AI Medicine
https://www.sciltp.com/journals/aim

Article

Auditable Symbolic Candidate Generation for Biomedical Ontology Matching in Medical AI Systems

Ruichao Xia [1], Xingsi Xue [2], Haonan Li [3] and Jianfeng Wang [4,*]

[1] Faculty of Radiophysics and Computer Technologies, Belarusian State University, Nezavisimosti Avenue 4, 220030 Minsk, Belarus
[2] School of Computer and Data Science, Fujian University of Technology, Fuzhou 350118, China
[3] Faculty of Computer Science and Information Technology, Universiti Malaya, Kuala Lumpur 50603, Malaysia
[4] College of Software, Taiyuan University of Technology, No.79 West Street Yingze, Taiyuan 030024, China
* Correspondence: wangjianfeng@tyut.edu.cn

How To Cite: Xia, R.; Xue, X.; Li, H.; et al. Auditable Symbolic Candidate Generation for Biomedical Ontology Matching in Medical AI Systems. *AI Medicine* **2026**, *3*(2), 10. https://doi.org/10.53941/aim.2026.100010

Received: 1 July 2026
Revised: 2 September 2026
Accepted: 10 September 2026
Published: 28 September 2026

Abstract: Biomedical ontology matching is a key step for semantic interoperability in medical AI systems, supporting clinical terminology mapping, cross-resource search, biomedical knowledge graph construction, and phenotype-aware data integration. This paper presents ARB, an auditable symbolic candidate-generation module designed for settings where external biomedical resources, learned embeddings, or reference-alignment calibration are unavailable or undesirable. ARB starts from a token inverted-index blocker, rescues missed candidates through hierarchy-pooled and neighbour-label evidence under a purity gate, and applies same-source structural dominance pruning to reduce lexically similar false siblings. The module is evaluated on seven OAEI 2013 biomedical ontology-matching tasks using candidate-pool metrics and a fixed downstream Hungarian matcher, and on five Bio-ML 2024 local-ranking tasks as a matcher-independent scoring check. Across the OAEI tasks, the rescue mechanism improves candidate recall by 2.0–10.8 percentage points while limiting gated pool growth to 5–24%. On Anatomy, structural dominance pruning alone raises downstream F1 from 0.197 to 0.366 under the fixed matcher, whereas the full ARB configuration (rescue plus pruning) reaches 0.375. Pruning is less reliable on structurally heterogeneous SNOMED-related tasks. On Bio-ML 2024, ARB achieves a mean MRR of 0.813 and mean Hits@1 of 0.760 using only ontology-internal symbolic evidence. The module is most useful when lexical blocking leaves meaningful recall headroom; recall-saturated phenotype tasks instead require conservative rescue. These results support ARB as an auditable candidate-generation and cleaning component for biomedical knowledge integration, rather than as a complete ontology matcher or a final clinical mapping tool.

Keywords: entity resolution; blocking; candidate provenance; SNOMED CT; semantic interoperability; offline biomedical knowledge integration

1. Introduction

Biomedical ontology matching is a necessary step in many medical AI workflows, because clinical terminologies, disease vocabularies, phenotype ontologies, anatomy resources, and oncology knowledge bases are usually developed independently. Links between these resources support terminology normalisation, cross-resource search, biomedical knowledge graph construction, cohort discovery, phenotype analysis, and downstream model development. Recent clinical NLP work likewise uses multi-ontology coding to standardise unstructured pathology reports across SNOMED CT, LOINC, and ICD-11 [1]. These examples motivate ontology alignment as an infrastructure problem for medical knowledge integration, not as evidence that the present blocker has been clinically deployed. If such links are missing or difficult to inspect, errors may propagate silently into later stages of medical data integration. For this reason, biomedical ontology matching should not be treated only as a benchmark task in ontology engineering, but also as a practical component of semantic interoperability in medical AI systems. This study evaluates blocking behaviour only on public ontology benchmarks; it does not test clinical deployment, clinical safety, or patient-level decision support.

In clinical and institutional settings, the matching process often needs to be traceable. A candidate pair may have to be justified by showing which labels, synonyms, ancestor concepts, neighbour labels, or graph relations caused it to be generated. It may also be necessary to confirm that no reference mapping, external service, or hidden supervision influenced the decision. This is particularly relevant when ontology files are updated, local terminology extensions are introduced, or mapping results are reviewed for data-governance purposes. In this paper, an auditable candidate-generation module means that each retained candidate can be linked to explicit ontology-internal evidence, such as token buckets, hierarchy context, neighbour labels, score components, and pruning status.

The scale of biomedical ontologies makes candidate generation indispensable. Matching two ontologies with tens of thousands, or even hundreds of thousands, of concepts would require an impractical number of pairwise comparisons if every source concept were compared with every target concept. Blocking reduces this search space by producing a smaller candidate set before scoring and one-to-one assignment. A simple token inverted-index blocker is attractive because it is deterministic, scalable, and easy to inspect, but it has two well-known weaknesses in biomedical data. It may miss equivalent concepts expressed with different surface forms, such as clinical and ontology-specific variants of the same disease or anatomical entity. It may also admit many lexically similar siblings that differ by laterality, subtype, disease stage, or anatomical scope, leaving the downstream matcher to resolve conflicts with limited evidence.

Existing biomedical ontology matchers address vocabulary mismatch and structural ambiguity in different ways. Systems such as LogMap [2] and AML [3] combine lexical matching, structural checks, repair procedures, and biomedical or lexical mediators. Neural systems such as BERTMap [4] and Matcha [5] use learned representations to improve semantic matching. These systems are valuable as complete matchers, but their assumptions are not always suitable for a blocking module that must run offline, avoid external biomedical resources, and expose candidate-level provenance. Meta-blocking [6, 7] can prune an already generated candidate graph, but it cannot recover a true pair that the initial blocker failed to generate. The resulting chain is therefore narrow: external-resource matchers and learned matchers can improve final semantic alignment, while meta-blocking can clean an existing candidate graph, but the blocking stage still lacks an offline, auditable, reference-independent mechanism for recovering selected lexical misses.

This paper proposes Auditable Rule-based Blocking (ARB), a symbolic candidate-generation module for biomedical ontology matching in medical AI systems. ARB starts from a token inverted-index base, adds purity-gated context rescue using hierarchy-pooled and neighbour-label evidence, and applies same-source structural dominance pruning to reduce selected lexical conflicts. The module is a deterministic optimisation and rule-based blocking pipeline rather than a trained machine-learning model: it uses only the two input ontologies during inference and does not calibrate its decisions on the test reference alignment. The contribution is therefore auditable blocking and candidate-pool diagnosis, not an end-to-end ontology matching system. We evaluate ARB on OAEI biomedical tasks using candidate recall, candidate purity, pool growth, runtime, and memory, and report final F_1 only as a downstream check with a fixed Hungarian matcher. We also use Bio-ML 2024 local-ranking tasks to examine symbolic scoring without a downstream matcher. The intended contribution is a transparent candidate-generation and cleaning component for biomedical knowledge integration, not a complete ontology matcher or a final clinical mapping tool.

2. Related Work

Biomedical ontology matching is already embedded in medical knowledge infrastructure. Resources such as SNOMED CT, NCI, FMA, DOID, ORDO, HP, and MP are linked in terminology services, ontology repositories, biomedical knowledge graphs, and benchmark campaigns such as OAEI LargeBio and Bio-ML equivalence tasks [8–12]. A recent SNOMED CT-aligned ontology used in a hearing-loss decision-support workflow illustrates how terminology alignment can support interpretable clinical data exchange, while still requiring external clinical validation before deployment [13]. The practical need in these settings is not only to maximise final alignment scores, but also to explain which lexical or structural evidence caused a candidate to enter the matching pipeline.

Among symbolic biomedical ontology matching systems, LogMap and AML remain the most established complete matchers in OAEI-style evaluation. LogMap [2] and AML [3] combine lexical matching, structural checks, logical repair, and biomedical background resources, and they remain central reference systems in recent OAEI campaigns [9–11]. These systems are appropriate comparators for final alignment quality. They are not direct blocking-stage baselines here because their outputs reflect complete matching pipelines, often including mediators such as UMLS or WordNet, repair modules, and final alignment filtering rather than an exposed candidate-generation table.

Neural and language-model ontology matchers reduce vocabulary mismatch with representations that are richer than surface tokens. BERTMap fine-tunes BERT for ontology alignment and combines neural scoring

with extension and repair [4]; Matcha adds neural re-ranking to a modular ontology-matching architecture [5]; OWL2Vec4OA adapts ontology embeddings for alignment [14]. Recent preprint systems such as OLaLa, Agent-OM, and LLMs4OM explore prompting, retrieval, and representation choices for OAEI-style tasks [15–17]. Retrieval- or generation-augmented preprints, including KROMA and GenOM, add definitional context or generated descriptions before matching [18, 19]. OAEI-LLM and OAEI-LLM-T isolate hallucination behaviour in LLM-based ontology matching under biomedical and schema-level settings [20, 21]. More broadly, healthcare AI literature emphasises that generated outputs require traceability and governance rather than uncritical operational use [22]. These systems target semantic matching quality when model access, embedding infrastructure, prompts, or external retrieval are acceptable. A deployment that requires offline execution, stable reruns, and candidate-level provenance places a narrower constraint on them.

Interactive and active-learning matchers treat expert feedback as part of the matching loop. DualLoop combines tunable heuristic matchers with short- and long-term active learners to reduce the query cost of finding difficult residual matches [23]. That setting fits deployments where expert queries are available during matching. The module studied here assumes a different interface: the blocker receives two ontology files and fixed parameters, then emits a candidate pool without reference pairs or run-time human labels.

Large-scale entity-resolution pipelines usually separate candidate generation, or blocking, from candidate scoring. Sorted Neighbourhood [24] and token blocking summarised in the entity-resolution survey of Papadakis et al. [25] reduce comparisons by assigning records to blocking keys or token buckets. Meta-blocking then reweights an already generated candidate graph and removes weak edges [6, 7]. Recent blockers replace hand-designed keys with learned representations or specialised execution engines: AutoBlock learns similarity-preserving representations for nearest-neighbour blocking [26], SC-Block uses supervised contrastive learning inside full ER pipelines [27], dense blockers such as UniBlocker seek broader application across domains [28], and HyperBlocker optimises rule-based blocking on GPUs [29]. Sudowoodo uses contrastive self-supervised learning for matching-oriented data integration tasks [30]. The usual inputs for these blockers include labelled pairs, learned embeddings, approximate-neighbour indexes, or rule tuning outside the two ontologies being matched. These designs improve retrieval power, but they weaken candidate-level audit trails when each retained pair must be traced back to observable ontology evidence.

Graph propagation methods that exploit structural context predate neural ontology alignment. Similarity flooding propagates evidence through graph edges and can exploit schema structure beyond labels [31]. GraphMatcher similarly learns graph-aware representations around ontology classes when lexical evidence is insufficient [32]. Propagation methods either materialise large pairwise similarity structures, depend on seed alignments, or require stopping and filtering conditions whose effect is difficult to separate from final matching. For blocking evaluation, the structural signal, candidate proposal rule, and pruning threshold must remain separately observable.

Benchmarks and evaluation protocols also shape candidate-pool assumptions. The OAEI LargeBio and Anatomy tracks provide long-running biomedical alignment benchmarks, while recent OAEI campaigns add broader tracks and newer system families [8–11]. The Bio-ML benchmark was introduced to support machine-learning-friendly biomedical matching tasks, including local-ranking evaluation settings [12]. These benchmark designs impose different candidate-pool assumptions. OAEI systems are commonly compared through final alignments, whereas Bio-ML local-ranking tasks provide candidate lists in advance. Candidate recall, purity, pool size, and cost are not interchangeable with final precision, recall, and F_1.

Table 1 summarises the deployment assumptions discussed above. The comparison is not a ranking of final alignment quality. It asks whether a method exposes candidate-level provenance, avoids labelled training pairs at deployment time, avoids external biomedical resources, can run offline inside an institution, and reports candidate-pool metrics.

Table 1. Positioning of ARB against related systems. A check (✓) means the property holds for the named configuration; a cross (✗) means it does not. "Candidate provenance" means the emitted pair can be traced to labels, buckets, local graph context, or pruning status rather than only to a final alignment score. Qualifiers indicate configuration-dependent assumptions rather than universal properties of every system variant. For LogMap, "std." denotes the standard configuration and "BK" denotes the background-knowledge configuration.

System	No Labelled Pairs	No Biomedical Ext. Resource	Candidate Provenance	Offline Feasible	Candidate Metrics
LogMap [2]	✓	std. ✓ BK ✗	✗	✓	✗
AML [3]	✓	✗	✗	✓	✗
BERTMap [4]	✓ [a]	✓	✗	✓	✗
Matcha [5]	✗	✓	✗	✓	✗
Agent-OM [16]	✗	✗	✗	✗	✗
LLM/RAG OM [17–19]	✗	✗	✗	✗	✗
Interactive OM [23]	✗	✓	✓	✓	✗
MetaBlocking [7]	✓	✓	✓	✓	✓
Dense ER blockers [26–28,30]	✗	✓	✗	✓	✓
Token blocking (base)	✓	✓	✓	✓	✓
ARB (this work)	✓	✓	✓	✓	✓

[a] BERTMap does not require benchmark alignment pairs at deployment time; its training signal comes from ontology synonym text and neural pretraining rather than labelled mapping pairs.

3. Problem Setting and Engineering Requirements

3.1. Task Definition and Blocking Metrics

The task starts from two parsed biomedical ontologies. Let $O_s = s_1, \ldots, s_m$ and $O_t = t_1, \ldots, t_n$ denote the source and target concept sets. Each ontology provides one or more labels for a concept, represented by label functions $\lambda_s : O_s \to 2^{\Sigma^*}$ and $\lambda_t : O_t \to 2^{\Sigma^*}$, and a subsumption relation $\preceq$ that defines the parsed hierarchy. When a benchmark reference is available, the gold equivalence alignment is denoted by $G \subseteq O_s \times O_t$.

A blocking module produces a candidate set $C \subseteq O_s \times O_t$ before downstream scoring and assignment. Its first objective is high candidate recall (pair completeness, PC),

$$\mathrm{PC}(C) = \frac{|C \cap G|}{|G|}, \tag{1}$$

because a true pair that never enters C cannot be recovered by any later matcher. The second objective is to keep the candidate pool much smaller than the Cartesian product. We report blocking reduction as

$$\mathrm{RR}(C) = 1 - \frac{|C|}{|O_s| \cdot |O_t|}. \tag{2}$$

The third objective is to control pool noise. Candidate purity (pair quality, PQ) is defined as

$$\mathrm{PQ}(C) = \frac{|C \cap G|}{|C|}, \tag{3}$$

with the understanding that this quantity is a benchmark-side estimate and is incomplete for partial-reference tasks. A highly polluted pool can burden the scorer with false candidates and can also create larger one-to-one conflict components. For this reason, blocking metrics PC, PQ, RR, and $|C|$ are reported separately from final precision, recall, and F_1 under the fixed downstream assignment procedure described in Section 4.

