

Robust and Scalable Lexical Candidate Generation for Knowledge Graph Matching: A Cross-Domain Empirical Study

Savita Sawant

swntsvt@gmail.com

Rajiv Gandhi Institute of Information Technology <https://orcid.org/0009-0002-2873-229X>

Sushopti Gawade

Vidyalankar Institute of Technology <https://orcid.org/0000-0001-5630-0895>

Mukesh Shrimali

Pacific Academy of Higher Education and Research University <https://orcid.org/0009-0004-6462-0417>

Research Article

Keywords: Knowledge Graph Matching, Ontology Alignment, Candidate Generation, BM25, TF-IDF, Character N-Gram Retrieval, OAEI

Posted Date: October 20th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11391013/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: The authors declare no competing interests.

Robust and Scalable Lexical Candidate Generation for Knowledge Graph Matching: A Cross-Domain Empirical Study

Savita Kiran Sawant^{1,2*}, Sushopti Gawade³ and Mukesh Shrimali¹

^{1*}Department of Computer Engineering, Pacific Academy of Higher Education and Research University, Udaipur, 313001, Rajasthan, India.

²Department of Information Technology, MCT Rajiv Gandhi Institute of Technology, Mumbai, 400053, Maharashtra, India.

³Department of Information Technology, Vidyalankar Institute of Technology, Mumbai, 400037, Maharashtra, India.

*Corresponding author(s). E-mail(s): swntsvt@gmail.com;

Contributing authors: sushopti2207@gmail.com;
mukesh_shrimali@yahoo.com;

Abstract

Knowledge Graph (KG) and ontology matching systems operate under strict scalability constraints, requiring efficient candidate generation to reduce the search space for downstream matching algorithms. Despite its critical role, the behaviour of widely used lexical candidate generation methods under varying configurations and across domains remains insufficiently understood, particularly from a deployment perspective.

This paper presents a robust and scalable evaluation framework for lexical candidate generation in KG and ontology matching. Four methods—exact label matching, word-level TF-IDF, BM25, and character n-gram retrieval—are systematically analysed across 34 datasets spanning five OAEI benchmark tracks, covering heterogeneous domains including biomedical ontologies, fan-wiki Knowledge Graphs, and conference ontologies. A stability-aware configuration selection strategy is introduced to identify parameter settings that balance performance and cross-domain consistency.

Experimental results show that BM25 provides stable performance across configurations, while word-level TF-IDF exhibits significant sensitivity, with certain configurations leading to substantial degradation in recall. Character n-gram

retrieval emerges as a strong zero-configuration baseline, achieving competitive or superior performance across domains and entity types. The study further demonstrates that high recall can be maintained while reducing the search space by over 99% in large-scale instance matching tasks.

The findings provide actionable insights for the design of scalable KG matching pipelines, highlighting robust default choices and identifying failure modes in commonly used retrieval techniques. The proposed framework and empirical analysis contribute towards improving the reliability and efficiency of candidate generation in real-world knowledge integration systems.

Keywords: Knowledge Graph Matching, Ontology Alignment, Candidate Generation, BM25, TF-IDF, Character N-Gram Retrieval, OAEI

1 Introduction

Knowledge Graph (KG) and ontology matching are fundamental tasks in enabling interoperability across heterogeneous data sources. As independently constructed KGs increasingly describe overlapping domains, identifying correspondences between their entities becomes essential for data integration, knowledge fusion, and downstream applications.

In practice, this task must be performed under strict scalability constraints, as modern Knowledge Graphs may contain hundreds of thousands to millions of entities. Exhaustive pairwise comparison is therefore computationally infeasible, motivating the use of multi-stage matching pipelines in which candidate generation plays a central role.

This process can be understood intuitively as searching for a specific face in a large crowd: rather than examining every individual, one first identifies a shortlist of likely candidates based on coarse features before applying more detailed comparison. Similarly, candidate generation reduces the search space from all possible entity pairs to a manageable subset, enabling scalable and efficient matching. However, any true match excluded at this stage is permanently lost, making candidate generation a critical determinant of overall system correctness.

1.1 Knowledge Graph Matching as a Multi-Stage Pipeline

A Knowledge Graph is a structured representation of real-world entities and the relationships between them, organised as a graph of nodes (entities) and edges (relations) [1]. When two independently constructed KGs describe overlapping domains—for example, two encyclopaedic wikis both covering the Star Wars universe, or two biomedical databases both cataloguing human diseases—*KG matching* is the task of identifying corresponding entities across the two graphs. Correspondences may exist at the schema level, between classes (categories of entities) and predicates (types of relationships), and at the instance level, between individual entities. Discovering these correspondences enables data integration, knowledge fusion, and interoperability across heterogeneous data sources [1].

