Which Model, Which Prompt? Selecting and Configuring LLMs as Oracles for Ontology Matching

Jonathon Dilworth[1], Sviatoslav Lushnei[2], Dmytro Shumskyi[2], Severyn Shykula[2], Pedro Cotovio[1,3], Artur d'Avila Garcez[1] and Ernesto Jiménez-Ruiz[1,*]

[1]City St George's, University of London, UK
[2]Ukrainian Catholic University, Ukraine
[3]University of Lisbon, Portugal

Abstract

Large language models (LLMs) are viable oracles for use during interactive ontology matching, but their adoption raises three practical questions: (i) *Which model do we select?* (ii) *How should the prompt be structured?* (iii) *What should the prompt contain?* This work responds to the first question by benchmarking a large selection of language models via LogMap's interactive matching mode on OAEI tasks, recommending models that balance performance, cost, latency, reliability, openness, and provider diversity; we publish the benchmark as a community resource via an online results explorer. For the second and third, we build on prior work, extending existing prompt templates for class, property, and instance matching, investigate zero-shot versus RAG-based few-shot *ontology-driven prompting*, and introduce *collective anchors* for *few-shot-RAG ontology-driven prompting*. We apply these methods to an existing system design, LogMapLLM, and evaluate over five OAEI tracks, spanning a variety of domains. We find that no single configuration consistently wins out, but the oracle's failure mode helps explain which interventions work: demonstrations recover sensitivity in conservative oracles, whereas mutual subsumption improves specificity in permissive ones. Ultimately, the optimal configuration is model-, prompt-, and domain-specific, motivating ongoing work on automatic selection, where the most preferable model $\times$ prompt configuration is recommended given a matching task.

Keywords

ontology matching, large language models, interactive matching, ontology-driven prompts, OAEI

1. Introduction

Ontology matching is vital for semantic interoperability between knowledge representations with distinct terminological, conceptual, and structural properties. Automatic matchers, such as LogMap [1], often rely on lexical and structural heuristics, leading to semantic ambiguities during matching, whose resolution is non-trivial. Interactive matching escalates such cases to an oracle, typically a human expert whose time is often scarce and expensive [2]. Prior work has shown that escalating ambiguous correspondences to a large language model (LLM) instead of a person might be a viable alternative [3]. However, three practical questions emerge. *Which model do we select?* The LLM landscape is broad and volatile; models differ in cost, latency, and reliability. Moreover, when access to model weights is restricted, expected behaviour can change, usage may be repriced, and models can be withdrawn without notice [4], making reproducibility particularly challenging. *How should the prompt be structured, and what should it contain?* Per our earlier work, the oracle only sees entity pairs via serialised *descriptive context*, *i.e.*, the labels, synonyms, and local structure that the prompt draws from the input ontologies. Too little context leaves it guessing from labels, whereas too much may introduce noise and inflate cost. The same can be said of *demonstrative context*, *i.e.*, example correspondences and near-misses that guide the model and condition its response.

In response this work makes three contributions. First, we publish an oracle benchmark of 86

OM 2026: The 21st International Workshop on Ontology Matching collocated with the 25th International Semantic Web Conference (ISWC 2026), October 25th, 2026, Bari, Italy
ernesto.jimenez-ruiz@citystgeorges.ac.uk (E. Jiménez-Ruiz)
https://www.citystgeorges.ac.uk/about/people/academics/ernesto-jimenez-ruiz (E. Jiménez-Ruiz)
0000-0002-9083-4599 (E. Jiménez-Ruiz)

language models over 3,792 runs on Ontology Alignment Evaluation Initiative[1] (OAEI) tasks [5]; we measure discriminative ability, cost, latency, and reliability—justifying our model selection—browsable via an online results explorer [6] (Section 5). Second, using our selected models, we evaluate prompting strategies with fixed *descriptive context*, varying the equivalence form and the *demonstrative context*, over 11 tasks from 3 OAEI tracks—Anatomy [7], Bio-ML [8], and Knowledge Graph [9]; extend our sweep to two additional tracks (Conference [10] and Circular Economy [11]), while building on prior work and introducing additional entity-specific prompt templates for classes, properties, and instances (Section 4.1.1). Specifically, we compare zero-shot and few-shot-RAG-based ontology-driven prompting with collective anchor pools (Section 4). Third, we show how the oracle's failure direction helps explain the effects of prompt strategies: few-shot demonstrations recover sensitivity when oracles are conservative, whereas querying equivalence as mutual subsumption (with two conjunct prompts) trades sensitivity for specificity (Section 6.2). These findings motivate ongoing work on automatic model $\times$ prompt selection for a given pair of input ontologies (Section 7).

2. Preliminaries

Ontology matching. Ontology matching is the task of finding a set of *correspondences*, referred to as an *alignment*, denoted $\mathcal{M}$, between two or more ontologies, such that entities from one ontology *correspond to* those in another [12]. A *correspondence* between a *source ontology* $\mathcal{O}_{src}$ and *target ontology* $\mathcal{O}_{tgt}$ is defined as a tuple $\langle e_{src}, e_{tgt}, r, c\rangle$ where e_{src} is drawn from $\mathrm{Sig}(\mathcal{O}_{src})$ and e_{tgt} from $\mathrm{Sig}(\mathcal{O}_{tgt})$; r is a relation in $\{\equiv, \sqsubseteq, \sqsupseteq\}$, and c is a confidence value between $[0, 1]$. Note that, in this work, we fix r to $\equiv$ and write $\langle e_{src}, e_{tgt}\rangle$ to mean an equivalence correspondence.

Interactive matching. An automatic matching program typically follows a fixed convention (*e.g.*, refusing borderline candidates: "better safe than sorry") on the candidate correspondences that its evidence leaves undecided. As such, *interactive matching* has historically included a *human-in-the-loop* such that a second source of evidence can be considered for such cases.

Oracles. Let $\mathcal{M}_{ask}$ denote the set of *mappings to ask*, *i.e.*, the subset of correspondences that an automatic matcher has generated but cannot fully commit to without further evidence. An *oracle* is a function $\mathrm{Or}\colon \mathcal{M}_{ask} \to \{\text{true}, \text{false}\}$ which the matcher consults, accepting the correspondences the oracle confirms and discarding the rest. Note that a *perfect oracle* (0% error rate) would answer according to *a perfect* reference alignment, should such a thing exist. In this work, we use an LLM as an oracle and score its answers against a published reference alignment $\mathcal{M}_{RA}$ provided by the OAEI.

Evaluation metrics. Oracle quality is measured on $\mathcal{M}_{ask}$ against reference judgements by *sensitivity* (true-positive rate), *specificity* (true-negative rate), and *Youden's index* $J = \text{sensitivity} + \text{specificity} - 1$, which summarises discriminative ability independently of class balance (J=1 is perfect; J=0 is uninformative). While there are no universally accepted cut-off values for J, some studies use thresholds of 0.3, 0.5, and 0.7 to indicate low, moderate, and high effectiveness, respectively (*e.g.*, [13]). The final alignment produced by the matcher is evaluated against the reference alignment by precision, recall, and their harmonic mean F_1. We report F_1 as the downstream (secondary) metric.

3. Related Work

Matching systems have traditionally decided correspondences by assessing lexical and structural evidence. For instance, LogMap [1] and AgreementMakerLight [14] are both long-standing participants in the OAEI campaigns [5], through which standardised benchmarks and evaluation protocols are established. However, when such evidence is inconclusive, human intervention is often necessary. In

[1] https://oaei.ontologymatching.org/

particular, interactive matching treats a domain expert as an authoritative oracle over low-confidence candidates, an arrangement whose cost is expert time [2]. Building on prior work [3], our proposed setting keeps this oracle-centred architecture but replaces the person.

Approaches based on language models have received increasing attention in ontology matching. Early representation-learning methods embed entities using general-purpose word, sentence, or graph embeddings and score candidate pairs by vector similarity. However, because these embeddings lack task-specific signals, they offer only limited improvements over traditional systems [15]. More recent approaches instead learn representations tailored to the matching task, typically by fine-tuning pre-trained encoders, as in BERTMap [16] and BERTSubs [17]. Meanwhile, decoder-based approaches prompt generative LLMs directly, often following a *retrieve-and-select* pattern in which a retriever proposes the top-k target candidates for each source entity and the LLM selects or verifies among them, as in OLaLa [18] and LLMs4OM [19]. Other systems use LLMs to enrich entity descriptions with generated text (*e.g.*, HybridOM [20]) or delegate the matching pipeline to an autonomous agent (*e.g.*, Agent-OM [21]).