3.2. Auditability and Reference-Independent Requirements

The engineering setting considered in this paper is deliberately restricted. ARB operates on the two input ontologies alone. It does not query UMLS, WordNet, external synonym dictionaries, embedding services, or language models during inference. This restriction is not a claim that external resources are intrinsically unsuitable: they can recover abbreviation–expansion and other weakly lexical correspondences. Rather, it isolates a deployable baseline for environments in which resource licences, versions, coverage, access controls, or audit requirements make external knowledge unavailable or undesirable. The restriction also keeps each emitted candidate traceable to evidence present

in the input ontologies; losses such as RNA–Ribonucleic acid are reported explicitly as a boundary of that design.

Reproducibility is another requirement. For a fixed pair of ontology files and a fixed configuration, the same candidate table should be produced. ARB therefore uses deterministic parsing, fixed hash functions in the inference path, and deterministic tie-breaking. The implementation sets `PYTHONHASHSEED = 0`. Parameter values such as `accept`, `rescue_floor`, `structural_floor`, `min_channels`, `struct_margin`, `lex_slack`, K, b, and α are fixed before test evaluation, either as engineering priors or as development-fixed values. As detailed in Section 5, development-fixed does not mean selected on disjoint benchmark families. The Otsu structural threshold θ^* is the only per-task adaptive parameter; it is computed from the unlabelled structural-overlap distribution of score-qualified candidates and does not use the reference alignment.

3.3. Module Boundary and Candidate Provenance

ARB is an intermediate component in a biomedical ontology-matching pipeline, not a complete matcher. It sits between ontology ingestion and downstream scoring or one-to-one assignment. Given two ontology files and a configuration object, it constructs a feature store, builds and expands a candidate pool, applies optional pruning, and returns a candidate table that can be consumed by an external scorer or assignment method. Each row records at least the source concept identifier, target concept identifier, generation channel, lexical score, structural score, gate status, and pruning status. These fields provide candidate-level provenance: they show why a pair was generated and whether it was later affected by pruning.

In deployment, ARB writes two main artifacts. The first is the candidate table used by the downstream matcher. The second is a diagnostic report for audit and capacity planning. The diagnostic report records skipped token buckets, skipped LSH buckets, empty-label concepts, isolated concepts, candidate counts, rescue counts, pruning counts, runtime, memory, and assignment fallback status. These outputs are intended to make scale and quality issues visible before the candidate pool is used in a medical AI integration workflow.

3.4. Parser Scope and Edge-Case Handling

The parser treats each class URI as the stable concept identifier. Human-readable labels are extracted from `rdfs:label` and available synonym-like annotation properties. All parsed label strings are stored as a multi-label bag rather than collapsed into a single preferred label. Subsumption edges are read from `rdfs:subClassOf` axioms whose object is a named class. Blank-node restrictions and complex class expressions are not expanded into inferred parents. Duplicate labels attached to the same concept are removed after normalisation, whereas duplicate labels across different concepts are retained because they represent genuine ambiguity in biomedical terminologies.

Concepts with at least one label enter the own-label token index. Concepts without labels remain in the graph and may contribute ancestor or neighbour context to labelled concepts, but they do not generate own-label token candidates. Concepts with multiple synonyms contribute all synonym tokens to the same concept-level label bag while preserving the original URI in every emitted candidate. Isolated concepts, defined as classes with no parsed parent and no parsed child, receive empty ancestor and neighbour contexts and participate only through their own labels. If an isolated concept is also unlabelled, it remains in the diagnostic report but cannot generate a candidate without additional metadata. Buckets above the configured maximum size are skipped and counted, so over-frequent biomedical tokens are exposed in diagnostics rather than hidden inside the scorer.

3.5. Evaluation Protocols

The OAEI evaluation uses blocking metrics and downstream matching metrics under a fixed validation setup. Candidate recall PC measures how much of the reference alignment enters the candidate pool, candidate purity PQ measures how much of the pool is confirmed by the reference, and pool growth is computed relative to the token-base configuration:

$$\mathrm{growth}(C) = \frac{|C| - |C_{\mathrm{base}}|}{|C_{\mathrm{base}}|}. \tag{4}$$

Final $P/R/F_1$ is reported only after applying the same downstream Hungarian matcher across all ablation cells. These final scores are therefore used as a controlled downstream check, not as a claim that ARB is a complete ontology-matching system.

For Bio-ML local ranking, the benchmark supplies a predefined candidate list for each source concept. ARB ranks that list by its symbolic score, and the evaluation reports mean reciprocal rank (MRR) and Hits@1. This protocol tests whether the same ontology-internal evidence can order medically plausible candidates inside an existing pool. It does not evaluate blocking, and it does not require a downstream one-to-one matcher.

4. Proposed Blocking Framework

4.1. Overview and Design Rationale

Auditable Rule-based Blocking (ARB) treats biomedical ontology blocking as two controlled operations: recovering missed candidates and reducing ambiguous competitors. A token blocker is kept as the base because it is deterministic, scalable, and easy to inspect. However, token overlap alone cannot handle cases where equivalent biomedical concepts use different surface forms, abbreviations, or context-dependent labels. ARB therefore adds context rescue through ancestor and neighbour evidence. Because context expansion may also introduce unsupported candidates, the rescue step is constrained by a purity gate and a per-source budget. After rescue, same-source lexical siblings may still compete for the same source concept; ARB addresses this with structural dominance pruning based on local parent and child evidence.

Figure 1 gives the data flow. The solid path is the deployable inference path, while the dashed path involving the reference alignment is used only for evaluation. The framework takes parsed OWL/RDF concept records as input, emits a pruned candidate table as its main output, and leaves final one-to-one alignment to a fixed Hungarian matcher used only for downstream validation.

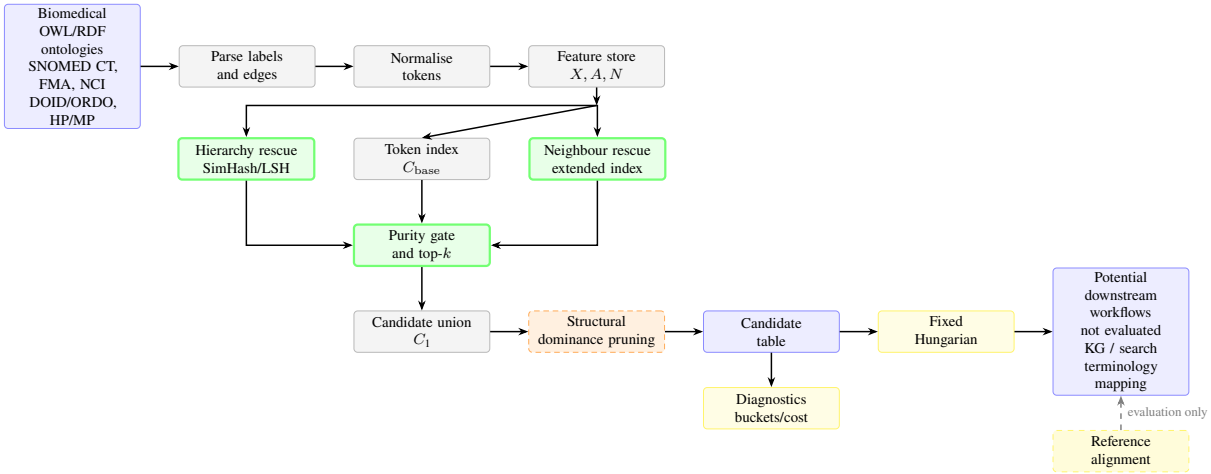

Figure 1. Detailed ARB data flow for biomedical ontology matching. SNOMED CT, FMA, NCI, disease, and phenotype ontologies are parsed into concept records, multi-label bags, and parent–child edges; preprocessing builds own-label, ancestor, and neighbour features. The token index creates C_{base}. A SimHash/LSH hierarchy channel and an extended-label neighbour channel propose rescue candidates; the purity gate admits only score-, structure-, or redundancy-supported pairs before union with C_{base}. Same-source structural dominance pruning removes selected competitors. Gold mappings enter only after candidate generation and pruning, for evaluation. Thick-outline boxes mark ARB additions, the dashed-outline box marks I2 pruning, solid arrows mark the inference path, and the dashed arrow marks the evaluation-only path. The retained candidate table can feed downstream knowledge-graph integration, semantic search, or terminology-mapping workflows, but those workflows are not evaluated as clinical deployments here.

4.2. Feature Construction and Scoring

For each concept v, ARB builds a tokenised label bag from `rdfs:label` and available synonym-like annotations. Labels are lowercased, accents are stripped, and strings are split on non-alphanumeric boundaries. Tokens of length one are discarded unless they contain a digit, and duplicate tokens attached to the same concept are collapsed after normalisation. The resulting own-label token multiset is denoted by $L(v)$. The ancestor set $\mathrm{Anc}_d(v)$ contains named-class ancestors within depth $d = 2$, and the neighbour set $M(v)$ contains immediate parents and children. Empty-label concepts remain in the graph but have $L(v) = \emptyset$.

TF–IDF weighting is computed over the union of parsed source and target concepts. Let $V = O_s \cup O_t$. For token i, the document frequency is

$$\mathrm{df}_i = |v \in V : i \in L(v)| \,. \tag{5}$$

The smoothed inverse document frequency is

$$\mathrm{idf}_i = 1 + \log \frac{1 + |V|}{1 + \mathrm{df}_i}, \tag{6}$$

so idf_i lies in $[1,\ 1 + \log(1 + |V|)]$. Term frequency is binary at the concept level:

$$\mathrm{tf}_{v,i} = \begin{cases} 1, & \text{if token } i \text{ appears in any normalised label of } v, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

The own-label vector is $X_i(v) = \mathrm{tf}_{v,i}\mathrm{idf}_i$. Before cosine or SimHash projection, $X(v)$ is L_2-normalised when non-zero; weighted Jaccard uses the unnormalised non-negative weights.

ARB uses ancestor and neighbour context for different purposes. Ancestor evidence supports candidate rescue, whereas immediate neighbour labels support local disambiguation. The ancestor-pooled vector is

$$A(v) = X(v) + \sum_{a \in \mathrm{Anc}_d(v)} \alpha^{\mathrm{dist}(v,a)} X(a), \qquad \alpha = 0.60,\ d = 2. \tag{8}$$

The neighbour token set is

$$N(v) = \bigcup_{u \in M(v)} L(u). \tag{9}$$

Ancestor pooling is used by the hierarchy rescue channel and the context term in the score. Neighbour tokens are used by the neighbour rescue channel and by structural pruning.

For non-negative vectors u and v, weighted Jaccard is defined as

$$J_w(u, v) = \frac{\sum_i \min(u_i, v_i)}{\sum_i \max(u_i, v_i)}, \tag{10}$$

with $J_w = 0$ when both vectors are zero. The blocking-stage pair score is

$$\mathrm{score}(s, t) = \beta J_w(X(s), X(t)) + (1 - \beta) J_w(A(s), A(t)), \qquad \beta = 0.70. \tag{11}$$

Both terms lie in $[0, 1]$, so the score also lies in $[0, 1]$. The first term rewards direct lexical agreement, while the second gives limited credit to shared ancestor context. The local structural signal used by pruning is kept outside Equation (11). This separation makes a pruning decision auditable as a structural-disambiguation decision rather than as an opaque score change.

4.3. Token Base and SimHash Indexing

The token inverted-index base is a prior-art starting point. For each own-label token, ARB indexes the source and target concepts that contain it. If a bucket contains at most `max_bucket = 60` concepts in total, all cross-ontology pairs in that bucket are added to C_{base}. Larger buckets are skipped and counted in diagnostics. This keeps high-frequency biomedical tokens from creating a quadratic local blow-up while preserving the traceability of every emitted base pair.

The hierarchy rescue channel uses SimHash over ancestor-pooled vectors, following the locality-sensitive similarity-estimation idea of Charikar [33]. For each bit position $j \in 1, \ldots, 128$ and token i, ARB derives a deterministic sign $r_{ij} \in -1, +1$ from BLAKE2b(token, j). The jth hash bit for concept v is

$$h_j(v) = \mathbf{1}\left[\sum_i A_i(v) r_{ij} \geq 0\right]. \tag{12}$$

The 128-bit code is partitioned into 12 contiguous LSH bands: the first eight bands contain 11 bits each, and the remaining four contain 10 bits each. Two concepts collide in the hierarchy channel when they share at least one band signature and belong to different ontologies. As with token buckets, over-large hash buckets are skipped and reported.

4.4. Purity-Gated Context Rescue

Context expansion is needed when a true pair shares no own-label token, but unfiltered expansion can produce many false candidates. ARB therefore proposes rescue candidates through two channels and admits only candidates with lexical, structural, or cross-channel support. The hierarchy channel creates C_{hier} from SimHash/LSH collisions over $A(v)$. The neighbour channel creates C_{nbr} by indexing the extended token set $L(v) \cup N(v)$. The raw rescue set is

$$\Delta C_{\mathrm{raw}} = (C_{\mathrm{hier}} \cup C_{\mathrm{nbr}}) \setminus C_{\mathrm{base}}. \tag{13}$$

For a raw rescue pair (s, t), let $m(s, t) = \mathbf{1}[(s, t) \in C_{\mathrm{hier}}] + \mathbf{1}[(s, t) \in C_{\mathrm{nbr}}]$ denote the number of rescue

channels that proposed it. The pair passes the gate if at least one of the following conditions holds:

$$\begin{aligned} &\mathrm{score}(s,t) \geq \texttt{rescue\_floor} = 0.35, \quad \mathrm{struct}(s,t) \geq \texttt{structural\_floor} = 0.10, \\ &\text{or} \quad m(s,t) \geq \texttt{min\_channels} = 2. \end{aligned} \tag{14}$$

For each source concept, passing candidates are ranked by

$$g(s,t) = \mathrm{score}(s,t) + 0.25\,\mathrm{struct}(s,t) + 0.05\,\mathbf{1}[m(s,t) \geq 2], \tag{15}$$

where 0.25 is the fixed structural weight and 0.05 is the fixed cross-channel redundancy bonus. Candidates are sorted by decreasing g, then decreasing score, then decreasing struct, and finally by ascending target URI; this complete order is deterministic. At most $k = 3$ candidates are retained for each source concept. The retained rescue set ΔC_{pass} is then unioned with the base:

$$C_1 = C_{\mathrm{base}} \cup \Delta C_{\mathrm{pass}}. \tag{16}$$

By set inclusion, rescue cannot lower candidate recall before pruning. The gate and per-source budget limit the purity risk before the union with C_{base} is emitted. Algorithm 1 summarises the complete rescue procedure.

Algorithm 1 Purity-Gated Context Rescue

Input: C_{base}, feature store X, A, N, hierarchy parameters (K, b), gate thresholds, and rescue budget k.
Output: C_1 and retained rescue set ΔC_{pass}.

