Supervised Asymmetric Ontology Alignment Classifier Based on NLP

Era Alcani
Politecnico di Torino
Turin, Italy
s338197@studenti.polito.it

Francesco Dal Cero
Politecnico di Torino
Turin, Italy
s342631@studenti.polito.it

Francesco Mastrosimone
Politecnico di Torino
Turin, Italy
s348159@studenti.polito.it

Abstract

Ontology alignment is a key step for harmonizing heterogeneous data sources, yet remains challenging in applied settings where short, noisy attributes must be mapped to a fixed, semantically rich reference ontology. In this work, we propose a production-oriented framework for *asymmetric ontology alignment*, in which extracted attributes are aligned to a stable target ontology in a supervised setting. The framework combines offline preprocessing, exact-match shortcuts, hybrid lexical and semantic candidate retrieval, and supervised Cross-Encoder re-ranking trained with hard negatives. By isolating computationally expensive operations offline, the system enables scalable and reproducible inference with controllable runtime budgets. We evaluate the proposed approach on two OAEI benchmarks, ENVO–SWEET and FLOPO–TO, using ranking-based metrics that assess both retrieval coverage and re-ranking quality. Results show substantial improvements in Precision@1 and MRR over lexical baselines, together with strong zero-shot generalization to unseen ontology pairs.

The full implementation of the framework is publicly available on **GitHub**.

1 Introduction

Ontology alignment (OA) is a key component of data integration pipelines, enabling heterogeneous attributes extracted from multiple sources to be mapped onto a shared semantic ontology. This paper focuses on a concrete, production-driven setting arising from a real-world use case at **Repertorio**, a startup that aggregates and harmonizes patient-preference data across heterogeneous studies. In Repertorio's pipeline, attributes extracted from individual studies are often short, noisy, and weakly contextualized, yet must be automatically mapped to a proprietary internal ontology that is fixed and semantically rich. This asymmetric structure—a variable list of extracted attributes aligned to a stable target ontology—is central to enabling cross-study comparison and downstream structured analysis.

To address this task, we propose an end-to-end framework for *supervised* asymmetric ontology alignment, explicitly engineered for reproducible and scalable deployment. The framework separates expensive preprocessing from runtime inference by constructing offline artifacts for the target ontology, including exact-match structures, lexical indices, and semantic embeddings. At inference time, candidate generation combines symbolic shortcuts with hybrid lexical and semantic retrieval, followed by supervised Cross-Encoder re-ranking trained with hard negatives to resolve fine-grained semantic distinctions.

We evaluate the proposed framework on two benchmarks from the Ontology Alignment Evaluation Initiative (OAEI), ENVO–SWEET and FLOPO–TO, using ranking-based metrics that reflect both candidate coverage and final decision quality. Results show that hard negative training and inference-time budget allocation substantially improve Precision@1 and MRR while preserving high recall, and that the optimized configuration exhibits strong zero-shot generalization to an unseen ontology pair. Overall, this work contributes a practical and production-oriented alignment framework tailored to Repertorio's operational requirements, with a clear separation between offline and online components and controllable inference-time costs.

2 Related Work

Ontology alignment (OA) has traditionally been addressed through systems that combine lexical similarity with structural and logical constraints. Representative matchers such as LogMap and AgreementMakerLight (AML) rely on token-based similarity, lexical indexation, and lightweight reasoning or repair mechanisms to ensure logical coherence at scale [1, 4]. While these approaches are strong baselines in Ontology Alignment Evaluation Initiative (OAEI) benchmarks, their dependence on surface-form overlap limits its robustness when labels are short, heterogeneous, or weakly overlapping, a common scenario in biomedical and survey-derived data [2].