Following from our ongoing work [3], we continue to extend LogMap directly, such that LogMapLLM delegates candidate generation, anchor computation, and repair processes entirely to LogMap. The LLM only answers for $\mathcal{M}_{ask}$, the correspondences the matcher itself flags as uncertain, *i.e.*, those that would ordinarily be routed to a human expert during interactive matching. We contribute to the state-of-the-art by addressing the following questions: *(i)* which model do we select when cost, latency, reliability, and weight openness all vary; *(ii)* what descriptive context should the prompt carry for classes, properties and instances; *(iii)* how the equivalence question itself is best posed; and *(iv)* where do few-shot demonstrations come from when no partial reference alignment is available.

4. Methods & System Design: LogMapLLM

LogMapLLM [3, 22] bridges LogMap with a large language model serving as an *oracle*, replacing the role of a human expert during interactive matching. Rather than specify a hard similarity threshold dictating which entities in $\mathcal{O}_{src}$ and $\mathcal{O}_{tgt}$ are matched, the system instead recognises borderline correspondences—candidate matches falling into a particular similarity range. These correspondences are considered *uncertain* and make up the set of *mappings to ask*, $\mathcal{M}_{ask}$. Each candidate correspondence in $\mathcal{M}_{ask}$ is serialised—alongside its descriptive and demonstrative context—into a prompt, and each prompt is routed to an LLM, whose answer (*accept* or *reject*) is folded back into the matching procedure before it produces the final alignment (Figure 1). Thus, the LLM is only consulted when the matcher's own evidence is insufficient to commit to a candidate correspondence confidently.

4.1. Ontology-Driven Prompting

Judging a candidate correspondence $\langle e_{src}, e_{tgt} \rangle \in \mathcal{M}_{ask}$ requires the LLM to combine what it already knows (parametric knowledge) with what the prompt shows it (in-context learning) [23]. *Ontology-driven prompting* [3] controls the latter by drawing the prompt's content from the input ontologies. A context function $\mathrm{ctx}(\cdot)$ maps each entity of the candidate correspondence to a textual serialisation of its annotations and local structure; the assembled query then comprises a task instruction, a few-shot slot holding zero or more demonstrations, the serialisations $\mathrm{ctx}(e_{src})$ and $\mathrm{ctx}(e_{tgt})$ as *descriptive context*, and a closing question asking whether r holds (Figure 2). Parsing the completion into $\{\mathsf{true}, \mathsf{false}\}$ yields the oracle's verdict $\mathrm{Or}(\langle e_{src}, e_{tgt} \rangle)$. The design of $\mathrm{ctx}(\cdot)$ must consider how much of the ontology's contextual information to include. In particular, too little leaves the model guessing, whereas too much dilutes discriminative evidence and inflates cost per query.

Few-shot-RAG ontology-driven prompting. The few-shot slot in Figure 2 is filled by retrieved examples. Following the retrieval-augmented generation pattern [24], the anchor correspondences that LogMap commits to without consultation are embedded into a vector store. At inference time, the

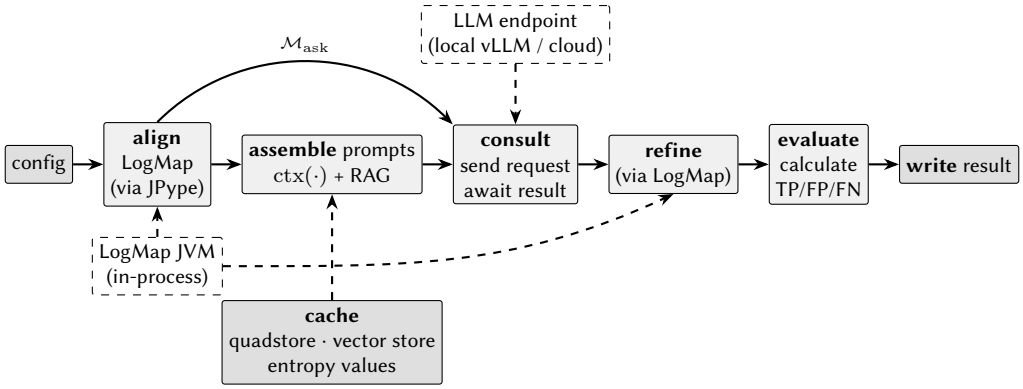

Figure 1: The LogMapLLM pipeline. From a run configuration, **align** runs LogMap in-process (via JPype) to produce an initial alignment and the uncertain set $\mathcal{M}_{ask}$; **assemble** builds one prompt per uncertain correspondence using descriptive and *(optionally)* demonstrative context. Where possible, a cache is maintained, providing efficient access to ontology objects via a quadstore, embeddings via a vector store, and precomputed entropy values; **consult** routes each prompt to an LLM endpoint (a local vLLM instance or external cloud provider); accepted answers are folded into LogMap's **refine** step, and the final alignment is **evaluate**d against the reference before results are written.

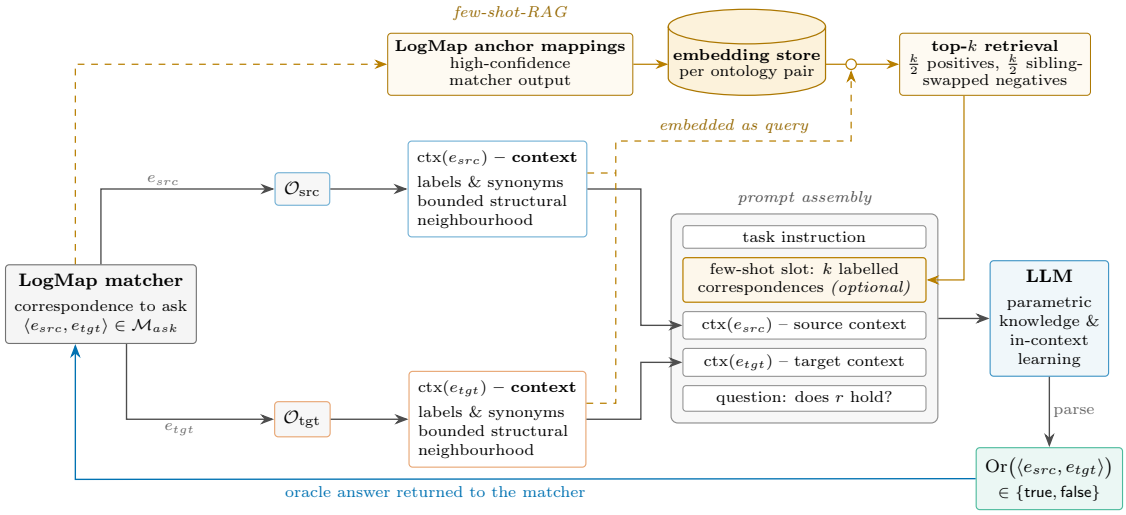

Figure 2: Ontology-driven prompting for a single candidate $\langle e_{src}, e_{tgt}\rangle \in \mathcal{M}_{ask}$. For each entity, $\text{ctx}(\cdot)$ serialises labels, synonyms, and a bounded structural neighbourhood into text. The assembled query places a task instruction first, then the few-shot slot (k demonstrations drawn from LogMap's anchors; empty at $k = 0$), the two entity serialisations, and finally the question of whether r (here, equivalence) holds. The parsed completion supplies the verdict $\text{Or}(\langle e_{src}, e_{tgt}\rangle)$ to the matcher.

current candidate from $\mathcal{M}_{ask}$ is scored against that store, so each prompt receives the demonstrations most similar to the entity pair under judgement. Half of the k examples are retrieved anchors serving as positives, the other half are negatives, produced by corrupting a positive (either by substituting its target with one of the target's siblings, *i.e.*, a *sibling negative*, or by taking the target from a different anchor, *i.e.*, a *donor negative*). Because anchors are the matcher's own high-confidence output, no curated examples are needed and $\mathcal{M}_{RA}$ is never consulted; the method is self-supervised.