Initialise $C_{\mathrm{hier}} \leftarrow \emptyset$, $C_{\mathrm{nbr}} \leftarrow \emptyset$, and $\Delta C_{\mathrm{pass}} \leftarrow \emptyset$.
For each concept $v \in O_s \cup O_t$, compute the 128-bit SimHash code from $A(v)$.
For each of the 12 LSH bands, group concepts by band signature.
For each non-skipped band bucket containing both ontologies, add all cross-ontology pairs to C_{hier}.
Build an inverted index over $L(v) \cup N(v)$ for all concepts.
For each non-skipped extended-label bucket containing both ontologies, add all cross-ontology pairs to C_{nbr}.
Set $\Delta C_{\mathrm{raw}} \leftarrow (C_{\mathrm{hier}} \cup C_{\mathrm{nbr}}) \setminus C_{\mathrm{base}}$.
For each source concept s with at least one pair in ΔC_{raw}, collect targets t satisfying Equation (14).
Rank the collected targets by $g(s,t)$ in Equation (15); retain the first k pairs and add them to ΔC_{pass}.
Return $C_1 \leftarrow C_{\mathrm{base}} \cup \Delta C_{\mathrm{pass}}$ and ΔC_{pass}.

4.5. Same-Source Structural Dominance Pruning

The pruning step targets a different error mode from rescue. Biomedical ontologies often contain sibling concepts whose labels share most tokens but differ in laterality, anatomical scope, or subtype. When several such targets compete for one source concept, a lexical scorer may assign near-equal scores, leaving the downstream assignment step to make a brittle choice. ARB uses local graph evidence before assignment through a same-source dominance rule.

For source concept s and target concept t, structural overlap is

$$\mathrm{struct}(s,t) = \max\big(J(N_P(s), N_P(t)), \quad J(N_D(s), N_D(t))\big), \tag{17}$$

where $N_P(v)$ and $N_D(v)$ are the token sets collected from parent labels and child labels of v, respectively, and $J(A,B) = |A \cap B|/|A \cup B|$ when $A \cup B \neq \emptyset$, with $J(A,B) = 0$ when either set is empty (including when both are empty). Thus, an unavailable parent or child neighbourhood contributes zero rather than being skipped; struct retains the informative direction, if any. Parent and child overlap are not averaged because one side of a biomedical ontology may be shallow in one direction and informative in the other.

Otsu calibration [34] supplies a per-task threshold for structural evidence. ARB first forms the score-qualified set

$$Q = (s,t) \in C_1 : \mathrm{score}(s,t) \geq \texttt{accept} = 0.45. \tag{18}$$

It then histograms $\mathrm{struct}(s,t) : (s,t) \in Q$ into 256 bins over $[0,1]$ and selects θ^* by minimising the within-class variance. The threshold is adaptive but not gold-calibrated: it uses only scores and structures computed from the two input ontologies. This per-task calibration avoids imposing a single structural cutoff on Anatomy, FMA–NCI, FMA–SNOMED, and HP–MP, whose neighbourhood-depth and label-overlap distributions differ.

A candidate (s,t) is removed only when a same-source competitor has substantially stronger structure and is not lexically worse:

$$\exists, t' \in Q_s : \mathrm{struct}(s, t') - \mathrm{struct}(s, t) \geq \texttt{struct\_margin} = 0.10 \tag{19}$$

and

$$\mathrm{score}(s, t') \geq \mathrm{score}(s, t) - \texttt{lex\_slack}, \qquad \texttt{lex\_slack} = 0.10. \tag{20}$$

The lexical slack is a guardrail: a structurally supported competitor cannot delete a much stronger lexical match. Algorithm 2 summarises this pruning procedure.

Algorithm 2 Same-Source Structural Dominance Pruning

Input: candidate set C_1, score function, structural-overlap function, `accept`, `struct_margin`, and `lex_slack`.
Output: pruned candidate set C_{pruned}.

Set $Q \leftarrow (s, t) \in C_1 : \mathrm{score}(s, t) \geq \texttt{accept}$.
Compute $\mathrm{struct}(s, t)$ for every $(s, t) \in Q$.
Compute θ^* by Otsu thresholding on the structural values in Q.
Initialize $R \leftarrow \emptyset$.
For each source concept s represented in Q, let $Q_s = t : (s, t) \in Q$.
For each target $t \in Q_s$ with $\mathrm{struct}(s, t) < \theta^*$, search $Q_s \setminus t$ for a competitor t' satisfying Equations (19) and (20).
If such t' exists, add (s, t) to removal set R.
Return $C_{\mathrm{pruned}} \leftarrow C_1 \setminus R$.

4.6. End-to-End Pipeline and Fixed Matcher

Algorithm 3 summarises the full pipeline and makes the module boundary explicit. The blocking framework stops at C_{pruned}. The Hungarian assignment is retained as a fixed downstream check rather than as the paper's main contribution. It is applied component-wise using `scipy.optimize.linear_sum_assignment` [35]. Ties are broken deterministically by $(-\mathrm{score}, \mathrm{src_u ri}, \mathrm{tgt_u ri})$.

Algorithm 3 ARB Blocking Pipeline

Input: source ontology O_s, target ontology O_t, and configuration parameters.
Output: candidate table C_{pruned}, diagnostics, and optional downstream alignment A for evaluation.

Parse OWL/RDF into concept identifiers, label bags, and parent–child edges.
For each concept $v \in O_s \cup O_t$, build $L(v)$, $X(v)$, $A(v)$, and $N(v)$.
Build token buckets from own-label tokens.
For each non-skipped token bucket containing source and target concepts, add all cross-ontology pairs to C_{base}.
Run Algorithm 1 to obtain C_1.
Run Algorithm 2 to obtain C_{pruned}.
Emit candidate rows with source URI, target URI, generation channel, $\mathrm{score}(s, t)$, $\mathrm{struct}(s, t)$, gate status, and pruning status.
If downstream validation is requested, run the fixed component-wise Hungarian matcher on score-qualified retained candidates.
Load the reference alignment only after the above steps, and compute blocking and final matching metrics.

4.7. Parameter Provenance

Complete implementation and reproducibility details are provided in Appendix A. Table 2 lists the hyperparameters and their provenance. These hyperparameters define a fixed, auditable pipeline configuration, not learned model parameters. The method is not parameter-free, and the benchmark families named in Section 5 and Appendix A.4 informed the development sweep for selected thresholds. However, reference alignments were not read by the inference pipeline and were not used to retune parameters during the formal reported runs; after each output was produced, references were used only for metric computation and post-hoc error labels. Values such as $K = 128$, $b = 12$, `max_bucket` = 60, and `depth` = 2 are engineering priors chosen for deterministic memory and runtime control. The acceptance and rescue thresholds were fixed through the development sweep before the reported evaluation tables were finalised. The rescue budget $k = 3$ is examined by the reported sensitivity sweep in Section 6: larger budgets increase recall on recall-bottleneck tasks but inject false candidates on HP–MP. The Otsu threshold θ^* is the only per-task adaptive parameter, and its input is the unlabelled structural-overlap distribution of score-qualified candidates. The study does not claim that the development-fixed parameters are independently calibrated for unseen ontology pairs.

Table 2. ARB hyperparameters and provenance. Values are fixed before test evaluation unless marked as runtime adaptive. "Engineering prior" means a resource-control or representation choice made before benchmark scoring; "development-fixed" means fixed during the archived development sweep rather than adjusted on reported test references.

Parameter	Symbol	Default	Determination
SimHash code length	K	128	Engineering prior
LSH bands	b	12	Engineering prior
Max bucket size	max_bucket	60	Runtime/cost prior
Score context weight	$1-\beta$	0.30	Development-fixed
Ancestor decay	α	0.60	Development-fixed
Ancestor depth	depth	2	Engineering prior
Acceptance threshold	accept	0.45	Development-fixed
Rescue lexical floor	rescue_floor	0.35	Development-fixed
Rescue struct. floor	structural_floor	0.10	Development-fixed
Rescue min channels	min_channels	2	Redundancy prior
Rank structural weight		0.25	Development-fixed
Rank redundancy bonus		0.05	Development-fixed
Rescue budget	k	3	Sensitivity-supported
Dominance struct. margin	struct_margin	0.10	Development-fixed
Dominance lex. slack	lex_slack	0.10	Development-fixed
Otsu pruning threshold	θ^*	per-task	Runtime adaptive

5. Experimental Design

5.1. Biomedical Benchmark Settings and Preprocessing

The evaluation uses two complementary biomedical benchmark settings. The first is a set of seven OAEI tasks covering ontology families commonly used in biomedical AI integration: anatomy resources (FMA, SNOMED CT, and Anatomy), oncology terminology (NCI), disease vocabularies (DOID and ORDO), and phenotype resources (HP and MP). These tasks are not based on the newest ontology releases, but they provide stable OWL/RDF files and fixed reference alignments, which are necessary for controlled blocking-stage evaluation [8]. The LargeBio tasks, including FMA–NCI-small, FMA–NCI-full, FMA–SNOMED-small, and SNOMED–NCI-small, pair FMA, SNOMED CT, and NCI under UMLS-derived reference alignments. Anatomy belongs to the OAEI Anatomy track, while DOID–ORDO and HP–MP represent disease and phenotype matching settings. Table 3 reports the parsed concept counts and reference sizes used by the local experimental pipeline.

Table 3. Biomedical benchmark coverage. Counts are parsed labelled classes and reference pairs from the local benchmark files used by the experimental pipeline; HP-MP is marked partial because its 696 reference pairs are a known subset.

Task	Source	Target	$\lvert O_s \rvert$	$\lvert O_t \rvert$	Ref.	Completeness	Biomedical Domain/Source
Anatomy	Mouse anatomy	Human anatomy	3082	9402	1516	Full	Anatomy; OAEI Anatomy
FMA-NCI-small	FMA	NCI	3696	6488	2931	Full	Anatomy/oncology; OAEI LargeBio
FMA-NCI-full	FMA	NCI	78,988	66,724	2931	Full	Anatomy/oncology; OAEI LargeBio
FMA-SNOMED-small	FMA	SNOMED CT	10,157	13,412	8941	Full	Anatomy/clinical body structures; OAEI LargeBio
SNOMED-NCI-small	SNOMED CT	NCI	51,128	23,958	18,476	Full	Clinical terminology/oncology; OAEI LargeBio
DOID-ORDO	DOID	ORDO	10,879	13,525	1237	Full	Disease/rare disease; public biomedical benchmark
HP-MP †	HP	MP	12,786	11,940	696	Partial	Human/model-organism phenotype; public biomedical benchmark

† HP-MP reference pairs are interpreted as known-reference pairs rather than a complete alignment.

HP–MP is treated as a partial-reference task throughout the study. Its 696 reference pairs are interpreted as a known subset rather than as a complete alignment. Therefore, candidate purity, precision, recall, and F_1 on HP–MP are known-reference estimates. The task is included to observe ARB's behaviour on phenotype ontologies, but its scores are not used to make full-reference claims and are not directly compared in strength with Anatomy, LargeBio, or DOID–ORDO.

The second benchmark setting is Bio-ML 2024, which complements the OAEI suite with a ranking-oriented biomedical retrieval protocol [12]. All five Bio-ML 2024 equivalence tasks are used: OMIM–ORDO, NCIT–DOID, SNOMED–FMA (Body), SNOMED–NCIT (Pharm), and SNOMED–NCIT (Neoplas). In this protocol,

each source concept is supplied with a predefined candidate list, and the method only ranks the candidates by its score. Mean reciprocal rank (MRR) and Hits@1 are then used to evaluate whether the symbolic score can order medically plausible candidates inside an existing pool. This setting does not evaluate candidate generation and does not introduce a downstream matcher.

For preprocessing and version control, all experiments use the ontology files distributed with the corresponding benchmark package: OAEI 2013 files for the OAEI tasks and Bio-ML 2024 files for the local-ranking tasks. Newer ontology releases are not mixed with older reference alignments, because this would create version mismatch between concept identifiers and reference pairs. If later ontology snapshots are used in future work, the snapshot identifier and the corresponding reference alignment should be reported as a separate dataset condition.

For the experiments reported here, every named class in the distributed OWL/RDF files is retained, including classes marked as obsolete or deprecated when they remain in the benchmark file. All `rdfs:label` values and synonym-like annotation strings parsed by the OWL/RDF reader are used as label evidence. No manual label selection or external synonym expansion is applied. Blank-node restrictions and complex class expressions are not expanded into additional inferred parents, consistent with the problem setting in Section 3. These choices keep every emitted candidate traceable to ontology-internal biomedical evidence, which is the audit condition studied in this paper.

5.2. Reference Isolation and Parameter Control

Reference isolation is part of the experimental protocol rather than an implementation-side claim. In an audit-sensitive biomedical pipeline, each candidate should be justified by ontology labels and local graph context, not by hidden supervision from the benchmark alignment. Therefore, reference alignments are withheld from candidate generation, scoring, purity gating, Otsu thresholding, pruning, and Hungarian assignment. For each run, the candidate table and optional downstream alignment are produced before the corresponding reference file is loaded. The reference is then used only for evaluation: computing blocking metrics, computing final $P/R/F_1$ under the fixed downstream matcher, and assigning post-hoc error categories in Section 7.

Parameter values are fixed before the reported test evaluations. ARB does not learn model parameters and therefore does not use a training/validation/test split in the machine-learning sense. The reported evaluation applies a fixed, auditable configuration to the benchmark tasks. Representation and resource-control parameters, including $K = 128$, $b = 12$, `max_bucket = 60`, and ancestor depth $d = 2$, are set as engineering priors. Thresholds and weights labelled as development-fixed in Table 2 were selected from a development sweep conducted before the reported manuscript tables were finalised. The sweep used the same benchmark configuration families later reported in the paper—Anatomy, FMA–NCI, FMA–SNOMED, SNOMED–NCI, DOID–ORDO, and HP–MP—not disjoint held-aside ontology pairs. These benchmark families therefore participated in parameter development, although the sweep was used to inspect parameter behaviour, not to establish transfer to unseen ontology pairs. After this provenance check, all development-fixed values were frozen. In the formal reported runs, reference alignments were loaded only after the candidate and downstream alignment outputs had been produced, for metric computation and post-hoc error labels. The rescue budget $k = 3$ is examined by the reported sensitivity sweep in Section 6. The Otsu threshold θ^* is the only per-task adaptive value; it is computed at runtime from the unlabelled structural-overlap distribution of score-qualified candidates and is not fitted to reference pairs.

5.3. Ablation Design, Baselines, and Evaluation Metrics

The main OAEI ablation follows a 2×2 factorial design. The generation factor is either the token base alone, denoted as *own*, or the token base with purity-gated context rescue, denoted as *own + I1*. The purification factor is either disabled or enabled through same-source structural dominance pruning, denoted as I2. The downstream Hungarian matcher is held fixed across all four cells, so changes in final $P/R/F_1$ can be interpreted together with changes in candidate recall, candidate purity, and pool size.