Recent work has explored neural representations to overcome these limitations. BERTMap introduces a transformer-based framework that fine-tunes BERT on ontology-derived corpora and combines contextualized text representations with efficient candidate selection based on a sub-word inverted index [3]. The authors show that purely lexical retrieval may miss valid correspondences when no sub-word overlap exists and propose refinement and repair strategies to improve recall and precision [3]. In parallel, graph-based approaches leverage ontology structure through Graph Neural Networks, enriching semantic representations at the cost of increased computational complexity.

A related line of research in large-scale NLP retrieval adopts a two-stage *filter-then-rank* architecture, where a high-recall retrieval module generates a small candidate set that is subsequently re-ranked by a more precise cross-encoder [5]. This paradigm offers an effective trade-off between scalability and accuracy in large search spaces. Our work follows this principle for asymmetric ontology alignment, combining lightweight lexical and dense semantic retrieval with cross-encoder re-ranking trained using hard negatives, targeting scenarios in which short, noisy attributes must be aligned to semantically rich biomedical concepts.

3 Method

We propose an end-to-end alignment framework designed to map short, context-limited study attributes to semantically rich ontology concepts. The method integrates standardized ontology processing, supervised cross-encoder training, and a two-stage inference pipeline that combines symbolic shortcuts with learned semantic re-ranking. To ensure reproducibility and efficiency, computationally expensive operations are isolated in an offline preprocessing phase, while inference relies on lightweight retrieval and scoring over precomputed artifacts.

3.1 Ontology Loading & Dataset Construction

To evaluate our approach, we relied on official data from the Ontology Alignment Evaluation Initiative (OAEI), focusing on the ENVO–SWEET benchmark. The benchmark provides two OWL/RDF ontologies as separate resources (ENVO and SWEET), together with a reference alignment file containing gold correspondences expressed as matched entity pairs across the two ontologies.

Ontology loading and unified representation. We implement an end-to-end ontology loading pipeline that extracts one record per class from each ontology. OWL/RDF ontologies are parsed with `owlready2`, collecting identifiers (IRI and local name), preferred labels, parent relations and restrictions, as well as logical signals such as equivalence and disjointness. In addition, all available annotation properties are extracted into dedicated columns.

Because ontologies encode metadata heterogeneously, raw extracted tables are normalized into an ontology-agnostic *unified view*. Each class is mapped to a compact schema with fields *iri*, *local_name*, *label*, *description*, *synonyms*, *parents_label*, *equivalent_to*, and *disjoint_with*. Definitions are selected using a prioritized list of common properties, while synonyms are aggregated by concatenating values from all columns whose name contains the substring `synonym`. This harmonization ensures that downstream components operate on consistent inputs independently of ontology-specific modeling conventions.

Text construction policy (configurable source, fixed target). Starting from the unified view, we construct the textual representations used by retrieval and supervised training. The source text is configurable through a set of flags controlling which semantic attributes are included (label, description, synonyms, parent context, equivalence, disjointness). The target text is built with a fixed policy that concatenates the available metadata into a single representation. This design enables controlled manipulation of the information content on the source side, while keeping the target representation stable. In our experiments we always enabled all available source flags, training on the richest dataset configuration.

Supervised dataset construction. Positive pairs are drawn exclusively from the reference alignment. Negative pairs are sampled from non-aligned entity pairs, combining two complementary mechanisms. Random negatives provide broad coverage of the non-match space, while hard negatives are mined from semantically and lexically close candidates to create challenging near-miss examples. To avoid class imbalance, we construct a balanced dataset where the total number of negative pairs (random + hard) matches the number of positive alignments. The resulting dataset is split into

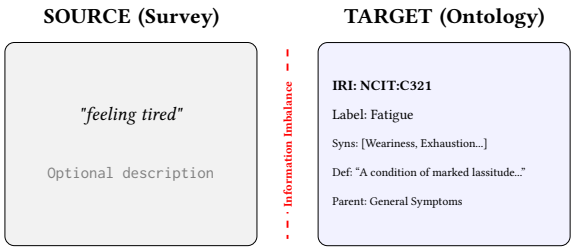

Figure 1: Data Asymmetry.

training, validation, and test partitions using a stratified procedure that preserves the proportion of positive and negative pairs across splits.