4.1.1. Prompt Templates

The prompt assembly contains the system prompt—which we refer to as the task instruction—optional *demonstrative context* (*i.e.*, few-shot examples), entity-specific *descriptive context* drawn from the input ontologies themselves, and the question of whether the relation r holds. The context function $\text{ctx}(\cdot)$ can obtain different types of descriptive context depending on whether the matching system is uncertain

about a class, property, or instance. Thus, the library provides entity-specific prompt templates, of which we evaluate a fixed selection.

Task instruction. The LLM system prompt–also known as the developer prompt–provides the initial task instruction prior to any descriptive or demonstrative context and is the first message in the prompt assembly. It is sometimes necessary to modify the task instruction, since we evaluate over multiple domain ontologies, entity types (*i.e.*, classes, properties, and instances), and relations (see below). Thus, a simple rule specifies the preferred task instruction based on the entity type and correspondence relation. They all share a common shape, however, such that they name the task as an alignment exercise, state which entity type will be presented, warn that additional context may or may not accompany it, and note that responses should be constrained to a single binary token.

Class-based templates. All tasks require at least some class-level correspondences. We organise class-based templates into (i) equivalence-only, and (ii) mutual subsumption—inferred equivalence. Every template operates over the same set of uncertain mappings provided via $\mathcal{M}_{ask}$. They may vary in presentation and may include different structural contexts. Varying the structural context within each prompt tests how well each context guides the model to semantically disambiguate domain-specific terminology. Specifically, we include synonym and parent-based prompts, as well as prompts with additional structural context; an example is shown below.

Class equivalence: labels, synonyms and parents

We have two entities from different ["biomedical", "conference", ""] ["ontologies", "knowledge graphs"]. The first one is "Trapezoid", also known as "Trapezoid bone", "Trapezoid", "Os trapezoideum", which falls under the category "Short Bone". The second one is "Trapezoid", also known as "Trapezium", "Trapezoid", which falls under the category "Quadrilateral". Do they mean the same thing? Respond with ["True" or "False"]["Yes" or "No"].

Class equivalence by mutual subsumption. We include a set of prompts which are used to infer equivalence by means of mutual subsumption. Specifically, two prompts are used to test for subsumption in each direction, and equivalence is decided as the conjunction of the two answers:

$$\mathrm{Or}_{\equiv}(e_{src}, e_{tgt}) := \mathrm{Or}_{\sqsubseteq}(e_{src}, e_{tgt}) \ \wedge \ \mathrm{Or}_{\sqsubseteq}(e_{tgt}, e_{src}). \tag{1}$$

The rule is motivated by the description-logic equivalence $A \equiv B \iff A \sqsubseteq B \wedge B \sqsubseteq A$; note, however, that each conjunct is a fallible LLM judgement rather than a logical assertion; *i.e.*, this is a decision procedure, not a proof. This approach doubles the number of oracle calls per uncertain correspondence, but we hypothesise that it will improve the reliability of LLM responses and can be evaluated at relatively low cost using locally deployed open-weight models.

Class equivalence by mutual subsumption ($A \sqsubseteq B$ case): labels, synonyms and parents

We have two entities from different ["biomedical", "conference", ""] ["ontologies", "knowledge graph"]. The first one is "Trapezoid", also known as "Trapezoid bone", "Trapezoid", "Os trapezoideum", which falls under the category "Short Bone". The second one is "Trapezoid", also known as "Trapezium", "Trapezoid", which falls under the category "Quadrilateral". If something is a "Trapezoid" and a "Short Bone", is it also a "Trapezoid" and a "Quadrilateral"? Respond with ["True" or "False"]["Yes" or "No"].

Property-based templates. The property prompt templates are key when matching ontologies rich in property definitions. Four property templates provide increasing context: labels only; labels with domain and range; labels with domain, range and synonyms (shown below); and a fourth that additionally supplies the inverse property and declared property characteristics.

Property labels with domain, range, and synonyms

We have two properties from different ["biomedical", "conference", ""] ["ontologies", "knowledge graphs"]. The first property is "writePaper" (also known as "writes"), connecting elements of type "Author" (also known as "Writer") to elements of type "Paper" (also known as "Manuscript", "Written Work"). The second property is "is_author_of", connecting elements of type "Person" (also known as "Individual") to elements of type "Contribution" (also known as "Article", "Publication"). Do these properties represent the same relationship? Respond with ["True" or "False"]["Yes" or "No"].

Instance-based templates. These templates focus on asking whether two individuals denote the same real-world entity. Here $\mathrm{ctx}(\cdot)$ typically has more *descriptive context* to leverage than for classes, but also opportunities to include *noisy context*—an individual often has dozens of relational predicates, many of which may be uninformative or insufficiently discriminative for a particular correspondence in $\mathcal{M}_{ask}$. Our instance templates therefore vary along two axes. The first is *breadth*, where we explore the use of labels, labels with asserted types, types with data-property attributes (*i.e.*, relational predicates that resolve to data literals), and *full context*—where we include all of the above, and also object properties with the label of the referenced instance (an example is included below).

Instance labels with full context

We have two entities from different knowledge graphs. The first is "Luke Skywalker", which is a "Characters" and has attributes: is "gender" "Male" and is "era" "Rebellion era" and is "height" "1.72" and has relationships: is "species" "Human" and is "affiliation" "Rebel Alliance" and is "homeworld" "Tatooine". The second is "Luke Skywalker", which is a "Characters" and has attributes: is "gender" "Male" and has relationships: is "species" "Human" and is "profession" "Jedi" and is "homeworld" "Tatooine". Do these refer to the same entity? Respond with ["True" or "False"]["Yes" or "No"].

The second axis is *selection*, where we attempt to filter out noisy context by restricting $\mathrm{ctx}(\cdot)$ to serialise only predicates shared by the two individuals. We also rank those shared predicates by Shannon entropy, keeping only the most discriminative. A full description of this process is provided in Appendix A.

Section 6 fixes labels, synonyms and one level of parents for classes; labels with domain and range, where declared, for properties; and full context with entropy-ranked shared predicates for instances (Appendix A). Fixing the descriptive context across tasks and models allows us to compare demonstrations (*i.e.*, few-shots), as well as the equivalence by mutual subsumption prompting strategy. These experiments do not seek an optimal template; a systematic ablation of the 38-template library is left to future work.

4.2. Demonstrative Context and Collective Anchors

The preceding sections mostly concern descriptive context, that is, what the prompt can express about the two entities of a candidate correspondence in $\mathcal{M}_{ask}$. The few-shot slot (Figure 2) instead carries *demonstrative context*—worked examples that may condition the oracle's judgement. In practice, however, the existence of labelled examples should not be assumed, and a system matching two ontologies ought not to rely on a sampled reference alignment. Demonstrations must therefore be sought from evidence the matcher already produces.

LogMap's *anchors* [25, 22] are that evidence. They are the high-confidence equivalences the matcher commits to without consultation, and are thus well-suited for use as demonstrative context (*i.e.*, as examples of what a correct correspondence looks like). In recent work, our few-shot RAG ontology-driven prompting approach retrieved demonstrations from LogMap's anchors but limited the example space to anchors from the pair of input ontologies under consideration [6]. In this paper, we instead propose *collective anchors*, which pool the anchors of every task (or every other task in the case of

Table 1
Adopted and proposed oracle models. Metrics are provided per model satisfying a failure rate of $\leq 5\%$, with ≥ 30 predictions on HUMAN–MOUSE (254 candidate pairs per run). Cost is token usage priced at live OpenRouter rates, normalised per $1,000$ candidate pairs; time is mean seconds per oracle call where logged. "–" denotes no measurement. † marks model weights that are open, but API calls are used during experiments. Values for cost and latency are provided (where available) regardless of routing.

Model	Weights	J	F_1	Acc.	Fail %	$/1k pairs	s/call	Description
gemini-2.5-flash	API	0.74	0.92	0.89	0.0	0.877	10.3	availability
qwen3.5-122b-a10b	open	0.74	0.92	0.89	0.0	–	–	strong backup
deepseek-v4-flash	open†	0.74	0.92	0.89	0.0	0.028	7.1	cost-effective
qwen3-235b-a22b-2507	open	0.62	0.87	0.83	0.0	0.014	10.6	best open-weight
mistral-small-3.2-24b	open†	0.59	0.88	0.84	0.0	0.011	–	family diversity
gemma-3-27b	open	0.56	0.82	0.78	0.0	0.010	4.3	low-latency
qwen3-8b	open	0.59	0.86	0.81	0.0	–	–	easily self-hosted
llama-3.1-70b	open	0.51	0.87	0.82	0.0	–	–	prior use

`gemini-2.5-flash`'s best configuration uses increased reasoning, which accounts for $\sim 94\%$ of its measured cost. `llama-3.1-70b` and `qwen3-8b` were used in earlier experiments; `qwen3-8b` is easily self-hostable; some `qwen` models exhibited high call-failure rates ($> 80\%$) within our benchmark, however, self-hosted runs were reliable.

