$	$\underline{0.00}$	$\underline{0.00}$	$\underline{0.00}$	$\underline{0.00}$	0.25	**0.00%**
$\mathcal{MD}_{\text{Company}}$	0.50	0.58	0.79	0.42	0.50	0.46	$\underline{7.93\%}$
$\mathcal{MD}_{\text{Film}}$	0.36	0.54	**0.97**	0.63	0.88	0.39	1.68%
$\mathcal{MD}_{\text{Food}}$	0.35	**0.84**	0.89	0.60	0.69	0.27	5.61%
$\mathcal{MD}_{\text{MeansOfTransport}}$	0.31	0.25	0.48	0.30	0.48	0.33	0.17%
$\mathcal{MD}_{\text{MusicalWork}}$	0.29	0.57	**0.96**	0.58	0.80	0.22	0.51%
$\mathcal{MD}_{\text{Politician}}$	0.43	0.53	0.90	0.45	0.50	0.13	0.55%
$\mathcal{MD}_{\text{Scientist}}$	0.48	0.69	0.86	0.55	0.58	0.32	3.36%
$\mathcal{MD}_{\text{SportsTeam}}$	0.55	0.57	0.87	0.77	0.80	0.42	6.84%
$\mathcal{MD}_{\text{University}}$	0.47	0.78	0.86	0.12	0.20	0.28	1.13%
$\mathcal{MD}_{\text{WrittenWork}}$	0.27	0.27	0.46	0.14	0.15	0.24	0.00%

per class s. (1) Average graph size ($|G_s|$, $|G_s^{NLG}|$) and average text length ($\overline{len(W_s)}$, $\overline{len(W_s^{NLG})}$). (2) Jaccard similarity of the property sets and example-specific pattern sets observed in each corpus:

$$J_{sim}(\mathcal{P}(s)) = \frac{|\mathcal{P}_G(s) \cap \mathcal{P}_{G^{NLG}}(s)|}{|\mathcal{P}_G(s) \cup \mathcal{P}_{G^{NLG}}(s)|}, \quad J_{sim}(\mathbb{P}(s)) = \frac{|\mathbb{P}_G(s) \cap \mathbb{P}_{G^{NLG}}(s)|}{|\mathbb{P}_G(s) \cup \mathbb{P}_{G^{NLG}}(s)|} \tag{19}$$

(3) Sentence-BERT cosine similarity between the average text embeddings of each corpus:

$$\widehat{Sim}_{\text{sBERT}}(W_s, W_s^{NLG}) = \frac{\text{Embed}(W_s) \cdot \text{Embed}(W_s^{NLG})}{\|\text{Embed}(W_s)\| \cdot \|\text{Embed}(W_s^{NLG})\|} \tag{20}$$

Wikipedia abstracts are consistently longer than WebNLG texts (by a factor of 3–5×), and their associated graphs contain slightly more properties on average. The Jaccard similarity of *property sets* ($J_{sim}(\mathcal{P}(s))$) is generally high (∼70% shared), indicating that most properties seen at training time also appear in WebNLG. The Jaccard similarity of *example-specific patterns* ($J_{sim}(\mathbb{P}(s))$) is, however, much lower (5–36%), meaning models must generalise to new property combinations at test time. This explains much of the performance drop on classes such as `dbo:Politician` ($J_{sim}(\mathbb{P}) = 0.05$) and `dbo:Athlete` ($J_{sim}(\mathbb{P}) = 0.09$). Conversely, `dbo:City` ($J_{sim}(\mathbb{P}) = 0.25$) and `dbo:University` ($J_{sim}(\mathbb{P}) = 0.36$) show the highest pattern overlap, correlating with their stronger WebNLG results.

Table 13: Comparison of KastorKG test sets and WebNLG sets per class. Bold/underline = best/worst.

Shape s	$\lvert G_s^{NLG}\rvert$	$\lvert G_s^{W}\rvert$	$\overline{len}(\text{NLG})$	$\overline{len}(W)$	$J_{sim}(\mathcal{P}(s))$	$J_{sim}(\mathbb{P}(s))$	$\widehat{Sim}_{\text{sBERT}}$
Airport	2.41	$\underline{2.12}$	117	488	0.60	0.10	0.74
Artist	2.73	2.62	111	327	0.67	0.08	0.63
Athlete	2.42	2.59	109	324	0.60	0.09	0.71
Building	2.22	2.58	115	454	0.75	0.31	0.66
CelestialBody	2.22	2.19	139	473	0.75	0.33	0.66
City	$\underline{2.01}$	2.01	117	501	0.60	0.25	0.70
Company	2.14	2.35	$\underline{97}$	494	$\underline{0.33}$	0.07	0.67
Film	2.70	3.10	130	343	0.82	0.18	0.59
Food	2.21	2.31	127	440	0.75	0.16	0.72
MeansOfTransport	2.62	2.33	110	476	0.47	0.14	0.67
MusicalWork	2.90	2.83	150	277	0.80	0.18	**0.77**
Politician	2.82	3.09	128	482	0.40	$\underline{0.05}$	0.57
Scientist	**2.92**	3.06	138	$\underline{208}$	0.67	0.17	0.64
SportsTeam	2.21	2.76	185	367	**0.86**	0.12	0.68
University	2.54	**3.16**	**254**	**510**	0.67	**0.36**	0.70
WrittenWork	2.24	2.48	127	407	0.54	0.17	$\underline{0.55}$

Comparison with LLM baselines. The Text2KGBench baselines are Vicuna-13B and Alpaca-LoRA-13B ($> 100\times$ more parameters than the BART-base models), evaluated via 4-shot in-context learning with an ontology verbalisation and an illustrative graph–text example as context. To enable a fair comparison, an RDF evaluation set was built from the WebNLG portion of Text2KGBench, restricted to triples whose subject is the main described entity and whose relations are instantiated in DBpedia (the only relations Kastor can produce). Since the Text2KGBench authors did not penalise out-of-ontology relations when computing F_1^-, both evaluation contexts are reported in Table 14; recomputing their metrics on the restricted subset yielded no significant differences. Hallucination metrics from Text2KGBench show that 12–16% of LLM-generated subjects and 7–9% of relations are hallucinated, whereas Kastor models produce none.

The Kastor models outperform the LLM baselines on 11 out of 16 classes (global F_1^-: 0.44 vs. 0.30 for Alpaca and 0.25 for Vicuna). Classes where Kastor leads—`dbo:Airport`, `dbo:Artist`, `dbo:City`, `dbo:Food`, `dbo:MusicalWork`, `dbo:University`, and `dbo:WrittenWork`—all exhibit higher pattern overlap ($J_{sim}(\mathbb{P}) \geq 0.16$) and stronger sBERT alignment, as reported in Table 13. Classes where LLMs prevail—`dbo:Athlete`, `dbo:Company`, and `dbo:Politician`—have the lowest example-specific pattern overlap ($J_{sim}(\mathbb{P}) \leq 0.09$), so the domain shift penalises the Kastor models disproportionately.

Two classes stand out as counterexamples. `dbo:CelestialBody` and `dbo:SportsTeam` both achieve excellent F_1^- on KastorKG ($\geq$0.92) yet underperform markedly on WebNLG (0.39 and 0.04, respectively). For `dbo:SportsTeam`, despite the highest property-set overlap ($J_{sim}(\mathcal{P}) = 0.86$), the example-specific pattern overlap is low ($J_{sim}(\mathbb{P}) = 0.12$): almost all properties are shared, but the *combinations* in WebNLG were rarely seen during Wikipedia training, and properties such as founding dates and venue names are typically expressed in Wikipedia infoboxes—text the Kastor models never saw. A similar mechanism applies to `dbo:CelestialBody` ($J_{sim}(\mathbb{P}) = 0.33$,

Table 14: Text2KGBench comparison: Alpaca-13B, Vicuna-13B, Kastor (139 M). Bold = best; underline = worst per row.

	Alpaca-13B			Vicuna-13B			KastorKG (139 M)		
Class	F_1^-	Prec.	Rec.	F_1^-	Prec.	Rec.	F_1^-	Prec.	Rec.
Airport	0.20	0.28	0.18	0.27	**0.33**	0.24	**0.43**±5	0.29±5	**0.85**±10
Artist	0.26	0.35	0.22	0.23	0.30	0.21	**0.54**±3	**0.62**±5	**0.48**±3
Athlete	**0.32**	**0.38**	**0.30**	0.29	0.33	0.26	0.27±4	0.29±5	0.26±5
Building	0.29	0.35	0.27	0.38	**0.48**	0.33	**0.52**±4	**0.51**±5	**0.53**±6
CelestialBody	**0.47**	**0.52**	**0.44**	0.46	0.48	0.46	0.39±6	0.41±8	0.38±7
City	0.09	0.10	0.09	0.12	0.12	0.12	**0.56**±38	**0.48**±36	**0.69**±47
Company	0.25	0.35	0.21	**0.35**	**0.49**	**0.37**	0.19±4	0.19±4	0.21±5
Film	0.15	0.18	0.14	0.20	0.23	0.19	**0.42**±5	**0.53**±9	**0.35**±5
Food	0.31	0.38	0.30	0.39	0.43	0.39	**0.88**±2	**0.94**±3	**0.83**±6
MeansOfTransport	0.10	0.13	0.09	0.18	0.22	0.17	**0.38**±7	**0.36**±9	**0.44**±8
MusicalWork	0.15	0.18	0.15	0.18	0.20	0.18	**0.44**±6	**0.48**±6	**0.42**±10
Politician	0.30	**0.39**	0.27	**0.32**	**0.39**	**0.28**	0.26±4	0.32±6	0.22±3
Scientist	0.34	0.42	0.33	0.46	0.52	0.43	**0.59**±5	**0.78**±3	**0.48**±6
SportsTeam	0.31	0.41	0.28	**0.42**	**0.52**	**0.91**	0.04±2	0.04±3	0.03±2
University	0.20	0.29	0.16	0.23	0.31	0.19	**0.63**±3	**0.71**±6	**0.59**±6
WrittenWork	0.36	0.39	0.35	0.36	0.40	0.34	**0.58**±3	**0.70**±6	**0.51**±2
Global	0.25	0.31	0.23	0.30	0.35	0.31	**0.44**	**0.47**	**0.45**

$\widehat{Sim}_{\text{sBERT}} = 0.66$), compounded by its minimal pattern inventory (four distinct patterns in $\mathcal{K}_{\text{dist}}$). These cases demonstrate that *property-set overlap alone is insufficient to predict transferability*: example-specific pattern overlap and register alignment are stronger predictors.