Each training instance is represented as a pairwise tuple (x_s, x_t, y), where x_s denotes the source text, x_t the target text, and $y \in \{0, 1\}$ indicates whether the pair corresponds to a gold alignment.

Hard negative mining. Hard negatives are generated through dense semantic search: we encode all source and target texts using a BioBERT-based `SentenceTransformer` model, fine-tuned on scientific Natural Language Inference (NLI) and Semantic Textual Similarity (STS) benchmarks. We retrieve, for each source, the top-20 most similar target candidates using efficient semantic search. Candidate pairs are filtered by similarity thresholds and by alignment exclusion (any gold pair is removed). We additionally apply lexical safety filters to discard cases that are too close and likely to create false negatives (e.g., label containment or synonym containment across the two sides). The final hard-negative set is sampled reproducibly with a fixed random seed.

Overall, this construction process—based on standardized ontology loading, explicit and configurable text encoding, gold-alignment positives, and principled negative sampling—provides a robust foundation for training and evaluating our alignment framework in a setting that remains practically aligned with the intended downstream use case.

3.2 Cross-Encoder Training

This section describes the supervised training procedure adopted for learning the alignment model. We train a supervised cross-encoder to discriminate valid from invalid source–target pairs constructed as described in Section 3.1. We formulate the task as a *pairwise classification problem* and implement the model as a *cross-encoder*. In this setting, the two texts are jointly encoded by a single Transformer model, allowing full attention across the source and target sequences. This design enables the model to capture fine-grained semantic interactions between short, noisy source attributes and semantically rich ontology concepts, which is critical for high-precision alignment.

Cross-encoder architecture. During training, the source and target texts are concatenated into a single input sequence, separated by special tokens, as shown in Figure 2, and fed to the cross-encoder for joint semantic representation and classification. The contextual representation associated with the `[CLS]` token is passed to a classification head that outputs a single logit, interpreted as the confidence that the pair represents a valid alignment. This formulation

directly reflects the implementation used during training, where each example consists of a (`source_text`, `target_text`) pair with a binary supervision signal.

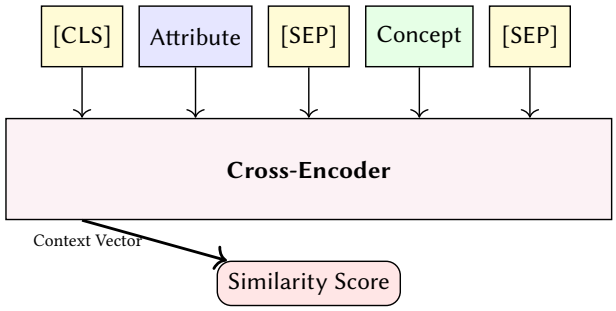

Figure 2: Cross-Encoder Input Representation.

Training setup. Training is performed using supervised fine-tuning on the pairwise dataset described in Section 3.1. The model is optimized with binary cross-entropy loss over positive and negative pairs. We employ mini-batch training with configurable batch size, learning rate, and weight decay, and optionally apply hyperparameter optimization using Optuna on a held-out validation set. When a validation split is available, early stopping and model selection are driven by validation performance, and the best checkpoint is retained. Mixed-precision training (FP16) is enabled to improve computational efficiency on GPU hardware.

Model selection. The training pipeline is designed to be model-agnostic: different pretrained Transformer encoders can be plugged into the cross-encoder architecture without changing the training logic. In our experiments, we evaluated multiple pretrained language models and selected the final cross-encoder based on the average precision achieved on the validation set. Details of this comparison and the final choice are reported in the experimental evaluation section. Overall, this training strategy yields a model that is explicitly optimized to discriminate true correspondences from hard negative pairs, making it well suited for re-ranking candidate alignments in large ontology matching scenarios.