The token base (*own*, off) is the primary baseline: standard token inverted-index blocking without rescue or pruning. Character 3-gram blocking is included as a second lexical blocking baseline. Weighted edge pruning (WEP) is included as a meta-blocking baseline applied on top of the token candidate graph. The SimHash base uses own-label SimHash banding without the token base or rescue channels and is used as an efficiency ablation. Across six tasks, this SimHash-base variant produces 4–27$\times$ more candidates than the token base at equal or lower candidate recall, which leaves token blocking both simpler and more efficient as the primary symbolic base.

The primary blocking metrics are candidate recall PC, candidate purity PQ, candidate-pool size $|C|$, and pool growth percentage relative to the base configuration. These metrics are reported for all ablation cells. Secondary downstream metrics for the OAEI tasks are precision P, recall R, and F_1, computed after applying the same fixed Hungarian matcher. For the Bio-ML local-ranking tasks, the reported metrics are MRR and Hits@1.

All experiments are run on a local workstation with an Apple M4 Pro 12-core CPU and 24 GB memory (Apple Inc., Cupertino, CA, USA). The implementation uses Python 3 with `numpy`, `scipy`, and `xml.etree`; no GPU is used. SimHash projections are generated with a seeded BLAKE2b hash function, and `PYTHONHASHSEED = 0` is set for deterministic dictionary ordering. These implementation settings support reproducibility, while reference isolation follows from the experimental protocol described above.

5.4. Uncertainty Reporting

The pipeline is deterministic for fixed inputs and configuration, so repeated runs would reproduce the same cell rather than estimate sampling variability. For the principal token-base comparisons, we therefore report nonparametric 95% bootstrap intervals over source concepts (1000 resamples; paired resampling for configuration differences). This quantifies uncertainty with respect to the benchmark reference units, not implementation randomness or clinical generalisability. The intervals are not used as random-algorithm significance tests; the main target remains candidate-generation behaviour under a fixed configuration. The bootstrap analysis is reported for candidate recall and fixed-validator F_1; HP–MP retains its partial-reference qualification.

6. Results and Ablation Study

This section reports candidate-pool quality before discussing downstream matching scores. The main evaluation separates blocking behaviour from final assignment: candidate recall, candidate purity, and pool size describe the candidate-generation stage, whereas final precision, recall, and F_1 are reported only under the same fixed Hungarian validator.

6.1. OAEI 2013 Main Ablation Results

Table 4 reports the full 2×2 ablation on the seven OAEI biomedical tasks. The two axes are candidate generation and pruning. Candidate generation is either the token inverted-index base (*own*) or the same base with purity-gated context rescue (*own + I1*); pruning is either disabled or performed by same-source structural dominance pruning (I2). Focused channel, gate, and order variants are reported in a later subsection.

Table 4. Ablation over all OAEI tasks (token-inverted-index base). gen = own: base only; gen = own + I1: with purity-gated context rescue. purify = off/I2: without/with same-source structural dominance pruning (Otsu). Columns: PC = candidate recall; Pur = candidate purity; $|C|$ = candidate-pool size; growth is relative to the own/off token base for the same task; ΔC = candidates removed by I2 (known-gold removed in parentheses); final P/R/F1 against full reference (HP-MP: partial reference †). All configurations use the same component-wise Hungarian matcher.

Task	Config	PC	Pur	$\lvert C\rvert$	Growth	ΔC (Gold)	P	R	F1
Anatomy	own, off	0.837	0.0142	89,560	0.0%	—	0.198	0.196	0.197
	own, I2	0.834	0.0144	88,186	−1.5%	1374 (4)	0.373	0.359	0.366
	own + I1, off	0.893	0.0144	94,296	+5.3%	—	0.204	0.213	0.209
	own + I1, I2	0.890	0.0145	92,908	+3.7%	1388 (4)	0.372	0.377	**0.375**
FMA-NCI (small)	own, off	0.750	0.0298	73,748	0.0%	—	0.870	0.516	0.648
	own, I2	0.748	0.0299	73,374	−0.5%	374 (8)	0.882	0.518	0.652
	own + I1, off	0.800	0.0286	82,110	+11.3%	—	0.869	0.521	0.652
	own + I1, I2	0.797	0.0286	81,734	+10.8%	376 (8)	0.882	0.523	**0.657**
FMA-NCI (full)	own, off	0.331	0.0064	150,509	0.0%	—	0.550	0.240	0.334
	own, I2	0.329	0.0064	150,142	−0.2%	367 (5)	0.569	0.243	0.340
	own + I1, off	0.439	0.0069	187,199	+24.4%	—	0.517	0.311	0.388
	own + I1, I2	0.438	0.0069	186,869	+24.2%	330 (3)	0.531	0.314	**0.395**
FMA-SNOMED (small)	own, off	0.710	0.0246	258,494	0.0%	—	0.884	0.366	0.518
	own, I2	0.703	0.0244	257,982	−0.2%	512 (59) *	0.889	0.366	0.518
	own + I1, off	0.794	0.0249	285,402	+10.4%	—	0.872	0.400	**0.549**
	own + I1, I2	0.787	0.0247	284,833	+10.2%	569 (66) *	0.876	0.399	0.548
SNOMED-NCI (small)	own, off	0.548	0.0314	321,945	0.0%	—	0.878	0.347	0.497
	own, I2	0.546	0.0314	321,566	−0.1%	379 (31)	0.881	0.347	0.498
	own + I1, off	0.629	0.0310	375,189	+16.5%	—	0.860	0.383	0.530
	own + I1, I2	0.627	0.0309	374,788	+16.4%	401 (33)	0.863	0.383	**0.531**
DOID-ORDO	own, off	0.971	0.0066	181,465	0.0%	—	0.576	0.936	0.713
	own, I2	0.971	0.0067	179,852	−0.9%	1613 (0)	0.589	0.941	**0.724**
	own + I1, off	0.991	0.0062	196,328	+8.2%	—	0.560	0.952	0.705
	own + I1, I2	0.991	0.0063	194,668	+7.3%	1660 (0)	0.577	0.958	0.720
HP-MP †	own, off	0.963	0.0035	192,439	0.0%	—	0.454	0.908	**0.605**
	own, I2	0.963	0.0035	191,999	−0.2%	440 (0)	0.461	0.915	0.613
	own + I1, off	0.991	0.0032	215,969	+12.2%	—	0.431	0.928	0.588
	own + I1, I2	0.991	0.0032	215,478	+12.0%	491 (0)	0.436	0.934	0.595

* See Section 7 for categorical analysis of removed gold pairs; † HP-MP uses partial reference (696 pairs); purity is a known-reference proxy.

Table 5 adds source-level bootstrap intervals for the archived headline candidate-recall and fixed-validator F_1 differences. The sampling unit is the source concept; we use 1000 nonparametric bootstrap resamples, paired resampling for configuration differences, and two-sided 95% percentile intervals. These intervals cover six source-resampled comparisons; the full-scale FMA–NCI run is reported deterministically in Table 4. The bootstrap candidate-recall gains are positive across the full-reference comparisons included in the table. In contrast, the F_1 evidence is heterogeneous: the Anatomy effect is large, the FMA–NCI-small effect is small, the FMA–SNOMED-small gain is driven by rescue rather than I2, and the HP–MP full-configuration interval crosses zero under the partial reference. We therefore treat sub-0.01 changes as small or uncertain rather than as decisive improvements.

Table 5. Source-level bootstrap uncertainty for the full ARB configuration versus the token base (1000 paired resamples). Values are absolute differences; intervals quantify reference-unit sampling uncertainty, not run-to-run variation. HP–MP uses a partial reference.

Task	Δ PC	95% CI for Δ PC	ΔF_1 (95% CI)
Anatomy	+0.0528	[0.0410, 0.0645]	+0.1779 [0.1589, 0.1960]
FMA–NCI-small	+0.0467	[0.0389, 0.0546]	+0.0089 [0.0041, 0.0137]
FMA–SNOMED-small	+0.0769	[0.0706, 0.0830]	+0.0302 [0.0254, 0.0348]
SNOMED–NCI-small	+0.0799	[0.0757, 0.0839]	+0.0334 [0.0304, 0.0362]
DOID–ORDO	+0.0202	[0.0128, 0.0294]	+0.0069 [−0.0010, 0.0148]
HP–MP †	+0.0287	[0.0178, 0.0418]	−0.0104 [−0.0207, 0.0007]

† Candidate purity and final metrics are known-reference estimates for HP–MP.

Purity-gated context rescue improves candidate recall on all evaluated tasks, but its downstream effect depends on how much recall headroom remains. On Anatomy, I1 raises candidate recall from 0.837 to 0.893, a 5.6 percentage-point gain, while the candidate pool grows from 89k to 94k pairs (+5.3%) and candidate purity increases slightly from 0.0142 to 0.0144. On FMA–NCI-full, the recall gain is larger, rising from 0.331 to 0.439, or about 10.8 percentage points, at +24.4% pool growth. HP–MP is the exception. Its token base already reaches candidate recall 0.963; rescue raises this value to 0.991, but final F_1 falls from 0.605 to 0.588 after I1 and remains below the own-only value after I2 (0.595). In the pool-control run, gated rescue adds 23,530 candidate pairs while recovering only about 20 additional known-reference pairs under the partial HP–MP reference.

Same-source structural dominance pruning has its strongest effect on Anatomy. Adding I2 to the own-only base raises final F_1 from 0.197 to 0.366 (+0.169), precision from 0.198 to 0.373 (+0.175), and recall from 0.196 to 0.359 (+0.163). Of the 1374 candidates removed by I2, only 4 are known-gold pairs (0.3%), indicating that the rule mainly removes false siblings on this task. On FMA–NCI-small, I2 removes 374 candidates, including 8 gold pairs, and gives a smaller F_1 gain of 0.005. On HP–MP, where the reference is partial, I2 removes 440 candidates with no known-gold loss and improves F_1 by 0.008 in the own-only setting.

The full ARB configuration, *own + I1* with I2, obtains $F_1 = 0.375$ on Anatomy, which is +0.178 over the own-only base. It reaches $F_1 = 0.657$ on FMA–NCI-small and $F_1 = 0.395$ on FMA–NCI-full. On DOID–ORDO, it reaches $F_1 = 0.720$, supported by high candidate recall (0.991) and modest pruning gains. These final F_1 values should be interpreted with caution. They are obtained with a simple fixed Hungarian validator and are not intended to represent full-system ontology matching performance. The stronger evidence for ARB lies at the candidate-pool level.

On the SNOMED-related tasks, I1 gives visible candidate-recall gains, but final recall remains limited. On FMA–SNOMED-small and SNOMED–NCI-small, the full ARB configuration records fixed-validator F_1 differences of 0.030 and 0.034 over the own-only base, respectively, rather than establishing a broad performance improvement. Final precision is high, around 0.88, but final recall remains low, around 0.37–0.40, holding F_1 near 0.50–0.53. This reflects the remaining gap between candidate recall (0.63–0.79) and the full reference alignment. The I2-only movement on FMA–SNOMED-small is not presented as a positive effect: its F_1 change is at most 0.0003 and I2 removes 59–66 known-reference pairs, depending on the generation setting. Additional candidate-recall improvement, or a stronger downstream scorer, would be needed to close this gap.

6.2. Candidate-Pool Growth and Purity Control

Table 6 isolates the effect of the purity gate on three representative tasks. Without the gate, rescue inflates the candidate pool by 455% on Anatomy (89k → 497k pairs), 521% on FMA-NCI-small (74k → 458k pairs), and 1065% on HP-MP (192k → 2242k pairs), with candidate purity falling by 5–11×. With the gate, pool growth contracts to 5.3%, 11.3%, and 12.2% respectively; purity rises on Anatomy (0.01417 → 0.01435), changes from 0.02982 to 0.02855 on FMA–NCI-small, and changes from 0.00348 to 0.00319 on HP–MP. The ungated variant

shows a 5–11× collapse.

Table 6. Effect of the purity gate on candidate pool size and purity (candidate-pair precision). Ungated rescue grows the pool by 455–1065% and lowers purity by 5–11×; the purity gate constrains growth to 5–12% and keeps purity close to the base value (far above the ungated case).

Task	Configuration	Pairs	Growth	Purity	Cand. Recall
Anatomy	Base (token)	89,560	—	0.01417	0.837
	+Rescue (no gate)	496,736	+455%	0.00280	0.916
	+Rescue (gated)	94,296	+5.3%	0.01435	0.893
FMA-NCI (small)	Base (token)	73,748	—	0.02982	0.750
	+Rescue (no gate)	457,688	+521%	0.00537	0.839
	+Rescue (gated)	82,110	+11.3%	0.02855	0.800
HP-MP	Base (token)	192,439	—	0.00348	0.963
	+Rescue (no gate)	2,241,758	+1065%	0.00031	0.994
	+Rescue (gated)	215,969	+12.2%	0.00319	0.991

Figure 2 renders the three scenarios after the tabular comparison.

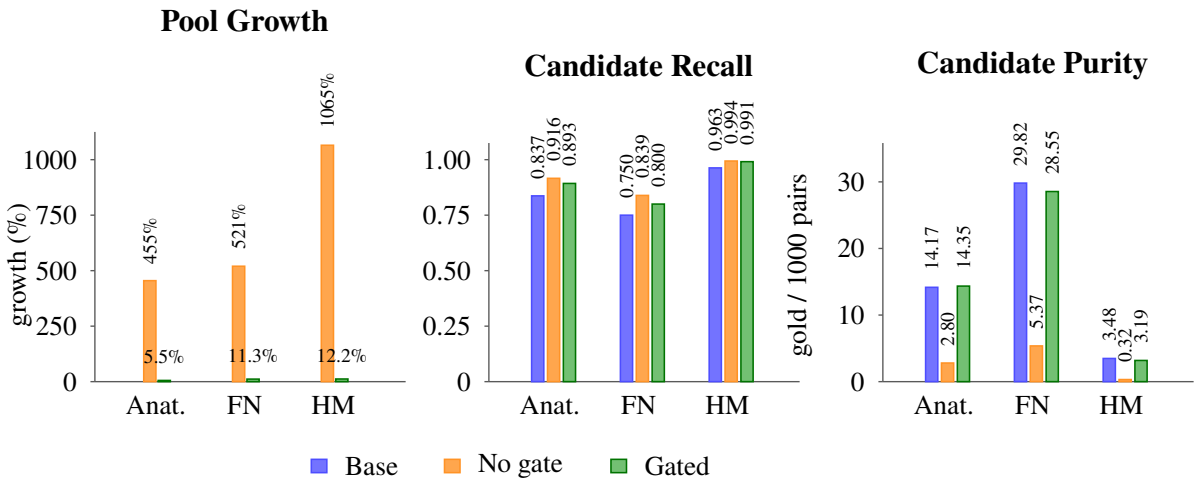

Figure 2. Effect of the purity gate across three biomedical tasks. Anat. denotes Anatomy, FN denotes FMA–NCI-small, and HM denotes HP–MP. Ungated rescue gives the largest candidate-recall gains but inflates the candidate pool by 455–1065% and lowers candidate purity. The gate keeps pool growth to 5–12%, preserves most of the recall gain, and avoids the purity collapse; HP-MP remains the recall-saturated exception.