Object property analysis (WebNLG). Table 15 extends the semantic error metrics of Section 5 to the WebNLG setting. Pattern-conformance ($r_{\hat{\mathbb{G}}^{\leftrightarrow}_{D}}$) and PEC are both lower than in the Wikipedia experiments, confirming domain shift; $\mathcal{MD}_{\text{Airport}}$ and $\mathcal{MD}_{\text{City}}$ are the only models retaining $r_{\hat{\mathbb{G}}^{\leftrightarrow}_{D}} > 0.87$. The low PEC values indicate systematic under-generation, with the exception of $\mathcal{MD}_{\text{City}}$ ($PEC = 0.87$). False-positive AlignScores are comparable to the Wikipedia setting, and the rate of predicted URIs matching existing DBpedia entities is even higher, suggesting that models still produce plausible objects under domain shift. By contrast, false-negative AlignScores are higher in WebNLG than in Wikipedia: the omitted triples are frequently supported by the evaluation text, revealing that the models fail to extract properties expressed in infobox-style language absent from their Wikipedia training. This shortcoming is further reflected in the wrong-relation rate (WRC), which is substantially elevated for several models—most strikingly `dbo:Film` (44% wrong assignments).

6.4.5 Synthesis: cross-domain transfer

Three structural findings consolidate the results above.

Finding 1 — Domain shift reduces pattern conformance and coverage. Both $r_{\hat{\mathbb{G}}^{\leftrightarrow}_{D}}$ and PEC are lower in WebNLG than in the Wikipedia setting: models trained on longer encyclopaedic texts produce smaller, less pattern-conformant graphs

Table 15: Object property metrics (WebNLG test sets, 10-fold average). Bold = best; underline = worst.

Model	$r_{\hat{\mathbb{G}}_D^{\updownarrow}}$	PEC	$AS(w, FP)$	$r_{\hat{\mathcal{O}}^{+}(FP_{new})}$	$Sem_{sim}(\hat{g}, s^*)$	$r_{\hat{\mathcal{O}}^{+}(FP_{diff})}$	$Sem_{sim}(\hat{g}, g)$	$AS(w, FN)$	$WRC(FP_{new})$
$\mathcal{MD}_{\text{Airport}}$	0.87	0.44	0.68	0.65	**0.99**	0.59	0.79	0.79	5.00%
$\mathcal{MD}_{\text{Artist}}$	0.45	0.02	0.54	0.68	0.97	0.46	0.72	0.69	13.31%
$\mathcal{MD}_{\text{Athlete}}$	0.21	0.03	0.20	0.63	0.84	0.87	0.42	0.79	7.23%
$\mathcal{MD}_{\text{Building}}$	0.32	0.13	**0.76**	0.90	0.97	0.78	0.58	0.74	11.00%
$\mathcal{MD}_{\text{CelestialBody}}$	0.25	0.10	<u>0.09</u>	0.07	<u>0.50</u>	0.40	0.40	0.59	21.75%
$\mathcal{MD}_{\text{City}}$	**0.90**	**0.87**	0.38	0.90	0.84	0.48	**0.90**	<u>0.90</u>	**0.00%**
$\mathcal{MD}_{\text{Company}}$	0.39	<u>0.00</u>	0.59	0.81	0.98	0.84	0.74	0.81	9.09%
$\mathcal{MD}_{\text{Film}}$	0.17	0.03	0.39	0.74	**0.99**	0.87	0.31	0.77	<u>44.75%</u>
$\mathcal{MD}_{\text{Food}}$	0.75	0.11	0.57	**0.95**	**0.99**	0.72	0.40	**0.53**	10.46%
$\mathcal{MD}_{\text{MeansOfTransport}}$	0.33	0.19	0.36	0.69	0.83	0.58	0.22	0.86	11.16%
$\mathcal{MD}_{\text{MusicalWork}}$	0.24	0.11	0.43	<u>0.36</u>	0.77	0.26	0.28	0.66	10.06%
$\mathcal{MD}_{\text{Politician}}$	<u>0.09</u>	0.02	0.47	0.55	0.86	0.39	0.59	0.75	3.30%
$\mathcal{MD}_{\text{Scientist}}$	0.35	0.01	0.30	0.86	0.98	0.88	0.89	0.78	29.99%
$\mathcal{MD}_{\text{SportsTeam}}$	0.16	0.05	0.37	0.44	0.89	<u>0.21</u>	<u>0.10</u>	0.77	12.82%
$\mathcal{MD}_{\text{University}}$	0.21	0.05	0.33	0.97	0.94	0.97	**1.00**	0.71	32.20%
$\mathcal{MD}_{\text{WrittenWork}}$	0.36	0.01	0.32	0.54	**0.90**	0.89	0.13	0.76	5.91%

on shorter, structured descriptions. High false-negative AlignScores in WebNLG confirm that many omitted triples *are* textually supported, but the corresponding property combinations were absent from the Wikipedia training distribution (low $J_{sim}(\mathbb{P}(s))$; see Table 13).

Finding 2 — Register mismatch drives class-specific failures. Classes whose WebNLG descriptions rely on infobox-style language (founding dates, venue names, orbital parameters) suffer the largest performance drops, because Kastor models were exposed only to Wikipedia running prose. Property-set Jaccard alone is a poor predictor: `dbo:SportsTeam` has the highest $J_{sim}(\mathcal{P}) = 0.86$ yet the lowest F_1^- on WebNLG (0.04), confirming that *example-specific pattern overlap* and *register alignment* are stronger transfer predictors.

Finding 3 — Ontological constraints make the task harder than in-domain benchmarks. BERT-base joint extraction systems such as TDEER [Li et al., 2021], CasRel, TPLinker, and SPN4RE—trained and tested on the standard WebNLG split—report micro-F_1 in the 0.52–0.86 range. These figures are not directly comparable for three cumulative reasons:

1. **Flat relation labels vs. DBpedia URIs.** Standard systems predict short surface labels that match their training vocabulary. KastorKG must produce full URIs (e.g., `dbo:birthPlace` with `dbr:Paris`), a stricter target where any string mismatch is penalised.
2. **Closed vs. ontologically constrained evaluation.** In-domain systems are

never penalised for relations outside their training set. Kastor is constrained to triples expressible within the SHACL-specified DBpedia schema and is penalised for any relation present in the text but absent from the ontology.

3. **In-domain vs. zero-shot cross-domain transfer.** BERT-based systems train directly on WebNLG text; Kastor trains on Wikipedia abstracts and evaluates zero-shot on WebNLG, systematically reducing recall for infobox-style properties not seen during training.

These constraints compound: a global F_1^- of 0.44 on WebNLG with zero hallucinations reflects the difficulty of the ontological setting, not a limitation of the model architecture. **Summary (RQ3a–RQ3b).** Within the Wikipedia domain (**RQ3a**), shape-aware SLMs generalise well across 16 SHACL shapes (average $F_1^- = 0.87$), with performance governed primarily by the number of distinct example-specific patterns in $\mathcal{K}_{\text{dist}}$. Across domains (**RQ3b**), compact 139 M-parameter Kastor models outperform >13 B-parameter LLM baselines on 11 of 16 classes with zero hallucination; and transferability is best predicted by example-specific pattern overlap ($J_{sim}(\mathbb{P}(s))$) and register similarity ($\widehat{Sim}_{\text{sBERT}}$) rather than property-set overlap alone.

7 Addressing the Long-Tail Property Distribution

The Text2KGBench experiments revealed persistent gaps between F_1^+ and F_1^- caused by the unbalanced distribution of properties. Several strategies are investigated to address this for the well-studied `dbo:Person` shape.

7.1 Scale, Stratification, and Custom Loss

7.1.1 Training at Scale and Oversampling

Models are trained on three scaled datasets: $D2$ (2,400 examples), $D4$ (4,800 examples), $D10$ (12,000 examples), and a rare-property-focused sample D_- (requiring at least one property from s^*_- per example).

7.1.2 Stratified Sampling

To ensure rare properties appear in every fold, training examples are stratified by rare property. Each graph is assigned to the stratum of its least-represented rare property. Algorithm 1 formalises this procedure.

