How Many Relation Types Do You Need? Defining a Corpus-Relative Optimum for Text-to-Graph Schemas

Anonymous authors
Paper under double-blind review

Abstract

Every text-to-graph pipeline commits to a schema before extraction, yet prior work offers no principled answer to how large that schema should be. We introduce **SchemaGain**, which scores a candidate relation inventory by the surprisal that its extracted graph removes from the corpus, minus the language-model cost of describing the schema itself. Across all 19 DBpedia-WebNLG corpora in Text2KGBench, SchemaGain has an interior optimum and aligns with expert-designed inventories (Pearson $r = 0.70$; 0.66 across seeds), while alternative selectors based on clustering, frequency, or direct LLM judgment show weaker agreement and greater variation across seeds. Randomly merging relations through otherwise identical machinery reduces the correlation to 0.12. On Text2KGBench's extraction task, using any schema improves F1 by 61% over schema-free extraction and reduces subject hallucination from 41% to below 1%. The cost of larger inventories becomes apparent downstream: in a routing benchmark over the same graphs, relation retrieval peaks near 35 types, and at 60 types a label router's top-1 accuracy is half that at 30. A full sweep takes about four hours on one consumer GPU. Together, these results suggest that schema size is a measurable modeling choice rather than an arbitrary design parameter.

1 Introduction

A growing class of systems answers questions by reading a corpus through a knowledge graph. They convert text into a graph, retrieve or traverse subgraphs at query time, and generate answers grounded in the retrieved structure (Edge et al., 2024; Guo et al., 2024). Agent-memory systems run the same cycle continuously (Rasmussen et al., 2025). All of them begin with the same commitment, made before any extraction happens: a schema, the inventory of relation types the extractor may use. Everything downstream inherits it. A relation the schema cannot express never enters the graph, is invisible to traversal, and cannot ground a generated answer.

Yet no prior work says how large that inventory should be, and deployed systems set it by convention. LightRAG fixes four generic entity types (Guo et al., 2024), GraphRAG adjusts its type list by prompt tuning (Edge et al., 2024), and schema-induction pipelines let the model propose whatever types it likes (Bai et al., 2026; Zhang & Soh, 2024; Mo et al., 2025). Systems that take the schema from a human (Caufield et al., 2024; Mihindukulasooriya et al., 2023) inherit the human's guess. Recent surveys and system papers list granularity control as an open problem (Babaei Giglou et al., 2026; Oyewale & Soru, 2026), and a systematic review finds no established evaluation standard for the choice (Li et al., 2026). Every induction pipeline we know of resolves it with a heuristic: a frequency threshold, a clustering cutoff, or the model's whim. In this paper, "types" means relation types (edge labels). Entity types enter only through each relation's domain and range.

Neither extreme is usable, so any useful schema lies between them. A schema with one type per surface phrase memorizes the corpus, so the graph merely restates the text. A single generic type says nothing, since every edge is `relatedTo`. What is missing is a definition of where a schema should sit: a quantity, computable from the corpus alone, whose optimum is the right size.

We give such a definition, SchemaGain, by treating the schema as a lossy code for the corpus. The types a schema allows determine what the extracted graph can express, and therefore how much of

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

the text it can explain. SchemaGain scores a candidate schema by the surprisal its extracted graph removes from the text, net of the cost of writing the schema down. One frozen language model measures both terms in nats, so they are commensurable and no trade-off weight is tuned.

SchemaGain borrows from the minimum description length principle (Rissanen, 1978; Grünwald, 2007) but is not a strict two-part code: it does not charge for transmitting the extracted fragments (§3.3). Because open-weights language models assign a likelihood to any text in any context, this 1978 principle becomes computable here (Delétang et al., 2024). To our knowledge, SchemaGain is the first computable definition of the size a relation inventory should have. It is a selection rule, not a new extractor: the extractor is held fixed, and only the rule for sizing its type inventory changes.

A definition cannot be checked against a true size, since none exists. It can, however, be checked for the behavior any sensible definition must show. First, an interior optimum must exist, which the construction does not guarantee. Second, where experts have written a complete inventory, the optimum should land near its size. This tests construct validity, not the exactness of the expert-chosen sizes. Third, the agreement should come from the objective itself, not from the surrounding machinery.

Text2KGBench (Mihindukulasooriya et al., 2023) lets us run all three checks. It pairs 19 DBpedia-WebNLG corpora with ontologies whose sizes, between 18 and 68 relation types, were fixed by human experts. Its sentences were written as verbalizations of those ontologies' triples, so the reference is complete by construction. Starting from one open extraction pass, we greedily merge types and track SchemaGain along the whole trajectory.

SchemaGain passes all three checks. On all 19 corpora the gain curve rises, peaks, and falls, never peaking at either extreme (§5.1). The peak size correlates with the expert-chosen size at $r = 0.70$ under greedy decoding, and at 0.66 (range 0.60–0.71) across three sampled extraction seeds, with a mean error of about eight types (§5.2). Choosing merges at random through the identical machinery collapses the correlation to 0.12, and clustering, frequency, and ask-the-model baselines are less accurate on both metrics in every seed (§5.3). Among the selectors we compare, SchemaGain's agreement also varies least under extraction resampling. Its answer sharpens with better evidence: upgrading only the extractor moves the selected size toward the expert-chosen size on three of the four corpora where we tried it, while swapping the scoring model or the serialization leaves the optimum in place (§5.4).

Downstream extraction confirms that schemas matter, but it cannot by itself say which size is right. On the benchmark's own extraction task, any reasonable schema beats schema-free extraction by a wide margin, and under-sized schemas measurably hurt. Above that floor, however, size-controlled F1 does not separate selectors (§5.5), which leaves agreement with complete expert references as the evidence for size. The case of an incomplete reference is taken up in §6 and App. L.

We make three contributions. First, we pose schema-size selection as a problem with a computable answer (§3.1) and define that answer, SchemaGain, with a cached greedy search that runs on one consumer GPU. Second, against 19 human-authored ontologies from as many domains, we give the first evidence that such a definition tracks expert judgment of schema granularity. Third, baselines and ablations show that this agreement is specific to the objective and robust to the main implementation choices.

2 Related work

A large family of text-to-graph systems takes the schema as an input. OntoGPT/SPIRES grounds extraction in user-authored LinkML schemas and reports that schema grounding is what separates usable output from free-form generation (Caufield et al., 2024). Text2KGBench (Mihindukulasooriya et al., 2023) benchmarks exactly this setting, with the ontology given. KARMA aligns multi-agent extraction to an existing domain schema (Lu et al., 2025), and iText2KG takes a user-supplied "blueprint" of types, its authors noting explicitly that the schema "depends on user preferences" (Lairgi et al., 2024). Deployed systems make the same move in production form. LightRAG (Guo et al., 2024) and GraphRAG (Edge et al., 2024) fix or prompt-tune their type lists (§1), and Graphiti falls back to a generic relation type when no user-defined schema is supplied (Rasmussen et al., 2025). The prevailing methodological position is that this is as it should be: language models accelerate ontology engineering while the schema itself remains a human responsibility (Shimizu &

Hitzler, 2025). All of these systems presuppose the inventory whose size we select, and none asks whether the human's choice of size was right.

Systems that induce the schema instead choose its granularity by heuristic. Ontology learning from text was posed in 2001 (Maedche & Staab, 2001), stratified into the classic "layer cake" (Buitelaar et al., 2005), and pursued with lexico-syntactic patterns (Hearst, 1992) whose high-precision, low-recall profile motivated the field's later turn to statistical methods. Retrospectives of that line of work record granularity control as unsolved (Wong et al., 2012). In the LLM era, the induction mechanisms multiply. AutoSchemaKG induces types by conceptualization at web scale (Bai et al., 2026). EDC extracts openly, then defines and self-canonicalizes relations (Zhang & Soh, 2024), and KGGen clusters entities and relations with an LLM judge, naming over- and under-clustering as its main open problem (Mo et al., 2025). SAC-KG iterates generate–verify–prune loops over openly generated relations (Chen et al., 2024). TRACE-KG (Abolhasani et al., 2026) and DIAL-KG (Bao et al., 2026) induce evolving schemas during extraction, and ontology construction itself is the target for Oyewale & Soru (2026) and Pi et al. (2025). Earlier relation canonicalization by embedding clustering (Vashishth et al., 2018), which supplies our strongest clustering baseline, cuts its clusters at a fixed threshold. All of these choose granularity through clustering thresholds, frequency filters, or prompt design. None states an objective for the size of the type inventory, and that gap is our subject.

Work that evaluates induced ontologies scores a schema against a reference, but it cannot choose one where no reference exists, which is the situation in any new domain. The LLMs4OL challenge (Babaei Giglou et al., 2023; 2024) and OntoLearner (Babaei Giglou et al., 2026) benchmark typing and taxonomy subtasks. OLLM contributes graph-similarity metrics against a gold ontology, along with a memorisation baseline that exposes how much literal matching rewards pretraining exposure (Lo et al., 2024). Human evaluations report that 29–67% of elements in LLM-generated ontologies are superfluous (Lippolis et al., 2025), the too-many-types failure mode observed in the wild. The field's systematic review finds no established evaluation standard for granularity at all (Li et al., 2026).

Description length has a long record of sizing discrete structures, but it has not been used to size an extraction schema. The MDL principle (Rissanen, 1978; Grünwald, 2007) has chosen, among other things, the size of word clusterings in NLP (Li & Abe, 1998). Related two-part codes measure deep models (Blier & Ollivier, 2018), and the information bottleneck (Tishby et al., 1999) is the sibling framework for sizing a discrete intermediate representation. The modern reading of language models as general-purpose compressors (Delétang et al., 2024) has made description length practical for probing (Voita & Titov, 2020) and prompt compression (Nagle et al., 2024), though not, to our knowledge, for extraction schemas. Closest in spirit, GraphMDL (Bariatti et al., 2020) and KG-MDL (Bariatti et al., 2023) select graph patterns that compress an *existing* graph. Our object is different: we encode a *corpus through* a schema, using a frozen LM as the measuring device, so the code being optimized is the type inventory of a text-to-graph extractor.