6.3. Focused Ablations and Component Contributions

The focused ablations in Table 7 identify which parts of I1 carry the recall gain. In the SimHash-base single-channel runs, the neighbour channel is consistently stronger than the hierarchy channel. On FMA–SNOMED-small, candidate recall rises from 0.585 to 0.705 with the neighbour channel, compared with 0.648 with the hierarchy channel; combining both channels reaches 0.723. The same pattern appears on Anatomy and FMA–NCI-small. This indicates that local neighbour evidence contributes more missed candidates than ancestor evidence alone in these tasks.

Table 7. Focused ablations for rescue channels and gate predicates. Single-channel runs use the SimHash-base variant, so their absolute values are not directly comparable with the token-base ablation. Gate-predicate runs use the token base and report PC/F1.

Study	Task	Own	+Hierarchy	+Neighbour	+Both/All
Single channel PC/F1	Anatomy	0.803/0.181	0.836/0.184	0.869/0.186	0.876/0.186
	FMA-NCI-small	0.639/0.655	0.675/0.661	0.742/0.668	0.746/0.669
	FMA-SNOMED-small	0.585/0.532	0.648/0.547	0.705/0.553	0.723/0.557
Gate predicate PC/F1	Anatomy	Lex 0.872/0.374	Struct 0.887/0.382	Channel 0.853/0.373	All 0.893/0.375
	FMA-NCI-small	Lex 0.759/0.656	Struct 0.797/0.656	Channel 0.755/0.654	All 0.800/0.657
	FMA-SNOMED-small	Lex 0.760/0.549	Struct 0.790/0.547	Channel 0.721/0.523	All 0.794/0.548

The gate-predicate ablation shows that the structural condition is the strongest admission signal among the tested predicates. With the token base, the structural-only predicate recovers candidate recall close to the full gate on all three tasks: 0.887 versus 0.893 on Anatomy, 0.797 versus 0.800 on FMA–NCI-small, and 0.790 versus 0.794 on FMA–SNOMED-small. The channel-only predicate is weaker, especially on FMA–SNOMED-small, where it reaches 0.721 candidate recall and $F_1 = 0.523$. These results suggest that local structural evidence is most useful as a gate for controlling which rescued candidates are admitted.

The interaction tests show little order dependence. On FMA–NCI-small, running I1 before I2 gives PC/F_1 = 0.7997/0.6565, while running I2 before I1 gives PC/F_1 = 0.7994/0.6565. The two orders are therefore nearly indistinguishable on this task. In contrast, removing the gate has a large effect: as shown in Table 6, ungated rescue expands the pool by 455–1065% and reduces candidate purity by 5–11×. Overall, the focused ablations indicate that the neighbour channel and structural gate account for most of the measured rescue gain.

6.4. Rescue Budget and Acceptance-Threshold Sensitivity

Figure 3 and Table 8 show the effect of the per-source rescue budget k. This sweep was run on the SimHash-base variant described in Section 5, so its absolute PC and F_1 values differ from the token-base results in Table 4. The purpose of this experiment is to examine the response to the rescue budget rather than to compare absolute scores.

Table 8. Rescue budget sensitivity (k = pairs per source retained from rescue channels), run on the SimHash-base variant; absolute PC/F1 differ from the token-base Table 4. Candidate recall rises with k on recall-bottleneck tasks (Anatomy, FMA-NCI) while final F1 stays flat; on HP-MP (recall-saturated) larger budgets lower F1. The budget response, not the absolute level, is the finding.

Task	*k*	PC	Purity	P	R	F1
Anatomy	1	0.819	0.00331	0.385	0.382	0.383
	2	0.848	0.00341	0.375	0.375	0.375
	3	0.857	0.00343	0.372	0.375	0.374
	5	0.868	0.00345	0.371	0.375	0.373
FMA-NCI (small)	1	0.678	0.00660	0.889	0.516	0.653
	2	0.705	0.00680	0.889	0.519	0.656
	3	0.721	0.00688	0.888	0.521	0.657
	5	0.734	0.00689	0.888	0.521	0.657
HP-MP	1	0.994	0.00037	0.461	0.953	0.622
	2	0.994	0.00037	0.452	0.948	0.612
	3	0.994	0.00037	0.451	0.950	0.612
	5	0.994	0.00037	0.449	0.948	0.609

On Anatomy, candidate recall increases from 0.819 at $k = 1$ to 0.868 at $k = 5$, while final F_1 remains between 0.373 and 0.383. On FMA–NCI-small, the same pattern appears: candidate recall rises with larger k, but final F_1 stays near a plateau. HP–MP behaves differently. Its F_1 decreases from 0.622 at $k = 1$ to 0.609 at $k = 5$, because the additional candidates mostly enlarge an already recall-saturated pool. The default value $k = 3$ is therefore a conservative operating point: it preserves the FMA plateau while avoiding the largest HP–MP degradation.

The acceptance threshold is more sensitive than the rescue budget (Table 9). Across the 0.35–0.55 sweep, FMA–NCI-small remains relatively stable, with F_1 ranging from 0.6388 to 0.6623. FMA–SNOMED-small is much more sensitive: its F_1 declines from 0.5483 at the default 0.45 to 0.3426 at 0.55. This default-relative drop of 0.206 is the largest adverse movement among the full-reference recall-bottleneck tasks. DOID–ORDO and HP–MP move in the opposite direction, improving at higher thresholds. These results indicate that `accept` = 0.45 should be treated as a development-fixed operating point, not as a universally calibrated threshold for biomedical ontology matching.

Table 9. Sensitivity of final F_1 to the acceptance threshold `accept`. The threshold is the largest observed fixed-threshold source of variation. Relative to the default 0.45, FMA-SNOMED-small drops by 0.206 at 0.55, while DOID-ORDO and HP-MP improve at higher thresholds.

`Accept`	Anatomy	FMA-NCI	FMA-SNOMED	SNOMED-NCI	DOID-ORDO	HP-MP †
0.35	0.3416	0.6623	0.5955	0.5367	0.5993	0.4497
0.40	0.3678	0.6581	0.5823	0.5321	0.6525	0.5133
0.45	0.3748	0.6565	0.5483	0.5308	0.7199	0.5947
0.50	0.3867	0.6463	0.4633	0.5143	0.7720	0.6888
0.55	0.3832	0.6388	0.3426	0.4957	0.8241	0.7778

† HP-MP uses a partial reference; final F_1 is a known-reference estimate.

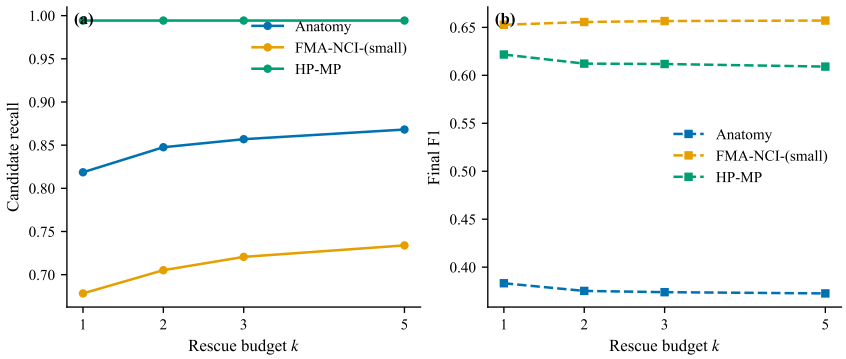

Figure 3. Rescue budget sensitivity on three tasks: (**a**) candidate recall versus rescue budget k; (**b**) final F_1 versus rescue budget k. Candidate recall rises smoothly on recall-bottleneck tasks such as Anatomy and FMA–NCI. Final F_1 remains relatively flat on those tasks but declines on HP–MP, where the base candidate pool is already recall-saturated (PC ≥ 0.96).

6.5. Baseline Comparison and Runtime Profile

The baseline comparison first examines alternative symbolic blocking choices. Token blocking is a stronger base than character 3-gram blocking for these biomedical labels. Across six OAEI tasks, token blocking obtains higher candidate recall on five tasks, produces smaller candidate pools on all tasks, and achieves higher candidate-pair purity on all tasks (Table 10). The only candidate-recall exception is FMA–NCI-small, where 3-gram blocking raises PC from 0.7503 to 0.8025. This gain comes at a high cost: the pool expands from 73,748 to 400,996 pairs and purity falls from 0.02982 to 0.00587.

Table 10. Token blocking versus character 3-gram blocking on six OAEI tasks. PC is candidate recall and Pur is candidate-pair purity. Token blocking is the stronger symbolic base in five of six tasks and keeps the pool 1.65–5.44× smaller with higher purity. The ratio column is 3-gram pairs divided by token pairs.

Task	Token PC	Token Pairs	Token Pur	3-Gram PC	3-Gram Pairs	3-Gram Pur	Ratio
Anatomy	0.8371	89,560	0.01417	0.6801	195,667	0.00527	2.18
FMA-NCI-small	0.7503	73,748	0.02982	0.8025	400,996	0.00587	5.44
FMA-SNOMED-small	0.7097	258,494	0.02455	0.6280	544,323	0.01032	2.11
SNOMED-NCI-small	0.5475	321,945	0.03141	0.3085	531,451	0.01072	1.65
DOID-ORDO	0.9709	181,465	0.00662	0.8294	629,499	0.00163	3.47
HP-MP †	0.9626	192,439	0.00348	0.7888	610,244	0.00090	3.17

† HP-MP uses a partial reference; purity is a known-reference proxy.

Table 11 adds a weighted edge pruning (WEP) meta-blocking baseline. WEP raises candidate-pair purity and reduces the token pool by 2.58–3.14× across the reported tasks, but it usually lowers candidate recall. The most favourable cases are DOID–ORDO and HP–MP, where WEP preserves the same known-reference candidate recall while substantially increasing purity. On FMA–SNOMED-small, however, WEP reduces PC from 0.7097 to 0.6839. This result shows a similar caution to I2: pruning can clean false siblings, but it can also remove valid cross-vocabulary biomedical candidates.

Table 11. Weighted edge pruning (WEP) meta-blocking baseline on the token candidate graph. The pruning threshold is the mean symbolic edge score for the task, computed without reference labels. RR is the blocking reduction ratio. The runtime and peak-memory column reports the shared token candidate-graph construction cost used by both rows for each task; WEP-specific reweighting overhead was not separately instrumented in this diagnostic run.

Task	Method	Threshold	PC	PQ	Pool Size	RR	Runtime/Peak Memory
Anatomy	Token blocking	–	0.8371	0.01417	89,560	0.996909	1.92 s/86.4 MB
Anatomy	WEP meta-blocking	0.1994	0.8245	0.03596	34,758	0.998800	1.92 s/86.4 MB
FMA-NCI-small	Token blocking	–	0.7503	0.02982	73,748	0.996925	1.47 s/54.0 MB
FMA-NCI-small	WEP meta-blocking	0.2224	0.7195	0.08104	26,024	0.998915	1.47 s/54.0 MB
FMA-NCI-full	Token blocking	–	0.3306	0.00644	150,509	0.999971	11.21 s/316.7 MB
FMA-NCI-full	WEP meta-blocking	0.1698	0.3299	0.01793	53,922	0.999990	11.21 s/316.7 MB
FMA-SNOMED-small	Token blocking	–	0.7097	0.02455	258,494	0.998102	5.23 s/173.6 MB
FMA-SNOMED-small	WEP meta-blocking	0.1765	0.6839	0.06911	88,479	0.999350	5.23 s/173.6 MB
SNOMED-NCI-small	Token blocking	–	0.5475	0.03141	321,945	0.999737	9.96 s/307.4 MB
SNOMED-NCI-small	WEP meta-blocking	0.1957	0.5272	0.08153	119,431	0.999902	9.96 s/307.4 MB
DOID-ORDO	Token blocking	–	0.9709	0.00662	181,465	0.998767	6.80 s/158.9 MB
DOID-ORDO	WEP meta-blocking	0.1689	0.9709	0.02077	57,834	0.999607	6.80 s/158.9 MB
HP-MP †	Token blocking	–	0.9626	0.00348	192,439	0.998739	6.25 s/161.4 MB
HP-MP †	WEP meta-blocking	0.1718	0.9626	0.00963	69,568	0.999544	6.25 s/161.4 MB

† HP-MP purity is a known-reference estimate.

LogMap and AML are used as full-system context rather than stage-level blocking baselines. Their published OAEI scores combine candidate generation, scoring, repair, filtering, and, in some configurations, biomedical background knowledge. A strict blocking-stage comparison would require instrumenting their intermediate candidate pools under the same preprocessing, resource, and evaluation protocol. Table 12 therefore marks the boundary between complete ontology matchers and the blocking module studied here: LogMap and AML report final precision, recall, and F_1, whereas ARB reports candidate-pool quality and auditable candidate cleaning.

Table 12. Official OAEI 2013 LargeBio full-matcher context for ontology pairs used in this study. Values are final alignment precision/recall/F_1 from the official LargeBio result tables [8], not candidate-pool metrics. ARB F_1 is shown only to mark the boundary between a blocking module plus a fixed Hungarian validator and complete biomedical ontology matchers.

Task	ARB Final F_1	LogMap P/R/F_1	AML P/R/F_1	Best Official OAEI 2013 F_1
FMA–NCI small	0.657	0.952/0.851/0.899	0.947/0.834/0.887	0.915 (LogMap-BK)
FMA–NCI full	0.395	0.874/0.795/0.832	0.880/0.730/0.798	0.872 (YAM++)
FMA–SNOMED small	0.548	0.966/0.656/0.782	0.943/0.720/0.816	0.836 (YAM++)
SNOMED–NCI small	0.531	0.896/0.672/0.768	0.895/0.642/0.747	0.770 (LogMap-BK)

Component-level profiling locates the main engineering cost. On a local workstation with an Apple M4 Pro 12-core CPU and 24 GB memory, using the Python implementation, FMA–SNOMED-small completes in 30.85 s and FMA–NCI-full completes in 119.52 s (Table 13). Candidate generation dominates wall time: 21.541 s (69.8%) on FMA–SNOMED-small and 77.234 s (64.6%) on FMA–NCI-full. On FMA–SNOMED-small, Otsu thresholding, I2 pruning, and Hungarian assignment together take 0.178 s, or 0.58% of 30.85 s; on FMA–NCI-full they take 0.129 s, or 0.11% of 119.52 s.

Table 13. Component-level runtime and peak memory on Apple M4 Pro (12-core CPU, 24 GB memory), Python implementation. Percentages are fractions of total wall time.