1.2 The Combinatorial Explosion Problem

To appreciate why candidate generation matters, consider the scale of the problem. Given a source KG with $|\mathcal{S}|$ entities and a target KG with $|\mathcal{T}|$ entities, naïve exhaustive comparison requires evaluating $|\mathcal{S}| \times |\mathcal{T}|$ pairs. For the real-world KGs used in this study, target instance pools reach up to 532,781 entities. Even restricting attention to schema elements, a source ontology with several hundred classes must be compared against a target with a similar number, yielding tens of thousands of candidate pairs before any filtering.

Candidate generation addresses this by reducing the effective search space from $|\mathcal{S}| \times |\mathcal{T}|$ to $|\mathcal{S}| \times k$, where k is the candidate list size, typically $k \leq 50$. For a large instance pool, a top-50 candidate list reduces the search space by over 99%. This reduction makes downstream neural or structural matching tractable at scale, and is one of the central practical contributions a candidate generation stage can offer.

1.3 Lexical Retrieval for Candidate Generation

Among the signals available for candidate generation, textual similarity over entity labels remains one of the strongest and most broadly applicable, particularly at schema level [1, 2]. Entity labels—the human-readable names assigned to classes, predicates, and instances in a KG—carry direct semantic content. A class labelled `StarshipClass` in one KG is a strong candidate match for a class labelled `Starship_Type` in another. Lexical retrieval methods exploit this signal by treating each entity’s label as a short document and ranking target entities by their textual similarity to each source query.

Three families of lexical retrieval methods are widely used in practice:

- **Exact and normalised label matching:** the simplest approach, checking whether labels match after normalisation (lowercasing, punctuation removal). Fast and precise, but brittle to any variation in naming convention.
- **Word-level TF-IDF and BM25:** term-frequency-based ranking that scores target labels by their overlap with the source query at the word token level, weighted by inverse document frequency. BM25 extends TF-IDF with principled term frequency saturation (k_1) and document length normalisation (b) [3]. Both are widely deployed in production information retrieval systems and through Lucene/Elasticsearch backends in matching tools [4, 5].
- **Character n -gram retrieval:** TF-IDF applied at the level of character subsequences rather than whole words. This approach is robust to morphological variation, compound terms, and minor spelling differences, making it well suited to the heterogeneous label conventions found across independently constructed KGs and ontologies.

Despite their widespread use in production systems and benchmarking tools, lexical candidate generation methods are typically deployed with default or heuristically chosen configurations, without systematic validation of their robustness across domains. This creates a critical gap from a systems perspective: poorly chosen configurations may lead to silent failure modes, where true matches are not retrieved during candidate generation and cannot be recovered by downstream components.

In large-scale Knowledge Graph matching pipelines, such failures are particularly problematic, as they directly impact end-to-end recall while remaining difficult to detect during system development. The absence of empirically grounded, deployment-oriented guidance for selecting and configuring lexical retrieval methods therefore represents a significant barrier to building reliable and scalable matching systems.

1.4 Research Questions

To address these challenges, this study investigates the robustness, sensitivity, and cross-domain behaviour of commonly used lexical candidate generation methods under realistic deployment conditions. We investigate four research questions:

RQ1 How do exact label matching, character n -gram retrieval, word-level TF-IDF, and BM25 compare as candidate generators across diverse KG and ontology matching benchmarks?

RQ2 How sensitive is BM25 candidate recall to the choice of hyperparameters k_1 and b , and what configuration is most robust across domains?

RQ3 How sensitive is TF-IDF candidate recall to its hyperparameters— n -gram range, minimum document frequency, and sublinear term frequency scaling—and are there configurations that pose a risk of failure in unsupervised pipelines?

RQ4 What deployment-oriented guidelines can be derived for selecting and configuring lexical candidate generation methods in scalable KG and ontology matching pipelines?

1.5 Contributions

This paper develops a robust and generalizable evaluation framework for lexical candidate generation in Knowledge Graph and ontology matching, with a focus on cross-domain stability, scalability, and deployment-oriented configuration selection.