3 METHOD

3.1 PROBLEM

A corpus is a sequence of chunks $c_1, \ldots, c_n$ (here, one sentence each; Figure 1 shows the pipeline). A schema O is a set of relation types. An extractor E maps a chunk and a schema to a graph fragment $g_i = E(c_i, O)$, a list of triples whose relations are drawn from O. A serializer π renders a fragment as plain text. A frozen language model θ is used only to assign log-probabilities. It never generates. With these objects fixed, the question this paper poses is well defined:

Problem (schema-size selection). Given a corpus C, a frozen extractor E, a serializer π, a frozen scorer θ, and a family $\mathcal{O}$ of candidate schemas, find

$$k^\star = \arg\max_k G(k), \qquad G(k) = \max_{O \in \mathcal{O},\, |O|=k} \mathrm{Gain}(O), \tag{1}$$

with Gain the description-length saving defined next (Eq. 4). The argument is a size and the value is in nats, so "how many types?" becomes an optimization rather than a matter of taste. Here $\mathcal{O}$ is the set of schemas reachable by merging the extractor's induced vocabulary (§3.4). The greedy

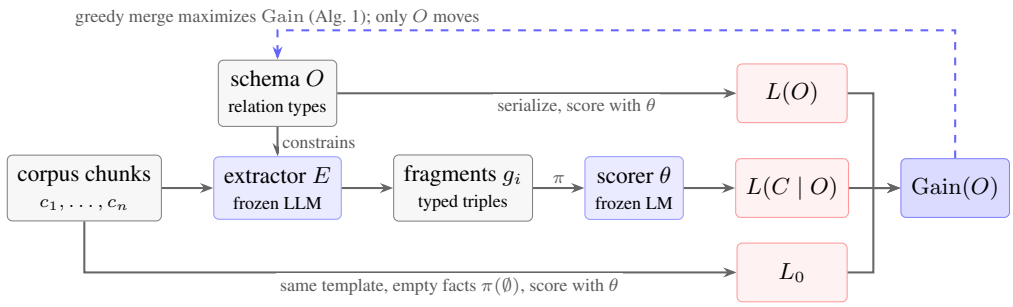

Figure 1: Schema cost and residual text cost. A frozen extractor turns each chunk into a typed fragment under schema O; a frozen scorer θ prices the schema ($L(O)$), the text given its facts ($L(C \mid O)$), and the same-template empty-graph baseline (L_0). The greedy search (Alg. 1) moves only the schema.

trajectory supplies one schema per size, so the curve we report lower-bounds $G(k)$, and we take its argmax as $k^\star$.

One further gap between this statement and the search should be named here rather than left for the reader. The search of §3.4 does not re-extract: it approximates $E(c_i, O)$ by $M_O(E_0(c_i))$, one open pass followed by a schema-dependent relabeling of the cached fragments. That treats a schema as changing only the relation partition, whereas §5.5 shows it changes more, since extraction run under a schema yields different triples and not merely the same ones relabeled. We therefore read $k^\star$ as the optimum of the merge family, and as an approximation to the optimum under schema-conditioned extraction.

3.2 Schema cost and residual text cost

We score a schema by the description length it saves on the corpus:

$$L(O) = -\log P_\theta\big(\text{serialize}(O)\big), \tag{2}$$

$$L(C \mid O) = \textstyle\sum_i -\log P_\theta\big(c_i \mid \pi(g_i)\big), \tag{3}$$

$$\text{Gain}(O) = L_0 - \big[L(O) + L(C \mid O)\big], \qquad L_0 = \textstyle\sum_i -\log P_\theta\big(c_i \mid \pi(\emptyset)\big). \tag{4}$$

$L(O)$ is the cost of writing the schema down. $L(C \mid O)$ is the cost of the text once its extracted facts are in context, and L_0 is the same quantity with an empty fact list. Gain is thus the number of nats the schema saves, net of its own cost. Both parts are measured by the *same* frozen θ, which puts them in the same unit and removes any tunable trade-off weight. We call this quantity, and the selection criterion that maximizes it, *SchemaGain*.

Two design details are correctness-critical. First, only the chunk's tokens are scored: the conditional in $L(C \mid O)$ is computed by one forward pass over $\pi(g_i)$ concatenated with c_i, summing negative log-probabilities over the c_i tokens alone (a hand-checkable unit test appears in App. A). Second, L_0 uses the identical template with an empty fact section. Comparing against a bare, unconditioned chunk would conflate the schema's value with the template's.

3.3 What the criterion optimizes

Any distribution P over token sequences induces a lossless code assigning x the length $-\log P(x) + O(1)$ nats (arithmetic coding), and a conditional code of length $-\log P(x \mid y)$ when context y is shared by encoder and decoder (Grünwald, 2007; Delétang et al., 2024). Every "cost" in this paper is therefore a literal codelength under the single frozen θ. With this reading in place, rearranging Eq. 4 decomposes the objective as

$$\text{Gain}(O) = \underbrace{\textstyle\sum_i \log \frac{P_\theta\left(c_i \mid \pi(g_i)\right)}{P_\theta\left(c_i \mid \pi(\emptyset)\right)}}_{=:\, \widehat{I}_\theta(G \to C)} - L(O), \tag{5}$$

Algorithm 1 Greedy SchemaGain search

Require: chunks $c_{1..n}$; open-extraction fragments $g_{1..n}$; cap K; frozen scorer θ

$O \leftarrow$ the K most frequent relation types in $\{g_i\}$; drop triples outside O
$\ell_i^0 \leftarrow -\log P_\theta(c_i \mid \pi(\emptyset))$; $\ell_i \leftarrow -\log P_\theta(c_i \mid \pi(g_i))$ for all i
$G \leftarrow \sum_i \ell_i^0 - L(O) - \sum_i \ell_i$; record $(|O|, G)$
while $|O| > 1$ **do**
  $P \leftarrow$ type pairs within 10-NN by embedding cosine (top 30 by similarity)
  for all $(a, b) \in P$ **do** ▷ cached across steps
    keep the more frequent label; rescore *only* chunks containing the absorbed one
    $\Delta G_{ab} \leftarrow -\big(\Delta L(O) + \Delta \sum_i \ell_i\big)$
  end for
  merge $\arg\max_{(a,b)} \Delta G_{ab}$: relabel fragments, update cached ℓ_i and $L(O)$
  $G \leftarrow G + \max \Delta G_{ab}$; record $(|O|, G)$
end while
return the trajectory $\{(|O|, G)\}$

where $G = \{g_i\}$ is the extracted graph and $\widehat{I}_\theta(G \to C)$ is the total codelength reduction its fragments provide for the corpus, an LM-measured information gain of the graph about the text. The criterion therefore selects the schema whose induced graph is most informative about the corpus, net of the schema's own description length.

The same objective admits a rate–distortion reading, with the schema as the codebook, $L(O)$ as the rate, and the residual surprisal of the text as the distortion. App. M spells it out.

When is the optimum of Eq. 1 interior? Along a trajectory, the marginal value of the $(k{+}1)$-th type is $\Delta(k) = G(k{+}1) - G(k)$: its additional information gain minus its additional description cost. An interior optimum exists exactly when this marginal value is positive at small k and eventually negative, that is, when early distinctions pay for themselves and marginal ones do not. Nothing in the construction guarantees the crossing. It is a joint property of corpus, extractor, and scorer. That the crossing occurs on every corpus we test, and that its location tracks the expert-chosen size, are the paper's empirical findings (§5.1–5.2).

Two limits of the formal picture should be stated plainly. The objective does not charge for transmitting the fragments themselves; the strict two-part code would add $\sum_i L(g_i \mid O)$, a term that grows with schema richness and would push selection toward smaller schemas, so our criterion is the permissive member of that family, measuring the schema's explanatory value rather than full transmission cost. And $\widehat{I}_\theta$ and L are estimates under one model: we prove no distributional guarantees, and the θ-dependence is measured rather than bounded (§5.4, where selected sizes move by roughly four types across a 1.5B–4B scorer range).

3.4 Search: greedy merging over induced types

One open extraction pass (no schema constraint) over the corpus yields the model's spontaneous relation vocabulary. We keep the most frequent types (cap 60, App. C) as the initial O. We then repeatedly merge the pair of types whose merge maximizes $\Delta\mathrm{Gain}$, relabeling the cached fragments rather than re-extracting, until one type remains, and record the entire trajectory $(|O|, \mathrm{Gain})$ (Alg. 1). Candidate pairs are restricted to each type's ten nearest neighbours by sentence-embedding cosine (Reimers & Gurevych, 2019). After a merge, only chunks containing the absorbed type are rescored, and per-chunk scores are cached. The trajectory, not just its argmax, is the object of study.

3.5 Cost

Scoring needs forward passes only, so θ can be small (3–4B), and it is batched. Extraction runs once per corpus and is reused by every candidate schema, every baseline, and every ablation arm. The full 19-ontology sweep (extraction, search, scoring) takes roughly four hours on one RTX 4080.

4 EXPERIMENTAL SETUP

We evaluate on the DBpedia-WebNLG half of Text2KGBench (Mihindukulasooriya et al., 2023): 19 corpora, 4,860 sentences in total (92–398 per corpus), each paired with a human-authored ontology of 18–68 relation types. That number is the corpus's *gold size*, k_{gold}. This half has a property our question requires: the WebNLG sentences are human-written verbalizations of the reference triples, so the gold schema covers the text *by construction*. The reference is complete, and agreement with its size is a fair test. The benchmark's other half (Wikidata-TekGen) lacks this property. It derives from a distant-supervision corpus (Agarwal et al., 2021) whose creators state it "should not be used as gold standard for evaluation" (Google Research, 2021), and its references are narrow domain slices of 4–15 Wikidata properties. Consistent with prior work (Abolhasani et al., 2026), we exclude it from quantitative comparison. App. L reports it as a qualitative case study, where it behaves exactly as this difference predicts.

The extractor is Qwen3-4B-Instruct-2507 (bf16) and the scorer θ is Qwen2.5-3B-Instruct (fp16), both open-weights with pinned revisions (Yang et al., 2024; 2025). The main result uses greedy decoding. Variance is estimated by resampling extraction at temperature 0.7 with three seeds. Prompts and serialization templates are in App. B.

Two potential confounds are held fixed in every condition: chunking (one sentence) and entity matching (exact strings, no resolution). Both change Gain on their own and would confound the schema comparison. The extraction and scoring prompts are identical across all schema sizes. A schema enters only through the relation list and the relabeled fragments.

The comparison selectors, the constant anchor, and the random-merge ablation are defined in App. G.

For each selector we report mean absolute error $|k - k_{\text{gold}}|$ and Pearson correlation with the gold sizes across the 19 ontologies. Under resampling, we report mean and range over the three seeds.