Stage	FMA-SNOMED-Small (10,157 × 13,412)	FMA-NCI-Full (78,988 × 66,724)
I/O parsing	0.544 s (1.8%)	2.857 s (2.4%)
Preprocessing (IDF + anc.)	0.583 s (1.9%)	3.649 s (3.1%)
Index/channel build	5.975 s (19.4%)	33.985 s (28.4%)
Candidate generation I1	21.541 s (69.8%)	77.234 s (64.6%)
Scoring	2.027 s (6.6%)	1.666 s (1.4%)
Otsu threshold	0.058 s (0.2%)	0.042 s (0.0%)
I2 pruning	0.083 s (0.3%)	0.052 s (0.0%)
Hungarian assignment	0.037 s (0.1%)	0.035 s (0.0%)
Total	30.85 s	119.52 s
Peak memory	975 MB	3173 MB

Table 14 summarises why the module remains practical at the reported biomedical scale. The most expensive operations are candidate-emitting indexes and rescue generation. In contrast, audit-sensitive pruning and downstream validation operate only on score-qualified candidates or connected conflict components. The diagnostics also make scale problems visible: skipped token buckets, skipped LSH buckets, rescue counts, Otsu thresholds, and assignment fallback status are reported rather than hidden inside the matcher.

Table 14. Stage-wise computational profile of ARB. m and n are source and target concept counts, T is the number of indexed tokens, B is the emitted candidate-pair count, R is the raw rescue candidate count, Q is the score-qualified candidate count, and c is a connected assignment component.

Stage	Main Cost	Audit/Scale Control
OWL/RDF parsing	$O(m+n+E)$ over labels and subclass edges	reports labelled concepts, isolated concepts, and parsed edge counts
Token inverted index	$O(T+B)$ after tokenisation	skips over-large buckets and records skipped bucket counts
Hierarchy SimHash rescue	$O((m+n)K+H)$ for K hash bits and LSH bucket emissions	deterministic BLAKE2b signs; skipped band buckets are reported
Neighbour rescue	$O(T_N+R_N)$ over neighbour-token index and emitted pairs	extended-label buckets above `max_bucket` are skipped
Purity gate	$O(R)$ scoring of raw rescue candidates plus per-source top-k selection	retains at most $k=3$ rescue candidates per source in the reported setting
Structural dominance pruning	$O(Q+\sum_s d_s^2)$ over score-qualified same-source competitors	Otsu threshold and removed/gold-removed counts are logged per task
Hungarian validation	$O(c^3)$ per connected conflict component	used only for fixed downstream validation; greedy fallback threshold was not triggered

6.6. Bio-ML 2024 Local Ranking

Bio-ML 2024 evaluates scoring on predefined candidate lists rather than candidate generation. On the five equivalence tasks, ARB attains a mean MRR of 0.813 and a mean Hits@1 of 0.760 (Table 15). Per-task MRR ranges from 0.724 on OMIM–ORDO to 0.942 on SNOMED–NCIT Pharm. The SNOMED–FMA Body task, whose source ontology contains about 88k concepts, records MRR 0.763, within the range of the smaller tasks. These figures are a matcher-independent check on symbolic scoring; they are not the primary blocking result.

Table 15. Bio-ML 2024 local ranking results (MRR and Hits@1). The system uses no embeddings or external knowledge; gold mappings are used only for evaluation. SNOMED–NCI and SNOMED–FMA are large tasks (>10k concepts each side).

Task	$\lvert O_s \rvert$	$\lvert O_t \rvert$	MRR	Hits@1
OMIM–ORDO	7000	12,000	0.724	0.670
NCIT–DOID	12,682	9405	0.833	0.777
SNOMED–FMA (Body)	88,329	21,119	0.763	0.688
SNOMED–NCIT (Pharm)	22,085	15,250	0.942	0.922
SNOMED–NCIT (Neoplas)	10,156	9803	0.804	0.741
Mean	—	—	**0.813**	**0.760**

Table 16 adds self-implemented local-ranking baselines on the same official candidate lists. Against the strongest single lexical baseline for each task, ARB differs by −0.003 on OMIM–ORDO, −0.008 on NCIT–DOID, +0.035 on SNOMED–FMA, +0.010 on SNOMED–NCIT Pharm, and −0.002 on SNOMED–NCIT Neoplas. Thus, ARB is above the strongest lexical baseline on two tasks and below it on three, while remaining within 0.010 MRR on four tasks. The ancestor-context-only score is weak by itself, indicating that local graph evidence is more useful as a disambiguating component than as a standalone biomedical ranking signal. These baselines do not evaluate candidate generation; they test whether the same symbolic evidence used by ARB gives a useful ordering inside Bio-ML's predefined biomedical candidate sets.

Table 16. Self-implemented Bio-ML 2024 local-ranking baselines on the same official candidate lists used for ARB scoring. Values are MRR except for the mean column, which reports MRR/Hits@1. No training mappings, embeddings, or external biomedical resources are used.

Method	OMIM–ORDO	NCIT–DOID	SNOMED–FMA	SNOMED–NCIT Pharm	SNOMED–NCIT Neoplas	Mean MRR/Hits@1
Label Jaccard	0.712	0.774	0.727	0.894	0.756	0.773/0.720
Character 3-gram	0.710	0.841	0.716	0.932	0.806	0.801/0.741
BM25 label	0.727	0.787	0.728	0.897	0.757	0.779/0.729
Ancestor context only	0.048	0.096	0.107	0.040	0.037	0.066/0.018
ARB combined score	**0.724**	0.833	**0.763**	**0.942**	0.804	**0.813/0.760**

The official Bio-ML unsupervised leaderboard gives additional context for this local-ranking result (Table 17). ARB trails the mean MRR of BERTMapLt, Matcha, HybridOM, and BERTMap, which is expected because those systems are complete biomedical matchers or richer ranking pipelines. The SNOMED–NCIT Pharm task is the exception: ARB reaches MRR 0.942, above BERTMapLt and Matcha under the same MRR/Hits@1 reporting format. This comparison does not make ARB a Bio-ML full-system competitor. It places the symbolic scorer beside stronger biomedical systems while keeping the paper's main claim at the candidate-generation stage. Because BERTMap and Matcha were introduced after OAEI 2013 LargeBio, Table 12 uses LogMap and AML for same-benchmark LargeBio context, while Table 17 uses Bio-ML 2024 for later neural and modular systems.

Table 17. Bio-ML 2024 official unsupervised local-ranking context (MRR/Hits@1). ARB uses only symbolic ontology-internal evidence; the other systems are reported as benchmark context, not as blocking-only baselines.

Task	ARB	BERTMapLt	Matcha	HybridOM	BERTMap
OMIM–ORDO	0.724/0.670	0.766/0.716	0.815/0.782	0.849/0.792	0.880/0.830
NCIT–DOID	0.833/0.777	0.890/0.861	0.902/0.873	0.952/0.928	0.959/0.937
SNOMED–FMA (Body)	0.763/0.688	0.892/0.865	0.950/0.935	0.907/0.861	0.944/0.920
SNOMED–NCIT (Pharm)	0.942/0.922	0.849/0.773	0.936/0.921	0.964/0.936	0.969/0.951
SNOMED–NCIT (Neoplas)	0.804/0.741	0.891/0.859	0.889/0.936	0.911/0.870	0.954/0.928
Mean	0.813/0.760	0.858/0.815	0.898/0.889	0.917/0.877	0.941/0.913

7. Error Analysis and Engineering Discussion

Blocking errors are important in biomedical AI because they determine which anatomy, disease, phenotype, and clinical terminology concepts are allowed to enter later matching stages. A missed candidate removes a potentially valid biomedical correspondence before scoring begins, whereas an added competitor may redirect a one-to-one decision toward a medically different concept. To make these effects inspectable, we categorise errors post hoc, outside the ARB inference path, into four types: *blocking loss*, where a reference pair never enters C; *pruning loss*, where a pair enters C but is removed by I2; *scoring loss*, where a pair remains in C but falls below `accept`; and *decision loss*, where a pair scores above `accept` but loses to a competitor during one-to-one assignment. Table 18 summarises the two aggregate negative cases before the URI-level analyses. The following analysis combines this evidence with Tables 4 and 6, and diagnostics exported by the ARB pipeline.

Table 18. Aggregate evidence for the two negative cases discussed in Section 7. HP–MP is evaluated against a partial reference, so "non-reference" means not present in the known 696-pair reference.

Task	Observed Counts	Interpretation
FMA–SNOMED-small	I2 removes 512 candidates in the own-only setting, including 59 reference pairs; after rescue it removes 569 candidates, including 66 reference pairs.	Gold-pair loss among removed candidates is 11.5%; the corresponding rates are 0.3% for Anatomy and 2.1% for FMA–NCI-small. I2 is therefore risk-controlled, but not lossless, on this task.
HP–MP	The token base has PC 0.963. Gated rescue raises PC to 0.991 and expands the pool from 192,439 to 215,969 pairs, adding 23,530 candidates. Against the 696 known-reference pairs, this corresponds to about 20 additional known-reference pairs and about 23.5k additional candidates not present in the known reference.	The extra candidates create new competitors in the same Hungarian conflict components. Precision falls enough that final F_1 declines from 0.605 to 0.595, despite the higher PC.

Blocking loss occurs when no symbolic path places a reference pair in the candidate pool. One confirmed URI-level example discussed below pairs the FMA concept 'Flocculonodular lobe" with the SNOMED concept 'Archicerebellar structure". The pair never enters C, and its recorded structural overlap is 0.0. The two labels refer to the same anatomical region at different descriptive levels, but neither token indexing nor the depth-limited structural channels are able to connect them. In a biomedical integration pipeline, this type of loss is severe because the correspondence is unavailable to every downstream matcher, regardless of how strong that matcher may be. It also marks a natural boundary of pure symbolic blocking: when lexical overlap and local graph neighbourhood evidence both vanish, rescue requires either a deeper graph strategy or an external synonym source, both of which fall outside the current no-external-resource setting.

Pruning loss is most visible on FMA–SNOMED-small. I2 removes 59 reference pairs from the own-only candidate set, corresponding to 0.7% of the 8941-pair reference alignment; after rescue, the diagnostic audit records 66 removed reference pairs. This loss is much larger than on Anatomy, where only 4 known-gold pairs are removed, and FMA–NCI-small, where 8 are removed. The post-hoc category analysis (Table 19) identifies 34 child-sparse cases and 18 laterality-related cases among the 66 post-rescue removed reference pairs. Table 20 then makes the recall cost explicit by reporting candidate recall before and after pruning. Across the full-configuration rows, I2 changes PC by -0.0026 on Anatomy, -0.0027 on FMA–NCI-small, -0.0010 on FMA–NCI-full, -0.0018 on SNOMED–NCI-small, and 0.0000 on DOID–ORDO and HP–MP. The largest loss is FMA–SNOMED-small, where PC falls from 0.7940 to 0.7866 and 66 gold pairs are removed. The composition of the removed set is especially revealing: on FMA–SNOMED-small, 59 of 512 removed pairs are reference pairs, or 11.5%, whereas the corresponding rate is only 0.3% on Anatomy. This pattern is consistent with structural asymmetry rather than uniform pruning noise. FMA is a fine-grained anatomical model, whereas SNOMED CT is organised primarily for clinical coding and may use body-structure wrappers, laterality branches, and different depths. Consequently, reference-equivalent body-part concepts can have non-comparable parent and child neighbourhoods. In such cases, an FMA leaf and a SNOMED "Structure of X" wrapper may be reference-equivalent but still have weak local child overlap. The dominance rule and `lex_slack` = 0.10 reduce this risk, but they cannot fully compensate for differences in hierarchy depth and modelling purpose.

The same pruning mechanism can also produce useful biomedical corrections. For the FMA concept 'Non-articular part of tubercle of rib", the retained gold target 'Structure of non-articular part of tubercle of rib" has

structural support 0.938, whereas numbered-rib siblings remain at 0.250. In this case, the rule removes lexically close but anatomically over-specific siblings and preserves the generic rib-part correspondence. This contrast explains the role of I2 more precisely: it is an optional, precision-oriented cleaning step whose benefit depends on whether local structural evidence is aligned across the two ontologies; it should not be treated as a default recall-preserving operation.

Table 19. Post-hoc classification of the 66 FMA-SNOMED-small reference pairs removed by I2 after rescue.

Category	Count	Share
Child-sparse hierarchy	34	51.5%
Laterality mismatch	18	27.3%
Other	9	13.7%
Both sparse	3	4.5%
Subtype/grade	2	3.0%

Table 20 gives the per-task Otsu pruning audit. In the own-generation rows, the score-qualified set ranges from 3324 to 11,784 candidates; after rescue, it ranges from 3359 to 13,435 candidates. Anatomy shows the intended behaviour: I2 removes 1388 candidates after rescue, only 4 of which are known-gold pairs, reduces PC only from 0.8924 to 0.8898, and raises final F_1 by 0.166. This is the clearest case where the precision-oriented pruning benefit is obtained at a small recall cost. FMA–SNOMED-small is the risk case: after rescue, I2 removes 569 candidates, including 66 known-gold pairs, reduces PC from 0.7940 to 0.7866, and final F_1 changes only from 0.5486 to 0.5483. This trade-off is not favourable enough to treat pruning as automatically acceptable on structurally heterogeneous ontology pairs. These results indicate that structural dominance pruning is auditable and precision-oriented, but it should not be described as recall-safe across biomedical ontologies.

Table 20. Otsu structural-pruning diagnostics under the token-base pipeline, including candidate recall before and after pruning. $|Q|$ is the number of score-qualified candidates with `score >= accept` before pruning. ΔPC and ΔF_1 compare each I2 row against the corresponding unpruned generation setting, and "gold removed" counts reference pairs deleted by pruning.

Task	Generation	θ^*	—Q—	Removed	Gold Removed	PC Before	PC After	ΔPC	PQ After	ΔF_1
Anatomy	own	0.4004	5047	1374	4	0.8370	0.8344	−0.0026	0.01435	+0.1689
Anatomy	own + I1	0.4004	5277	1388	4	0.8924	0.8898	−0.0026	0.01452	+0.1662
FMA-NCI-small	own	0.4434	3324	374	8	0.7502	0.7475	−0.0027	0.02986	+0.0046
FMA-NCI-small	own + I1	0.4434	3359	376	8	0.7997	0.7970	−0.0027	0.02858	+0.0050
FMA-NCI-full	own	0.2988	3469	367	5	0.3306	0.3289	−0.0017	0.00642	+0.0058
FMA-NCI-full	own + I1	0.3652	4454	330	3	0.4394	0.4384	−0.0010	0.00688	+0.0066
FMA-SNOMED-small	own	0.3770	6356	512	59	0.7097	0.7031	−0.0066	0.02437	+0.0003
FMA-SNOMED-small	own + I1	0.3848	7233	569	66	0.7940	0.7866	−0.0074	0.02469	−0.0003
SNOMED-NCI-small	own	0.2988	11,784	379	31	0.5475	0.5458	−0.0017	0.03135	+0.0005
SNOMED-NCI-small	own + I1	0.2988	13,435	401	33	0.6291	0.6273	−0.0018	0.03091	+0.0009
DOID-ORDO	own	0.1777	8365	1613	0	0.9709	0.9709	0.0000	0.00668	+0.0110
DOID-ORDO	own + I1	0.1895	8917	1660	0	0.9911	0.9911	0.0000	0.00630	+0.0147
HP-MP †	own	0.4277	3527	440	0	0.9626	0.9626	0.0000	0.00349	+0.0077
HP-MP †	own + I1	0.4395	3881	491	0	0.9914	0.9914	0.0000	0.00320	+0.0064

† HP-MP is evaluated against a partial known-reference set.