3.3 Offline Preprocessing

The offline preprocessing phase performs all computationally heavy but reusable operations, with the goal of making inference fast, deterministic, and portable across runs. Given a target ontology, we first parse and normalize its content into the unified view (Section 3.1), ensuring that each class is represented consistently through identifiers, preferred labels, synonyms, and the text field consumed by downstream components.

Exact-match structures (label and synonym normalization). To enable a constant-time shortcut for trivial mappings, labels and synonyms are preserved as first-class objects rather than being merged into a single text string. For each target class c, we construct a set of textual forms $\Omega(c)$ containing the preferred label and all available synonyms. Each form undergoes lightweight normalization and is stored in a per-class dictionary (`class_data`). From this structure, we derive a reverse lookup map `label2classes`, which associates each normalized string with the set of target class IRIs exposing it. This allows exact matches to be resolved symbolically, without invoking learned scoring.

Lexical candidate selection (subword inverted index and IDF weighting). When exact matches are not available, the offline phase prepares lexical structures for efficient high-recall candidate retrieval. Using the same subword tokenizer adopted at scoring time, we tokenize the normalized forms in $\Omega(c)$ and build a token set $T(c)$ for each class. An inverted index maps each subword token to the classes that contain it. We further compute inverse document frequency (IDF) statistics,

$$\mathrm{idf}(t) = \log_{10}\left(\frac{|C|}{\mathrm{df}(t)}\right),$$

where $|C|$ denotes the number of target classes and $\mathrm{df}(t)$ the number of classes containing token t. These structures allow inference-time lexical retrieval to efficiently rank candidates by weighted token overlap, while maintaining tokenization consistency with the downstream model.

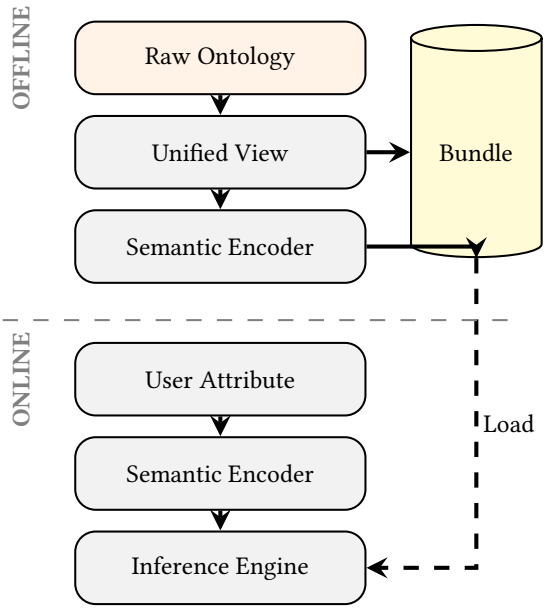

Figure 3: Inference Pipeline Architecture with Offline–Online Separation.

Semantic index (precomputed embeddings). Lexical retrieval can fail when the correct match exhibits little or no subword overlap with the query, as in the case of paraphrases, terminological shifts, abbreviations, or noisy extraction. To mitigate these failure modes, we additionally build a semantic index offline. Specifically, we compute one dense vector per target class by encoding each class text independently with a HuggingFace Transformer encoder specified by `SemanticIndexConfig`. Class representations are obtained via mean pooling over the last hidden states with attention masking and can be optionally L2-normalized to support cosine similarity

search. For efficiency and portability, embeddings are stored separately on disk (as `.npy` files), while the offline bundle retains only the associated metadata, enabling memory-mapped loading when required.

Packaging and reproducibility. All offline artifacts are serialized into a single `offline_bundle.pkl`, which serves as the central resource for inference. The bundle contains: (i) exact-match structures (`class_data`, `label2classes`); (ii) lexical candidate-selection components (`T`, `inverted_index`, `idf`, `num_classes`); and (iii) semantic index metadata with references to the embedding store. By computing these components once and persisting them as a versioned artifact, the offline phase avoids redundant preprocessing and ensures that inference relies on consistent and reproducible structures.