5 RESULTS

5.1 THE INTERIOR OPTIMUM EXISTS

Every gain trajectory has an interior optimum. Figure 2 overlays the full trajectory of all 19 corpora on a common axis: schema size is expressed relative to each corpus's gold size, which therefore sits at x=1 on all curves. Each curve rises as redundant induced types merge (e.g. `wasCreatedBy` into `createdBy`), peaks at an intermediate size, and collapses once merges destroy real distinctions. The final forced merges each cost hundreds of nats. The per-corpus peaks (dots) cluster around that line, with a median at $0.92\,k_{\text{gold}}$. Per-corpus curves in original units appear in App. D (Fig. 5).

5.2 THE OPTIMUM AGREES WITH EXPERT INVENTORIES

Where experts wrote complete inventories, the optimum lands close to their size. Figure 3 plots the peak size against the gold size. Under greedy decoding the correlation is $r = 0.70$ (Spearman $\rho = 0.66$; bootstrap 95% CI $[0.34, 0.90]$; 0.69 after controlling for corpus size, App. D) with mean absolute error 7.3 types. At $n = 19$ that interval is wide, so the direction and the ablations below carry the argument, not the point value. The gain at the gold size is within 2% of the peak's range on 13 of 19 corpora. Under extraction resampling (Table 5) the correlation is 0.66 with a tight range (0.60–0.71), so the agreement is not an artifact of one decoding path. We read this agreement as construct-validity evidence: SchemaGain tracks what experts judge, without assuming that their counts are exact.

5.3 THE OBJECTIVE, NOT THE MACHINERY

The agreement comes from the objective, not from the machinery around it. Choosing merges *at random* while keeping the entire scoring and search apparatus intact destroys it ($r = 0.12$). All six baseline selectors have larger error than SchemaGain in each individual seed, and no executable baseline or count-only selector matches its correlation on average (Table 5 in App. G).

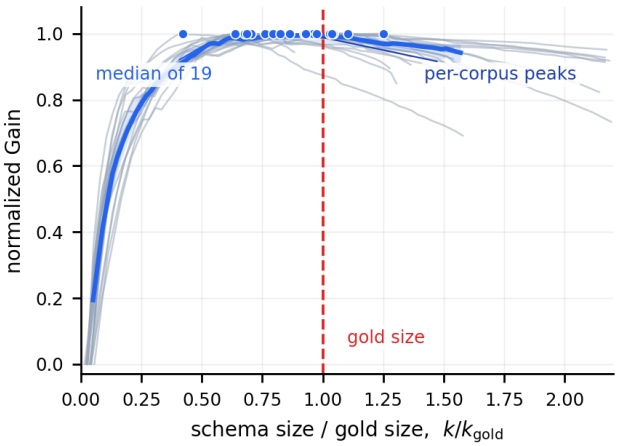

Figure 2: All 19 gain trajectories on one axis. x is schema size relative to the gold size; y is per-corpus min–max-normalized Gain. Gray: individual corpora; blue: median with interquartile band; dots: each corpus's peak; dashed red: the gold size. Every corpus has an interior optimum, and the peaks cluster near it.

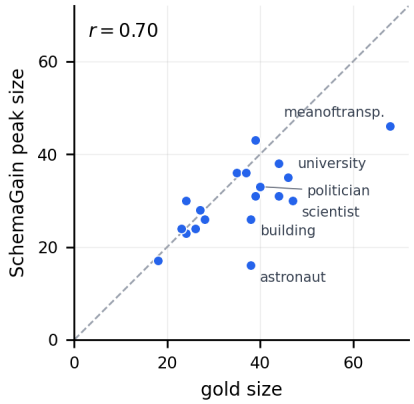

Figure 3: SchemaGain peak size vs. gold size, one point per corpus (greedy decoding); the six labeled corpora are the misses discussed in §6.

The simpler selectors fail in three recognizable ways: their granularity comes from a constant rather than from the corpus, they are blind to size, or, for the LLM family, they are unreliable. A constant sets the granularity of the strongest baseline, a fixed-threshold cut of hierarchical agglomerative clustering (threshold-HAC, $r = 0.55$). It is credible: at $n = 19$ a paired test does not separate it from SchemaGain, and lower thresholds push its error below SchemaGain's, while its correlation stays lower at every threshold (Tables 6 and 7). Three selectors are blind to size. Silhouette is size-*insensitive* (19–33 types whether the human chose 23 or 68). The gap statistic, the textbook selector, fails outright ($r = -0.18$), and the frequency elbow collapses to 2–17 types. Asked for the schema (LLM-direct), the LLM barely canonicalizes (keeping 40–59 of 60 relations, with r swinging from -0.23 to $+0.64$ across samples). Asked for just the number (LLM-count), it is mediocre ($r = 0.29$). The best constant in hindsight, k=38 (the median gold size), beats every baseline on error (MAE 8.6) and comes within 0.2 types of SchemaGain's seeded mean. It sees the gold sizes, so it is a diagnostic anchor, not a competitor. It shows how little corpus signal several selectors use, and why correlation, which no constant can earn, carries the comparison.

A selector must also be stable, because LLM graph construction is run-dependent. Identical-parameter reruns agree at lexical Jaccard ≈ 0.33 even at temperature zero (Giordano & Razniewski, 2025), and different models on one corpus overlap at edge Jaccard 0.06–0.11 (Xie et al., 2026). SchemaGain's correlation moves least across extraction seeds, by 0.11. Threshold-HAC's moves by 0.14, silhouette's halves relative to its greedy value, and LLM-direct's spans 0.87. To our knowledge, SchemaGain is the first schema-size selector shown to be stable under LLM extraction noise.

5.4 Robustness and the role of evidence quality

The optimum is a property of the corpus and schema rather than of the measuring model. With the extractions fixed, scorers of 1.5B, 3B, and 4B parameters (the last from a different model generation) select nearly the same sizes ($r = 0.629$, 0.700, 0.675 and MAE 8.7, 7.3, 7.4). Per-corpus peaks move 3.6–3.8 types on average against plateaus 10–25 wide, and a 1.5B scorer suffices to find the optimum. Serialization behaves similarly: rewriting fragments in a different template shifts absolute gains but preserves the conclusion ($r = 0.613$, MAE 10.6). It does contribute modest variance, which argues for reporting the peak plateau rather than a single argmax. Expectations for the 4B-scorer and serialization arms were pre-registered.

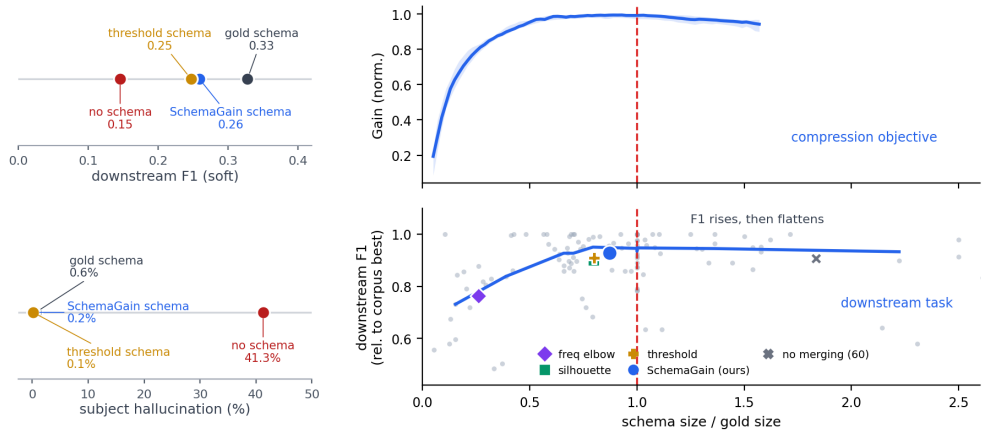

Figure 4: Downstream effects of the schema. Left: swapping the schema source with the extractor fixed (means over 19 corpora): any schema beats none by a wide margin on F1, and the schema constraint collapses subject hallucination. Right: the compression objective (top) and downstream utility (bottom, per-corpus normalized) rise and then flatten on a common size axis, with selector choices marked. The frequency elbow is the only selector below the plateau.

The extractor is where quality matters. On the four corpora where we ran both a 3B and the 4B extractor (scorer fixed), the upgrade moved three peaks toward the gold size and left the fourth in place, cutting mean error from 6.3 to 1.5 types (App. E). Better evidence thus moves selection toward the gold sizes, which suggests that extraction quality, not the objective, limits the agreement.

5.5 What downstream extraction can and cannot show

Downstream extraction confirms that schema constraints matter, while its F1 is too blunt to validate the selected size. The task is schema-guided extraction on the benchmark's held-out test split (2,014 sentences), scored against the ground-truth triples with embedding-matched relation labels and label-free hallucination rates (protocol and full tables in App. H). With the extractor fixed, we swap only the schema *source*. The SchemaGain schema lifts downstream F1 over schema-free extraction by 61% under strict label mapping and 77% under soft matching (0.259 vs 0.146), and it collapses subject hallucination from 41.3% to 0.2%. The gold schema, whose labels are the benchmark's own vocabulary, reaches 0.328 under the same soft matching (Fig. 4, left). The gains over schema-free extraction belong to having a schema, not to how its size was chosen: the threshold-HAC baseline's schema reaches the same mapped F1 (Table 8).

To isolate size, each selector supplies only its chosen k, and we take the schema at that size from the merge trajectory. The resulting utility curve (Fig. 4, right) rises steeply and then plateaus. The frequency elbow's $k \approx 9$ sits about 15% below the plateau, while every other choice, SchemaGain's included, sits on it. Downstream F1 therefore penalizes under-sized schemas but cannot rank the rest. Which size is right must be judged by the agreement of §5.2, not by F1.

6 Analysis and Limitations

When SchemaGain misses, it under-shoots, and only on large ontologies. All six corpora where the gold size falls outside the peak plateau lie *below* the diagonal of Fig. 3, with gold sizes of 38–68 types. Two explanations fit our data. First, SchemaGain is corpus-relative: with 92–398 sentences, rare relation types may not repay their description cost *in this sample*, while the gold size encodes domain knowledge beyond it. Second, the extractor may under-produce rare relations, hiding evidence the criterion never sees. Both are at work (App. F): on the six misses the criterion merges away 10% of the gold relations, while 41% never reach the search, lost to extraction or to the

vocabulary cap. Raising the cap moves all six peaks toward the gold size and then stops (App. C). The first is a property, not a bug: the criterion answers "how many types does *this corpus* justify."