- **A robust cross-domain evaluation framework.** We design and implement a reproducible evaluation framework for lexical candidate generation across 34 datasets spanning five OAEI benchmark tracks. The framework explicitly separates development and held-out evaluation, enabling reliable assessment of cross-domain generalisation and avoiding configuration overfitting.
- **A stability-aware configuration selection strategy.** We introduce a variance-penalised configuration selection approach that balances mean performance with cross-domain consistency. This strategy enables the identification of deployment-ready configurations that are robust across heterogeneous datasets, rather than optimised for a single benchmark.
- **Systematic characterisation of retrieval robustness and failure modes.** Through exhaustive hyperparameter analysis, we identify fundamental differences in robustness between lexical retrieval methods, demonstrating that BM25 exhibits stable behaviour across configurations while TF-IDF is highly sensitive and prone to catastrophic failure under certain settings.
- **Character n -gram retrieval as a strong zero-configuration baseline.** We establish character n -gram retrieval as a reliable and competitive baseline that

requires no hyperparameter tuning, consistently achieving strong performance across domains and entity types.

- **Multi-entity-type evaluation for Knowledge Graph matching.** We provide a systematic evaluation of lexical candidate generation across classes, predicates, and instances in Knowledge Graph datasets, quantifying both retrieval effectiveness and search-space reduction.
- **Practical guidelines for scalable KG matching pipelines.** We derive actionable recommendations for selecting and configuring candidate generation methods in real-world systems, highlighting robust default choices and identifying high-risk configurations that may lead to failure in unsupervised settings.

Novelty and positioning. Unlike prior work that evaluates candidate generation methods primarily as components within end-to-end matching systems, this study treats lexical candidate generation as a first-class object of investigation. The proposed framework introduces a stability-aware configuration selection strategy and provides a systematic characterisation of robustness and failure modes across domains and entity types. This perspective enables the identification of deployment-relevant properties that are not captured by conventional single-benchmark evaluations.

The remainder of this paper is organised as follows. Section 2 reviews related work on ontology and KG matching, candidate generation strategies, and the use of lexical retrieval in matching systems. Section 3 describes the retrieval methods, hyperparameter grids, evaluation metrics, and experimental design. Section 4 describes the datasets used across all three evaluation tiers. Section 5 presents the development analysis of hyperparameter sensitivity for BM25 and TF-IDF. Section 6 presents the held-out evaluation results across all entity types and domains. Section 7 discusses the findings and their implications for pipeline design. Section 8 concludes with directions for future work.

2 Background and Related Work

This section situates the present study within the broader literature on KG and ontology matching. We first introduce the matching task and the pipeline architecture that motivates candidate generation as an independent concern. We then review the specific strategies used for candidate generation, the systems that deploy them, and the alternatives to lexical retrieval that have emerged in recent years. We close by identifying the gap that this paper addresses.

2.1 Ontology and Knowledge Graph Matching

Ontology matching—also called ontology alignment—is the task of discovering correspondences between entities of independently constructed ontologies or Knowledge Graphs. A correspondence typically takes the form of an equivalence or subsumption relation between a source entity s and a target entity t , expressed as $(s, \equiv, t, c)$, where c is a confidence value [1]. Matching applies to entities at different levels of granularity: classes (categories), properties or predicates (relation types), and instances (individual entities). Most practical systems handle these types in sequence or in combination.

The problem is motivated by the fragmented nature of knowledge representation. Independent teams building ontologies or KGs for the same domain will make different modelling choices, use different vocabularies, and organise concepts at different levels of granularity. Without alignment, these independently constructed resources cannot interoperate. Applications in biomedical data integration, Linked Open Data, enterprise knowledge management, and semantic web services all depend on reliable cross-KG correspondences [1, 6].

The scale of the problem has grown substantially over the past decade. Early OAEI evaluations [7] dealt with ontologies of a few thousand classes, where even quadratic pairwise comparison was tractable. Today, the OAEI Knowledge Graph Track [2, 8] includes KGs with hundreds of thousands of instances, and biomedical resources such as SNOMED CT contain hundreds of thousands of concepts. AgreementMakerLight (AML) [9], one of the longest-running high-performing systems in OAEI history, explicitly highlights scalability as a core design constraint, replacing the all-against-all similarity matrix of its predecessor with hash-based lexical indices that achieve linear rather than quadratic time complexity. This architectural shift from exhaustive comparison to targeted candidate retrieval is now a standard design principle in the field.

2.2 The Multi-Stage Matching Pipeline

A recurring observation across KG and ontology matching research is that no single matching strategy is both fast enough and accurate enough to handle large-scale alignment end to end. This has led to the now-standard decomposition of the matching task into a pipeline [1]:

1. A **candidate generation** (or blocking) stage that efficiently retrieves, for each source entity, a short list of plausible target matches, reducing the $O(|\mathcal{S}| \times |\mathcal{T}|)$ comparison space to a tractable $O(|\mathcal{S}| \times k)$ with $k \ll |\mathcal{T}|$.
2. A **scoring** stage that applies more expensive similarity functions—structural, semantic, or learned—to the candidate pairs.
3. An optional **refinement** stage that enforces global consistency constraints, repairs logical conflicts, and filters low-confidence mappings [9, 10].