3.4 Inference Phase

The inference phase maps unseen study attributes to target ontology classes through a two-stage pipeline designed to be efficient at runtime. It leverages the offline artifacts computed once in Section 3.3 and applies a supervised cross-encoder only to a restricted candidate set.

Stage A: High-recall candidate retrieval. Stage A retrieves a shortlist of candidate target classes for each attribute, with the goal of maximizing recall before expensive scoring. The attribute string is first normalized and checked against the exact-match dictionary (`label2classes`). If an exact match is found, the system immediately returns the corresponding class IRI with score 1.0, bypassing the learned scorer. If no exact match exists, candidates are generated

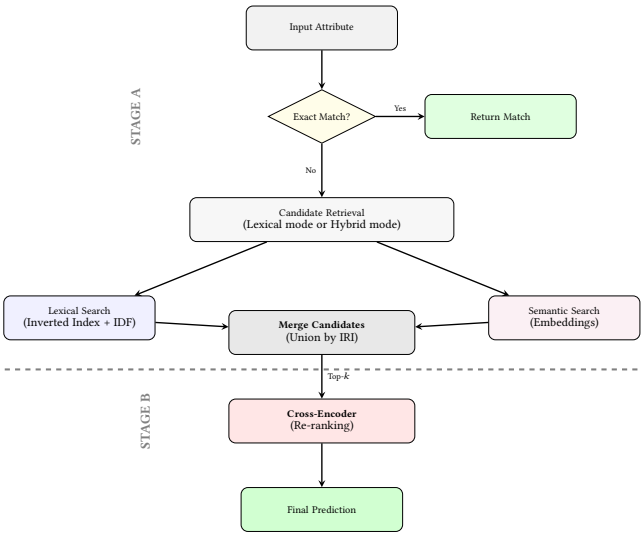

Figure 4: Inference logic of the proposed alignment system.

through (i) lexical retrieval and, optionally, (ii) semantic retrieval. Lexical retrieval uses a subword inverted index and IDF-weighted token overlap to efficiently rank target classes without scanning the full ontology. Semantic retrieval embeds the attribute text in batch and ranks classes by cosine similarity against the precomputed target-class embeddings stored in the offline bundle.

Two retrieval strategies are supported. In `lexical` mode, the system returns the lexical shortlist and falls back to the semantic shortlist only when lexical retrieval produces no candidates. In `hybrid` mode, lexical and semantic candidates are always combined with a fixed budget split and deduplicated by IRI, producing a merged shortlist that balances surface overlap and semantic similarity.

Stage B: Cross-encoder scoring and selection. Stage B improves precision by re-ranking a limited number of retrieved candidates using the *fine-tuned* sequence-classification cross-encoder learned during training. Only the top-k candidates per attribute (with k controlled by `cross_top_k`) are scored. Scoring is performed in batches over all (attribute, class-text) pairs, and exact-match cases are excluded from scoring by design.

The cross-encoder outputs a scalar score for each attribute–class pair. Candidates are ranked according to this score, and the final prediction is the target IRI with the highest score.

Outputs. Inference produces a CSV with one row per attribute, reporting the predicted class IRI and score, the retrieval strategy used (exact shortcut, lexical retrieval, or hybrid retrieval), and summary metadata such as the number of retrieved and scored candidates. Optionally, the top-N scored candidates can be exported for inspection and downstream analysis.

4 Results & Evaluation

This section reports the experimental evaluation of the proposed alignment framework on two OAEI benchmarks, ENVO–SWEET and FLOPO–TO. We follow a structured evaluation protocol that includes model screening, inference-time parameter optimization, and quantitative analysis on held-out test data. Results are presented using ranking-based metrics to separately assess candidate retrieval coverage and final re-ranking quality across different configurations of the pipeline.