Disagreement with a reference in either direction, under-shoot with thin evidence and over-shoot against thin references, is what a measure of the corpus rather than of the curator should produce. The worst miss (`astronaut`: peak 16, gold 38) shows the under-shoot. The reference is not at fault: the annotations attest the full nominal ontology on 18 of 19 corpora and 44 of 46 relations on the last. One relation (`bornIn`) is 23% of the corpus's extracted triples and the top five are 46%. Of the 38 gold relations, 4 are never extracted, 11 surface only under labels too rare to enter the starting vocabulary, and the criterion merges 9 before the peak (Table 4). At this sample size, rare distinctions cost more to declare than they save, so the criterion's answer of 16 reflects evidence concentration, not a failure to find structure. The TekGen case study (App. L) shows the over-shoot. Where the reference is a known-incomplete slice, the criterion, like every baseline selector, finds *more* structure than the reference annotates, and the excess is attested.

Downstream accuracy validates the selection problem, not SchemaGain's answer. §5.5 shows that the low end of the size axis is harmful and that a common heuristic, the frequency elbow, lands there. Above the floor the task cannot distinguish selectors, so the evidence for SchemaGain's size remains agreement with the gold sizes. Extraction F1 never uses the graph, so it cannot show what too many types cost. A routing benchmark on the same graphs can (App. I): relation-retrieval F1 peaks near 35 types and is 17% lower at 60, where a lookup checks 6.5 labels against 3.5 at 30, and two sampled extraction runs agree on a shared fact's label 88% of the time against 97% at 10 types (App. K). Embedding clustering at matched sizes shows the same shape and random merging does not, so this supports the trade-off, not SchemaGain's size. Costs at scale are unmeasured.

Pretraining contamination is real, but it is not an obvious confound for *selection*. DBpedia derives from Wikipedia and WebNLG has been public since 2017, so we assume our models have seen both: a pure memorisation baseline reconstructs the Wikipedia ontology at 0.837 fuzzy F1 in Lo et al. (2024), and LLMs memorise ontological identifiers in proportion to their web frequency (Bombieri et al., 2024). The exposure plausibly helps the scorer predict sentences, but it applies identically at every schema size and to every baseline. A post-cutoff corpus with an expert ontology would settle it; we know of none.

On scope, our schemas contain relation types only. Entity types enter through domain and range but are not selected. The corpora are short, fact-dense English sentences of one benchmark family, whose text was written from the reference triples. Whether the agreement holds for text written independently of its ontology, or for narrative and argumentative genres, is untested, and we know of no second benchmark with complete expert inventories. One model family provided both extractor and scorer (a second generation was tested as scorer), so cross-family replication is future work.

7 Conclusion

The size of an extraction schema has been left to taste. We have given it a definition. SchemaGain scores a schema by the surprisal its extracted graph removes from the text, net of the schema's own cost, both measured by one frozen language model. Across 19 corpora with expert-sized ontologies, the definition has an interior optimum on every corpus. That optimum agrees with the expert-chosen sizes better than clustering, frequency, or direct LLM judgment, and its agreement varies least under extraction resampling. Downstream extraction confirms that schemas matter but cannot choose their size. Two directions follow. Because the criterion measures the corpus rather than the curator, the gap between its optimum and a benchmark's reference size doubles as an audit of annotation completeness, as our case study observed on references incomplete by construction. And nothing in the argument is specific to relation types. Wherever a discrete vocabulary mediates between data and representation (topics, taxonomies, codebooks), description length under a frozen model prices its size. The schema case is the instance we have now tested.

Reproducibility statement

All models are open-weights with pinned revisions (Qwen3-4B-Instruct-2507 for extraction, Qwen2.5-3B-Instruct for scoring); decoding settings, prompts, and serialization templates are given in the appendix and shipped as machine-readable configuration files. Every result file in our released

code embeds the exact configuration, library versions, git commit, GPU model, and wall-clock time that produced it. The complete experimental suite (main sweep, seeds, baselines, ablations, and the case study) totals roughly 30 GPU-hours on a single RTX 4080 (16 GB); App. N itemizes it. The release also includes the step-by-step reproduction specification the authors themselves worked from, which states the method's invariants (chunk-only scoring, the shared empty-graph template, fixed chunking and entity matching) and the order in which results should be checked, so that the experiments can be reproduced by a human or by an LLM-based coding agent.

THE USE OF LARGE LANGUAGE MODELS

LLMs are the object of study (extractor and scorer) as described in the paper. The research question, the description-length criterion, and the experimental design are the authors' own. An LLM coding assistant was used, under the authors' direction, to implement the experiment code and the analysis scripts that produce the tables and figures, and to draft and edit prose. All experiments were run by the authors on their own hardware; the code was reviewed and tested, and all analyses, claims, numbers, and citations were checked by the authors, who take full responsibility for the content.

REFERENCES

Mohammad Sadeq Abolhasani, Yang Ba, Yixuan He, and Rong Pan. Beyond predefined schemas: TRACE-KG for context-enriched knowledge graph generation, 2026. arXiv:2604.03496; accepted at the Workshop on Graph Foundation Models @ ICML 2026.

Oshin Agarwal, Heming Ge, Siamak Shakeri, and Rami Al-Rfou. Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training. In *NAACL*, 2021. arXiv:2010.12688.

Hamed Babaei Giglou, Jennifer D'Souza, and Sören Auer. LLMs4OL: Large language models for ontology learning. In *ISWC*, 2023. arXiv:2307.16648.

Hamed Babaei Giglou, Jennifer D'Souza, and Sören Auer. LLMs4OL 2024 overview: The 1st large language models for ontology learning challenge. In *Open Conference Proceedings, Vol. 4 (1st LLMs4OL Challenge @ ISWC 2024)*. TIB Open Publishing, 2024. arXiv:2409.10146.

Hamed Babaei Giglou, Jennifer D'Souza, Andrei Aioanei, Nandana Mihindukulasooriya, and Sören Auer. OntoLearner: A modular Python library for ontology learning with large language models. *arXiv preprint arXiv:2607.01977*, 2026.

Jiaxin Bai, Wei Fan, Qi Hu, Qing Zong, Chunyang Li, Hong Ting Tsang, Hongyu Luo, Yauwai Yim, Haoyu Huang, Xiao Zhou, Feng Qin, Tianshi Zheng, Xi Peng, Xin Yao, Huiwen Yang, Leijie Wu, Yi Ji, Gong Zhang, Renhai Chen, and Yangqiu Song. AutoSchemaKG: Autonomous knowledge graph construction through dynamic schema induction from web-scale corpora. In *ACL*, 2026. arXiv:2505.23628.

Weidong Bao, Yilin Wang, Ruyu Gao, Fangling Leng, Yubin Bao, and Ge Yu. DIAL-KG: Schema-free incremental knowledge graph construction via dynamic schema induction and evolution-intent assessment. In *DASFAA*, 2026. arXiv:2603.20059.

Francesco Bariatti, Peggy Cellier, and Sébastien Ferré. GraphMDL: Graph pattern selection based on minimum description length. In *Advances in Intelligent Data Analysis XVIII (IDA)*, volume 12080 of *Lecture Notes in Computer Science*, pp. 54–66. Springer, 2020.

Francesco Bariatti, Peggy Cellier, and Sébastien Ferré. KG-MDL: Mining graph patterns in knowledge graphs with the MDL principle, 2023. arXiv preprint arXiv:2309.12908.

Léonard Blier and Yann Ollivier. The description length of deep learning models. In *NeurIPS*, 2018. arXiv:1802.07044.

Marco Bombieri, Paolo Fiorini, Simone Paolo Ponzetto, and Marco Rospocher. Do LLMs dream of ontologies? *ACM Transactions on Intelligent Systems and Technology*, 15:41:1–41:26, 2024. arXiv:2401.14931.

Paul Buitelaar, Philipp Cimiano, and Bernardo Magnini. *Ontology Learning from Text: Methods, Evaluation and Applications*. IOS Press, 2005. Frontiers in Artificial Intelligence and Applications, vol. 123.

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):btae104, 2024. doi: 10.1093/bioinformatics/btae104.

Hanzhu Chen, Xu Shen, Qitan Lv, Jie Wang, Xiaoqi Ni, and Jieping Ye. SAC-KG: Exploiting large language models as skilled automatic constructors for domain knowledge graphs. In *ACL*, 2024. arXiv:2410.02811.

Grégoire Delétang, Anian Ruoss, Paul-Ambroise Duquenne, Elliot Catt, Tim Genewein, Christopher Mattern, Jordi Grau-Moya, Li Kevin Wenliang, Matthew Aitchison, Laurent Orseau, Marcus Hutter, and Joel Veness. Language modeling is compression. In *ICLR*, 2024. arXiv:2309.10668.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph RAG approach to query-focused summarization, 2024. arXiv preprint arXiv:2404.16130.

Luca Giordano and Simon Razniewski. Foundations of LLM knowledge materialization: Termination, reproducibility, robustness, 2025. arXiv preprint arXiv:2510.06780.

Google Research. The KELM corpus: TEKGEN training corpus. `https://github.com/google-research-datasets/KELM-corpus`, 2021. "[TEKGEN] is a corpus generated with distant supervision and should not be used as gold standard for evaluation.".

Peter D. Grünwald. *The Minimum Description Length Principle*. MIT Press, 2007.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. LightRAG: Simple and fast retrieval-augmented generation, 2024. arXiv preprint arXiv:2410.05779.

Marti A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *COLING*, 1992.

Yassir Lairgi, Ludovic Moncla, Rémy Cazabet, Khalid Benabdeslem, and Pierre Cléau. iText2KG: Incremental knowledge graphs construction using large language models. In *WISE*, 2024. arXiv:2409.03284.

Hang Li and Naoki Abe. Word clustering and disambiguation based on co-occurrence data. In *COLING-ACL*, 1998.

Jiayi Li, Daniel Garijo, and María Poveda-Villalón. Large language models for ontology engineering: A systematic literature review. *Semantic Web*, 17(4), 2026. doi: 10.1177/22104968261465514.

Anna Sofia Lippolis, Mohammad Javad Saeedizade, Robin Keskisärkkä, Sara Zuppiroli, Miguel Ceriani, Aldo Gangemi, Eva Blomqvist, and Andrea Giovanni Nuzzolese. Ontology generation using large language models. In *ESWC*, volume 15718 of *Lecture Notes in Computer Science*, pp. 321–341, 2025. arXiv:2503.05388.