This pipeline structure means that the candidate generation stage imposes a hard ceiling on end-to-end recall: a true match excluded from the candidate list at stage one cannot be recovered later, no matter how powerful the downstream scorer. Maximising candidate recall at stage one is therefore not merely a performance optimisation—it is a correctness requirement for the overall system. Despite this, the candidate generation stage receives far less research attention than the scoring and refinement stages, and its hyperparameter choices are rarely reported or evaluated systematically.

The same pipeline structure appears in the related field of entity resolution and record linkage [11], where the candidate generation step is called *blocking*. Blocking methods in entity resolution range from standard blocking (grouping records by a single key) to sorted neighbourhood methods, token blocking, and attribute clustering, all sharing the same goal: reduce the candidate space while preserving true matches.

The KG and ontology matching communities have converged on similar strategies, typically implemented as inverted lexical indices over entity labels.

2.3 Lexical Candidate Generation in Practice

LogMap [10] was among the first scalable systems to make lexical indexing central to its architecture. LogMap constructs inverted lexical indices over class and property labels, using the intersection of these indices to compute initial anchor mappings—high-confidence lexical correspondences that seed an iterative matching and repair process. Its lightweight variant LogMapLt retains only the lexical string-matching component and has remained a competitive baseline in OAEI evaluations for over a decade, demonstrating the persistent value of simple lexical methods even as more complex approaches have emerged.

AgreementMakerLight [9] similarly uses a multi-tier lexical index as its primary matcher. Labels, synonyms, and name variants are indexed; candidate pairs are generated by looking up source labels in the target index and retrieving matches above a similarity threshold. Secondary matchers—word-level similarity, structural propagation, and background knowledge from sources such as UMLS or WordNet—are then applied only to the candidate pairs surfaced by the primary lexical stage, not to the full cross-product. This two-tier design, common across many production systems, illustrates exactly the role that fast candidate generation plays: it gates the more expensive work.

Several OAEI systems route lexical candidate generation through production information retrieval infrastructure. Systems based on Lucene or Elasticsearch implicitly use BM25 as their scoring function when ranking label matches, since this is the default retrieval model in both platforms [3]. HybridOM [4], for example, explicitly combines BM25-based keyword search with dense vector similarity to retrieve candidate concepts before applying LLM-based description augmentation and reasoning. OLaLa [5], which uses Large Language Models for ontology matching, employs a retrieval stage to surface candidate pairs before LLM-based verification, with the quality of this retrieval stage directly conditioning the recall of the overall system.

2.4 Dense and Hybrid Retrieval for Candidate Generation

Beyond sparse lexical methods, embedding-based dense retrieval has emerged as a candidate generation strategy in recent years. Dense retrieval encodes entities as vectors in a shared semantic space and retrieves candidates via approximate nearest neighbour (ANN) search, enabling semantic matching beyond exact or near-exact label overlap [12, 13].

BERTMap [14], introduced at AAAI 2022, uses sub-word inverted indices for candidate selection and a fine-tuned BERT classifier for scoring. Its lightweight variant BERTMapLt retains only the sub-word string matching component, making it effectively a character n -gram-based candidate generator. BERTMap demonstrated that combining a fast lexical candidate selection step with a neural scorer substantially outperforms either component alone, and the system became a reference baseline in the OAEI BioML track [15].

More recent systems have pushed towards fully neural candidate generation. LLMs4OM [16] uses Sentence-BERT encodings of ontology concepts to retrieve candidate pairs, then prompts a Large Language Model to verify each candidate. MILA [17] embeds source and target ontologies into vector knowledge bases using sentence encoders and retrieves candidates via similarity search, then applies a prioritised depth-first search strategy that escalates to an LLM only for borderline cases, achieving state-of-the-art results on the 2023 and 2024 OAEI BioML challenges. GenOM [18] takes this further by using an LLM to generate enriched semantic descriptions of each concept, embedding these into a vector space for candidate retrieval, and combining embedding-based candidates with exact matching for final alignment.

Despite the appeal of dense retrieval, evidence from the broader information retrieval literature suggests that lexical methods remain highly competitive for entity retrieval tasks, particularly for short labels and long-tail entities. Dense retrievers have been shown to perform worse than or only on par with strong lexical models such as BM25 on entity retrieval benchmarks [19], and require substantially more computational resources and domain-specific fine-tuning to deploy. For the KG matching setting, where entity labels are often short, noisy, and domain-specific, and where no task-specific training data may be available, lexical methods retain a practical advantage that motivates the systematic study presented here.