4.1 Evaluation Metrics

Given the asymmetric nature of the alignment task and its intended use in a semi-automated mapping setting, we adopt ranking-based evaluation metrics that explicitly reflect both candidate coverage and final decision quality.

- **Hits@K:** The proportion of test attributes for which the correct ontology concept appears within the top-K candidates generated by the retrieval stage. Hits@K measures the recall of the candidate generation process and indicates whether the correct match is made available to the re-ranking stage.
- **Precision@1:** The proportion of test attributes for which the top-ranked prediction produced by the system corresponds to the ground-truth alignment. This metric captures the accuracy of fully automated predictions.
- **Mean Reciprocal Rank@K (MRR@K):** The mean reciprocal rank of the correct concept within the top-K ranked candidates. MRR@K provides a summary measure of ranking quality, assigning higher scores when the correct match is ranked closer to the top of the list.

4.2 Inference Parameters and Budget Allocation

The inference pipeline is governed by a small set of runtime parameters that control how candidates are generated, combined, and ultimately evaluated by the Cross-Encoder. These parameters define the trade-off between candidate coverage, computational cost, and final ranking accuracy.

During **Stage A**, the parameters *retrieval lexical top k* and *retrieval semantic top k* specify the maximum number of candidates produced by the lexical and semantic retrieval modules, respectively. In *hybrid* mode, candidates from the two sources are merged and deduplicated by IRI, up to a maximum size defined by *retrieval merged top k*. The parameter *hybrid ratio semantic* controls the relative contribution of semantic versus lexical candidates in this merged shortlist, allocating a fixed fraction of the retrieval budget to each source.

During **Stage B**, the parameter *cross top k* defines the maximum number of (attribute, candidate) pairs that are actually scored per attribute. In hybrid mode, the same lexical/semantic budget ratio is applied when selecting candidates for re-ranking, ensuring a balanced representation of both sources among the scored pairs.

Overall, retrieval parameters determine the diversity and coverage of the candidate pool, while *cross top k* directly controls the scope of the final decision process.

4.3 Phase 1: Model Screening (Pre-Tuning)

As a first step, we conducted a comparative screening of different Transformer architectures to select the Cross-Encoder backbone used in subsequent experiments. We evaluated three pretrained models—**BERT-base**, **PubMedBERT**, and **SciBERT**—under identical inference conditions on the ENVO–SWEET validation set.

To ensure a controlled comparison, all models were tested using the same inference parameters, reported below:

- **retrieval lexical top k = 100**: number of candidates retrieved via the symbolic subword inverted index;
- **retrieval semantic top k = 100**: number of candidates retrieved via dense semantic search;
- **retrieval merged top k = 150**: maximum number of unique candidates obtained after merging retrieval sources;
- **hybrid ratio semantic = 0.5**: balanced budget split between lexical and semantic retrieval in Hybrid Mode;
- **cross top k = 20**: number of candidates per attribute passed to the Cross-Encoder for re-ranking.

Table 1: Model screening results on the ENVO–SWEET validation set.

Model	Hits@20	Precision@1	MRR@20
SciBERT	0.6966	0.6180	0.6440
PubMedBERT	0.7303	0.5955	0.6411
BERT-base	0.7191	0.5843	0.6333

Table 1 reports the validation performance obtained by each backbone in terms of candidate coverage and ranking quality. **SciBERT** achieved the highest Precision@1 and competitive MRR, was therefore selected as the Cross-Encoder backbone for all subsequent experiments.

4.4 Phase 2: Inference Parameter Optimization

After selecting SciBERT as the Cross-Encoder backbone, we optimized the inference-time parameters on the validation split using Optuna. The optimization targeted the allocation of the retrieval budget between lexical and semantic candidates, as well as the size of the candidate set passed to the re-ranking stage.