Andy Lo, Albert Q. Jiang, Wenda Li, and Mateja Jamnik. End-to-end ontology learning with large language models. In *NeurIPS*, 2024. arXiv:2410.23584.

Yuxing Lu, Wei Wu, Xukai Zhao, Rui Peng, and Jinzhuo Wang. KARMA: Leveraging multi-agent LLMs for automated knowledge graph enrichment. In *NeurIPS*, 2025. arXiv:2502.06472.

Alexander Maedche and Steffen Staab. Ontology learning for the semantic web. *IEEE Intelligent Systems*, 16(2):72–79, 2001.

Nandana Mihindukulasooriya, Sanju Tiwari, Carlos F. Enguix, and Kusum Lata. Text2KGBench: A benchmark for ontology-driven knowledge graph generation from text. In *ISWC (Resources Track)*, 2023. arXiv:2308.02357.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *ACL*, 2009.

Belinda Mo, Kyssen Yu, Joshua Kazdan, Joan Cabezas, Proud Mpala, Lisa Yu, Chris Cundy, Charilaos Kanatsoulis, and Sanmi Koyejo. KGGen: Extracting knowledge graphs from plain text with language models. In *NeurIPS*, 2025. arXiv:2502.09956.

Alliot Nagle, Adway Girish, Marco Bondaschi, Michael Gastpar, Ashok Vardhan Makkuva, and Hyeji Kim. Fundamental limits of prompt compression: A rate-distortion framework for black-box language models. In *NeurIPS*, 2024. arXiv:2407.15504.

Abdulsobur Oyewale and Tommaso Soru. LLM-driven ontology construction for enterprise knowledge graphs. In *ICSC*, 2026. arXiv:2602.01276.

Xinyu Pi, Qisen Yang, and Chuong Nguyen. LOGOS: LLM-driven end-to-end grounded theory development and schema induction for qualitative research, 2025. arXiv preprint arXiv:2509.24294.

Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. Zep: A temporal knowledge graph architecture for agent memory, 2025. arXiv preprint arXiv:2501.13956.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *EMNLP*, 2019.

Sebastian Riedel, Limin Yao, and Andrew McCallum. Modeling relations and their mentions without labeled text. In *ECML-PKDD*, 2010.

Jorma Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

Ville Satopää, Jeannie Albrecht, David Irwin, and Barath Raghavan. Finding a "kneedle" in a haystack: Detecting knee points in system behavior. In *ICDCS Workshops*, 2011.

Cogan Shimizu and Pascal Hitzler. Accelerating knowledge graph and ontology engineering with large language models. *Journal of Web Semantics*, 85:100862, 2025. arXiv:2411.09601.

Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B*, 63(2):411–423, 2001.

Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method. In *Proceedings of the 37th Annual Allerton Conference on Communication, Control and Computing*, pp. 368–377, 1999. arXiv:physics/0004057.

Shikhar Vashishth, Prince Jain, and Partha Talukdar. CESI: Canonicalizing open knowledge bases using embeddings and side information. In *WWW*, 2018.

Elena Voita and Ivan Titov. Information-theoretic probing with minimum description length. In *EMNLP*, 2020.

Wilson Wong, Wei Liu, and Mohammed Bennamoun. Ontology learning from text: A look back and into the future. *ACM Computing Surveys*, 44(4):20:1–20:36, 2012.

Qiang Xie, Yang Fang, and Yanan Hu. Large language models build inconsistent knowledge graphs from one immune-checkpoint corpus. *Frontiers in Immunology*, 2026. doi: 10.3389/fimmu.2026.1937268.

An Yang, Baosong Yang, Beichen Zhang, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

An Yang, Anfeng Li, Baosong Yang, et al. Qwen3 technical report, 2025. arXiv preprint arXiv:2505.09388.

Bowen Zhang and Harold Soh. Extract, define, canonicalize: An LLM-based framework for knowledge graph construction. In *EMNLP*, 2024. arXiv:2404.03868.

APPENDIX

A THE CHUNK-ONLY SCORING TEST

The failure mode that silently invalidates every number in this paper is scoring the serialized fragment's tokens along with the chunk's. Our unit test scores a three-token toy chunk against a fixed template prefix and verifies four properties token by token. First, exactly the chunk tokens are summed. Second, each chunk token's conditional NLL matches an independent shorter forward pass. Third, extending the chunk leaves the original tokens' contribution unchanged (causality). Finally, batched scoring equals sequential scoring on mixed-length inputs within fp16 kernel tolerance (exact at batch size one, ≈ 0.005 nats/token drift with padding). The test runs against every scoring configuration before any experiment. A bf16 scorer variant was rejected when this check exposed $5\times$ larger batched-kernel noise. Actual output for prefix `"Facts:\nnone\nText:\n"` and chunk `"Paris is nice"`:

```
idx  token             nll  scored?
  0  'F'              (t0)  no
  1  'acts'         4.8799  no
  2  ':\n'          5.5932  no
  3  'none'         9.3512  no
  4  '\n'           3.0347  no
  5  'Text'        15.6836  no
  6  ':\n'          0.7197  no
  7  'Paris'        8.4271  YES
  8  ' is'          0.7420  YES
  9  ' nice'        9.0135  YES

chunk_nll (sum of YES rows) = 18.1827 over 3 chunk tokens
[check 1] independent per-token recomputation: 18.1842  -> PASS
[check 2] different facts prefix changes score: 12.7778 -> PASS
[check 3] extended text, original tokens sum unchanged  -> PASS
[check 4] batched == sequential, max diff 0.0545        -> PASS
```

Check 2 is also a miniature of the paper's central effect: adding the fact `Paris capitalOf France.` to the prefix lowers the chunk's cost from 18.18 to 12.78 nats: conditioning on extracted facts compresses text.

B PROMPTS AND SERIALIZATION TEMPLATES

The open-extraction prompt (the seed pass):

```
Extract knowledge-graph triples from the sentence.
Write one triple per line, exactly in this format:
subject | relation | object
Choose a concise camelCase relation name yourself (e.g. bornIn,
createdBy). Use entity names as they appear in the sentence.
Output only triple lines, nothing else.

Sentence: {sentence}
```

The schema-guided prompt (used for extractor-quality probes; relations listed as `label (domain -> range)`):

```
Extract knowledge-graph triples from the sentence.
Allowed relations, shown as: relation (subject type -> object type).
Use no other relations.
{relation list}
Write one triple per line, exactly in this format:
```

```
subject | relation | object
The subject must be the entity of the relation's subject type and the
object the entity of its object type, even when the sentence mentions
them in the other order. Use entity names as they appear in the
sentence. Output only triple lines, nothing else.

Sentence: {sentence}
```

Output lines without exactly two pipe characters are discarded. In the schema-guided mode, triples whose relation is outside the schema are dropped (case-insensitive match).

The main scoring serialization π uses prefix template `"Facts:\n{facts}\nText:\n"` with one fact per line as `"{s} {r} {o}."`, and the empty graph serializes as `none`. The chunk follows the final newline. $L(O)$ scores the comma-separated relation list against the constant prefix `"The schema relations are:\n"`.

The serializer-ablation variant uses template `"Known facts:\n{facts}\nThe text:\n"` with facts as `"({s}, {r}, {o})"` and empty graph `(none)`.

The LLM-direct baseline's prompt:

```
Below are relation names extracted from a text corpus, with how often
each occurred. Design the schema for this corpus: merge synonyms and
near-duplicates, drop junk, and output the final list of relation
types, one name per line. Use exactly as many types as this corpus
needs - no more, no fewer. Output only the list.

{relation: count, one per line}
```

C THE SEARCH CAP

The starting vocabulary is capped at the 60 most frequent induced types. One ontology (`meanoftransportation`) has a gold size of 68, exceeding the cap, so its trajectory could not reach the gold size. Re-run with a cap of 80, its peak lies at 60 with a 2%-plateau of 45–69, which contains the gold size. The gain at gold is 1.8% below the peak. All other corpora have gold sizes well under the cap.

Having the largest gold size, this corpus also has the most leverage on the correlation of §5.2. Leaving it out lowers r from 0.70 to 0.58, the lowest value in a leave-one-out range of 0.58–0.80 over the 19 corpora. Substituting its cap-80 peak raises r to 0.78. The headline figure uses the uncorrected cap-60 peak.

The cap matters beyond that one corpus. The accounting of App. F finds that a fifth to a quarter of the gold relations surface only under labels outside the 60 most frequent. We therefore reran all 19 searches from 80 types, reusing the same extractions, with expectations pre-registered (Table 1). All six misses move toward the gold size, by 8.5 types on average, and the gold size lies in the 2%-plateau on 16 of 19 corpora instead of 13. Over the 19 corpora r rises from 0.70 to 0.80 (bootstrap 95% CI $[0.51, 0.93]$; 0.69 against 0.58 without `meanoftransportation`) and MAE falls from 7.3 to 5.5. The mean signed error moves from -5.9 to -0.5 types, so the systematic under-shoot of the main configuration is largely a consequence of the cap. The larger cap is not free. The thirteen hits drift upward as well, by 3.9 types on average. Their error rises from 3.6 to 4.3, and `food` now over-shoots its gold size by more than its plateau. The dependence is bounded, however. From 80 to 100 starting types the peaks move by 0.6 on average (mean absolute shift 2.3, within the 3–4 types by which the scorer and serialization arms move them), r is 0.76 (CI $[0.43, 0.91]$), MAE is 6.5, and the gold size lies in the plateau on 17 of 19 corpora. The mean peak is 36 types at both caps, against a mean gold size of 36. The selectors of App. G rerun at cap 80 also improve: threshold-HAC reaches MAE 6.6 and $r = 0.65$, and its paired differences from SchemaGain again include zero (ΔMAE 1.2 $[-2.1, 4.7]$, Δr 0.16 $[-0.27, 0.40]$), while silhouette's correlation falls to 0.25 and LLM-direct keeps 84 labels on average. The main text keeps the pre-specified cap of 60 throughout. Both larger caps were run after the main result was known.

Table 1: All 19 searches rerun from 80 and from 100 starting types instead of 60 (greedy decoding, same extractions). Two corpora have fewer induced types than the cap (`comicscharacter` 75, `celestialbody` 66) and start from all of them. Plateau and gap are for cap 80. Misses of the main configuration are marked with an asterisk.