"leave-one-task-out" collective anchors), spanning multiple campaigns, into a single corpus, encoded once, from which every query retrieves.

Pooling widens what any query can retrieve. LogMap tags every candidate as a class, object property, data property, or instance, and retrieval is restricted by the type of entity being queried; this ensures that a property-matching example is not fetched for a class-matching prompt, and so forth.

A demonstration (example) *pair* is a positive correspondence followed by a negative. *Donor negatives* take their target entity from another (already existing) anchor, whereas *sibling negatives* use the sibling of the positive's target, falling back to the donor when the target has no sibling. This is most useful in tasks that heavily involve instance matching, such as the OAEI Knowledge Graph track, where siblings are often unavailable. Note that a negative is rejected if it happens to coincide with any known positive in the pooled anchors, and that example demonstrations are presented in a fixed P, N order.

5. Language Model Selection

The LLM oracle introduces a component whose cost, latency, and availability are outside of our control when delegated to a third-party provider; self-hosting open weights is the alternative, which we also explore. LogMap's matching behaviour is influenced by the responses provided by the oracle. As such, model and provider choice directly shapes the final alignment. To inform this choice, we benchmark 86 candidate models across $3,792$ experiment runs on Anatomy's HUMAN–MOUSE task (with 254 candidate pairs per run). We assess four measurable qualities: *(i)* performance in J, F_1, and accuracy, *(ii)* cost, *(iii)* wall-clock time, and *(iv)* reliability. We also consider openness for reproducibility and provider diversity. Our adopted panel is given in Table 1, with the entire benchmark openly published in our repository [6]. Note that Table 1 reports each model under its *best* benchmark configuration; the experiments in Section 6 instead fix a single prompt template per entity type with reasoning disabled (or set to minimal); models are also quantised as per Appendix B. Thus, per-model scores may differ between the two. The benchmark screens models on a single class-only biomedical task, so its ranking does not necessarily transfer to other domains or entity types; we therefore treat it as support for the adopted panel rather than as a general ranking. For example, Gemma 3 27B has a lower benchmark J than Gemini 2.5 Flash, yet it leads the zero-shot KG comparison in Table 2.

Reliability. A run is *reliable* if at most 5% of its oracle calls errored (provider failure or unparseable output) measured over at least 30 predictions. Of the 86 models, 11 produced *no* scoreable prediction,

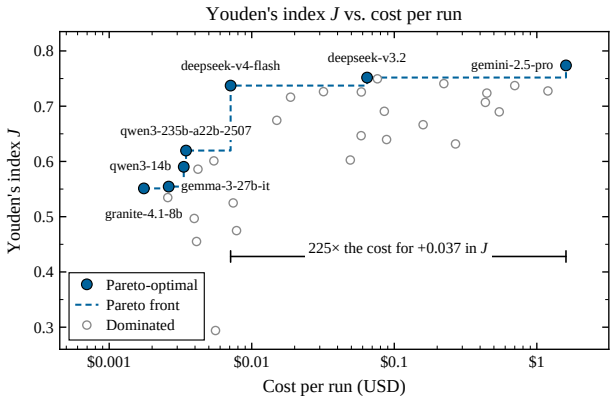

Figure 3: Oracle discriminability against price on HUMAN–MOUSE. Each point is one benchmarked model, plotting Youden's index J against mean cost per run in USD. The dashed line traces the Pareto front over all provider families. Full results available in our repository [6].

and several others post their highest raw scores on runs where 40–70% of calls failed; high scores on low-coverage runs should not be considered reliable.

Cost and timing. We measured the price of token usage at live OpenRouter rates from a single snapshot taken at benchmark time over 244 runs for which usage was recorded. Call latency was logged for 32 models. Across those runs, cost per unit of Youden's index ranges over roughly three orders of magnitude, and Figure 3 plots discriminability against price. The Pareto front rises steeply through the cheap open-weight models and then flattens almost completely.

6. Experiments

6.1. Experimental Setup

Data. Our primary campaign data spans eleven tasks across three OAEI 2025 tracks [5]. Anatomy contributes its single HUMAN–MOUSE task [7]. Bio-ML and KG contribute five tasks each. Anatomy and Bio-ML are class-only, whereas the KG tasks additionally exercise property and instance prompts. Anatomy is reported as a single task, Bio-ML and KG as unweighted macro averages. In total, the oracle adjudicates 259; 4, 103; and 686 candidate pairs across the three primary tracks, retrieving from anchor pools of 1, 145; 17, 910; and 20, 507, respectively; the mean prompt length is 225 tokens at $k = 0$, rising to 504 ($k = 2$) and 783 ($k = 4$); note that figures are quoted for a single negative sampling strategy. Two further tracks are included for domain breadth, at zero-shot only: Conference (21 tasks) and Circular Economy (2 tasks), both of which are micro-averaged due to *(i)* track evaluation conventions, and *(ii)* only a small number of correspondences are escalated to $\mathcal{M}_{ask}$ by LogMap. Where a task leads to an empty $\mathcal{M}_{ask}$ set, sensitivity and specificity are undefined for that task, making a macro average uninformative.

Bio-ML and Anatomy are scored against complete equivalence reference alignments rather than the OAEI leaderboard's semi-supervised protocol, which withholds 30% of the reference as training mappings. Thus, the BioML results are not directly comparable to systems that have been evaluated only on its test set. In addition, the KG references are *partial*, meaning pairs outside the reference are of unknown status, not known negatives. KG specificity is therefore computed only over labelled negatives.

LogMap baseline. Our baseline is LogMap itself, *i.e.*, the end-to-end alignment the matcher produces with no oracle attached. The baseline is measured as a classifier over $\mathcal{M}_{ask}$, such that LogMap (in automatic mode) is scored as predicting a candidate positive if it appears in its final alignment, and

negative otherwise. We deliberately do not tabulate other OAEI systems. The comparison made within this work is between *oracle configurations* inside a fixed matcher (LogMap).

Decoding and serving. All oracle calls use greedy decoding (a temperature of 0.0 and top-p of 1.0) with reasoning disabled in the self-hosted setting and set to the minimum for cloud API calls. Self-hosted models are served via vLLM [26]; model-specific quantisations and hardware specifications are provided under Appendix B. Failed or unparseable oracle calls (which account for $\sim 0.003\%$ of responses) are excluded from oracle-level metrics rather than scored as incorrect.

6.2. Results

Table 2 reports the matched sweep over $k \in \{0, 2, 4\}$. The oracle is asked to respond using a `True` or `False` answer format. k=0 disables retrieval entirely; k=2 and k=4 draw demonstrations using SapBERT [27] for Anatomy and Bio-ML and `all-MiniLM-L12-v2` [28] for Knowledge Graph. Anatomy is a single task; Bio-ML and Knowledge Graph are unweighted task averages. On the class-matching tracks (Anatomy and Bio-ML), zero-shot oracles are already sensitivity-saturated (sensitivity 0.91–0.99), and example demonstrations shift J only modestly (*e.g.*, $0.622 \rightarrow 0.658$ and $0.510 \rightarrow 0.531$). The situation reverses on Knowledge Graph, where several zero-shot oracles are conservative (sensitivity as low as 0.53), and retrieved demonstrations deliver promising sensitivity-driven gains (*e.g.*, Gemini 2.5 Flash rises from 0.534 to 0.937 in sensitivity and from 0.505 to 0.865 in J at $k = 4$ with donor negatives). As expected, the two negative-construction strategies behave similarly throughout. F_1 scores move in the same direction as J, but the overall impact on the matching task depends on the relative size of $\mathcal{M}_{ask}$.