Algorithm 1: Stratification pseudo-algorithm

```
Data: shape $s^*$, sample size $N$, knowledge base $\mathcal{K}$
Result: stratified sample $D_c$
1 begin
2     Initialise $P_{no} = \{(p_i, 0)\}_{i=1}^{n}$ from $s^*$;
3     Sample $D$ of size $N$ from $\mathcal{K}$;
4     Count occurrences: $P_{no}[p_j]$ += 1 for each $p_j \in g_i$, $(g_i, t_i) \in D$;
5     Compute mean freq. $\mu_p$;
6     Build rare set $s_- = \{p_i \mid \text{freq}(p_i) < \mu_p\}$;
7     Assign each $(g, t)$ to stratum of least-represented rare property in $g$;
```

7.1.3 Custom Loss

Beyond standard cross-entropy (CE), the following losses are evaluated:

$$Loss_{WCE}(y, \hat{y}) = \omega_y \cdot Loss_{CE}(y, \hat{y}), \quad \omega_y = \log\left(\frac{\sum_i |c_i|}{|c_y|}\right) \tag{21}$$

$$Loss_{WF}(y, \hat{y}) = -\omega_y (1 - p_t)^{\gamma} \log p_t, \quad \gamma = 2 \tag{22}$$

where c_y is the stratum of sequence y and ω_y is an inverse log-frequency weight.

7.2 Cross-Evaluation Setup

Four independent 200-example test sets are used: d_N (unseen post-2021 articles), d_+ (frequent properties only), d_- (at least one rare property), d_R (random).

7.3 Results: Scale and Stratification

Table 16 shows the number of triples per property in each dataset. $D4SE_{all}$ is the only configuration in which every property—including the rarest (`dbo:alias`: only 156 occurrences in the seed set)—reaches or exceeds the 1,000-example threshold.

Table 16: Number of triples per property across augmented datasets.

		$D10$	$D10A_{all}$	$D10A_{KR0}$	$D10A_{KR1}$	$D4SE_{all}$
s_+^*	birthYear	10,508	11,396	11,296	11,625	2,718
	birthPlace	9,887	9,948	10,757	10,913	1,115
	birthDate	9,553	10,387	10,305	10,450	2,361
	label	8,211	9,158	8,989	9,277	2,321
	deathYear	4,804	5,185	4,962	5,570	1,605
	deathDate	4,237	4,544	4,387	4,877	1,353
s_-^*	deathPlace	3,181	3,197	3,279	3,777	1,024
	nationality	1,383	1,401	1,385	1,585	1,001
	birthName	1,222	1,434	1,296	1,728	1,001
	alias	$\underline{156}$	1,398	1,082	1,387	1,001

Table 17 shows that neither stratification nor weighted loss provides significant gains. Scaling the training data to $D10$ yields nearly a ten-point improvement on d_- for F_1^+ (77.3 vs. 59.2). All configurations perform well on d_+; performance on d_N confirms the absence of any memorisation effect.

The Focal Loss variant ($Loss_{WF}$, Eq. 22) was also evaluated but did not improve over the Weighted Cross-Entropy ($Loss_{WCE}$) baseline and is therefore omitted from Table 17 for brevity.

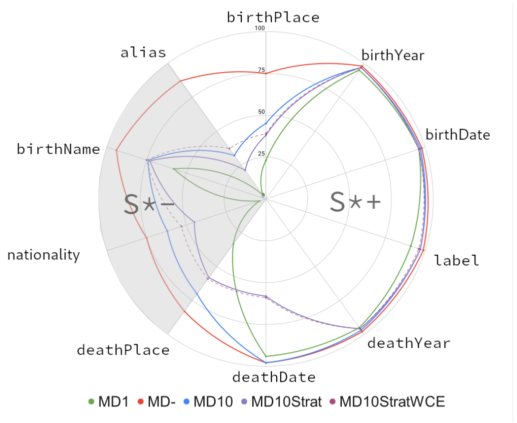
Figure 4 shows that scaling ($\mathcal{M}D10$) benefits rare properties but cannot match $\mathcal{M}D_-$, which was trained specifically on rare examples.

7.4 Sufficient-Exposure Sampling

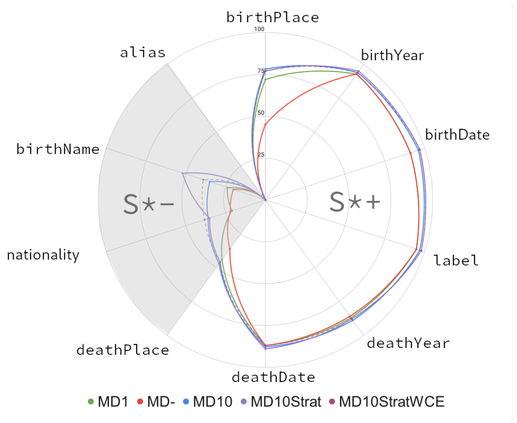
The `dbo:alias` property (only 1,398 occurrences in the full KG) consistently underperforms; three data augmentation strategies are investigated.

Table 17: F_1^- and F_1^+ across cross-evaluation sets. Std. dev. always < 5 pt.

Model	Config	F_1^- d_N	F_1^- d_+	F_1^- d_-	F_1^- d_R	F_1^+ d_N	F_1^+ d_+	F_1^+ d_-	F_1^+ d_R
$\mathcal{M}D1$	CE (10f)	82.3	91.7	<u>76.9</u>	83.9	63.1	96.7	<u>59.2</u>	62.0
$\mathcal{M}D_-$	CE (10f)	<u>72.7</u>	<u>76.8</u>	**95.1**	<u>75.5</u>	<u>57.5</u>	<u>84.9</u>	**90.7**	<u>58.1</u>
$\mathcal{M}D2$	CE (10f)	82.1	92.0	77.6	84.6	62.9	96.7	61.1	62.7
$\mathcal{M}D4$	CE (10f)	84.2	91.9	79.2	86.4	64.2	**97.1**	63.3	65.7
$\mathcal{M}D10$	CE (10f)	**84.4**	**91.9**	86.1	**86.8**	**67.1**	97.1	**77.3**	66.5
$\mathcal{M}D1_{Strat}$	CE-STRAT (5f)	81.4	91.7	75.7	84.5	60.1	94.8	56.5	61.3
$\mathcal{M}D10_{Strat}$	CE-STRAT (5f)	83.9	90.1	83.6	86.5	65.8	95.3	71.9	**67.8**
$\mathcal{M}D10_{StratWCE}$	WCE-STRAT (5f)	84.2	91.3	84.5	86.5	66.0	96.2	74.9	66.5



(a) Evaluated on d_- (rare properties).



(b) Evaluated on d_R (random).

Figure 4: Averaged micro-F1 by property and model (Steps 1 cross-eval).

All available data ($D10A_{all}$). All entities with `dbo:alias` from the distilled KG are added, yielding 13,244 entities.

Abstract-Template Knowledge Replacement. Starting from 156 seed entities with `dbo:alias`, we synthesise new text-graph pairs by (i) replacing property values in a template abstract with values from another entity sharing the same pattern (KR0, Algorithm 2, Fig. 5) or (ii) filling the template with values from an entity following a larger pattern (KR1, Algorithm 3, Fig. 6).

Algorithm 2: KR0 — Same-pattern knowledge replacement

```
Data: seed set $Seed_{data}$; target count $N$; base $\mathcal{K}$
Result: synthetic set $Synth_{data}$
1 begin
2     Count realised patterns in $Seed_{data}$; keep patterns with $\geq 2$ instances;
3     while $|Synth_{data}| < N$ do
4         Pick random pattern $Patt_p$; select entities $e_i \neq e_j$ both following $Patt_p$;
5         Create template $WT_i$ from $WMD_i$ by masking values of $g_i$;
6         Fill $WT_i$ with values from $g_j \to WKR0_{(i,j)}$; add to $Synth_{data}$;
7         Repeat symmetrically for $(j, i)$;
```

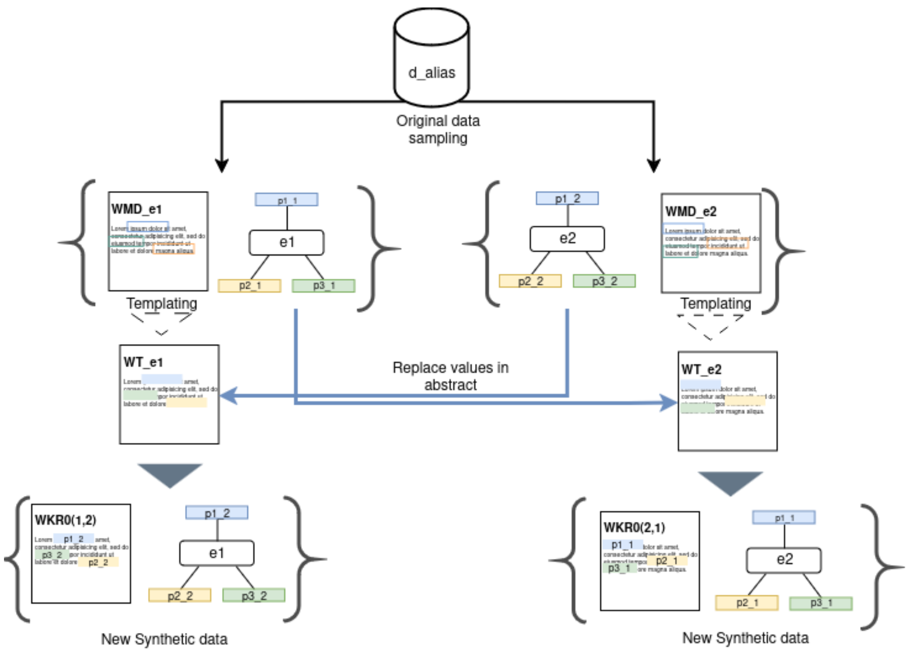


Figure 5: KR0 — Same-pattern knowledge replacement.