Corpus	Gold	Cap 60	Cap 80	Cap 100	2%-plateau, cap 80	Gap@gold (%)
comicscharacter	18	17	25	25	17–36	1.2
city	23	24	27	29	18–36	0.9
food	24	30	35	37	27–46	2.6
sportsteam	24	23	25	24	17–36	0.1
monument	26	24	24	22	15–31	0.3
celestialbody	27	28	30	30	19–43	0.2
company	28	26	29	32	18–34	0.3
musicalwork	35	36	38	39	30–53	0.1
athlete	37	36	32	35	28–48	0.6
astronaut*	38	16	24	23	19–27	7.8
building*	38	26	29	31	21–36	2.0
airport	39	31	34	33	18–48	0.9
artist	39	43	48	51	35–63	1.0
politician*	40	33	43	43	29–63	0.2
film	44	38	43	40	31–56	0.1
writtenwork	44	31	48	54	33–61	0.9
university*	46	35	47	39	33–55	0.0
scientist*	47	30	34	36	28–47	1.9
meanoftransportation*	68	46	60	63	45–69	1.8

D PER-ONTOLOGY RESULTS

Figure 5 shows every corpus's trajectory in original units. Table 2 gives the per-corpus numbers behind Figures 2 and 3 (greedy decoding, main configuration). "Gap@gold" is how far the gain at the gold size falls below the peak, as a percentage of the trajectory's gain range; the 2%-plateau is the set of sizes within 2% of the peak.

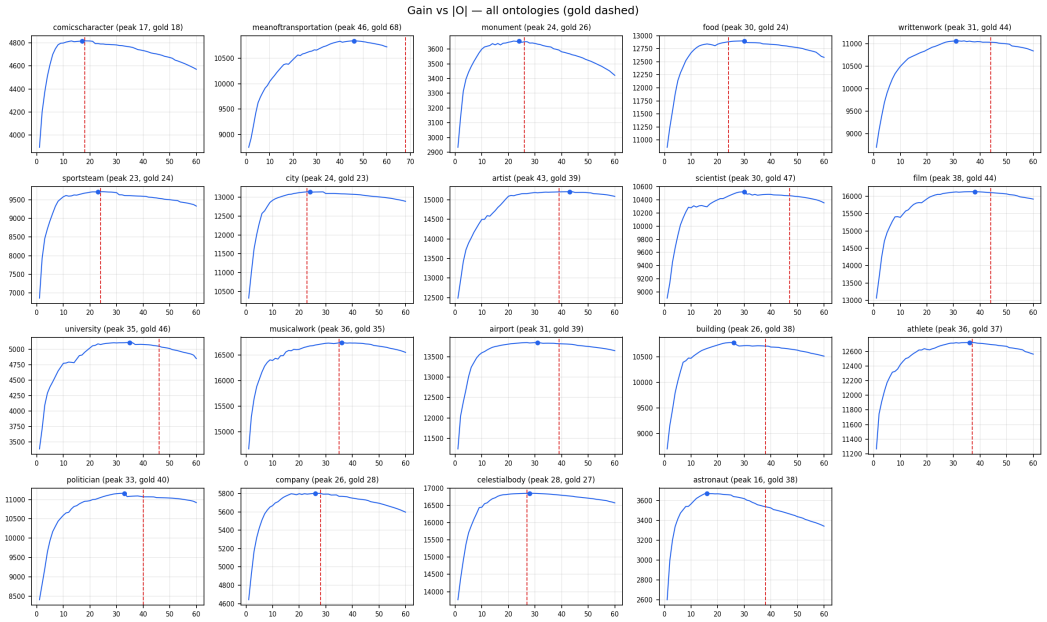

Figure 5: Gain vs. schema size for each corpus in original units (nats). Dots mark the peak; dashed lines the gold size.

Table 2: Per-ontology results (main configuration, cap 60). `meanoftransportation`'s gold size exceeds the cap; see App. C.

Ontology	Sentences	Gold	Peak	2%-plateau	Gap@gold (%)
comicscharacter	102	18	17	11–21	0.0
city	348	23	24	18–35	0.2
food	398	24	30	22–36	0.9
sportsteam	235	24	23	17–30	0.0
monument	92	26	24	20–29	0.8
celestialbody	194	27	28	17–40	0.0
company	153	28	26	16–34	0.0
musicalwork	290	35	36	27–47	0.3
athlete	293	37	36	27–44	0.1
astronaut	154	38	16	14–25	12.6
building	275	38	26	21–26	3.4
airport	306	39	31	19–44	1.1
artist	386	39	43	25–53	0.2
politician	319	40	33	27–33	3.3
film	264	44	38	26–49	0.9
writtenwork	322	44	31	27–46	1.3
university	156	46	35	23–41	3.5
scientist	259	47	30	27–30	3.8
meanoftransportation	314	68	46	38–53	n/a

The correlation interval quoted in §5.2 is a percentile bootstrap over the 19 corpus pairs in Table 2 ($B = 20{,}000$ paired resamples): Pearson $r = 0.70$ with 95% CI $[0.34, 0.90]$, Spearman $\rho = 0.66$ with 95% CI $[0.30, 0.88]$. Nineteen corpora place a wide interval around either coefficient, which is why the agreement is reported as construct-validity evidence rather than as a measured effect size.

Corpus size does not explain the agreement. Larger corpora could in principle have both larger ontologies and larger peaks, but the gold sizes are nearly unrelated to the number of sentences ($r = 0.28$, $p = 0.25$), and the partial correlation between peak and gold size, controlling for sentences, is 0.69 ($p = 0.001$) against 0.70 uncontrolled. The peaks themselves do grow with corpus size ($r = 0.57$, $p = 0.01$). For a corpus-relative definition that is expected, since with the extractor fixed more text supports more distinctions. It also means that peak sizes should not be compared across corpora of different sizes.

E Extractor upgrade

Four corpora were run end to end with two extractors, Qwen2.5-3B-Instruct and the Qwen3-4B-Instruct of the main configuration, under the same scorer, prompts, cap, and search (Table 3). The four were picked for a pilot before the 4B extractor was introduced, so they were not selected on the outcome. All have small gold ontologies, and none is among the six misses of §6. The three peaks that moved belong to the corpora where the better extractor lowered $L(C \mid O)$ most. The 3B-extractor runs predate batched scoring; the two scoring paths agree within the tolerance of App. A.

Table 3: Effect of upgrading only the extractor (scorer fixed, greedy decoding). $\Delta L(C \mid O)$ is the change in the cost of the text given its graph at the starting size of 60 types.

Corpus	Gold	Peak, 3B extractor	Peak, 4B extractor	$\Delta L(C \mid O)$
comicscharacter	18	13	17	-10.8%
monument	26	18	24	-7.5%
celestialbody	27	17	28	-12.3%
company	28	26	26	-1.7%
mean $\lvert\text{peak} - \text{gold}\rvert$		6.3	1.5	

F WHERE THE GOLD RELATIONS GO

To see why the criterion under-shoots on the six misses, we follow every gold relation through the pipeline and record the first stage at which it stops being a type of its own (Table 4). Gold and extracted triples are paired within each sentence by matching their entities after normalization, in either direction, and a gold relation's label is the induced label most often paired with it. The pairing is heuristic (it covers 69–89% of the gold triples per corpus), so the counts are estimates, computed identically for all 19 corpora.

On the six misses, 41% of the gold relations never reach the search. A few are never extracted (6%). Most of the rest are extracted only under labels too infrequent to enter the 60-type starting vocabulary (27%), and some share their induced label with another gold relation (8%). The criterion itself merges a further 10% before the peak, the same share as on the thirteen hits, and 43% of the misses' gold relations have fewer than five gold triples in their corpus. Both explanations of §6 are therefore at work, and the upstream loss is the larger one: on `university` and `meanoftransportation` the criterion merges none of the gold distinctions it is given. `astronaut` and `scientist` are the two corpora where its own merges matter most. The hits lose gold relations in the same ways, only less often upstream (33%). Their peaks match the gold sizes in number while holding other induced types in place of the lost ones, which is why we claim agreement in size and not in content.

Table 4: First stage at which each gold relation stops being a type of its own (greedy decoding, cap 60). *Never*: no extracted triple is paired with it. *Out of cap*: extracted only under labels outside the 60 most frequent. *Shared*: its induced label is also another gold relation's. *Merged*: distinct at the start, merged with another gold relation's label before the peak. *Kept*: a type of its own at the peak. Rows sum to the gold relations attested in the corpus's annotations: the gold size, except for `university`, where 2 of 46 never appear in a gold triple.

Corpus	Gold	Peak	Never	Out of cap	Shared	Merged	Kept
astronaut	38	16	4	11	1	9	13
building	38	26	1	9	4	6	18
politician	40	33	3	8	4	4	21
university	46	35	2	12	6	0	24
scientist	47	30	2	10	3	8	24
meanoftransportation	68	46	4	26	4	0	34
six misses, share of gold			6%	27%	8%	10%	48%
thirteen hits, share of gold			4%	20%	9%	10%	57%

G SELECTOR COMPARISON TABLE

As comparison points, all selectors consume the *same* cached open extraction, capped identically, and output a chosen size k. Two selectors cluster relation embeddings agglomeratively and choose k by silhouette (Vashishth et al., 2018; Rousseeuw, 1987) or by the gap statistic (Tibshirani et al., 2001). A third cuts the same clustering at a fixed cosine distance of 0.5. Vashishth et al. (2018) tune this threshold on held-out gold clusters, which no selector here may see. We therefore fix it between the two published values we know of: the default of their released code (0.42) and the WordNet-synonym calibration of Lo et al. (2024) for the embedding model used here (a similarity of 0.436, hence a distance of 0.56). Table 7 reports both. A fourth takes the elbow of the rank-frequency curve via knee detection (Satopää et al., 2011). Two more show the extractor LLM the induced relations with their counts. *LLM-direct* asks it to write out the final schema and takes the number of types it lists as k. *LLM-count* asks it only for the number of types the schema should have. Across all 19 corpora and four extraction runs it only ever answered 18, 20, or 30. The last entry is not a selector but a constant anchor: the best constant in hindsight, k=38, the median gold size, which minimizes MAE over the 19 corpora. It is the only entry that sees the gold sizes. As a mechanism ablation, the greedy search is rerun with the merged pair chosen uniformly at random (three seeds), leaving everything else identical.

Table 5 gives the full seeded comparison. We also attempted EDC's published self-canonicalization (Zhang & Soh, 2024) as an existing-system baseline. Its released configuration requires three 7B-class LLM roles plus a 7B embedding model, which exceeds our single-GPU budget. Substituting smaller models would no longer measure the published system. Comparative claims in §5.3 are therefore scoped to the executable baselines evaluated here.