2.5 The OAEI Benchmark Ecosystem

The Ontology Alignment Evaluation Initiative (OAEI) provides the community’s primary shared benchmarking infrastructure, offering a growing set of tracks that cover different domains, entity types, and matching challenges [2, 7, 15]. The five OAEI tracks used in this study are described in detail in Section 4.

2.6 The Role of BM25 in Information Retrieval

BM25 (Best Match 25) is the dominant scoring function in classical information retrieval, underlying the default search behaviour of production systems including Lucene, Elasticsearch, and Solr [3]. It extends TF-IDF with two principled corrections: term frequency saturation, controlled by parameter k_1 , which limits the contribution of a single term appearing many times in a document; and document length normalisation, controlled by parameter b , which prevents longer documents from receiving unfairly high scores simply because they contain more tokens.

The scoring function for a query Q against a document D is:

$$\text{BM25}(D, Q) = \sum_{t \in Q} \text{IDF}(t) \cdot \frac{f(t, D) \cdot (k_1 + 1)}{f(t, D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{\text{avgdl}}\right)} \quad (1)$$

where $f(t, D)$ is the frequency of term t in document D , $|D|$ is document length, and avgdl is the average document length in the corpus. Setting $k_1 = 0$ reduces BM25 to a pure IDF model; setting $b = 0$ disables length normalisation. The standard default values recommended in the literature are $k_1 \in [1.2, 2.0]$ and $b = 0.75$ [3].

A reproducibility study by Kamphuis et al. [20] examined how BM25 scoring variants differ across IR toolkits and found meaningful differences in ranking behaviour, but concluded that the qualitative recommendations for k_1 and b are broadly consistent across implementations. This study adapts that empirical approach to the KG matching context, where the documents are entity labels rather than full-length passages, and the queries are source entity labels rather than natural language search queries.

In the KG matching setting, entity labels are typically very short: one to five words for schema elements, and similarly short for most instance labels. This is qualitatively different from the passage retrieval setting that BM25 was designed for, which raises the question of whether the standard parameter recommendations transfer to this domain or require adjustment.

2.7 Gap in the Literature and Positioning of This Work

Reviewing the landscape above, a consistent pattern emerges. Systems that rely on lexical candidate generation—whether through Lucene/Elasticsearch backends, inverted label indices, or sub-word indexing—do so with default or informally chosen parameters, and do not report sensitivity analyses of those choices. End-to-end evaluations report precision, recall, and F1 over final alignment sets, but do not isolate candidate generation as a stage or measure candidate recall as an independent diagnostic. As a consequence, practitioners building new matching pipelines receive no empirically grounded guidance on which lexical retrieval method or configuration to use as a first stage, and which configurations carry a risk of silent failure.

Character n -gram retrieval occupies a particularly underexplored position in this landscape. It is used implicitly in BERTMapLt [14] through sub-word indexing and is known to be effective for fuzzy label matching in entity resolution [11], but it has not been evaluated as a standalone candidate generation method on OAEI benchmarks, nor compared systematically to BM25 and word-level TF-IDF in this context.

To the best of our knowledge, prior work has not systematically evaluated lexical candidate generation across multiple OAEI tracks with controlled hyperparameter analysis across multiple OAEI tracks, and this paper addresses that gap directly. The contribution is not a new matching system but a characterisation of the retrieval layer that underlies many existing systems—a layer that is often taken for granted but, as the results of this study show, can be the difference between robust and fragile pipeline behaviour depending on configuration choices that are rarely examined.

3 Methods

3.1 Task Definition

We treat candidate generation as a ranking problem over labelled graph entities. Given a source entity set $\mathcal{S}$ and a target entity set $\mathcal{T}$, each source entity $s \in \mathcal{S}$ is associated with a gold-standard target $g(s) \in \mathcal{T}$ drawn from the filtered alignment $\mathcal{G}_f$:

$$\mathcal{G}_f = \{(s, t) \in \mathcal{G}_{\text{raw}} \mid s \in \mathcal{S}_{\text{type}}, t \in \mathcal{T}_{\text{type}}\} \quad (2)$$

where $\mathcal{G}_{\text{raw}}$ is the full alignment and the subscript denotes filtering to the entity type under evaluation (class, predicate, or instance). We assume 1:1 source-to-target mappings; if duplicate source entries are present in an alignment file, the first valid target is retained.