Algorithm 3: KR1 — Upper-pattern knowledge replacement

```
Data: seed set $Seed_{data}$; target count $N$; base $\mathcal{K}$
Result: synthetic set $Synth_{data}$
1 begin
2     while $|Synth_{data}| < N$ do
3         Pick pattern $Patt_{p1}$; select entity $e_i$ following $Patt_{p1}$;
4         Pick subpattern $Patt_{p2} \subset Patt_{p1}$; select entity $e_j$ following $Patt_{p2}$;
5         Mask $g_j$-values in $WMD_j$; fill with $g_i$-values $\rightarrow WKR1_{(i,j)}$;
6         Restrict $g_i$ to properties in $g_j \rightarrow \hat{g}_i$;
7         Add $(e_i, e_j, \hat{g}_i, WKR1_{(i,j)})$ to $Synth_{data}$;
```

The two strategies differ in the diversity of the generated examples. KR0 operates *within* a single pattern: both the template entity and the replacement entity follow the same property combination. All synthetic examples therefore remain within the pattern space already seen during training, limiting the variety of exposures introduced. KR1 uses a *richer* entity (one following a larger pattern) to fill the template of a *simpler* entity, thereby exposing the model to a wider range of property combinations and better approximating the true data distribution. This explains why $\mathcal{M}D10A_{KR0}$ consistently underperforms $\mathcal{M}D10A_{KR1}$ across all test sets in Table 18.

Sufficient-Exposure Dataset ($D4SE_{all}$). Algorithm 4 builds a balanced sample guaranteeing $\lambda = 1{,}000$ examples for every property, yielding 3,472 entities.

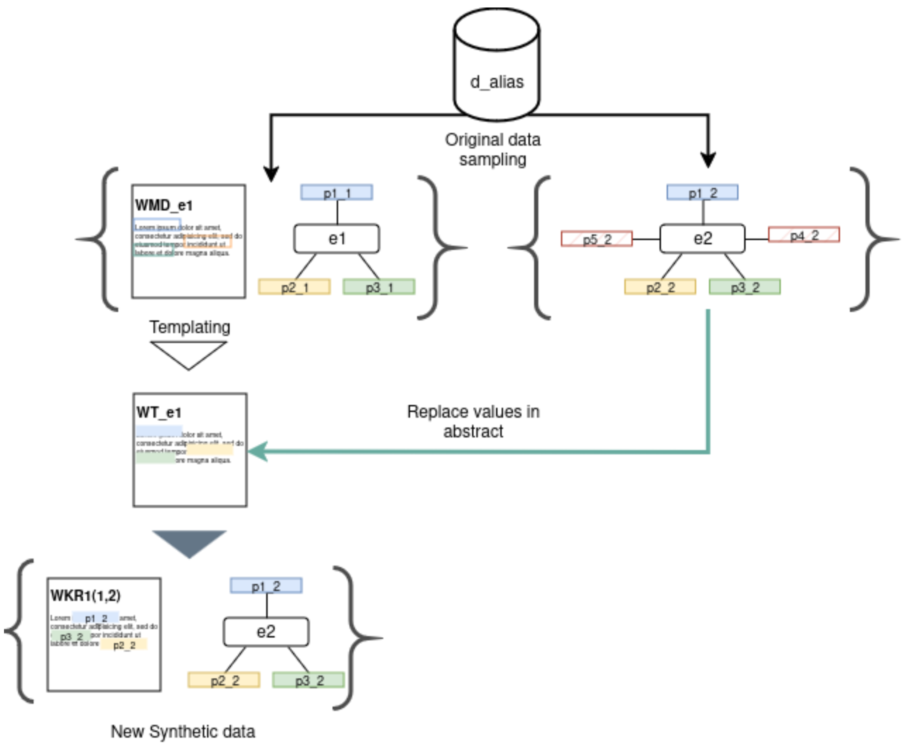


Figure 6: KR1 — Upper-pattern knowledge replacement.

Algorithm 4: Sufficient-Exposure Sampling

Data: shape s^*; threshold $\lambda = 1000$; base $\mathcal{K}$
Result: sample D_{SE}

```
begin
    Sort properties of $s^*$ by ascending frequency in $\mathcal{K}$;
    foreach property $p_i$ (least to most frequent) do
        while count($p_i$, $D_{SE}$) $< \lambda$ do
            Sample random $(g, t)$ with $p_i \in g$ and $(g, t) \notin D_{SE}$;
            Add $(g, t)$ to $D_{SE}$; update all property counts;
            Stop if all properties exceed $\lambda$;
```

7.5 Results: Sufficient Exposure

Evaluation covers the five cross-evaluation sets, including a corrected set d_C (manually annotated 1,470 FP and FN to remove KG noise). Table 18 shows that $\mathcal{MD}4SE_{all}$ outperforms all other configurations by 10 F_1 points on d_C, demonstrating the most balanced performance across properties. Figure 7 confirms that it is the only configuration capable of extracting `dbo:alias` on d_R, and the one with the fewest genuine errors on d_C.

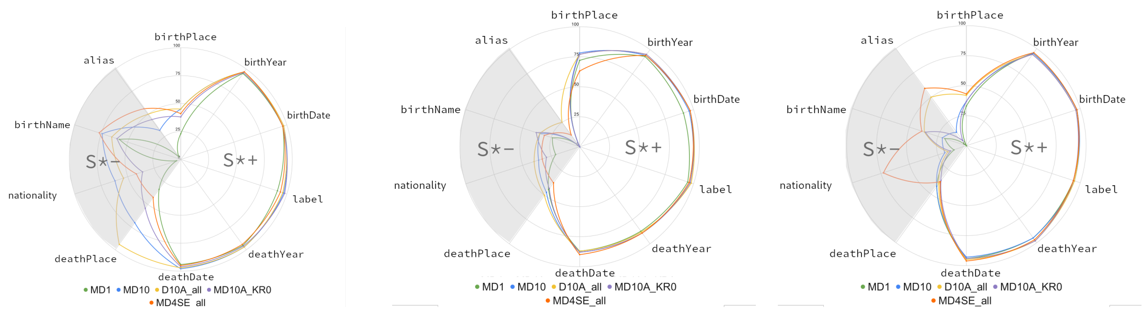
8 Discussion

8.1 Answers to Research Questions

RQ1 — Joint extraction of datatype and object properties. Table 3 confirms that pattern-based RE can be extended to jointly extract datatype *and* object

Table 18: Results on the five cross-evaluation sets. Std. dev. < 5 pt.

Model	F_1^-					F_1^+				
	d_N	d_+	d_-	d_R	d_C	d_N	d_+	d_-	d_R	d_C
$\mathcal{M}D10$	84.4	**91.9**	**86.1**	86.8	72.5	67.1	97.1	77.3	66.5	61.0
$\mathcal{M}D10A_{all}$	84.0	91.9	85.2	**87.2**	72.4	67.4	96.4	76.7	69.5	63.9
$\mathcal{M}D10A_{KR0}$	84.0	91.7	82.1	86.7	71.3	64.6	95.7	71.2	65.5	58.3
$\mathcal{M}D10A_{KR1}$	**86.9**	91.9	85.0	86.9	72.5	**67.3**	**97.1**	71.2	**85.0**	59.6
$\mathcal{M}D4SE_{all}$	76.4	79.1	85.7	79.7	**79.9**	65.1	89.2	**89.2**	64.6	**72.7**



(a) d_- (rare props.) (b) d_R (random) (c) d_C (corrected)

Figure 7: Micro-F1 per property and model (sufficient-exposure step).

properties from a single SHACL shape. Models trained on $D1$ achieve $F_1^- = 95\%$ on s^*, close to the datatype-only baseline ($F_1^- = 98\%$). Object properties introduce specific challenges (lower F_1^+, higher variance) owing to URI generation, but the framework handles both property types in a unified manner. The models are not affected by memorisation: performance on unseen post-2021 documents (d_N) is comparable to that on seen data.

RQ2 — Semantic evaluation of object property errors. The four metrics introduced in Section 5 reveal complementary aspects of model behaviour. AlignScore (AS) best discriminates between genuine hallucinations (FP^-, $AS \approx 0.12$) and unlabelled correct discoveries (FP^+, $AS \approx 0.76$), outperforming both the Triplet Critic and NLI models. High semantic similarity scores ($Sem_{sim} \geq 0.8$) show that even incorrect URIs tend to refer to ontologically plausible entities, indicating that strict triple matching underestimates true model quality.