HP–MP illustrates a different failure regime. The token base already covers 0.963 of the known partial reference, leaving little recall headroom for rescue. Gated rescue raises candidate recall to 0.991, but candidate purity drops from 0.00348 to 0.00319. In absolute terms, the pool grows by 23,530 pairs; only about 20 of these are known-reference pairs, while roughly 23.5k are not present in the known reference. Because HP–MP is a partial-reference task, these added candidates should not be treated as confirmed false positives. Nevertheless, they can still affect the fixed one-to-one assignment used for downstream validation. The Hungarian matcher operates inside connected conflict components, so additional phenotype neighbours can reallocate targets across multiple sources. The confirmed cases in Table 21 show that a source concept may lose its known gold target even when the gold target has a higher local score than the finally selected non-gold sibling. Thus, the final decline from 0.605 to 0.595 is best explained by the interaction between a recall-saturated candidate pool and the simple one-to-one matcher, not by candidate recall alone.

Table 21. Representative HP-MP decision losses. In these cases the gold target has a higher pair score for the listed source, but global one-to-one assignment reallocates the target within the conflict component.

Source (HP)	Gold Target Score	Winning Non-Gold Target	Winner Score
Abnormal sperm motility	0.925	abnormal activated sperm motility	0.629
Skeletal muscle atrophy	0.900	skeletal muscle fiber atrophy	0.669
Abnormal iris pigmentation	0.891	abnormal iris stromal pigmentation	0.654
Abnormal tongue morphology	0.850	abnormal tongue muscle morphology	0.673

Scoring loss occurs when a reference pair is present in C but its score falls below `accept` = 0.45. The clearest examples are abbreviation–expansion pairs. In the confirmed FMA–SNOMED case 'RNA" versus 'Ribonucleic acid", the structural diagnostic is 1.0, but the recorded score is 0.000 because the abbreviation and the expanded form have no token-level overlap under the current normalisation. A second example, 'GAG" versus 'Mucopolysaccharide", has the same abbreviation-expansion pattern and receives a score of 0.010. This is a biomedical vocabulary issue rather than a generic string-matching issue: curated ontologies often mix abbreviations, expanded chemical names, Latinised terms, and clinical shorthand. A lightweight abbreviation-expansion resource would help this category, but it would also change the no-external-resource condition evaluated in this study.

These loss cases matter for medical AI knowledge integration. A cross-vocabulary oncology mapping between a clinical phrase such as 'malignant neoplasm of breast" and an ontology label such as 'breast carcinoma" can determine whether cancer concepts from clinical coding, knowledge graphs, and cohort-retrieval queries are treated as the same disease entity. If such a pair is missed, breast-cancer evidence may fragment across resources. If a near sibling is retained or selected instead, downstream retrieval may mix tumour site, histology, laterality, or disease-stage distinctions. This is why candidate provenance is relevant to governed medical knowledge-integration workflows, although no clinical diagnostic system is evaluated here. For this reason, the ARB audit table records not only whether a candidate is correct under the reference, but also which channel generated it, which structural evidence supported it, and whether pruning or assignment changed the candidate's status.

Table 22 reports seven confirmed URI-level cases covering successful rescue, useful pruning, blocking loss, pruning loss, scoring loss, and decision loss. Table 23 separates the biomedical interpretation from the numerical diagnostics. The pruning examples show how local structural evidence can both help and harm. For 'Intermediate supraclavicular nerve", the gold pair has score 0.51, parent overlap 0.143, and child overlap 0.0, whereas the competitor 'Medial supraclavicular nerve" has structural support 0.67. A second laterality-sensitive case involves 'Lateral talocalcaneal ligament", where the gold target scores 0.573 but is removed because a structurally denser competitor reaches support 0.50 with a near-identical lexical score. These failures are medically interpretable: the labels remain close, but laterality and hierarchy design do not align well enough for the dominance rule. The HP–MP decision case with 'Abnormal sperm motility" shows a different mechanism. The source's known gold target has score 0.925, but the final assignment selects the finer-grained sibling "abnormal activated sperm motility" with score 0.629 because of conflict-component interactions. These examples do not replace aggregate evaluation; they explain how the aggregate losses arise inside the pipeline.

Two engineering implications follow from this analysis. First, context rescue should be used with attention to recall headroom. The current purity-gate threshold `rescue_floor` = 0.35 and rescue budget $k = 3$ work better for recall-bottleneck tasks such as Anatomy and FMA–NCI than for the recall-saturated HP–MP phenotype task. Before enabling I1 in a deployment, users should estimate whether additional candidates mainly recover missing correspondences or simply enlarge already dense conflict components. This can be done with a small held-aside probe set, task-specific diagnostics, or candidate-pool statistics computed before downstream assignment.

Second, `accept` remains a stronger calibration risk than the rescue budget. Table 9 shows a task split across the threshold sweep: raising `accept` to 0.55 reduces FMA–SNOMED-small F_1 from 0.5483 to 0.3426, while DOID–ORDO and HP–MP improve at higher thresholds. This pattern is consistent with biomedical ontology heterogeneity. Anatomy-to-clinical-coding mappings may need a lower lexical acceptance threshold because SNOMED CT and FMA organise body structures differently, whereas disease or phenotype tasks may benefit from stricter candidate acceptance. The current fixed threshold should therefore be viewed as a development-fixed operating point, not as a universally calibrated threshold for biomedical ontology matching. The same task-specific caution applies to pruning: I2 can be disabled or guarded when diagnostics show sparse hierarchy overlap or high known-reference loss.

Table 22. Confirmed URI-level candidate cases from ARB diagnostics: numerical evidence. Scores are the pipeline lexical scores; parent overlap and child overlap are structural diagnostics where available. "Gold" means present in the benchmark reference alignment.

Case	Source and Target/Competitor Labels	Channel/Status	Score	Structural Diagnostic	Decision/Reference
Successful rescue	Source: Caudal articular process of thoracic vertebra. Target: Structure of inferior articular process of thoracic vertebra.	neighbour	0.454	parent overlap 0.429; child overlap 0.800	rescued and retained; gold
Useful pruning	Source: Non-articular part of tubercle of rib. Gold target: Structure of non-articular part of tubercle of rib. Sibling: Structure of non-articular part of seventh rib.	token	0.587 vs 0.493	structural support 0.938 vs 0.250	gold retained; sibling pruned
Blocking loss	Source: Flocculonodular lobe. Target: Archicerebellar structure.	not generated	n/a	parent overlap 0.000; child overlap 0.000	absent from C; gold
Pruning loss	Source: Lateral talocalcaneal ligament. Gold target: Structure of lateral talocalcaneal ligament. Competitor: Talocalcaneal ligament.	token	0.573 vs 0.570	parent overlap 0.333; child overlap 0.000; competitor support 0.500	pruned; gold
Pruning loss	Source: Intermediate supraclavicular nerve. Gold target: Intermediate supraclavicular nerve. Competitor: Medial supraclavicular nerve.	token	0.510 vs n/a	parent overlap 0.143; child overlap 0.000; competitor support 0.670	pruned; gold
Scoring loss	Source: RNA. Target: Ribonucleic acid.	context candidate	0.000	structural support 1.000	below `accept`; gold
Decision loss	Source: Abnormal sperm motility. Gold target: abnormal sperm motility. Winner: abnormal activated sperm motility.	token/context	0.925 vs 0.629	not applicable	gold loses in Hungarian assignment; gold

Table 23. Confirmed URI-level candidate cases from ARB diagnostics: interpretation. The table separates biomedical interpretation from the numerical diagnostics in Table 22.

Case	Biomedical Interpretation
Successful rescue	Cross-vocabulary anatomical wording is recovered from shared thoracic-vertebra neighbourhood evidence.
Useful pruning	A numbered-rib sibling is removed, avoiding an anatomically over-specific link.
Blocking loss	The two labels refer to the same anatomical region at different descriptive levels, but neither lexical overlap nor depth-limited local structure connects them.
Pruning loss: lateral talocalcaneal ligament	Laterality and hierarchy-depth mismatch make a valid laterality-specific pair look structurally weaker than a more generic competitor.
Pruning loss: intermediate supraclavicular nerve	Closely related nerve labels create a same-source competition in which sparse local child overlap favours the medial sibling over the reference pair.
Scoring loss	Abbreviation–expansion variation is missed by the current lexical score despite strong structural support.
Decision loss	A finer-grained phenotype sibling can redirect one-to-one assignment inside a dense conflict component.

The Otsu threshold θ^* has a related assumption. It is most useful when the structural-overlap distribution contains a visible separation between weakly and strongly supported candidates. If the distribution is nearly unimodal, the split point becomes less reliable and I2 may either retain too many weak candidates or remove structurally sparse true pairs. This situation is likely when two ontologies encode the same biomedical domain for different operational purposes, as in FMA's anatomical modelling and SNOMED CT's clinical coding. Future versions should therefore include diagnostics for structural-overlap shape before applying structural dominance pruning.

8. Limitations

ARB is designed as an auditable candidate-generation and candidate-cleaning module for biomedical knowledge integration, not as a complete ontology matcher. It does not include a trained semantic scorer, logical repair, or external biomedical synonym resource. The low absolute final F_1 values on Anatomy (0.375) and FMA–NCI-full (0.395) should therefore be interpreted as the result of passing the candidate pool to a simple fixed Hungarian validator, rather than as full-system matching performance. The evidence reported in this study mainly supports

the value of ARB at the blocking stage, where candidate recall, candidate purity, pool size, provenance, runtime, and memory are directly relevant to audit-sensitive medical AI interoperability pipelines. These benchmark-based findings should not be interpreted as evidence of clinical safety, clinical utility, or readiness for hospital deployment.

A first limitation concerns recall-saturated and partial-reference tasks. On HP–MP, the token base already reaches a candidate recall of 0.963. Gated rescue further raises this value to 0.991, but it also adds 23,530 candidate pairs. Against the 696-pair partial reference, this means that only about 20 additional known-reference pairs are recovered, while many additional candidates are not present in the known reference. The larger pool can create larger one-to-one conflict components, allowing the Hungarian validator to reassign targets away from known gold pairs. This result shows that higher blocking recall does not necessarily lead to better downstream assignments when the base candidate pool is already near-saturated, especially under partial-reference evaluation.

A second limitation is structural asymmetry between biomedical ontologies. Same-source structural dominance pruning is a precision-oriented filter, not a guaranteed recall-preserving operation. On FMA–SNOMED-small, it removes 59 reference pairs, corresponding to 0.7% of the gold alignment; after rescue, the diagnostic audit records 66 removed reference pairs. Many of these losses involve child-sparse or laterality-related cases, where an FMA leaf concept and a SNOMED body-structure wrapper have weak local child overlap despite being reference-equivalent. The dominance rule and `lex_slack` reduce this risk, but they do not remove the underlying assumption that equivalent concepts usually share some local structural support. This limitation reflects ontology design as well as algorithm design, because FMA and SNOMED CT organise anatomical knowledge for different modelling purposes.

A third limitation is the coverage of the symbolic score. ARB relies on ontology-internal labels, ancestors, neighbours, and structural evidence. It therefore has limited ability to handle abbreviation-expansion pairs when no shared token or usable context is available. Pairs such as RNA–Ribonucleic acid require abbreviation knowledge that may not be encoded in the input labels or hierarchy. Adding such knowledge would likely improve coverage, but it would also change the no-external-resource setting evaluated in this study. Similar difficulties may appear in phenotype, disease, and laboratory vocabularies, where abbreviations and shortened clinical expressions are common.

The acceptance threshold is another source of calibration risk. A single development-fixed value cannot fully accommodate the lexical and structural differences among anatomy, disease, oncology, and phenotype ontologies. The sensitivity analysis indicates that this threshold can affect different tasks in different directions, so future versions should consider task-adaptive calibration. Possible directions include using unlabelled score distributions, small institutional probe sets, explicit operating-point selection, or pre-validation grid search before deployment in a biomedical integration pipeline. Such calibration should remain separate from test reference alignments if the reference-independent setting is to be preserved. Crucially, this revision does not provide an independent leave-one-task-out calibration experiment or an automatic calibration mechanism; the present parameter settings should therefore be interpreted as fixed settings for the reported benchmark configurations rather than as automatically calibrated settings for unseen ontology pairs.

Finally, the current evaluation is bounded by data format, language, ontology version, and assignment strategy. All reported experiments use OWL/RDF resources with English labels from the selected OAEI 2013 and Bio-ML 2024 settings. The OAEI 2013 tasks are used because their stable reference alignments support controlled blocking-stage evaluation; this does not test current-release ontology pairs or clinical deployment workflows. The system has not yet been tested on multilingual labels, alternative ontology serialisation formats, newer releases of the same ontology pairs, or local institutional terminology extensions. The downstream Hungarian step also has cubic cost in each connected conflict component of size c, i.e., $O(c^3)$. Components exceeding 250k cells fall back to deterministic greedy one-to-one matching. This fallback was not triggered in the reported tasks, but it could affect results on denser ontologies or on candidate pools generated under less conservative settings.

9. Conclusions

This paper presented Auditable Rule-based Blocking (ARB), a symbolic candidate-generation module for biomedical ontology matching in medical AI systems. ARB combines token indexing, purity-gated context rescue, and same-source structural dominance pruning to produce a candidate pool that can be traced back to ontology-internal evidence. Rather than replacing complete ontology matchers, ARB is designed to support the blocking stage in settings where external biomedical resources, learned embeddings, or reference-alignment calibration are unavailable or undesirable. Accordingly, the contribution of this paper is auditable blocking, candidate provenance, and pruning-risk diagnosis, not a complete ontology matching system. Across the evaluated OAEI biomedical tasks, ARB improved candidate recall by 2.0–10.8 percentage points while limiting gated pool growth to 5–24%. These results indicate that labels, ancestor context, neighbour labels, and conservative structural pruning can improve

candidate availability without relying on external resources or learned semantic models. At the same time, the results clarify the boundaries of the current design. Context rescue is less useful when the base candidate pool is already recall-saturated, especially under partial-reference evaluation. Structural pruning may also become less reliable when two biomedical ontologies organise similar concepts under different hierarchy patterns. In addition, abbreviation-expansion pairs and other weakly lexical matches remain difficult for the current symbolic scorer, and the Hungarian matcher used in this study serves only as a fixed downstream validator.

Future work will connect the same auditable candidate pool with stronger symbolic scoring and assignment methods while preserving the no-external-resource setting. Further development should also improve abbreviation handling, introduce hierarchy-aware safeguards for structurally asymmetric ontologies, and evaluate the module in downstream biomedical knowledge-integration tasks such as knowledge graph construction, semantic search, and terminology normalisation. Independent evaluation on unseen ontology pairs, automatic calibration from unlabelled score distributions, and pre-validation grid search are important future directions rather than claims established by the present benchmark study. These extensions would help clarify how auditable candidate generation can contribute to practical biomedical knowledge integration beyond benchmark-level matching.