A retrieval method produces, for each source s , a ranked candidate list P_s of up to k target entities. Performance is measured by two metrics. *Recall@k* measures the fraction of gold pairs for which the correct target appears within the top k candidates:

$$\text{Recall@}k = \frac{1}{|\mathcal{G}_f|} \sum_{s \in \text{dom}(\mathcal{G}_f)} \mathbf{1}[g(s) \in \text{TopK}(P_s, k)] \quad (3)$$

Mean Reciprocal Rank (MRR) rewards methods that rank the correct target higher within the candidate list:

$$\text{MRR} = \frac{1}{|\mathcal{G}_f|} \sum_{s \in \text{dom}(\mathcal{G}_f)} \begin{cases} \frac{1}{\text{rank}_s} & \text{if } g(s) \in P_s \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where rank_s is the position of $g(s)$ in P_s . Candidate list order defines rank, and source entities for which no prediction is produced contribute zero to both metrics. We evaluate *Recall@k* for $k \in \{1, 5, 10, 20, 50\}$ and report MRR as the primary configuration selection metric.

3.2 Text Representation

A single textual representation is constructed for each entity from its RDF description. Labels are extracted by preferring explicit `rdfs:label` literals; if none is present, the URI fragment or terminal path segment is used. The extracted label is then canonicalised with the following normalisation steps: CamelCase identifiers are split at case boundaries (e.g., `StarshipClass` becomes “*starship class*”); separator characters (`-`, `_`, `/`) are replaced with spaces; and all text is lowercased. No stemming or lemmatisation is applied, so that observed differences in retrieval performance are attributable to the retrieval methods and their hyperparameters rather than to linguistic preprocessing choices.

The TF-IDF and BM25 retrievers consume token streams derived from this normalised text. The exact match and character n -gram baselines apply the same normalisation pipeline, with the addition of punctuation collapsing and stopword preservation to ensure deterministic string comparisons.

3.3 Retrieval Methods

We compare four lexical retrieval strategies that together span fixed baselines and configurable hyperparameter grids. Exact match and character n -gram are fixed baselines and do not participate in hyperparameter selection; TF-IDF and BM25 are evaluated exhaustively over their respective grids.

Exact match indexes normalised labels and returns a deterministic hit whenever a source query’s normalised text equals a stored target label. It serves as a lower-bound reference that quantifies how many alignments are captured by strict equality filtering alone, without any tolerance for naming variation.

Character n -gram (`char_ngram`) applies TF-IDF over character n -grams using a `char_wb` analyser with a fixed window of 3–5 characters [21]. The `char_wb` analyser pads each token with word boundaries before extracting n -grams, which prevents spurious matches across token boundaries. This method operates on the same normalised labels as exact match but tolerates morphological variation, compound terms, and minor naming differences without requiring any hyperparameter tuning. It functions as a strong zero-configuration baseline.

TF-IDF uses scikit-learn’s `TfidfVectorizer` over whitespace-separated tokens [21]. We evaluate a grid of 16 configurations varying three hyperparameters:

- *N -gram range*: $\{(1,1), (1,2), (2,2), (1,3), (2,3)\}$, controlling whether unigrams, bigrams, or trigrams (or their combinations) are included as features.
- *Minimum document frequency* (`min_df`): $\{1,2\}$, controlling whether terms appearing in only one document are retained or discarded.
- *Sublinear TF scaling* (`sublinear_tf`): $\{\text{true}, \text{false}\}$, controlling whether term frequency is log-scaled to dampen the influence of very frequent terms.

The `max_df` parameter is fixed at 1.0 throughout. These hyperparameters interact materially: as shown in Section 5, certain combinations (bigram-only or trigram-only n -gram ranges) collapse retrieval performance to near zero on datasets where unigram configurations perform well.

BM25 is implemented via the BM25S library [22] and follows the scoring function described in Section 2.6 [3]. We evaluate 20 configurations with $k_1 \in \{0.0, 0.6, 1.2, 1.8, 2.4\}$ and $b \in \{0.0, 0.3, 0.75, 1.0\}$; setting $k_1=0$ reduces BM25 to a pure IDF model and $b=0$ disables length normalisation, as described in Section 2.6.

3.4 Experimental Design and Held-Out Protocol

The evaluation follows a strict development–held-out separation to ensure that configuration selection decisions do not influence the reported held-out results. Datasets are partitioned into three tiers as described in Section 4: 24 development datasets across the Biodiversity, Anatomy, and Conference tracks; 5 primary held-out datasets from the OAEI Knowledge Graph Track; and 5 secondary held-out datasets from the OAEI BioML Track.