RQ3 — Generalisation to Text2KGBench. The Kastor models outperform both LLM baselines on 11 out of 16 classes, with a global F_1^- of 0.44 vs. 0.30 (Alpaca) and 0.25 (Vicuna), despite having $> 100\times$ fewer parameters. The KastorKG models produce *zero* hallucinated subjects or relations, whereas LLMs hallucinate 12–16% of subjects and 7–9% of relations. The pattern of wins and losses correlates with domain shift: KastorKG prevails on classes whose WebNLG descriptions are semantically close to Wikipedia abstracts and whose property-pattern overlap is high (e.g., `dbo:City`, `dbo:Food`, `dbo:University`); LLMs perform better on classes where many properties derive from infoboxes not present in Wikipedia prose (e.g., `dbo:Athlete`, `dbo:CelestialBody`, `dbo:SportsTeam`, `dbo:Company`).

RQ4 — Long-tail mitigation. Neither stratification nor weighted loss provides significant gains. Scaling data to $D10$ (12,000 examples) improves rare-property performance by $\sim$10 F_1 points but cannot match a model trained specifically on rare properties ($\mathcal{M}D_{-}$). Sufficient-exposure sampling ($D4SE_{all}$, $\lambda = 1{,}000$ examples per property) achieves the best corrected score ($F_1^- = 79.9$ on d_C), outperforming $\mathcal{M}D10$ by seven F_1 points on this most reliable test set. This confirms the sufficient-exposure hypothesis [Allen-Zhu and Li, 2025] at the property level: guaranteeing at least 1,000 training examples per property is more effective than scaling or re-weighting strategies.

8.2 Key Findings

The experiments confirm that shape-aware SLMs can reliably extract RDF graphs—covering both datatype and object properties—with high performance on frequent properties. Memorisation is not observed: performance on unseen post-2021 documents (d_N) is comparable to that on seen data. Compact SLMs (139 M parameters) outperform LLM baselines (>13 B parameters) on the Text2KGBench benchmark across 11 of 16 classes, with zero hallucinated subjects or relations. Table 19 places

Table 19: Qualitative comparison of KastorKG (BART), REBEL, and LLM baselines.

Criterion	**KastorKG (BART)**	**REBEL**	**LLMs**
Output graph	Entity-centric, shape-constrained	Multi-subject	Multi-subject
Property set	SHACL shape-focused	Fixed vocabulary	Fixed vocabulary + unexpected
Hallucination	Very low	Low–medium	High
Data leakage	Fully traceable	Potential	Highly potential
RDF output	Yes (valid Turtle)	No (needs post-proc)	No (needs post-proc)
Inference cost	Low (139M)	Medium ($\sim$400M)	Very high (>13B)

these results in a broader perspective by comparing the three system families studied in this work: our SHACL-specialised SLMs (the Kastor models) vs. REBEL vs. LLMs, along six qualitative dimensions. Kastor models are the only ones to produce valid RDF Turtle directly, with fully traceable data provenance via Wikicheck and negligible hallucination risk thanks to SHACL-constrained generation. REBEL offers a compact alternative but requires post-hoc entity linking and covers only a fixed relation vocabulary that does not completely align with pattern-based relation extraction. LLMs are the most flexible but incur the highest inference cost, produce no native RDF output, and exhibit the highest hallucination and data-leakage risk. Taken together, these characteristics highlight that SHACL-guided SLMs occupy a distinct and practically advantageous niche: schema-compliant, hallucination-free, and lightweight.

8.3 Object Property Challenges

Extracting object properties is harder than extracting datatype properties, as reflected in lower F_1^+ scores and higher standard deviation. The semantic metrics introduced in Section 5 provide complementary insight: even when a generated URI does not exactly match the ground truth, it often refers to a valid, semantically close entity (high Sem_{sim}), indicating that strict triple matching underestimates model quality.

8.4 Limitations

1. **Domain shift.** Transferring Wikipedia-trained models to WebNLG text (shorter and more structured) degrades performance for some classes (e.g., `dbo:SportsTeam`), because WebNLG describes properties found in infoboxes that are absent from plain text.
2. **Stratification complexity.** The stratification focuses on a limited set of rare properties; richer stratification over full property combinations is computationally intractable.
3. **Naive data augmentation.** KR0/KR1 can introduce text–graph inconsistencies and duplicate entities, limiting quality gains.
4. **Mereological relations.** The Wu–Palmer similarity does not fully capture part-whole relationships (e.g., `dbo:City` $\subset$ `dbo:Place`), a known limitation of purely taxonomic similarity.

9 Perspectives and Future Work

Defining and Using More Complex Shapes. The shapes used in this work are relatively simple: the `dbo:Person` shape covers ten properties, and the Text2KGBench micro-ontologies, whilst potentially listing up to 68 properties, specify only enumerations of property paths without cardinality constraints. Richer shapes—with cardinality bounds, value lists, and inter-property dependencies—would more faithfully represent real-world ontological commitments but also pose new challenges for the extraction framework.

Automatic large-scale shape generation has been explored via the SheXer framework [Rabbani et al., 2022], which demonstrated the difficulty of reliably inferring shape constraints for noisy KGs such as DBpedia. Recent work [Zhang et al., 2025] showed that LLMs can generate ShEx constraints but struggle to assign correct cardinalities, a task where classical ML methods can be competitive. Such work may need to be extended to DBpedia to produce a set of high-quality, SHACL shapes.

Extending shapes with additional properties will also challenge the current framework: models trained on abstracts may need to process entire articles to capture rare or infobox-specific properties, and the extraction pipeline will need to support lists of objects and cardinality constraints.

Usage of Synthetic Data. The experiments confirmed the sufficient-exposure hypothesis [Allen-Zhu and Li, 2025]: guaranteeing at least 1,000 training examples per property is the most effective strategy for balanced performance. For classes with very few DBpedia instances (e.g., `dbo:ComicsCharacter`, `dbo:Astronaut`, `dbo:Monument`), however, even this threshold is difficult to reach with naturally occurring Wikipedia data.

A promising direction is to use LLMs as **constrained synthetic data generators**, following the teacher–student paradigm in which a large model generates labelled data used to fine-tune a smaller specialised model [Long et al., 2024, Patel et al., 2024]. In this context, LLMs would be conditioned on SHACL shape specifications to produce novel but semantically valid abstract–graph pairs, ensuring sufficient exposure for under-represented properties. An illustrative prompt integrating the target entity type, datatype constraints, and cloze-style textual templates is shown in Figure 8.

Figure 8: Prompt designed to generate synthetic graph-text pairs conditioned on a SHACL shape.

> **Instruction:** Generate 10 synthetic (abstract, Turtle graph) pairs for a `dbo:Person` instance. The graph must include exactly `rdfs:label` (string), `dbo:alias` (string), and `dbo:birthDate` (date). The subject URI should resemble a Wikipedia ID but must not be a real entity. Base your generation on the provided example graph and cloze template.
>
> **Input example graph:** `dbr:Duchnuf a dbo:Person; rdfs:label "Duchnuf"; dbo:alias "Duchduch"; dbo:birthDate "1992-08-25".`
>
> **Expected output format:** `[{"id":"ex1","graph":"...","text":"..."}, ...]`

Scaling from Abstracts to Full Documents. A fundamental limitation of the current framework is that it operates at the **Wikipedia abstract level**: both the distillation pipeline and the Wikicheck step rely on the short introductory paragraph of each article. As a result, the SHACL shapes targeted in this work are intentionally restricted to a small number of relations that are reliably expressed in abstracts—three object properties for `dbo:Person`, and 4–18 properties per shape across Text2KGBench. Properties mentioned only in article body text or infoboxes (e.g., career statistics, list-valued attributes, section-specific facts) cannot currently be extracted, which caps the coverage of the distilled KG regardless of model quality.

Scaling to the full document is the natural next step, but it raises two challenges: (i) BART's 1,024-token context window cannot accommodate a full Wikipedia article; and (ii) Wikicheck, which detects entity mentions in Markdown links, was designed for the abstract and does not transfer directly to multi-section HTML.

Two complementary approaches could address this. First, **Retrieval-Augmented Generation (RAG)**: for each property constraint in the SHACL shape, a retrieval step would identify the most relevant sentences or passages from the full article and supply only those to the extractor. This preserves the pattern-based framework and the 1,024-token limit whilst extending property coverage beyond the abstract. Second, **chunk-level extraction with SHACL-guided aggregation**: the article could be split into sentence- or paragraph-level chunks, each processed independently, with a post-hoc reconciliation step that merges and deduplicates competing triples under the shape constraints. Both directions require adapting the Wikicheck step—extending URI detection beyond the abstract's Markdown links—and the distillation pipeline to handle multi-passage supervision signals.

Making Shape-Based Relation Extraction Perform in the Wild. The KastorKG framework currently relies on the English editions of DBpedia and Wikipedia. Several natural extensions are envisaged:

1. other linguistic chapters of DBpedia and Wikipedia (multilingual extension);
2. Wikidata, which would require handling opaque identifiers and qualifiers;
3. web pages linked from DBpedia/Wikidata [Nadăş et al., 2025] as additional text sources;
4. domain-specific wikis, which share the wiki interface with Wikipedia and could be turned into dual knowledge graphs following the DBpedia extraction framework [Hertling and Paulheim, 2018];
5. scientific literature, for which a systematic literature review tool (Scilex) was developed alongside this work and could be combined with KastorKG for domain-science extraction.

Small Agents for Knowledge Engineering. The KastorKG framework could also serve as an agent component in larger knowledge engineering pipelines. In agentic systems, specialised small models have been shown to be more reliable than general-purpose LLMs for structured output tasks [Sharma and Mehta, 2025]. The complete lifecycle of a KG—population, validation, maintenance, and schema evolution—could benefit from autonomous agents able to produce SHACL-tailored extractors on demand. In this context, KastorKG could be viewed as an *extractor factory*: given a SHACL shape, it produces a fine-tuned SLM that participates in the lifecycle of the KG and its schema.