Table 3 extends the zero-shot condition to Conference and Circular Economy (CE). Across all 21 Conference tasks, LogMap escalates just 36 candidates to the oracle—31 of them property correspondences—with five tasks referring none at all, twenty of the twenty-one escalating fewer than five, and the largest (`conference-edas`) escalating seven. Similarly, Circular Economy refers 20, ten from each task. Thus, few-shot example strategies cannot reliably be tested on these pairs. These results serve as boundary observations (cases where an oracle may or may not be beneficial); on CE, LogMap attains $J = 0.778$ and no oracle exceeds it. Meanwhile, on Conference only four of the eight models improve on the baseline in J, three of which also improve F_1.

Table 4 presents the effect of pooling, where each pair of columns shows the same oracle answering the same candidates, differing only in whether its four demonstrations were retrieved from the task's own anchors (*per-pair*) or, under leave-one-task-out, from the pooled anchors of every other task (*pooled*). Note that this ablation selects donor negatives differently to the retrieval procedure tested in Table 2, so absolute values are not comparable between the two tables; the per-pair column is the matched control.

Table 5 compares direct equivalence prompts ("control") against equivalence decided as the conjunction of two directional subsumption prompts ("bidir"). This ablation is conducted at k=0 on the six class-only tasks (Anatomy and Bio-ML).

6.3. Discussion

The LogMap baseline varies across tasks—it retains all 259 escalated candidates on Anatomy, 82% on Bio-ML, and almost none on Knowledge Graph—but its discriminative ability is near zero throughout, with a macro Youden's $J \leq 0.092$ on the three primary tracks. Against that floor, of course, any discriminating oracle is an improvement. The best of the eight models reaches $J = 0.658$ on Anatomy, 0.531 on Bio-ML and 0.865 on the Knowledge Graph track, resulting in best-case improvements of $+0.66$, $+0.53$ and $+0.77$ over the baseline. Meanwhile, the F_1 scores for the same tracks improve in the same direction, but by a much smaller amount: $0.881 \rightarrow 0.898$, $0.700 \rightarrow 0.718$ and $0.781 \rightarrow 0.784$. This gap is expected, since $\mathcal{M}_{ask}$ is small relative to the full alignment, so the oracle's influence on end-to-end F_1 is bounded by the number of candidates it is shown.

Table 2
Oracle sensitivity, specificity and Youden's J on $\mathcal{M}_{ask}$, with final-alignment F_1, measured over 440 runs, 8 models $\times$ 11 tasks $\times$ 5 conditions. Best J within each track and condition is in **bold**. A *perfect oracle* that answers reference-labelled candidates in $\mathcal{M}_{ask}$ according to $\mathcal{M}_{RA}$ gives an approximate F_1 ceiling. Two DeepSeek `ncit-doid` cells are retained as *degraded*, where 1 failed call in 856 (99.88 % coverage) causes a coverage shortfall. Seven of the $201,920$ responses (0.003%) that were returned failed to parse as exactly `True` or `False`.

	Anatomy				Bio-ML				Knowledge Graph			
Model	sens.	spec.	J	F_1	sens.	spec.	J	F_1	sens.	spec.	J	F_1
LogMap	1.000	0.000	0.000	0.881	0.827	0.173	0.000	0.700	0.142	0.960	0.092	0.781
Perfect oracle *(upper bound)*	1.000	1.000	1.000	0.910	1.000	1.000	1.000	0.737	1.000	1.000	1.000	0.785
Zero-shot (k=0)												
DeepSeek V4 Flash (GMICloud FP8)	0.982	0.553	0.535	0.896	0.954	0.436	0.391	0.714	0.618	0.876	0.588	0.782
Gemini 2.5 Flash	0.964	0.638	0.602	0.897	0.918	0.581	0.499	0.717	0.534	0.976	0.505	0.782
Mistral Small 3.2 24B	0.982	0.532	0.514	0.895	0.941	0.429	0.370	0.713	0.560	0.982	0.538	0.783
Qwen3 235B	0.952	0.670	**0.622**	0.897	0.908	0.596	0.504	0.717	0.549	0.988	0.534	0.782
Qwen3.5 122B	0.982	0.638	0.620	0.898	0.934	0.575	**0.510**	0.718	0.727	0.982	0.705	0.784
Llama 3.1 70B	0.970	0.606	0.576	0.896	0.947	0.469	0.416	0.715	0.791	0.902	0.669	0.783
Gemma 3 27B	0.982	0.415	0.397	0.892	0.928	0.412	0.340	0.712	0.913	0.841	**0.839**	0.784
Qwen3 8B	0.988	0.404	0.392	0.892	0.940	0.341	0.280	0.710	0.617	0.959	0.565	0.783
Donor negatives (k=2)												
DeepSeek V4 Flash (GMICloud FP8)	0.945	0.681	**0.626**	0.897	0.898	0.628	**0.526**	0.717	0.685	0.948	0.620	0.783
Gemini 2.5 Flash	0.970	0.596	0.565	0.896	0.919	0.587	0.506	0.717	0.885	0.942	0.813	0.784
Mistral Small 3.2 24B	0.976	0.574	0.550	0.896	0.945	0.439	0.384	0.714	0.909	0.931	**0.822**	0.784
Qwen3 235B	0.976	0.617	0.593	0.897	0.925	0.484	0.409	0.714	0.833	0.942	0.761	0.783
Qwen3.5 122B	0.939	0.660	0.599	0.896	0.918	0.592	0.510	0.718	0.734	0.942	0.662	0.783
Llama 3.1 70B	0.939	0.670	0.610	0.896	0.926	0.555	0.482	0.717	0.876	0.885	0.732	0.784
Gemma 3 27B	0.976	0.521	0.497	0.894	0.913	0.428	0.341	0.712	0.795	0.855	0.614	0.784
Qwen3 8B	0.933	0.543	0.476	0.892	0.897	0.475	0.372	0.713	0.478	0.942	0.406	0.783
Donor negatives (k=4)												
DeepSeek V4 Flash (GMICloud FP8)	0.933	0.681	0.614	0.896	0.915	0.616	**0.531**	0.718	0.735	0.942	0.663	0.783
Gemini 2.5 Flash	0.970	0.628	0.597	0.897	0.920	0.572	0.492	0.717	0.937	0.942	**0.865**	0.784
Mistral Small 3.2 24B	0.970	0.574	0.544	0.895	0.947	0.429	0.376	0.713	0.946	0.931	0.860	0.784
Qwen3 235B	0.976	0.606	0.582	0.897	0.941	0.419	0.361	0.713	0.843	0.942	0.771	0.784
Qwen3.5 122B	0.952	0.702	**0.654**	0.898	0.918	0.595	0.513	0.718	0.893	0.936	0.813	0.784
Llama 3.1 70B	0.958	0.606	0.564	0.896	0.928	0.544	0.472	0.716	0.911	0.925	0.817	0.784
Gemma 3 27B	0.958	0.521	0.479	0.893	0.923	0.412	0.335	0.712	0.954	0.913	0.845	0.784
Qwen3 8B	0.927	0.660	0.587	0.895	0.882	0.519	0.401	0.713	0.669	0.942	0.597	0.783
Sibling negatives (k=2)												
DeepSeek V4 Flash (GMICloud FP8)	0.933	0.691	**0.625**	0.897	0.889	0.622	0.511	0.717	0.693	0.948	0.628	0.782
Gemini 2.5 Flash	0.964	0.617	0.581	0.896	0.914	0.596	0.511	0.717	0.869	0.942	0.796	0.784
Mistral Small 3.2 24B	0.976	0.574	0.550	0.896	0.934	0.463	0.397	0.714	0.909	0.931	**0.822**	0.784
Qwen3 235B	0.958	0.606	0.564	0.896	0.918	0.517	0.435	0.715	0.841	0.942	0.769	0.783
Qwen3.5 122B	0.939	0.649	0.588	0.896	0.905	0.613	**0.517**	0.718	0.885	0.942	0.813	0.784
Llama 3.1 70B	0.939	0.628	0.567	0.895	0.915	0.570	0.486	0.717	0.859	0.879	0.708	0.783
Gemma 3 27B	0.958	0.543	0.500	0.894	0.900	0.466	0.366	0.713	0.937	0.861	0.764	0.784
Qwen3 8B	0.921	0.543	0.464	0.891	0.889	0.501	0.390	0.713	0.528	0.982	0.506	0.783
Sibling negatives (k=4)												
DeepSeek V4 Flash (GMICloud FP8)	0.945	0.713	**0.658**	0.898	0.900	0.628	**0.528**	0.717	0.770	0.942	0.698	0.783
Gemini 2.5 Flash	0.982	0.628	0.609	0.898	0.915	0.594	0.509	0.717	0.887	0.936	0.808	0.784
Mistral Small 3.2 24B	0.970	0.574	0.544	0.895	0.932	0.465	0.397	0.713	0.920	0.931	0.834	0.784
Qwen3 235B	0.976	0.628	0.603	0.897	0.927	0.456	0.383	0.713	0.850	0.942	0.778	0.783
Qwen3.5 122B	0.939	0.702	0.642	0.897	0.899	0.617	0.516	0.718	0.859	0.982	0.837	0.784
Llama 3.1 70B	0.952	0.660	0.611	0.897	0.919	0.578	0.497	0.717	0.894	0.931	0.808	0.784
Gemma 3 27B	0.945	0.606	0.552	0.895	0.901	0.484	0.385	0.713	0.937	0.925	**0.843**	0.784
Qwen3 8B	0.903	0.660	0.563	0.894	0.869	0.551	0.420	0.714	0.636	0.988	0.621	0.783