The best-performing configuration favored a broader lexical retrieval pool combined with a higher proportion of semantic candidates in the merged shortlist. In addition, the effective re-ranking budget was increased, allowing the Cross-Encoder to evaluate a larger subset of retrieved candidates.

Specifically, the selected setup retrieves up to 200 lexical and 100 semantic candidates during Stage A, merges them into a shortlist of at most 150 unique candidates, and applies Cross-Encoder scoring to the top 100 candidates per attribute.

4.5 Quantitative Analysis

Following the inference-time optimization described above, we evaluate the resulting configurations on held-out test data to quantify their impact on retrieval coverage and ranking quality.

Ablation Study and Tuning (ENVO–SWEET). Table 2 reports the results obtained on the ENVO–SWEET test set for different configurations of the proposed framework. As a reference point, we report the performance of an Exact Match strategy evaluated as a standalone alignment approach that attempts to resolve all input attributes; attributes for which no exact correspondence is found are treated as incorrect predictions. Under this convention, Exact Match achieves a Precision@1 of 0.6119.

The Lexical Mode configuration, which integrates the Exact Match shortcut with subword-based lexical retrieval and Cross-Encoder re-ranking while disabling semantic retrieval, achieves a Precision@1 of 0.6716 and a Hits@20 of 0.7015.

The SciBERT (Standard) and SciBERT (Standard no HN) configurations achieve identical retrieval recall, with Hits@20 equal to 0.7015 in both cases, while differing in ranking metrics. The configuration trained without hard negatives yields lower Precision@1 and MRR values compared to the standard training setup.

The SciBERT (Optimized) configuration achieves the highest values across all reported metrics, with improvements observed in both retrieval recall and ranking quality.

Table 2: Performance comparison on the ENVO–SWEET test set. Best results in bold.

Method	Hits@20	Precision@1	MRR@20
Exact Match	-	0.6119	-
Lexical Mode (Inverted Index + IDF)	0.7015	0.6716	0.6841
SciBERT (Standard)	0.7015	0.6866	0.6940
SciBERT (Standard) no HN	0.7015	0.6418	0.6679
SciBERT (Optimized)	**0.7761**	**0.7164**	**0.7360**

Zero-Shot Generalization (FLOPO–TO). Table 3 reports the results obtained by the optimized configuration on the FLOPO–TO dataset, representing an unseen ontology pair from the biodiversity domain. The model achieves high values across all evaluation metrics, including Precision@1, Hits@20, and MRR.

Table 3: Generalization performance on FLOPO–TO (unseen ontology pair).

Metric	Value
Hits@20	0.9207
Precision@1	0.8634
MRR@20	0.8776

5 Discussion

The experimental results provide insight into how architectural choices, training strategy, and inference-time budget allocation jointly affect alignment performance in asymmetric ontology matching.

Backbone choice and ranking behavior. The model screening phase indicates that the choice of Cross-Encoder backbone primarily affects ranking quality rather than retrieval coverage. As shown in Table 1, all evaluated models achieve comparable Hits@20 values under identical retrieval conditions, indicating similar candidate recall. In contrast, SciBERT consistently attains higher Precision@1 and MRR, demonstrating superior discriminative ability once the correct candidate is present in the shortlist. This suggests that backbone selection mainly influences the re-ranking stage, supporting the architectural separation between high-recall retrieval and high-precision scoring adopted in the framework.

Budget allocation and re-ranking as the main bottleneck. The inference-time optimization results indicate that alignment quality is strongly influenced by how candidates are filtered before re-ranking. As shown in Table 2, increasing the size of the lexical retrieval pool improves robustness at Stage A, leading to higher candidate recall. At the same time, assigning a larger portion of the retrieval budget to semantic candidates highlights the complementary role of dense retrieval in recovering valid correspondences when lexical overlap is insufficient. Most importantly, expanding the effective re-ranking budget yields the largest gains in Precision@1 and MRR, despite more moderate increases in Hits@20. This confirms that the re-ranking stage constitutes the primary decision bottleneck of the pipeline: performance improvements are driven less by marginal recall gains than by ensuring sufficient candidate diversity reaches the Cross-Encoder.