10 Conclusion

Shape-aware small language models can reliably populate knowledge graphs from natural language at a fraction of the computational cost of large language models—with RDF output, no hallucinated subjects or relations, and strong generalisation across ontological classes. This paper demonstrates this at scale: 139 M-parameter SLMs outperform 13 B-parameter LLM baselines on 11 out of 16 Text2KGBench classes whilst producing zero subject- or relation-level hallucinations, versus 12–16% and 7–9% hallucination rates for the LLM baselines.

This journal article extends the KCAP 2025 conference paper [Ringwald et al., 2025d] with three major novel contributions. First, **KastorKG** was released as an open resource comprising 19 DBpedia SHACL Turtle shapes derived from Text2KGBench, a distilled knowledge graph of 141,096 Wikipedia abstract–RDF graph pairs across 16 DBpedia classes, and 16 shape-aware SLMs on HuggingFace. Second, four semantic evaluation metrics for object properties were introduced—covering URI validity, class-level semantic similarity, relation confusion, and hallucination estimation—and their utility was demonstrated both on the original `dbo:Person` experiments and on the cross-domain WebNLG evaluation. A cross-domain transfer analysis (Wikipedia → WebNLG) using Jaccard pattern similarity and sentence-BERT alignment further identified the structural predictors of per-class performance. Third, a comprehensive study of data construction strategies for long-tail distributions established sufficient-exposure sampling ($\lambda = 1{,}000$ examples per property) as the dominant mitigation approach—outperforming both scaling and re-weighting strategies by up to seven F_1 points on the noise-corrected test set d_C.

These results provide actionable guidance for practitioners building schema-constrained KG population systems. The most critical open challenge is **scaling from the abstract to the full article**: the current framework is intentionally restricted to properties reliably expressed in Wikipedia abstracts, and extending coverage to infobox-derived or body-level properties will require retrieval-augmented extraction and an adapted Wikicheck step. Further directions include extending sufficient-exposure training to cross-lingual settings, exploring curriculum-based fine-tuning over SHACL shape complexity, leveraging LLMs as constrained synthetic data generators guided by shape specifications, and adapting the KastorKG framework to dynamic ontologies spanning DBpedia, Wikidata, and YAGO.

Data Availability

All datasets, trained models, and evaluation artefacts are available at long-term stable URLs:

- **Code and scripts:** `https://github.com/datalogism/KastorKG`
- **Trained models (HuggingFace):** `https://huggingface.co/Datartisan` (Models follow the naming convention `Datartisan/Kastor<EntityType>`; see documentation at `https://github.com/datalogism/Kastor/blob/main/doc/UseKastorKGsModelsInTheWild.md`)
- **Datasets (Zenodo):** `https://doi.org/10.5281/zenodo.18862240`
- **SHACL shapes (Text2KGBench):** `https://github.com/datalogism/Kastor/tree/main/shapes/txt2kg`

Acknowledgments

This work was supported by the French government through Université Côte d'Azur, Inria, CNRS, I3S, France, and through the 3IA Côte d'Azur of the National Research Agency (ANR-23-IACL-0001). Computing resources were provided by IDRIS under allocation AD010614282 on the Jean Zay supercomputer.

Generative AI Disclosure: The authors used AI-assisted tools during the preparation of this manuscript. Claude (Anthropic) was used to support manuscript structuring, section drafting, and LaTeX editing. ChatGPT (OpenAI) was used for rephrasing and improving clarity of selected passages. Grammarly was used for grammar and typography checking. All AI-generated or AI-assisted content was reviewed, verified, and revised by the authors, who take full responsibility for the integrity and accuracy of the published work.

References

Zeyuan Allen-Zhu and Yuanzhi Li. Physics of Language Models: Part 3.3, Knowledge Capacity Scaling Laws. In *Proceedings of the 13th International Conference on Learning Representations*, ICLR '25, April 2025.

Varvara Arzt, Allan Hanbury, Michael Wiegand, Gábor Recski, and Terra Blevins. Relation extraction or pattern matching? unravelling the generalisation limits of language models for biographical re, 2025.

Julien Aubert, Oscar Molina, Edouard Pelissier, et al. Text2kgbench-LettrIA: A refined benchmark for text2graph systems. In *Proceedings of the Knowledge Graph Construction Workshop (KGC 2023), co-located with ESWC 2023*, 2023. URL https://ceur-ws.org/Vol-4041/paper3.pdf. CEUR Workshop Proceedings, Vol. 4041.

J Harry Caufield, Harshad Hegde, Vincent Emonet, Nomi L Harris, Marcin P Joachimiak, Nicolas Matentzoglu, HyeongSik Kim, Sierra Moxon, Justin T Reese, Melissa A Haendel, Peter N Robinson, and Christopher J Mungall. Structured prompt interrogation and recursive extraction of semantics (spires): a method for populating knowledge bases using zero-shot learning. *Bioinformatics*, 40(3), 2024.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. T-REx: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, may 2018. European Language Resources Association (ELRA). URL https://aclanthology.org/L18-1544.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. The WebNLG challenge: Generating text from RDF data. In Jose M. Alonso, Alberto Bugarín, and Ehud Reiter, editors, *Proceedings of the 10th International Conference on Natural Language Generation*, pages 124–133, Santiago de Compostela, Spain, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-3518. URL https://aclanthology.org/W17-3518/.

Qiyuan He, Yizhong Wang, Jianfei Yu, and Wenya Wang. Language models over large-scale knowledge base: on capacity, flexibility and reasoning for new facts. In *International Conference on Computational Linguistics*, pages 1736–1753, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics.

Sven Hertling and Heiko Paulheim. DBkWik: A consolidated knowledge graph from thousands of wikis. In *2018 IEEE International Conference on Big Knowledge (ICBK)*, pages 17–24, 2018. doi: 10.1109/ICBK.2018.00011.

Nicolas Hubert, Pierre Monnin, Armelle Brun, and Davy Monticolo. Sem@k: Is my knowledge graph embedding model semantic-aware? *Semantic Web*, 14(6): 1273–1309, 2023. doi: 10.3233/SW-233508. URL https://journals.sagepub.com/doi/abs/10.3233/SW-233508.

Pere-Lluís Huguet Cabot and Roberto Navigli. REBEL: Relation extraction by end-to-end language generation. In *Findings of the Association for Computational*

Linguistics: EMNLP 2021, pages 2370–2381, Punta Cana, Dominican Republic, November 2021. ACL. doi: 10.18653/v1/2021.findings-emnlp.204.

Pere-Lluís Huguet Cabot, Simone Tedeschi, Axel-Cyrille Ngonga Ngomo, and Roberto Navigli. RED$^{\mathrm{fm}}$: a filtered and multilingual relation extraction dataset. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4326–4343, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.237.

Chunyang Jiang, Chi-Min Chan, Wei Xue, Qifeng Liu, and Yike Guo. Importance weighting can help large language models self-improve. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(23):24257–24265, Apr. 2025.

Martin Josifoski, Nicola De Cao, Maxime Peyrard, Fabio Petroni, and Robert West. GenIE: Generative information extraction. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4626–4643, Seattle, United States, July 2022. ACL.

Martin Josifoski, Marija Sakota, Maxime Peyrard, and Robert West. Exploiting asymmetry for synthetic training data generation: SynthIE and the case of information extraction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1555–1574, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.96. URL https://aclanthology.org/2023.emnlp-main.96.

Xianming Li, Xiaotian Luo, Chenghao Dong, Daichuan Yang, Beidi Luan, and Zhen He. TDEER: An efficient translating decoding schema for joint extraction of entities and relations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8055–8064. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.635. URL https://aclanthology.org/2021.emnlp-main.635.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007, 2017.

Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. On LLMs-driven synthetic data generation, curation, and evaluation: A survey. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11065–11082. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.658.

Nandana Mihindukulasooriya, Sanju Tiwari, Carlos F. Enguix, and Kusum Lata. Text2kgbench: A benchmark for ontology-driven knowledge graph generation from text. In *Proceedings ISWC 2023*, pages 247–265, Cham, 2023. Springer. ISBN 978-3-031-47243-5.

Mihai Nadăş, Laura Dioşan, and Andreea Tomescu. Synthetic data generation using large language models: Advances in text and code. *IEEE Access*, 13:134615–134633, 2025. doi: 10.1109/ACCESS.2025.3589503.

Ajay Patel, Colin Raffel, and Chris Callison-Burch. DataDreamer: A tool for synthetic data generation and reproducible LLM workflows. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3781–3799. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.208.

Kashif Rabbani, Matteo Lissandrini, and Katja Hose. Shacl and shex in the wild: A community survey on validating shapes generation and adoption. In *Companion Proceedings of the Web Conference 2022*, WWW '22, pages 260–263. Association for Computing Machinery, 2022. doi: 10.1145/3487553.3524253.

Célian Ringwald. Learning pattern-based extractors from natural language and knowledge graphs: Applying large language models to wikipedia and linked open data. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(21): 23411–23412, 2024. URL https://doi.org/10.1609/aaai.v38i21.30406.