Table 5: Schema-size selection across 19 ontologies, aggregated over three sampled extraction seeds (mean, with min–max range). All selectors consume the same extractions.

Selector	MAE ↓	Pearson r ↑
SchemaGain peak (ours)	**8.4** (8.0–8.6)	**0.664** (0.598–0.708)
constant k=38 (hindsight anchor, sees gold)	8.6	n/a
threshold-HAC	10.4 (10.3–10.4)	0.550 (0.476–0.615)
silhouette clustering	11.7 (11.2–12.6)	0.378 (0.254–0.449)
LLM-count	12.5 (11.3–13.4)	0.285 (0.157–0.457)
LLM-direct	19.0 (13.3–26.6)	0.156 (−0.228–0.640)
gap statistic	25.2 (24.8–25.5)	−0.182 (−0.248–−0.077)
frequency elbow	27.5 (27.5–27.6)	0.266 (0.209–0.344)
Counts only, no LM scoring		
types with frequency ≥ 5	11.8 (11.6–12.2)	0.274 (0.265–0.290)
types with frequency ≥ 3	12.4 (11.7–12.8)	0.438 (0.411–0.469)
types with frequency ≥ 2	20.2 (19.8–20.5)	0.482 (0.478–0.486)
knee of the type–token curve	21.9 (21.1–22.8)	0.262 (0.189–0.341)
distinct induced types, uncapped	113.9 (113.6–114.5)	0.561 (0.557–0.569)
Counts rescaled with the gold sizes (leave-one-out)		
c·(distinct induced types)	8.5 (8.5–8.6)	0.505 (0.502–0.510)
$c \cdot \sqrt{\text{sentences}}$	8.6	0.205
random merge (ablation)	23.2	0.123

The two count blocks of Table 5 ask what LM scoring adds over counting. Counting the induced types that occur at least m times, or taking the knee of the type–token curve, gives neither the scale nor the ordering ($r \leq 0.48$). The raw number of distinct induced types orders the corpora better ($r = 0.56$), but it is not a size: it averages 150 types, more than four times the gold sizes. Rescaling a count needs a constant, and fitting that constant needs the gold sizes. We fit it leave-one-out for the vocabulary size and for $\sqrt{\text{sentences}}$. Like the hindsight constant, both then match SchemaGain's error (8.5 and 8.6 against 8.4) while correlating less (0.51 and 0.21 against 0.66). Error is cheap once the typical size is known. SchemaGain reaches the same error with no fitted constant and the highest correlation, although at $n = 19$ the paired interval for its correlation advantage over the raw vocabulary size includes zero (0.10, $[-0.11, 0.49]$).

Table 6 tests the differences in Table 5 with the corpus as the unit of analysis. For each corpus we take the absolute error $|k - k_{\text{gold}}|$, averaged over the three seeds, and report the mean paired difference from SchemaGain with a percentile-bootstrap 95% interval over the 19 corpora ($B = 20{,}000$) and a Wilcoxon signed-rank test. The same resamples give an interval for the difference in correlation. SchemaGain's error is reliably lower than that of LLM-count, LLM-direct, the gap statistic, and the frequency elbow, and lower than silhouette's by a margin whose interval just excludes zero. Against threshold-HAC both intervals include zero. Under greedy decoding the error difference from threshold-HAC is 3.5 types ($[0.6, 6.4]$, $p = 0.04$), while the correlation difference (0.26, $[-0.25, 0.61]$) again includes zero. Nineteen corpora therefore do not separate SchemaGain from the strongest clustering baseline on accuracy alone under resampling. The hindsight constant is likewise indistinguishable from SchemaGain on error, one more reason the argument does not rest on error.

Threshold-HAC's selected size is set largely by its constant. Table 7 cuts the same clustering at five thresholds. The mean selected size falls from 38 to 16 types as the threshold rises from 0.3 to 0.7, and the error moves with it, from 7.6 to 20.1. At 0.3 and 0.4 the error is below SchemaGain's 8.4: the mean selected size then sits near the median gold size, which is also what makes the hindsight constant accurate. At the two published values the error is 8.4 and 13.3 and the correlation 0.50

Table 6: Paired differences from SchemaGain over the 19 corpora (sampled extraction, errors averaged over three seeds). ΔMAE is the baseline's error minus SchemaGain's, and Δr is SchemaGain's correlation minus the baseline's, so positive values favor SchemaGain. Brackets are percentile-bootstrap 95% intervals.

Baseline	ΔMAE	Wilcoxon p	Δr
constant k=38 (hindsight)	$0.2\ [-3.6, 4.0]$	0.87	n/a
threshold-HAC	$2.0\ [-0.8, 4.8]$	0.19	$0.11\ [-0.22, 0.39]$
silhouette clustering	$3.3\ [0.6, 6.1]$	0.05	$0.29\ [-0.13, 0.65]$
LLM-count	$4.1\ [1.7, 6.7]$	0.008	$0.38\ [0.07, 0.61]$
LLM-direct	$10.6\ [4.2, 17.4]$	0.009	$0.51\ [0.17, 0.72]$
gap statistic	$16.8\ [11.1, 22.2]$	< 0.001	$0.85\ [0.57, 1.18]$
frequency elbow	$19.1\ [15.5, 22.7]$	< 0.001	$0.40\ [0.08, 0.86]$

and 0.52. At no threshold does the correlation reach SchemaGain's 0.66, and 0.5 is the threshold at which it is highest. The clustering sets a scale through its constant but follows the corpus-to-corpus variation in gold sizes less closely, and choosing the threshold by its error would require the gold sizes.

Table 7: Sensitivity of threshold-HAC to its cosine-distance threshold (mean over three sampled extraction seeds, range in parentheses). The comparison in Table 5 uses 0.5, fixed without gold access between the two published values.

Threshold	Mean k	MAE $\downarrow$	Pearson r $\uparrow$
0.3	37.9	7.6 (7.6–7.7)	0.374 (0.335–0.439)
0.4	32.8	7.8 (7.4–8.5)	0.528 (0.438–0.593)
0.42 (CESI code default)	31.6	8.4 (8.2–8.7)	0.504 (0.476–0.526)
0.5 (ours)	27.0	10.4 (10.3–10.4)	0.550 (0.476–0.615)
0.56 (WordNet synonyms, Lo et al., 2024)	23.1	13.3 (13.1–13.5)	0.516 (0.441–0.624)
0.6	21.0	15.1 (15.0–15.2)	0.514 (0.403–0.596)
0.7	16.0	20.1 (20.0–20.3)	0.428 (0.343–0.482)

H DOWNSTREAM EVALUATION DETAILS

The protocol is schema-guided extraction on the 2,014-sentence test split with the main extractor, greedy decoding, and a name-only prompt shared by every condition (the gold schema's domain/range information is withheld so only the relation inventory varies):

```
Extract knowledge-graph triples from the sentence.
Allowed relations (use no others):
{one relation name per line}
Write one triple per line, exactly in this format:
subject | relation | object
Use entity names as they appear in the sentence.
Output only triple lines, nothing else.

Sentence: {sentence}
```

Predictions are scored per sentence against the ground-truth triples. Subjects and objects match after normalization (lowercase, underscores to spaces, quotes stripped). Relations match two ways. Under *mapped* matching, each induced label is assigned once to its nearest gold label by MiniLM cosine. Under *soft* matching, a prediction matches a gold triple if the raw label's cosine to the gold relation exceeds τ, applied identically to every condition (gold's exact labels score 1). We report $\tau \in \{0.4, 0.5, 0.6\}$, bracketing the threshold calibrated for this embedding model by Lo et al. (2024) (0.436). An initial grid of $\{0.7, 0.8, 0.9\}$ proved miscalibrated for short camelCase labels, scoring below exact mapping, and was replaced. Hallucination rates are label-free: the fraction of predicted subjects (objects) absent from the normalized sentence.

Table 8 compares schema sources. The *threshold* schema names each threshold-HAC cluster by its most frequent member; *SchemaGain* replays the merge trajectory to its peak.

Table 8: Downstream comparison by schema source (means over 19 ontologies).

Schema	F1 (mapped)	soft F1@.5	soft F1@.4	subj. hall.	obj. hall.
gold	0.297	0.322	0.328	0.006	0.092
SchemaGain	0.206	0.208	0.259	0.002	0.039
threshold	0.206	0.203	0.247	0.001	0.039
none (open)	0.128	0.122	0.146	0.413	0.333

In the size-controlled runs, each selector supplies its chosen k per corpus. The schema at every k is the trajectory's schema at that size (nested family, same merge path), so size is the only variable. Table 9 aggregates per method.

Table 9: Downstream utility at each selector's chosen size (trajectory schemas; means over 19 ontologies).

k supplied by	mean k	F1 (mapped)	soft F1@.4
frequency elbow	8.6	0.165	0.222
silhouette	26.8	0.207	0.258
threshold-HAC	26.7	0.207	0.261
SchemaGain (ours)	30.2	0.207	0.259
gold size	35.6	0.209	0.260
cap (60, no merging)	60.0	0.217	0.252

SchemaGain's chosen size is the argmax size condition on 3 of 19 corpora and within 0.02 soft F1 of the best size on 13 of 19. This is the basis for the scope statement in §5.5 that downstream F1 does not separate selectors above the floor.

I A ROUTING BENCHMARK: THE COST OF SIZE AT USE TIME

Extraction F1 never uses the graph, so it cannot show what too many types cost. This benchmark does. For every corpus and every size on the merge trajectory, the cached extracted graph is relabeled with the schema at that size, with no re-extraction, and queried by gold relation. A query names a relation by its phrase ("birth place"), and a router ranks the schema's labels by the cosine of their MiniLM embeddings to that phrase. *Relation retrieval* returns every edge under the router's top label and is scored against the edges that the sentence-level pairing of App. F assigns to that gold relation. *Anchored lookup* starts from a subject and probes its edges label by label in router order until it reaches the paired edge. Only facts present in the graph are queried ($10{,}353$ lookups over 495 gold relations), so size is the only variable.

Table 10 shows a trade-off with an interior optimum. Precision rises with size and saturates near 0.60. Recall peaks at 20 types and then falls as one relation is split across several labels. Their F1 peaks near 35 types and is 17% lower with no merging. The best size is interior on all 19 corpora and averages 33 types, against a mean SchemaGain peak of 30 and a mean gold size of 36. Cost rises throughout: at 60 types the router's first label is right 27% of the time against 51% at 30, and a lookup checks 6.5 labels against 3.5.