On the development datasets, every configuration in the BM25 and TF-IDF grids is evaluated in full. The configuration that maximises a *lambda-penalised MRR score* is selected for held-out evaluation:

$$\text{score} = \mu - \lambda\sigma \quad (5)$$

where μ is the mean per-track normalised MRR across the three development tracks, σ is its standard deviation, and $\lambda = 0.5$ is a fixed penalty coefficient. Per-track normalisation ensures that no single track dominates selection due to scale differences

in absolute MRR. The penalty term $\lambda\sigma$ favours configurations that are both high-performing and cross-domain consistent; $\lambda = 0.5$ is fixed, giving equal weight to mean performance and stability, and is not itself tuned.

This stability-aware selection formulation explicitly prioritises configurations that generalise across domains, reflecting deployment requirements where consistent behaviour is preferred over dataset-specific optimisation. The selected configuration for each method is then applied without modification to both held-out evaluation tiers. The two fixed baselines (exact match and character n -gram) do not participate in this selection procedure and are applied identically to all datasets.

For the held-out evaluation, pairwise signed-rank inference is used to assess whether observed performance differences between methods are consistent across dataset-entity-type units. Paired deltas are computed across the 15 held-out units (5 KG pairs $\times$ 3 entity types) and the 3 macro entity-type means separately; 95% confidence intervals are derived from the bootstrap empirical distribution of non-zero paired differences, providing a non-parametric assessment consistent with the sign test.

3.5 Candidate Reduction Metric

Beyond recall and ranking quality, we report a *candidate reduction ratio* that quantifies the search-space savings delivered by the top- k candidate list:

$$\text{CandidateReduction} = 1 - \frac{|C_k|}{|\mathcal{T}|} \quad (6)$$

where $|\mathcal{T}|$ is the total number of target entities available for retrieval (the target pool size) and $|C_k| = \min(k_{\max}, |\mathcal{T}|)$ is the size of the candidate list retained after top- k retrieval, with $k_{\max} = 50$ fixed for reporting. A ratio of 0 means no reduction has occurred (the candidate list equals the full target pool); a ratio approaching 1 means almost the entire target pool has been pruned. This metric complements Recall@ k and MRR by making the efficiency argument for candidate generation concrete: high recall at high reduction ratio is the ideal operating point for a first-stage filter.

3.6 Implementation and Hardware

TF-IDF and character n -gram retrieval use scikit-learn’s `TfidfVectorizer` with `linear_kernel` for similarity computation [21]. BM25 uses the BM25S library [22]. All four methods share the same label normalisation and tokenisation utilities to ensure that differences in retrieval performance reflect the methods themselves and not incidental preprocessing inconsistencies.

Experiments were conducted on a Mac mini with an Apple M4 Pro chip, 64 GB RAM, macOS Tahoe 26.3.1, Python 3.12.12, and 12 CPU cores (8 performance, 4 efficiency cores), with no concurrent heavy workloads running during any experiment. The complete code and configuration files are publicly available at <https://github.com/swntsvt/kgcandgen> to support full reproducibility of all reported results.

4 Datasets

This study uses 34 datasets drawn from five OAEI benchmark tracks, organised into three evaluation tiers: development datasets used for hyperparameter sensitivity analysis, a primary held-out evaluation on the OAEI Knowledge Graph Track, and a secondary held-out evaluation on the OAEI BioML Track. The three-tier design ensures that the configuration selection procedure is informed only by development data, and that held-out results reflect genuine out-of-sample performance. Table 1 provides a consolidated overview of all 34 datasets.

4.1 Development Datasets

The 24 development datasets span three OAEI tracks chosen for their domain diversity and their established role as shared benchmarks in the matching community [7]. All three tracks provide class-level gold standard alignments. The development datasets collectively contain 2,588 gold-standard class correspondences, providing a statistically meaningful basis for hyperparameter sensitivity analysis across the full BM25 and TF-IDF configuration grids.

4.1.1 Biodiversity Track

The Biodiversity track [7] provides two ontology pairs drawn from environmental and plant science domains. The **FLOPO-PTO** pair aligns the Flora Phenotype Ontology with the Plant Trait Ontology, yielding 227 gold-standard class correspondences. The **ENVO-SWEET** pair aligns the Environment Ontology with the Semantic Web for Earth and Environmental Terminology ontology, with 606 gold-standard correspondences. Both ontologies use compound, domain-specific labels such as `leafAreaIndex` and `soilMoistureContent` that challenge word-level tokenisation but are well served by character-level retrieval methods. The two pairs span different subdomains, providing useful within-track variation.