Célian Ringwald, Fabien Gandon, Catherine Faron, Franck Michel, and Hanna Abi Akl. Well-written knowledge graphs: Most effective rdf syntaxes for triple linearization in end-to-end extraction of relations from texts (student abstract). *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(21):23631–23632, 2024. URL https://doi.org/10.1609/aaai.v38i21.30502.

Célian Ringwald, Fabien Gandon, Catherine Faron, Franck Michel, and Hanna Abi Akl. Kastor: Fine-tuned small language models for shape-based active relation extraction. In *European Semantic Web Conference*, volume 15718 of *Lecture Notes in Computer Science*, pages 94–115. Springer, 2025a. URL https://doi.org/10.1007/978-3-031-94575-5_6.

Célian Ringwald, Fabien Gandon, Catherine Faron, Franck Michel, and Hanna Abi Akl. Extensive benchmark of small language models for datatype properties extraction and rdf knowledge graph generation. *Semantic Web Journal*, 2025b. URL https://www.semantic-web-journal.net/content/extensive-benchmark-small-language-models-datatype-properties-extraction-and-rdf- under-review.

Célian Ringwald, Fabien Gandon, Catherine Faron, Franck Michel, and Hanna Abi Akl. A systematic review of relation extraction task since the emergence of transformers. *ACM Computing surveys*, 2025c. URL https://arxiv.org/abs/2511.03610. under-review.

Célian Ringwald, Fabien Gandon, Catherine Faron, Franck Michel, and Hanna Abi Akl. Overcoming the generalization limits of slm finetuning for shape-based extraction of datatype and object properties. *Knowledge Capture Conference 2025*, 2025d. URL https://hal.science/hal-05285428. accepted.

Raghav Sharma and Manan Mehta. Small language models for agentic systems: A survey of architectures, capabilities, and deployment trade-offs, 2025.

Zhibiao Wu and Martha Palmer. Verb semantics and lexical selection. In *32nd Annual Meeting of the Association for Computational Linguistics*, pages 133–138, Las Cruces, New Mexico, USA, June 1994. Association for Computational Linguistics. doi: 10.3115/981732.981751. URL https://aclanthology.org/P94-1019/.

Xin Xu, Xiang Chen, Ningyu Zhang, Xin Xie, Xi Chen, and Huajun Chen. Towards realistic low-resource relation extraction: A benchmark with empirical baseline study. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 413–427, Abu Dhabi, United Arab Emirates, December 2022. ACL. doi: 10.18653/v1/2022.findings-emnlp.29.

Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. DocRED: A large-scale document-level relation extraction dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 764–777, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1074.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. AlignScore: Evaluating factual consistency with a unified alignment function. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11328–11348. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.634.

Bohui Zhang, Yuan He, Lydia Pintscher, Albert Meroño Peñuela, and Elena Simperl. Schema generation for large knowledge graphs using large language models, 2025.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, 2017.

A Comparison of WEBNLG/Text2KGBench/KastorKG

Table 20: Comparative analysis of **WebNLG**, **Text2KGBench–DBpedia (T2KG-DBpedia)**, and **KastorKG** across four dimensions: dataset size, lexicalisation richness, syntactic variety, and aggregation/segmentation complexity. *CTTR* = Corrected Type–Token Ratio.

Metric	**WebNLG**	**T2KGBench–DBpedia**	**KastorKG**
Size			
# data–text pairs	25,298	4,860	141,096
# distinct inputs	9,674	4,860	141,096
Lexicalisation			
# properties	373	356	101
# domains	15	19	52,026
CTTR	6.51	9.96	58.35
# word types	6,547	4,408	308,109
Syntactic Variety			
# input patterns	4,129	2,093	1,136
# inputs / # patterns	2.34	2.32	124.20
# input shapes	62	12	3
Aggregation, GRE & Segmentation			
# inputs with 1–2 triples	11,111	718	121,779
# inputs with 3–4 triples	8,172	3,736	17,324
# inputs with 5–7 triples	6,015	406	1,921
# inputs with ≥8 triples	—	0	72
# texts with 1 sentence	16,740	3,183	25,130
# texts with 2 sentences	6,798	1,291	32,395
# texts with ≥3 sentences	1,760	386	83,571
words/text (avg / min / max)	22.69 / 4 / 80	20.15 / 3 / 78	98.80 / 6 / 2014

B Statistics on Text2KGBench and KastorKG

Table 21: Per-class statistics for Text2KGBench-DBpedia and KastorKG. OP/DP: distinct object/datatype property types; Tr: total number of triples across all pairs (T2KG: same triple counted once per text); DTr (T2KG only): number of unique ⟨sub, rel, obj⟩ tuples; Ent: distinct subject entities; Txt: entries with text; Sent: total number of sentences; Patt: distinct property-set patterns.

	Text2KGBench-DBpedia								**KastorKG**						
Class	OP	DP	Tr	DTr	Ent.	Txt	Sent	Patt	OP	DP	Tr	Ent	Txt	Sent	Patt
Airport	0	39	951	258	36	306	403	134	3	1	8305	5112	5112	15135	15
Artist	0	39	1148	271	41	386	515	134	7	5	50570	27275	27275	94742	173
Astronaut	0	38	520	92	14	154	202	129	0	0	0	0	0	0	0
Athlete	0	37	874	244	38	293	396	83	9	6	14681	10711	10711	28447	121
Building	0	38	897	232	30	275	426	129	6	1	9388	5838	5838	24465	37
CelestialBody	0	27	596	170	22	194	273	97	1	2	3291	2654	2654	10491	4
City	0	23	1049	234	37	348	476	95	1	3	4077	4034	4034	16006	9
ComicsCharacter	0	18	322	86	19	102	122	40	2	0	19	19	19	117	2
Company	0	28	457	87	10	153	195	120	7	1	6920	4924	4924	20408	44
Film	0	44	776	108	21	264	392	190	9	1	34707	19963	19963	75803	180
Food	0	24	1266	259	36	398	546	65	2	1	3131	2236	2236	9909	7
MeansOfTransport	0	68	923	268	41	314	417	161	8	10	5105	3449	3449	15079	68
Monument	0	26	348	41	9	92	163	90	2	0	1857	1173	1173	4524	3
MusicalWork	0	35	825	118	26	290	399	164	8	1	10923	4967	4967	22296	126
Politician	0	40	974	265	35	319	507	99	12	5	16302	8548	8548	32009	175
Scientist	0	47	711	113	18	259	355	195	9	2	22190	11393	11393	43173	199
SportsTeam	0	24	776	219	27	235	459	68	6	0	2941	2090	2090	7220	19
University	0	44	531	80	16	156	319	146	8	2	10625	6675	6675	25750	55
WrittenWork	0	44	938	238	34	322	505	109	9	3	25166	20035	20035	77684	73
Total	**0**	**683**	**14882**	**3383**	**510**	**4860**	**7070**	**2248**	**109**	**44**	**230198**	**141096**	**141096**	**523258**	**1310**

C Relation Distribution in Text2KGBench (T2KG) and KastorKG

Table 22: Airport — T2KG: 59 ent., KastorKG: 5112 ent. KastorKG uses 4 relation(s); 35 T2KG-only relation(s) not shown.

Relation	**T2KG**		**KastorKG**	
	Ent.	**%**	**Ent.**	**%**
icaoLocationIdentifier	3	5.1%	4407	86.2%
location	15	25.4%	2184	42.7%
city	2	3.4%	593	11.6%
owner	1	1.7%	162	3.2%

Table 23: Artist — T2KG: 51 ent., KastorKG: 27275 ent. KastorKG uses 12 relation(s); 27 T2KG-only relation(s) not shown.

Relation	**T2KG**		**KastorKG**	
	Ent.	**%**	**Ent.**	**%**
birthDate	4	7.8%	22163	81.3%
deathDate	1	2.0%	8092	29.7%
genre	25	49.0%	5733	21.0%
birthPlace	10	19.6%	4531	16.6%
activeYearsStartYear	9	17.6%	3327	12.2%
deathPlace	5	9.8%	1217	4.5%
recordLabel	6	11.8%	1210	4.4%
birthYear	1	2.0%	1180	4.3%
nationality	2	3.9%	476	1.7%
training	1	2.0%	150	0.5%
country	3	5.9%	70	0.3%
background	9	17.6%	11	0.0%

Table 24: Athlete — T2KG: 64 ent., KastorKG: 10711 ent. KastorKG uses 15 relation(s); 22 T2KG-only relation(s) not shown.

Relation	**T2KG**		**KastorKG**	
	Ent.	%	**Ent.**	%
birthDate	9	14.1%	10253	95.7%
birthPlace	13	20.3%	1371	12.8%
formerTeam	2	3.1%	573	5.3%
draftYear	1	1.6%	422	3.9%
draftRound	1	1.6%	310	2.9%
debutTeam	2	3.1%	247	2.3%
activeYearsStartYear	1	1.6%	207	1.9%
league	1	1.6%	193	1.8%
deathPlace	1	1.6%	180	1.7%
draftTeam	2	3.1%	167	1.6%
draftPick	1	1.6%	98	0.9%
college	1	1.6%	32	0.3%
club	19	29.7%	7	0.1%
birthYear	7	10.9%	6	0.1%
coach	5	7.8%	2	0.0%

Table 25: Building — T2KG: 54 ent., KastorKG: 5838 ent. KastorKG uses 7 relation(s); 31 T2KG-only relation(s) not shown.