Author Contributions

Conceptualization, X.X., J.W.; methodology, R.X.; software, R.X.; validation, X.X.; writing—original draft preparation, R.X., H.L.; writing—review and editing, J.W.; supervision, J.W. All authors have read and agreed to the published version of the manuscript.

Funding

This research received no external funding.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

The benchmark datasets are publicly available from the OAEI and Bio-ML releases cited in the manuscript. The revision package includes the archived aggregate outputs used for the bootstrap, pruning-diagnostic, and threshold-sensitivity analyses. The code repository containing the implementation, configuration defaults, experiment drivers, archived result files, and run commands is available at https://github.com/Sharpans/arb-candidate-generation under tag `v1.0.0-anonymous-review` and commit `3c54faefe2317569ce72e89540644350bc17520e`. At the time of proof correction, the repository remains private; access can be granted to the editor on request and the repository will be made public upon publication.

Conflicts of Interest

The authors declare no conflict of interest.

Use of AI and AI-Assisted Technologies

AI tools were used only for language editing, manuscript-structure checking, and formatting assistance. The study design, implementation, numerical results, medical-risk interpretation, and final manuscript decisions were reviewed and approved by the authors.

Appendix A. Reproducibility Details

Appendix A.1. Feature Construction

Each concept's own-label token vector $x(v)$ is built from all `rdfs:label` and synonym annotations attached to the concept. Tokens are lowercased, split on non-alphanumeric boundaries, and weighted by inverse document frequency computed over the union of both ontologies' label vocabularies. Empty-label concepts (concepts with no `rdfs:label` or synonym-like annotation) do not generate own-label token candidates, but they remain in the parsed graph and may supply context to labelled neighbours. Isolated nodes (no parent and no child) receive an empty ancestor pool and participate only via the own-label base.

Appendix A.2. SimHash and LSH Configuration

SimHash codes are computed by projecting $x(v)$ onto $K = 128$ random hyperplanes; each hyperplane's sign gives one bit. The hyperplanes are generated deterministically by hashing the token identity with BLAKE2b, so no random seed is required and the projection is identical across machines. LSH banding splits the 128-bit code into $b = 12$ bands; concepts sharing a band signature become hierarchy-channel candidates. Buckets larger than `max_bucket` = 60 concepts are skipped to avoid quadratic blow-up within a single oversized bucket.

Appendix A.3. Otsu Threshold Computation

For each task, the structural overlap values of score-qualified candidates are histogrammed into 256 bins over $[0, 1]$. Otsu's method scans all 255 possible split points and selects the one minimising the intra-class variance. The resulting θ^* is used for the dominance rule in Algorithm 2.

Appendix A.4. Parameter Provenance

All development-fixed parameters in Table 2 were fixed before the reported evaluation tables were finalised and were not adjusted during the formal reported runs. Here, "development sweep" refers to the development-stage parameter check over the acceptance threshold `accept`. It did not use disjoint held-aside ontology pairs; it used the same benchmark configuration families later reported in the paper to inspect parameter behaviour before the final protocol was frozen. These benchmark families therefore contributed to parameter development, even though reference alignments were not used by the inference pipeline. Table A1 reports the development-sweep results for the local interval around the default setting. The previously cited 0.085 value is the default-relative decrease on FMA–SNOMED-small when `accept` is raised from 0.45 to 0.50. This is distinct from Table 9, which reports the wider 0.35–0.55 benchmark sensitivity range. Engineering-prior parameters were set for deterministic memory and runtime control before benchmark scoring. The acceptance threshold `accept` = 0.45 is the single most sensitive parameter in the reported sensitivity table, so the manuscript treats threshold choice as a limitation rather than as evidence of automatic calibration or transfer to unseen ontology pairs.

Appendix A.5. Bootstrap Analysis

For the principal token-base comparisons, uncertainty intervals in Table 5 use 1000 source-level nonparametric bootstrap resamples and a fixed seed of 13. Each configuration is recomputed against the same resampled reference units, and the reported configuration differences use paired resampling. This procedure assesses benchmark-sampling uncertainty only; it does not turn the deterministic pipeline into a repeated-run experiment and does not establish automatic calibration for unseen ontology pairs.

Table A1. Development-sweep results for the acceptance threshold `accept`. The table reports the full values underlying the previously cited 0.085 movement: relative to the default 0.45, FMA–SNOMED-small changes by −0.085 at 0.50. These values document parameter behaviour before the reported tables were finalised; they are not results from disjoint held-aside ontology pairs and are not a task-specific retuning of the reported test results.

`accept`	Anatomy	FMA–NCI	FMA–SNOMED	SNOMED–NCI	DOID–ORDO	HP–MP †	Max Adverse ΔF_1
0.40	0.3678	0.6581	0.5823	0.5321	0.6525	0.5133	−0.081
0.45	0.3748	0.6565	0.5483	0.5308	0.7199	0.5947	0.000
0.50	0.3867	0.6463	0.4633	0.5143	0.7720	0.6888	−0.085

† HP–MP uses a partial reference; final F_1 is a known-reference estimate.

References

1. Abdaoui, H.; Barki, C.; Dergaa, I.; et al. Accurate Clinical Entity Recognition and Code Mapping of Anatomopathological Reports Using BioClinicalBERT Enhanced by Retrieval-Augmented Generation: A Hybrid Deep Learning Approach. *Bioengineering* **2026**, *13*, 30. https://doi.org/10.3390/bioengineering13010030.
2. Jiménez-Ruiz, E.; Cuenca Grau, B. LogMap: Logic-Based and Scalable Ontology Matching. In Proceedings of the 10th International Semantic Web Conference (ISWC), Bonn, Germany, 23–27 October 2011; pp. 273–288.
3. Faria, D.; Pesquita, C.; Santos, E.; et al. The AgreementMakerLight Ontology Matching System. In Proceedings of the OTM Confederated International Conferences, Graz, Austria, 9–13 September 2013; pp. 527–541.
4. He, Y.; Chen, J.; Antonyrajah, D.; et al. BERTMap: A BERT-Based Ontology Alignment System. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual Event, 22 February–1 March 2022; Vol. 36, pp. 5684–5691.
5. Faria, D.; Silva, M.C.; Cotovio, P.; et al. Results for Matcha and Matcha-DL in OAEI 2023. In Proceedings of the 18th

International Workshop on Ontology Matching Co-Located with the 22nd International Semantic Web Conference (ISWC 2023), CEUR Workshop Proceedings, Athens, Greece, 7 November 2023; Vol. 3591, pp. 164–169.

6. Papadakis, G.; Ioannou, E.; Niederée, C.; et al. Efficient Entity Resolution for Large Heterogeneous Information Spaces. In Proceedings of the 4th ACM International Conference on Web Search and Data Mining (WSDM), Hong Kong, China, 9–12 February 2011; pp. 535–544.
7. Papadakis, G.; Papastefanatos, G.; Palpanas, T.; et al. Scaling Entity Resolution to Large, Heterogeneous Data with Enhanced Meta-Blocking. In Proceedings of the 19th International Conference on Extending Database Technology (EDBT), Bordeaux, France, 15–18 March 2016; pp. 221–232.
8. Cuenca Grau, B.; Dragisic, Z.; Eckert, K.; et al. Results of the Ontology Alignment Evaluation Initiative 2013. In Proceedings of the 8th International Workshop on Ontology Matching Co-Located with the 12th International Semantic Web Conference (ISWC 2013), CEUR Workshop Proceedings, Sydney, Australia, 21 October 2013; Vol. 1111, pp. 61–100.
9. Ontology Alignment Evaluation Initiative. Results of the Ontology Alignment Evaluation Initiative 2022. Available online: https://oaei.ontologymatching.org/2022/results/ (accessed on 21 June 2026).
10. Ontology Alignment Evaluation Initiative. Results of the Ontology Alignment Evaluation Initiative 2023. Available online: https://oaei.ontologymatching.org/2023/results/ (accessed on 21 June 2026).
11. Ontology Alignment Evaluation Initiative. Results of the Ontology Alignment Evaluation Initiative 2024. Available online: https://oaei.ontologymatching.org/2024/results/ (accessed on 21 June 2026).
12. He, Y.; Chen, J.; Dong, H.; et al. Machine Learning-Friendly Biomedical Datasets for Equivalence and Subsumption Ontology Matching. In Proceedings of the 21st International Semantic Web Conference (ISWC), Hangzhou, China, 23–27 October 2022; pp. 575–591.
13. Kebsi, D.; Barki, C.; Dergaa, I.; et al. Personalized Hearing Loss Care Using SNOMED CT-Aligned Ontology and Random Forest Machine Learning: A Hybrid Decision-Support Framework. *Audiol. Res.* **2026**, *16*, 37. https://doi.org/10.3390/audiolres16020037.
14. Teymurova, S.; Jiménez-Ruiz, E.; Weyde, T.; et al. OWL2Vec4OA: Tailoring Knowledge Graph Embeddings for Ontology Alignment. In Proceedings of the International Knowledge Graphs and Semantic Web: 6th International Conference, KGSWC 2024, Lecture Notes in Computer Science, Paris, France, 11–13 December 2024; Vol. 15459, pp. 168–182.
15. Hertling, S.; Paulheim, H. OLaLa: Ontology Matching with Large Language Models. In Proceedings of the 12th Knowledge Capture Conference 2023, Pensacola, FL, USA, 5–7 December 2023; pp. 131–139.
16. Qiang, Z.; Wang, W.; Taylor, K. Agent-OM: Leveraging LLM Agents for Ontology Matching. In Proceedings of the VLDB Endowment, Guangzhou, China, 26–30 August 2024; Vol. 18, pp. 516–529.
17. Babaei Giglou, H.; D'Souza, J.; Engel, F.; et al. LLMs4OM: Matching Ontologies with Large Language Models. In Proceedings of the Semantic Web: ESWC 2024 Satellite Events, Lecture Notes in Computer Science, Crete, Greece, 26–30 May 2024; Vol. 15344, pp. 25–35.
18. Nguyen, L.; Barcelos, E.I.; French, R.H.; et al. KROMA: Ontology Matching with Knowledge Retrieval and Large Language Models. In Proceedings of the Semantic Web—ISWC 2025, Lecture Notes in Computer Science, Nara, Japan, 2–6 November 2025; Vol. 16140, pp. 629–649.
19. Song, Y.; Chen, J.; Schmidt, R.A. GenOM: Ontology Matching with Description Generation and Large Language Models. *World Wide Web* **2026**, *29*, 29. https://doi.org/10.1007/s11280-026-01413-y.
20. Qiang, Z.; Taylor, K.; Wang, W.; et al. OAEI-LLM: A Benchmark Dataset for Understanding Large Language Model Hallucinations in Ontology Matching. In Proceedings of the Special Session on Harmonising Generative AI and Semantic Web Technologies (HGAIS 2024), CEUR Workshop Proceedings, Baltimore, MD, USA, 13 November 2024; Vol. 3953.
21. Qiang, Z.; Taylor, K.; Wang, W.; et al. OAEI-LLM-T: A TBox Benchmark Dataset for Understanding Large Language Model Hallucinations in Ontology Matching. *arXiv* **2025**. arXiv:2503.21813.
22. Boussi Rahmouni, H.; Hassine, N.B.E.H.; Chouchen, M.; et al. Healthcare 5.0-Driven Clinical Intelligence: The Learn-Predict-Monitor-Detect-Correct Framework for Systematic Artificial Intelligence Integration in Critical Care. *Healthcare* **2025**, *13*, 2553. https://doi.org/10.3390/healthcare13202553.
23. Cheng, B.; Fürst, J.; Jacobs, T.; et al. Interactive Ontology Matching with Cost-Efficient Learning. *arXiv* **2024**. arXiv:2404.07663.
24. Hernández, M.A.; Stolfo, S.J. The Merge/Purge Problem for Large Databases. *ACM SIGMOD Rec.* **1995**, *24*, 127–138. https://doi.org/10.1145/568271.223807.
25. Papadakis, G.; Skoutas, D.; Thanos, E.; et al. Blocking and Filtering Techniques for Entity Resolution: A Survey. *ACM Comput. Surv.* **2021**, *53*, 31:1–31:42. https://doi.org/10.1145/3377455.
26. Zhang, W.; Wei, H.; Sisman, B.; et al. AutoBlock: A Hands-off Blocking Framework for Entity Matching. In Proceedings of the 13th ACM International Conference on Web Search and Data Mining (WSDM), Houston, TX, USA, 3–7 February 2020; pp. 744–752.
27. Brinkmann, A.; Shraga, R.; Bizer, C. SC-Block: Supervised Contrastive Blocking within Entity Resolution Pipelines. In Proceedings of the Semantic Web: 21st International Conference, ESWC 2024, Hersonissos, Greece, 26–30 May 2024; pp. 121–142.
28. Wang, T.; Lin, H.; Han, X.; et al. Towards Universal Dense Blocking for Entity Resolution. *arXiv* **2024**. arXiv:2404.14831.

29. Zhu, X.; Xie, M.; Deng, T.; et al. HyperBlocker: Accelerating Rule-Based Blocking in Entity Resolution Using GPUs. In Proceedings of the VLDB Endowment, Guangzhou, China, 26–30 August 2024; Vol. 18, pp. 308–321. https://doi.org/10.14778/3705829.3705847.
30. Wang, R.; Li, Y.; Wang, J. Sudowoodo: Contrastive Self-supervised Learning for Multi-purpose Data Integration and Preparation. In Proceedings of the 39th IEEE International Conference on Data Engineering (ICDE), Anaheim, CA, USA, 3–7 April 2023; pp. 1502–1515.
31. Melnik, S.; Garcia-Molina, H.; Rahm, E. Similarity Flooding: A Versatile Graph Matching Algorithm and Its Application to Schema Matching. In Proceedings of the 18th International Conference on Data Engineering (ICDE), San Jose, CA, USA, 26 February–1 March 2002; pp. 117–128.
32. Efeoglu, S. GraphMatcher: A Graph Representation Learning Approach for Ontology Matching. In Proceedings of the 17th International Workshop on Ontology Matching, CEUR Workshop Proceedings, Hangzhou, China, 23 October 2022; Vol. 3324, pp. 174–180.
33. Charikar, M.S. Similarity Estimation Techniques from Rounding Algorithms. In Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC), Montréal, QC, Canada, 19–21 May 2002; pp. 380–388.
34. Otsu, N. A Threshold Selection Method from Gray-Level Histograms. *IEEE Trans. Syst. Man Cybern.* **1979**, *9*, 62–66. https://doi.org/10.1109/TSMC.1979.4310076.
35. Kuhn, H.W. The Hungarian Method for the Assignment Problem. *Nav. Res. Logist. Q.* **1955**, *2*, 83–97. https://doi.org/10.1002/nav.3800020109.