Complementarity of retrieval strategies. The ablation study on ENVO–SWEET (Table 2) demonstrates that purely symbolic matching, while locally precise, is insufficient under terminology variation. Lexical retrieval combined with neural re-ranking already yields substantial improvements over Exact Match, while the optimized hybrid configuration further increases both retrieval recall and ranking quality. These findings indicate that lexical and semantic retrieval provide complementary signals: lexical search preserves surface-form evidence, while semantic retrieval introduces conceptually relevant candidates that would otherwise be missed, together maximizing the quality of the candidate pool passed to the Cross-Encoder.

Effect of hard negative training. Comparisons between the SciBERT models trained with and without hard negatives reveal the critical role of the training objective in shaping ranking behavior. Despite identical retrieval recall, the model trained without hard negatives exhibits a clear drop in Precision@1 and MRR, as reported in Table 2. Exposure to challenging near-miss examples enables the Cross-Encoder to better distinguish correct alignments from semantically related but incorrect candidates, such as parent or sibling concepts in the ontology hierarchy.

Generalization and practical applicability. The strong performance observed on the FLOPO–TO dataset (Table 3) demonstrates that the learned alignment strategy generalizes beyond the ontology pair used for model selection and tuning. High Precision@1 and MRR in this unseen setting indicate that the combination of lexical retrieval, semantic candidate generation, and Cross-Encoder re-ranking captures alignment signals that are not domain-specific. Together with the filter–then–rank design and offline preprocessing strategy, these results support the suitability of the framework for scalable and reproducible ontology alignment in real-world scenarios involving heterogeneous and evolving domains.

6 Conclusion

In this work, we presented a production-oriented framework for asymmetric ontology alignment, designed to automatically map short, heterogeneous study attributes to a fixed, semantically rich ontology. Unlike prior approaches that focus on aligning two ontologies in an unsupervised setting, our method addresses a supervised, real-world scenario in which extracted attributes must be harmonized against a stable target ontology. The framework combines offline preprocessing, hybrid candidate retrieval, and supervised Cross-Encoder re-ranking, achieving strong performance on OAEI benchmarks and robust zero-shot generalization to unseen ontology pairs. Beyond quantitative gains, the system is explicitly engineered for practical deployment, with clear separation between offline and online components and controllable inference-time budgets, making it suitable for large-scale industrial use at Repertorio.

Future work will explore extending the supervised training setup to larger and more diverse alignment corpora, for instance by jointly leveraging multiple OAEI tracks to improve robustness across domains. In addition, we plan to investigate human-in-the-loop strategies, where expert feedback on uncertain or ambiguous mappings can be incorporated to iteratively refine the model and the underlying alignment data. These directions aim to further strengthen the framework's applicability in real-world settings, where ontologies evolve and alignment quality must be continuously maintained.

References

[1] Daniel Faria et al. 2013. The AgreementMakerLight Ontology Matching System. In *On the Move to Meaningful Internet Systems: OTM 2013 Conferences*. Springer.

[2] Yuan He et al. 2021. Biomedical Ontology Alignment with BERT. In *Proceedings of the Ontology Matching Workshop (OM)*.

[3] Yuan He et al. 2022. BERTMap: A BERT-Based Ontology Alignment System. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[4] Ernesto Jiménez-Ruiz and Bernardo Cuenca Grau. 2011. LogMap results for OAEI 2011. In *Proceedings of the Ontology Matching Workshop (OM)*.

[5] Ledell Wu et al. 2020. Scalable Zero-shot Entity Linking with Dense Entity Retrieval. In *Proceedings of EMNLP*.