Relation	**T2KG**		**KastorKG**	
	Ent.	%	**Ent.**	%
location	19	35.2%	4933	84.5%
country	19	35.2%	801	13.7%
tenant	2	3.7%	471	8.1%
owner	5	9.3%	374	6.4%
buildingStartDate	7	13.0%	293	5.0%
architect	10	18.5%	231	4.0%
language	3	5.6%	3	0.1%

Table 26: CelestialBody — T2KG: 25 ent., KastorKG: 2654 ent. KastorKG uses 3 relation(s); 24 T2KG-only relation(s) not shown.

Relation	**T2KG**		**KastorKG**	
	Ent.	%	**Ent.**	%
discovered	2	8.0%	2589	97.6%
discoverer	8	32.0%	644	24.3%
epoch	21	84.0%	1	0.0%

Table 27: City — T2KG: 63 ent., KastorKG: 4034 ent. KastorKG uses 4 relation(s); 19 T2KG-only relation(s) not shown.

Relation	**T2KG**		**KastorKG**	
	Ent.	**%**	**Ent.**	**%**
country	45	71.4%	3983	98.7%
postalCode	1	1.6%	42	1.0%
utcOffset	2	3.2%	29	0.7%
areaCode	4	6.3%	18	0.4%

Table 28: ComicsCharacter — T2KG: 31 ent., KastorKG: 19 ent. KastorKG uses 2 relation(s); 16 T2KG-only relation(s) not shown.

Relation	**T2KG**		**KastorKG**	
	Ent.	**%**	**Ent.**	**%**
creator	15	48.4%	18	94.7%
voice	1	3.2%	1	5.3%

Table 29: Company — T2KG: 17 ent., KastorKG: 4924 ent. KastorKG uses 8 relation(s); 20 T2KG-only relation(s) not shown.

Relation	**T2KG**		**KastorKG**	
	Ent.	**%**	**Ent.**	**%**
location	5	29.4%	2477	50.3%
parentCompany	2	11.8%	1448	29.4%
regionServed	1	5.9%	846	17.2%
keyPerson	2	11.8%	364	7.4%
foundingDate	8	47.1%	257	5.2%
subsidiary	2	11.8%	144	2.9%
country	1	5.9%	120	2.4%
foundationPlace	2	11.8%	1	0.0%

Table 30: Film — T2KG: 24 ent., KastorKG: 19963 ent. KastorKG uses 10 relation(s); 34 T2KG-only relation(s) not shown.

Relation	**T2KG**		**KastorKG**	
	Ent.	**%**	**Ent.**	**%**
director	5	20.8%	14649	73.4%
writer	4	16.7%	8732	43.7%
language	1	4.2%	3292	16.5%
distributor	2	8.3%	2811	14.1%
musicComposer	3	12.5%	864	4.3%
starring	3	12.5%	732	3.7%
editing	3	12.5%	714	3.6%
releaseDate	1	4.2%	545	2.7%
cinematography	2	8.3%	542	2.7%
producer	3	12.5%	175	0.9%

Table 31: Food — T2KG: 48 ent., KastorKG: 2236 ent. KastorKG uses 3 relation(s); 21 T2KG-only relation(s) not shown.

Relation	**T2KG**		**KastorKG**	
	Ent.	**%**	**Ent.**	**%**
country	21	43.8%	1303	58.3%
region	17	35.4%	1158	51.8%
servingTemperature	2	4.2%	230	10.3%

Table 32: MeansOfTransport — T2KG: 60 ent., KastorKG: 3449 ent. KastorKG uses 18 relation(s); 50 T2KG-only relation(s) not shown.

Relation	**T2KG**		**KastorKG**	
	Ent.	**%**	**Ent.**	**%**
manufacturer	19	31.7%	1256	36.4%
shipLaunch	1	1.7%	958	27.8%
productionStartYear	2	3.3%	877	25.4%
productionEndYear	2	3.3%	609	17.7%
country	5	8.3%	509	14.8%
owner	4	6.7%	280	8.1%
status	3	5.0%	263	7.6%
relatedMeanOfTransportation	10	16.7%	124	3.6%
completionDate	1	1.7%	28	0.8%
designCompany	1	1.7%	15	0.4%
christeningDate	3	5.0%	12	0.3%
maidenFlight	1	1.7%	12	0.3%
engine	9	15.0%	10	0.3%
maidenVoyage	2	3.3%	7	0.2%
countryOrigin	3	5.0%	6	0.2%
transmission	3	5.0%	5	0.1%
finalFlight	1	1.7%	4	0.1%
comparable	1	1.7%	2	0.1%

Table 33: Monument — T2KG: 10 ent., KastorKG: 1173 ent. KastorKG uses 2 relation(s); 24 T2KG-only relation(s) not shown.

Relation	**T2KG**		**KastorKG**	
	Ent.	**%**	**Ent.**	**%**
location	5	50.0%	1120	95.5%
designer	2	20.0%	157	13.4%

Table 34: MusicalWork — T2KG: 28 ent., KastorKG: 4967 ent. KastorKG uses 9 relation(s); 26 T2KG-only relation(s) not shown.

Relation	**T2KG**		**KastorKG**	
	Ent.	**%**	**Ent.**	**%**
writer	1	3.6%	2732	55.0%
releaseDate	3	10.7%	1891	38.1%
artist	4	14.3%	1558	31.4%
album	2	7.1%	1386	27.9%
genre	11	39.3%	1159	23.3%
recordLabel	4	14.3%	586	11.8%
producer	9	32.1%	298	6.0%
language	1	3.6%	26	0.5%
recordedIn	1	3.6%	14	0.3%

Table 35: Politician — T2KG: 55 ent., KastorKG: 8548 ent. KastorKG uses 17 relation(s); 23 T2KG-only relation(s) not shown.

Relation	**T2KG**		**KastorKG**	
	Ent.	**%**	**Ent.**	**%**
birthDate	2	3.6%	7270	85.0%
deathDate	1	1.8%	3720	43.5%
party	14	25.5%	1542	18.0%
birthPlace	23	41.8%	1532	17.9%
deathPlace	8	14.5%	706	8.3%
almaMater	4	7.3%	355	4.2%
nationality	6	10.9%	249	2.9%
residence	1	1.8%	217	2.5%
spouse	4	7.3%	99	1.2%
office	13	23.6%	21	0.2%
country	1	1.8%	19	0.2%
award	3	5.5%	14	0.2%
birthYear	1	1.8%	11	0.1%
battle	9	16.4%	10	0.1%
region	1	1.8%	10	0.1%
deathYear	1	1.8%	9	0.1%
militaryBranch	7	12.7%	7	0.1%

Table 36: Scientist — T2KG: 18 ent., KastorKG: 11393 ent. KastorKG uses 11 relation(s); 36 T2KG-only relation(s) not shown.

Relation	**T2KG**		**KastorKG**	
	Ent.	**%**	**Ent.**	**%**
birthDate	5	27.8%	9475	83.2%
deathDate	2	11.1%	5762	50.6%
almaMater	3	16.7%	1971	17.3%
birthPlace	4	22.2%	1629	14.3%
deathPlace	1	5.6%	775	6.8%
award	1	5.6%	773	6.8%
nationality	2	11.1%	528	4.6%
doctoralAdvisor	1	5.6%	273	2.4%
residence	3	16.7%	243	2.1%
spouse	1	5.6%	66	0.6%
country	5	27.8%	1	0.0%

Table 37: SportsTeam — T2KG: 46 ent., KastorKG: 2090 ent. KastorKG uses 6 relation(s); 18 T2KG-only relation(s) not shown.

Relation	**T2KG**		**KastorKG**	
	Ent.	**%**	**Ent.**	**%**
location	3	6.5%	995	47.6%
league	11	23.9%	900	43.1%
city	1	2.2%	425	20.3%
owner	1	2.2%	63	3.0%
ground	14	30.4%	56	2.7%
manager	14	30.4%	18	0.9%

Table 38: University — T2KG: 22 ent., KastorKG: 6675 ent. KastorKG uses 10 relation(s); 35 T2KG-only relation(s) not shown.

Relation	**T2KG**		**KastorKG**	
	Ent.	**%**	**Ent.**	**%**
country	8	36.4%	3751	56.2%
city	7	31.8%	3011	45.1%
state	3	13.6%	2064	30.9%
affiliation	2	9.1%	1315	19.7%
officialSchoolColour	—	—	72	1.1%
motto	1	4.5%	69	1.0%
president	1	4.5%	19	0.3%
dean	1	4.5%	12	0.2%
director	1	4.5%	9	0.1%
rector	1	4.5%	8	0.1%

Table 39: WrittenWork — T2KG: 47 ent., KastorKG: 20035 ent. KastorKG uses 12 relation(s); 32 T2KG-only relation(s) not shown.

Relation	**T2KG**		**KastorKG**	
	Ent.	**%**	**Ent.**	**%**
author	15	31.9%	11981	59.8%
publisher	11	23.4%	5727	28.6%
firstPublicationYear	2	4.3%	2260	11.3%
country	14	29.8%	1056	5.3%
headquarter	1	2.1%	1053	5.3%
language	13	27.7%	925	4.6%
abbreviation	8	17.0%	487	2.4%
releaseDate	2	4.3%	444	2.2%
city	1	2.1%	301	1.5%
founder	1	2.1%	95	0.5%
editor	1	2.1%	7	0.0%
genre	1	2.1%	4	0.0%