When is an oracle worth using? To put these gains in context, we estimate the F_1 attainable by a *perfect oracle* by replacing the oracle's answers for reference-labelled correspondences in $\mathcal{M}_{ask}$ with their judgements recorded in $\mathcal{M}_{RA}$. The rest of the alignment remains unchanged; we do not re-run matching or repair. This gives approximate F_1 ceilings of 0.910 on Anatomy, 0.737 on Bio-ML and

Table 3
Domain-breadth beyond bio: 9 model entries × 23 tasks, **zero-shot only**. The oracle instruction is `Respond with True or False`. Both tracks are micro-averaged.

	Conference (21 tasks, 36 candidates)				Circular Economy (2 tasks, 20 candidates)			
Model	sens.	spec.	J	F_1	sens.	spec.	J	F_1
LogMap only *(no oracle)*	0.857	0.545	0.403	0.681	1.000	0.778	0.778	0.656
DeepSeek V4 Flash (GMICloud FP8)	0.929	0.318	0.247	0.677	1.000	0.722	0.722	0.651
Gemini 2.5 Flash	0.857	0.591	0.448	0.681	1.000	0.722	0.722	0.651
Mistral Small 3.2 24B	1.000	0.500	0.500	0.683	0.500	0.833	0.333	0.650
Qwen3 235B	0.857	0.636	0.494	0.682	1.000	0.778	0.778	0.656
Qwen3.5 122B	0.857	0.500	0.357	0.678	1.000	0.778	0.778	0.656
Llama 3.1 70B	0.929	0.409	0.338	0.678	1.000	0.611	0.611	0.641
Gemma 3 27B	0.929	0.545	0.474	0.682	1.000	0.667	0.667	0.646
Qwen3 8B	0.786	0.591	0.377	0.679	1.000	0.722	0.722	0.651

Table 4
Pooled anchors. The *per-pair* column represents examples drawn from the task's own anchors; *pooled* represents examples drawn from pooled anchors of every other task—the *"leave-one-task-out"* configuration. This is applied since intra-task anchors rank most similar and are therefore preferred in most cases over inter-task anchors. Values are Youden's J on $\mathcal{M}_{ask}$, averaged over the tasks in each track. DeepSeek ran Anatomy and one KG task; its incomplete KG result is excluded from the mean. Qwen3 235B did not run this experiment. $*$ marks an exclusion due to task incompleteness and unreliability.

	Anatomy			Bio-ML			Knowledge Graph		
Model	per-pair	pooled	diff.	per-pair	pooled	diff.	per-pair	pooled	diff.
DeepSeek V4 Flash	0.616	0.681	$+0.065$	–*	–*	–*	0.800*	1.000*	$+0.200$*
Gemini 2.5 Flash	0.603	0.604	$+0.001$	0.498	0.500	$+0.002$	0.808	0.815	$+0.007$
Qwen3.5 122B-A10B	0.632	0.640	$+0.008$	0.506	0.506	0.000	0.804	0.861	$+0.058$
Llama 3.1 70B	0.558	0.555	-0.003	0.470	0.457	-0.013	0.824	0.784	-0.041
Gemma 3 27B	0.464	0.461	-0.003	0.324	0.301	-0.023	0.790	0.862	$+0.071$
Mistral Small 3.2 24B	0.529	0.514	-0.015	0.375	0.343	-0.032	0.910	0.867	-0.042
Qwen3 8B	0.563	0.531	-0.032	0.397	0.376	-0.021	0.570	0.525	-0.044
Mean	0.566	0.569	$+0.003$	0.428	0.414	-0.015	0.784	0.786	$+0.002$

0.785 on KG. The best configurations therefore recover roughly half of the available improvement on the biomedical tracks; on KG, even the ceiling is only 0.004 above LogMap. Whether the observed improvements warrant consultation depends on the value of correcting the remaining errors relative to the cost and reliability of oracle calls.

Importantly, there is no one single model or configuration recommendation. Across each track × model × configuration, six of the eight models are *best performers* at least once. Extending the zero-shot condition to two further domains (Table 3) also shows that attaching an oracle is not always worthwhile. For Circular Economy, LogMap alone reaches $J = 0.778$ and none of the eight oracles beats it (two match it exactly and six are worse). For Conference, evaluation is more difficult, since five of its twenty-one ontology pairs escalate zero *mappings to ask*; under its micro-averaged metrics, we see marginal gains in both Youden's J and the corresponding F_1 in three cases. Where LogMap escalates a small number of mappings $\mathcal{M}_{ask}$, and already performs well, an oracle has little room to provide benefit.

Collective anchors. Pooling anchors from different tasks and campaigns does not generally change which anchors are selected during retrieval; this is mostly the case for Bio-ML, where the significant majority of retrieved examples are exactly the same whether you pool anchors or not. This is why we use "leave-one-task-out" (*i.e.*, we collect the anchors from all other tasks), forcing every example to come from another task. We see that this hurts performance on Bio-ML. Anatomy and KG, however, appear to be beneficiaries of this approach, such that pooled anchors drawn from Bio-ML actively

Table 5
Equivalence by mutual subsumption ("bidir", meaning *bidirectional*) versus a single (one-way) equivalence query ("control"), at $k{=}0$ on the six class-only tasks. $\mathcal{M}_{ask}$ is identical within a row; the two configurations differ only in the class prompts. ΔJ is bidirectional minus control.

		Sensitivity		Specificity		Youden's J		
Task	Model	control	bidir.	control	bidir.	control	bidir.	ΔJ
anatomy	Qwen3 8B	0.988	0.952	0.404	0.649	0.392	0.600	+0.208
	Gemma 3 27B	0.982	0.933	0.415	0.596	0.397	0.529	+0.132
	Llama 3.1 70B	0.970	0.915	0.606	0.745	0.576	0.660	+0.084
	Qwen3.5 122B	0.982	0.939	0.638	0.766	0.620	0.705	+0.085
ncit-doid	Qwen3 8B	0.951	0.882	0.303	0.585	0.253	0.467	+0.214
	Gemma 3 27B	0.931	0.802	0.362	0.657	0.293	0.459	+0.166
	Llama 3.1 70B	0.951	0.830	0.463	0.683	0.414	0.513	+0.099
	Qwen3.5 122B	0.904	0.819	0.610	0.811	0.514	0.630	+0.116
omim-ordo	Qwen3 8B	0.988	0.953	0.463	0.731	0.451	0.685	+0.234
	Gemma 3 27B	0.977	0.878	0.485	0.643	0.461	0.521	+0.060
	Llama 3.1 70B	0.994	0.919	0.678	0.872	0.673	0.791	+0.118
	Qwen3.5 122B	0.983	0.913	0.700	0.907	0.683	0.820	+0.137
snomed-ncit.pharm	Qwen3 8B	0.960	0.869	0.304	0.638	0.263	0.507	+0.244
	Gemma 3 27B	0.916	0.865	0.435	0.665	0.350	0.531	+0.180
	Llama 3.1 70B	0.943	0.879	0.454	0.631	0.397	0.510	+0.113
	Qwen3.5 122B	0.939	0.818	0.535	0.712	0.474	0.530	+0.056
snomed-ncit.neoplas	Qwen3 8B	0.942	0.828	0.176	0.408	0.118	0.236	+0.118
	Gemma 3 27B	0.925	0.852	0.276	0.514	0.201	0.366	+0.165
	Llama 3.1 70B	0.956	0.885	0.271	0.484	0.227	0.369	+0.142
	Qwen3.5 122B	0.936	0.839	0.428	0.589	0.364	0.429	+0.065
snomed-fma.body	Qwen3 8B	0.859	0.786	0.457	0.640	0.316	0.426	+0.110
	Gemma 3 27B	0.892	0.775	0.504	0.721	0.396	0.496	+0.101
	Llama 3.1 70B	0.892	0.802	0.476	0.666	0.368	0.468	+0.100
	Qwen3.5 122B	0.911	0.843	0.603	0.754	0.513	0.596	+0.083

help Anatomy in at least three cases (*i.e.*, with three different models). For KG, DeepSeek V4 Flash encountered reliability issues (discussed below), where the provider took the model offline during evaluation, meaning it could only complete one of its five tasks. For the remaining models, a marginal benefit is observed in three cases. On average, however, pooled scores barely shift (mean ΔJ of $+0.003$, -0.015 and $+0.002$ per track). Taken together, task-local anchors dominate retrieval, and cross-task pooling neither reliably helps nor hurts.