Two controls locate the source of this shape. Random merging through the same machinery has no interior optimum: its F1 only falls as types are merged, so merging as such does not help. Complete-link clustering of the label embeddings, cut at the same sizes, does as well as the SchemaGain trajectory (0.49 against 0.48 at 35 types), which is unsurprising for a router built on the same embeddings. The benchmark therefore supports the trade-off itself, that both too few and too many types cost something when the graph is used. It does not single out SchemaGain's merges or its chosen size. The graphs are small (92–398 sentences), so these are shapes, not serving costs at scale. App. J adds two deterministic costs.

Table 10: Routing benchmark, means over 19 corpora (greedy decoding). P, R, F1: relation retrieval through the router's top label on the SchemaGain trajectory. *Clustering*: the same F1 when the schema at each size comes from embedding clustering. *Random*: from random merges (mean of three seeds). *First right* and *labels checked*: anchored lookup on the SchemaGain trajectory. P, R, and F1 are averaged over gold relations within each corpus and then over corpora, so F1 is below the harmonic mean of the P and R columns.

Size	P	R	F1	F1, clustering	F1, random	First right	Labels checked
10	0.24	0.54	0.271	0.280	0.141	0.62	2.0
20	0.42	0.61	0.426	0.428	0.208	0.60	2.7
30	0.52	0.57	0.475	0.486	0.261	0.51	3.5
35	0.55	0.55	0.477	0.488	0.284	0.46	4.0
40	0.57	0.51	0.465	0.485	0.311	0.43	4.4
50	0.60	0.47	0.439	0.450	0.359	0.34	5.5
60	0.60	0.40	0.396	0.396	0.396	0.27	6.5

J TWO DETERMINISTIC COSTS OF SIZE

Two further costs can be read off the cached trajectories without running a model (Table 11). The first is prompt length. The relation list adds about 3.3 extractor tokens per type to every schema-guided prompt, roughly 200 tokens per call at 60 types against 100 at 30. This cost is certain but small at these sizes. The second is a proxy for over-segmentation. The share of types supported by fewer than three extracted triples rises from 2% at 10 types to 10% at 30 and 24% at 60, so a quarter of an unmerged schema rests on one or two triples per type. Neither number is a retrieval cost. App. I measures one on these small graphs, and costs at scale remain unmeasured (§6).

Table 11: Schema prompt length and thinly supported types along the merge trajectory (means over 19 corpora, greedy decoding). *Tokens*: length of the relation-name list under the extractor's tokenizer. *Thin types*: share of types with fewer than three supporting triples.

Schema size	Tokens	Thin types
10	33	1.6%
20	66	5.0%
30	98	10.4%
40	130	15.0%
50	162	20.2%
60	197	24.0%

K EXTRACTION SELF-CONSISTENCY

A larger schema also makes extraction less repeatable. For 25 fixed test sentences per corpus and the trajectory's schema at 10, 30, and 60 types, we ran schema-guided extraction twice with sampled decoding (temperature 0.7, two seeds) and compared the runs. Among entity pairs extracted in both runs of a sentence, the share given the same relation label falls from 97.3% at 10 types to 93.5% at 30 and 88.0% at 60, and the Jaccard overlap of the two triple sets from 0.90 to 0.85 (Table 12). Agreement is lower at 60 types than at 10 on 17 of 19 corpora. Part of the drop is mechanical, since more labels leave more room to differ, but that is the cost being measured: at 60 types one shared fact in eight receives a different label from run to run, against one in 37 at 10. The grid is coarse (three sizes, two runs), so we read the direction, not the slope. The expectation that agreement falls with size was recorded before the run.

Table 12: Agreement between two sampled extraction runs under the same schema, means over 19 corpora. *Label agreement*: share of entity pairs extracted in both runs that received the same relation label.

Schema size	Label agreement	Pairs in both runs	Triple Jaccard
10	97.3%	1,210	0.895
30	93.5%	1,396	0.869
60	88.0%	1,524	0.850

L CASE STUDY: THE DISTANT-SUPERVISION HALF OF THE BENCHMARK

The Wikidata-TekGen half of Text2KGBench cannot serve as a gold standard for our question (§4), for two reasons. First, its sentences come from a distant-supervision alignment whose creators warn against using it for evaluation (Agarwal et al., 2021; Google Research, 2021) (the benchmark itself ships only a small manually-verified subset). Distant supervision's core assumption is also known to fail frequently (Mintz et al., 2009; Riedel et al., 2010). Second, its references are deliberately narrow slices of 4–15 Wikidata properties, never intended to describe all the structure in the text. We therefore use this half the opposite way: as a case study of how the criterion behaves when the reference is *known* to be incomplete. Our pipeline consumes only the raw sentences and the reference sizes. It never uses the noisy sentence–triple alignments.

We apply an identical pipeline, prompts, models, and hyperparameters as the main experiments (cap 60) to all 10 corpora (8,747 unique sentences after removing the duplicates created by TekGen's one-pair-per-line format). We pre-registered two expectations before the run: interior peaks throughout, and peak locations plausibly *above* the narrow references.

Both expectations held (Table 13, Fig. 6). Eight of ten corpora show a clean interior peak. The other two (`movie`, `music`) are flat-topped. At a search cap of 80, movie's 2%-plateau reaches the edge of the searched range and music's argmax moves with the cap, so we report them as censored rather than as peaks. The two censored corpora are also the two largest (1,393 and 1,744 sentences). That more evidence supports more distinctions is itself consistent with the corpus-relative reading. The located peaks sit *at or above* the reference size in 9 of 10 corpora: the criterion finds more structure than the slice annotates, which is the predicted behavior when the reference is incomplete.

Table 13: TekGen case study (identical pipeline, cap 60). †: flat-topped/cap-sensitive; no decisive optimum in the searched range. Reference sizes are narrow Wikidata slices, not complete schemas; no quantitative claim is made from this table.

Corpus	Sentences	Reference slice	SchemaGain peak
computer	476	4	12
space	448	7	15
culture	362	8	13
military	501	9	16
politics	465	9	22
sport	1,075	11	10
book	1,221	12	14
music	1,393	13	49†
nature	1,062	13	23
movie	1,744	15	54†

The extra types are real. For `computer` (reference slice: *developer, creator, platform, operating system*), the 12-type schema at the peak is an interpretable superset of the slice:

developedBy (250) releasedFor (218) releasedIn (141) platform (126) portedTo (100) publishedBy (72) foundedIn (57) isA (54) expansionOf (30) basedOn (21) runsOn (17) designedBy (13)

(merged frequencies in parentheses). Publishers, release dates, ports and expansions are asserted throughout the corpus; the slice simply does not annotate them.

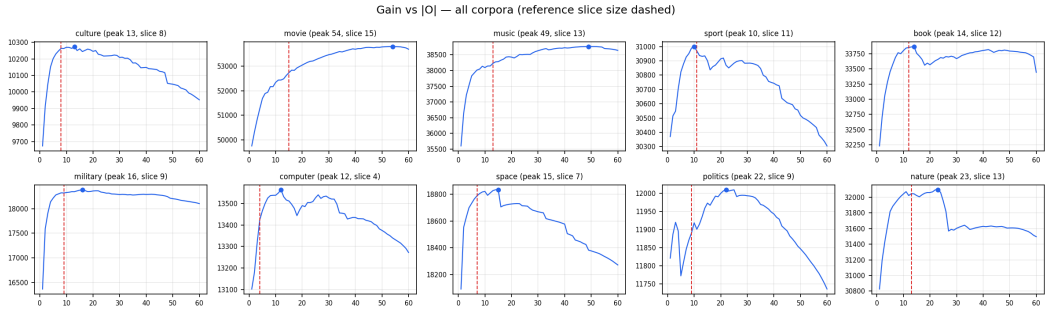

Figure 6: Gain trajectories for the ten TekGen corpora (reference slice sizes dashed). Interior peaks in 8/10; `movie` and `music` are flat-topped near the cap.

Every selector agrees the slice is too small: on `computer`, all seven selectors of App. G choose at least 10 types against the slice's 4 (freq 10, SchemaGain 12, silhouette 27, threshold-HAC 30, LLM-count 30, LLM-direct 54, gap 59). Methods that disagree with each other about everything else agree unanimously that the text contains more than the reference annotates.

We do not score this half as a leaderboard, because distance to an incomplete reference rewards bias, not understanding. The frequency-elbow baseline, for instance, attains the smallest mean distance to the slice sizes (3.5). It does so purely because its small-k bias happens to match small slices, while its correlation with them is -0.04. The same bias placed it last on the complete-reference benchmark (Table 5). For completeness, the SchemaGain peaks correlate with the slice sizes at $r = 0.68$. This value is dominated by the two censored corpora: excluding them gives $r = 0.38$ over a slice range of only 4–13. We therefore draw no conclusion from it in either direction.

In short, where the reference is complete by construction, the criterion recovers the gold size (§5.2). Where the reference is a known-incomplete slice, the criterion, like every other corpus-driven selector, exceeds it, and the excess is interpretable, attested structure. The criterion measures the corpus, not the curator.

M Rate–distortion reading of the objective

The objective of §3.2 admits a rate–distortion reading. Define the distortion of representing chunk c by fragment g as $d_\theta(c, g) := -\log P_\theta(c \mid \pi(g))$, the residual surprisal of the text once its extracted facts are in context. The schema then plays the role of the codebook: enlarging O refines the representable fragments and weakly lowers achievable distortion, while its description cost $L(O)$, the rate side, grows. Maximizing Gain is the Lagrangian trade-off of distortion reduction against rate with the multiplier fixed at one, since both sides are measured in nats by the same θ. The framework is the operational sibling of the information bottleneck (Tishby et al., 1999). Under this reading, the problem of §3.1 is a rate–distortion question. The merge trajectory of §3.4 then traces an empirical rate–distortion curve for the corpus. Each point $(|O|, \mathrm{Gain})$ records the net saving at one rate, and the interior optimum is the rate past which further codebook refinement no longer pays for its own description.

N Compute

All experiments ran on one NVIDIA RTX 4080 (16 GB). In measured wall-clock time, the main 19-corpus sweep took 238 minutes (182.5 extraction, 55.5 search and scoring). The three-seed resampling study took roughly 12 hours including all baselines. Ablation arms reuse cached extractions and cost about one hour each. The TekGen case study (App. L) adds roughly 10 hours. Every released result file embeds its configuration, git commit, library versions, GPU model, and elapsed time.