Equivalence by mutual subsumption. Equivalence can be tested directly—*do these two classes mean the same thing?*—or as the conjunction of two subsumption questions (Section 4.1.1). While this incurs twice as many oracle calls, effectively doubling the cost, there is a significant benefit realised when applying this technique to tasks that are already sensitivity-saturated, as shown in Table 5. In all twenty-four experiments, specificity rises, sensitivity falls, and Youden's J rises. The mean ΔJ is $+0.127$ on Anatomy (per-configuration range $+0.084$ to $+0.208$) and $+0.131$ on Bio-ML (range $+0.056$ to $+0.244$), which constitutes the largest and most consistent effect observed by a wide margin. Sensitivity is slightly reduced by 0.045 on Anatomy and 0.086 on Bio-ML while specificity rises by 0.173 and 0.216. Requiring two affirmative answers from the oracle makes its decision more conservative and, one might suppose, more logically grounded. However, this does not establish that the LLM reasons faithfully about subsumption; we infer equivalence from the conjunction of its two answers, resulting in fewer accepted correspondences. This trade is especially effective here because these tasks were *sensitivity-saturated.* At zero-shot, the oracle already recovers 0.86–0.99 of the true correspondences in $\mathcal{M}_{ask}$ while its specificity is 0.18–0.70, *i.e.*, there is almost nothing left to gain in one direction and a great deal in the other.

Table 6 summarises the two interventions suggested by the oracle's zero-shot failure direction.

Table 6
Prompt interventions suggested by the observed failure modes. Gains concern oracle judgements; their effect on final-alignment F_1 depends on the available headroom.

Zero-shot behaviour	Intervention	Observed effect and trade-off
Conservative: rejects true correspondences	Few-shot-RAG demonstrations	Sensitivity-driven KG gains depend on model and k; longer prompts (Table 2).
Permissive: accepts false correspondences	Class equivalence by mutual subsumption	Specificity rises and sensitivity falls in all 24 cases; twice the oracle calls (Table 5).

Open-weight model service provider reliability. Every API-routed model within our selection was pinned to a single provider; this ensures that prompt requests flow to the same model, running the same quantisation on similar hardware, between calls. While DeepSeek V4 Flash proved *reliable enough* during benchmarking, prolonged experiments suffered intermittent outages and routing failures. This highlights the reliability trade-off between services such as OpenRouter and self-hosting.

7. Conclusion

This work set out to establish which model to use as an oracle for interactive ontology matching, how to structure its prompt, and what to put in it. First, we contributed an openly published benchmark of 86 models over 3,792 runs, reporting discriminability against cost, latency and reliability, and note that reliability is of particular concern when pinning a specific service provider.

Investigating prompt structure and content, we found that no single configuration wins out consistently, but the oracle's zero-shot failure direction provides a useful starting point for choosing an intervention. Demonstrative context helps chiefly on tasks where oracles are conservative (specificity high), allowing the oracle to recognise true correspondences it would otherwise reject. It remains unclear, however, whether the benefit of in-context examples in this setting comes from showing the oracle the *form* of the task—what a correspondence, a near-miss, and an answer look like—or from task-specific information in the demonstrations themselves. Conversely, querying class equivalence as mutual subsumption traded sensitivity for specificity in every case we measured, greatly improving oracle discriminability on sensitivity-saturated tasks, at the cost of twice as many oracle calls per candidate. Pooling anchors from other tasks offered no average benefit over task-local retrieval. Nevertheless, it showed potential benefits on tasks such as Anatomy, where a larger candidate pool may be advantageous.

Since the optimal configuration is task-, model-, and prompt-specific, our ongoing work investigates automatic model $\times$ prompt selection for a given pair of input ontologies. In particular, we consider predictive selection using ontology-level features (*e.g.*, probing oracle sensitivity and specificity on small task-specific samples), as well as cheaper heuristics to rank configurations based on our benchmark data.

Limitations of this work include the varying coverage between experiments: the benchmark spans 86 models under many configurations, while eight models were evaluated in Tables 2 and 3, seven in Table 4, and four in Table 5. The single-task model benchmark and fixed descriptive templates also limit the extent to which our findings can be generalised. Additionally, the pooling experiment uses a different donor-negative composition than Table 2, so absolute values are not comparable.

Acknowledgments

This research was supported by the RAI for Ukraine program of the NYU Center for Responsible AI, and by Turing Innovations Limited and The Alan Turing Institute's Defence and Security Programme via the project GUARD. It was also supported by FCT through the fellowship https://doi.org/10.54499/2022.10557.BD (Pedro Cotovio), the LASIGE Research Unit, ref. UID/00408/2025 and the CancerScan

project, which received funding from the European Union's Horizon Europe Research and Innovation Action (EIC Pathfinder Open) under grant agreement No. 101186829. Views and opinions expressed are, however, those of the author(s) only and do not necessarily reflect those of the European Union or the European Innovation Council and SMEs Executive Agency. Neither the European Union nor the granting authority can be held responsible for them. We would also like to acknowledge the support of Dr Dave Herron in the creation of the preliminary LogMapLLM's pipeline.

Declaration on Generative AI

The authors used Grammarly for language checks and Google Search with Gemini for alternative phrasing, and reviewed and edited the content as needed, taking full responsibility for its final content.

References

[1] E. Jiménez-Ruiz, B. Cuenca Grau, LogMap: Logic-based and scalable ontology matching, in: International Semantic Web Conference, Springer, 2011, pp. 273–288.

[2] Z. Dragisic, V. Ivanova, P. Lambrix, D. Faria, E. Jiménez-Ruiz, C. Pesquita, User validation in ontology alignment, in: The Semantic Web – ISWC 2016, volume 9981 of *LNCS*, Springer, 2016, pp. 200–217.

[3] S. Lushnei, D. Shumskyi, S. Shykula, E. Jiménez-Ruiz, A. d. Garcez, Large Language Models as Oracles for Ontology Alignment, in: Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers), 2026, pp. 2435–2449.

[4] Anthropic, Statement on the US government directive to suspend access to Fable 5 and Mythos 5, 2026. URL: https://www.anthropic.com/news/fable-mythos-access.

[5] M. A. N. Pour, et al., Results of the Ontology Alignment Evaluation Initiative 2025, in: Proceedings of the 20th International Workshop on Ontology Matching co-located with the 24th International Semantic Web Conference (ISWC), CEUR Workshop Proceedings, CEUR-WS.org, 2025.

[6] City Artificial Intelligence Repository, LogMap-LLM experiments, https://city-artificial-intelligence.github.io/logmap-llm-exp, 2026. GitHub repository, accessed 31 July 2026.

[7] O. Bodenreider, T. F. Hayamizu, M. Ringwald, S. De Coronado, S. Zhang, Of mice and men: Aligning mouse and human anatomies, in: AMIA Annual Symposium Proceedings, volume 2005, 2005, p. 61.

[8] Y. He, J. Chen, H. Dong, E. Jiménez-Ruiz, A. Hadian, I. Horrocks, Machine Learning-Friendly Biomedical Datasets for Equivalence and Subsumption Ontology Matching, in: 21st International Semantic Web Conference, 2022.

[9] S. Hertling, H. Paulheim, The knowledge graph track at OAEI: Gold standards, baselines, and the golden hammer bias, in: European Semantic Web Conference, Springer, 2020, pp. 343–359.

[10] O. Zamazal, V. Svátek, The ten-year ontofarm and its fertilization within the onto-sphere, Journal of Web Semantics 43 (2017) 46–53.

[11] H. Li, E. Blomqvist, P. Lambrix, Initial and experimental ontology alignment results in the circular economy domain, in: KG4S 2024: The 2nd International Workshop on Knowledge Graphs for Sustainability co-located with the 21st Extended Semantic Web Conference (ESWC), Hersonissos, Greece, May 27th, 2024., volume 3753, CEUR Workshop Proceedings, 2024, pp. 79–85.

[12] J. Euzenat, P. Shvaiko, Ontology Matching, 2 ed., Springer, Berlin, Heidelberg, 2013. doi:10.1007/978-3-642-38721-0.

[13] S. Liu, Q. Tian, Y. Liu, P. Li, Joint Statistical Inference for the Area under the ROC Curve and Youden Index under a Density Ratio Model, Mathematics 12 (2024) 2118. URL: https://doi.org/10.3390/math12132118. doi:10.3390/math12132118.

[14] D. Faria, C. Pesquita, E. Santos, M. Palmonari, I. F. Cruz, F. M. Couto, The AgreementMakerLight

ontology matching system, in: OTM Confederated International Conferences" On the Move to Meaningful Internet Systems", Springer, 2013, pp. 527–541.
[15] J. Chen, O. Mashkova, F. Zhapa-Camacho, R. Hoehndorf, Y. He, I. Horrocks, Ontology Embedding: A Survey of Methods, Applications and Resources, IEEE Transactions on Knowledge and Data Engineering (2025) 1–20. URL: http://arxiv.org/abs/2406.10964. doi:10.1109/TKDE.2025.3559023, arXiv:2406.10964 [cs].
[16] Y. He, J. Chen, D. Antonyrajah, I. Horrocks, BERTMap: A BERT-Based Ontology Alignment System, in: Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI), AAAI Press, 2022, pp. 5684–5691. URL: https://doi.org/10.1609/aaai.v36i5.20510. doi:10.1609/AAAI.V36I5.20510.
[17] J. Chen, Y. He, Y. Geng, E. Jiménez-Ruiz, H. Dong, I. Horrocks, Contextual semantic embeddings for ontology subsumption prediction, World Wide Web-internet and Web Information Systems 26 (2023) 2569–2591.
[18] S. Hertling, H. Paulheim, OLaLa: Ontology Matching with Large Language Models, in: Proceedings of the 12th knowledge capture conference 2023, 2023, pp. 131–139.
[19] H. Babaei Giglou, J. D'Souza, F. Engel, S. Auer, LLMs4OM: Matching Ontologies with Large Language Models, in: European Semantic Web Conference, Springer, 2024, pp. 25–35.
[20] M.-H. Totoian, A. Marginean, P. Blohm, M. N. Hussain, HybridOM: Ontology Matching using Hybrid Search, in: OM@ ISWC, 2024, pp. 138–145.
[21] Z. Qiang, W. Wang, K. Taylor, Agent-OM: Leveraging LLM Agents for Ontology Matching, arXiv preprint arXiv:2312.00326 (2023).
[22] E. Jiménez-Ruiz, S. Lushnei, D. Shumskyi, S. Shykula, A. d'Avila Garcez, LogMap Family welcomes LogMapLLM in the OAEI 2025, in: Proceedings of the 20th International Workshop on Ontology Matching, volume 4144 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2025, pp. 178–181. URL: https://ceur-ws.org/Vol-4144/om2025-oaei-paper7.pdf.
[23] Q. Dong, L. Li, D. Dai, C. Zheng, J. Ma, R. Li, H. Xia, J. Xu, Z. Wu, B. Chang, et al., A survey on in-context learning, in: Proceedings of the 2024 conference on empirical methods in natural language processing, 2024, pp. 1107–1128.
[24] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, D. Kiela, Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks, 2021. URL: http://arxiv.org/abs/2005.11401. doi:10.48550/arXiv.2005.11401, arXiv:2005.11401 [cs].
[25] E. Jiménez-Ruiz, B. Cuenca Grau, Y. Zhou, I. Horrocks, Large-scale interactive ontology matching: Algorithms and implementation, in: Europ. Conf. on Artif. Intell. (ECAI), 2012.
[26] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. E. Gonzalez, H. Zhang, I. Stoica, Efficient memory management for large language model serving with pagedattention, in: Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles, 2023.
[27] F. Liu, E. Shareghi, Z. Meng, M. Basaldella, N. Collier, Self-alignment pretraining for biomedical entity representations, in: Proceedings of the 2021 conference of the North American chapter of the association for computational linguistics: human language technologies, 2021, pp. 4228–4238.
[28] W. Wang, H. Bao, S. Huang, L. Dong, F. Wei, MiniLMv2: Multi-head self-attention relation distillation for compressing pretrained transformers, in: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Association for Computational Linguistics, Online, 2021, pp. 2140–2151. URL: https://aclanthology.org/2021.findings-acl.188. doi:10.18653/v1/2021.findings-acl.188.

Appendix

A. Shannon entropy-based predicate filtering for instance matching

For a knowledge graph G and a *matchable* predicate p (whose IRI contains a vocabulary-specific substring, *e.g.*, `/property`), let $n_p(v)$ denote the number of triples $(s, p, o) \in G$ whose object serialises to the string v. Literals and object IRIs are treated uniformly via their string forms, with no further normalisation. The value distribution of p and its Shannon entropy are

$$P_G(v \mid p) = \frac{n_p(v)}{\sum_{v'} n_p(v')}, \qquad H_G(p) = -\sum_{v} P_G(v \mid p)\ \log_2 P_G(v \mid p), \tag{2}$$

where counts range over triple occurrences (a subject asserting several values contributes each of them) and a predicate with a single distinct value has $H_G(p) = 0$. Entropies are computed once per graph and cached (Figure 1). Across the two graphs, predicate identity—both for the shared-predicate restriction and for merging—is established by local name; a predicate occurring in both graphs inherits the larger of its two entropy values. The ranking thus prefers predicates whose values vary widely across the graph, which we assume carry more identifying information than near-constant ones; we note this assumption is imperfect, since an identifier-like predicate attains maximal entropy while offering little cross-graph comparability. In all cases, a fixed budget on the maximum number of properties serialised per entity gates the volume of descriptive context, bounding the prompt length regardless of how densely described the individuals are.

B. Model-specific quantisation and serving hardware

Self-hosted open-weight models include: Qwen3 235B (GPTQ Int4/Int8 mix), Qwen3.5 122B (GPTQ Int4), Llama 3.1 70B (AWQ Int4), and Gemma 3 27B (GPTQ 4-bit) run at 4-bit weights, whereas Qwen3 8B runs unquantised. API-routed models are pinned to a single provider (and quantisation), with DeepSeek V4 Flash and Mistral Small 3.2 24B at FP8, and Gemini 2.5 Flash with no quantisation declared. Note that cross-model comparisons therefore compare model-plus-serving stacks.

Hardware specifications. When oracle consultation is performed locally, we route requests for Qwen3 235B, Qwen3.5 122B, and Llama 3.1 70B to vLLM running on a server with the following hardware specification:

- **CPU:** AMD Ryzen 9975WX (24 cores, 48 threads, base clock: 4.2 GHz, boost clock: 5.4 GHz).
- **GPU:** 2 × NVIDIA RTX PRO 6000 Max-Q Edition 96GB (192GB available VRAM).
- **RAM:** 768GB DDR5 ECC RDIMM at 5600 MHz.
- **OS:** Ubuntu 24.04.4 LTS; CUDA version 13.2.

Requests for Gemma 3 27B and Qwen3 8B route to a second node with the following hardware specifications:

- **CPU:** Intel Xeon Sapphire Rapids (48 cores, 96 threads, base clock: 2.40 GHz, boost clock: 3.80 GHz).
- **GPU:** 4 × NVIDIA RTX PRO 4000 24GB (96GB available VRAM).
- **RAM:** 128GB DDR5 ECC RDIMM at 4800 MHz.
- **OS:** Ubuntu 24.04.4 LTS; CUDA version 13.2.