4.1.2 Anatomy Track

The Anatomy track provides a single, large matching task between the Mouse Adult Gross Anatomy Ontology (MA) and a corresponding fragment of the NCI Thesaurus (NCI) describing human anatomy [7]. The task contains 1,497 gold-standard class correspondences, making it the largest development dataset by gold count and the most statistically reliable basis for measuring retrieval performance. The two ontologies were constructed independently by different biomedical communities and exhibit characteristic naming variation: equivalent concepts are frequently expressed with different word orderings (e.g., “*hair of trunk*” versus “*trunk hair*”) or with different levels of specificity. This track has been part of OAEI evaluations since 2007 and is widely used as a reference benchmark for schema-level matching.

4.1.3 Conference Track

The Conference track provides 21 pairwise combinations drawn from a set of eight independently constructed ontologies describing academic conference organisation [7].

The ontologies model concepts such as submissions, reviews, programme committees, and author roles, but do so with substantially different vocabularies and structural choices. Gold-standard correspondences per pair are small, ranging from 4 to 23 class pairs (mean: 12.3), with a combined total of 258 pairs across all 21 datasets.

This small per-dataset gold count is an important characteristic to keep in mind when interpreting Conference track results. A single misclassified pair shifts a dataset’s Recall@ k by as much as 0.25, which would be negligible on the Anatomy track. Results on individual Conference datasets should therefore be treated as ordinal indicators rather than precise probability estimates. Aggregate performance across all 21 pairs provides a more reliable signal and is the primary basis for reporting in this study.

4.2 Primary Held-Out Datasets: OAEI Knowledge Graph Track

The five held-out datasets for primary evaluation come from the OAEI Knowledge Graph Track version 4 [2], built on the DBkWik corpus [23, 24]. DBkWik consolidates thousands of Fandom wikis into a unified Knowledge Graph through a semi-automated extraction pipeline that harvests infoboxes, categories, and textual content from MediaWiki instances. The resulting KGs are noisier and more structurally heterogeneous than curated ontologies, with variable label quality and large instance populations.

The five KG pairs span three fan-wiki domains:

- **Marvel Cinematic Universe – Marvel Database:** two KGs covering the Marvel fictional universe, extracted from different fan-wiki communities.
- **Memory Alpha – Memory Beta:** two Star Trek KGs representing the canonical and the expanded universe respectively.
- **Memory Alpha – Stexpanded:** the canonical Star Trek KG paired with the Star Trek Expanded Universe wiki.
- **Starwars – SWG:** the main Star Wars wiki paired with the Star Wars Galaxies game wiki.
- **Starwars – SWTOR:** the main Star Wars wiki paired with The Old Republic game wiki.

Unlike the development datasets, the KG Track is evaluated across three entity types: **classes** (schema categories), **predicates** (relation types), and **instances** (individual entities). Entity type partitioning follows the typing rules described in Section 3: class entities are identified by `rdf:type owl:Class` or by membership in the DBkWik `/class/` URI space; predicates by `rdf:type rdf:Property` or the DBkWik `/property/` URI space; and instances by the DBkWik `/resource/` URI space, after excluding anything already classified as a class or predicate.

The scale contrast across entity types is pronounced. Schema-level gold counts are small: 2–15 class pairs and 11–56 predicate pairs per KG pair, reflecting the compact size of the fan-wiki schemas. Instance-level gold counts are far larger, ranging from 1,096 to 9,295 pairs, and the target instance pools are substantially larger still, reaching 532,781 entities for the Marvel pair. This scale difference means that the candidate reduction ratio—the fraction of the target pool excluded by a top-50 candidate list—varies dramatically across entity types, a pattern explored in detail in Section 6.

4.3 Secondary Held-Out Datasets: OAEI BioML Track

The five secondary held-out datasets come from the OAEI BioML Track [15], which provides large-scale biomedical ontology matching tasks specifically designed to support machine learning-based matching systems. All five tasks involve class-level matching only. The datasets pair well-established biomedical terminologies:

- **NCIT–DOID**: the National Cancer Institute Thesaurus paired with the Disease Ontology (3,245 gold pairs; target pool: 8,448 classes).
- **OMIM–ORDO**: the Online Mendelian Inheritance in Man paired with the Orphanet Rare Disease Ontology (2,595 gold pairs; target pool: 8,838 classes).
- **SNOMED-Body – FMA-Body**: the body anatomy fragment of SNOMED CT paired with the Foundational Model of Anatomy (4,604 gold pairs; target pool: 64,726 classes).
- **SNOMED-Neoplasm – NCIT-Neoplasm**: the neoplasm fragment of SNOMED CT paired with the corresponding NCIT fragment (2,488 gold pairs; target pool: 13,956 classes).
- **SNOMED-Pharm – NCIT-Pharm**: the pharmacology fragment of SNOMED CT paired with the NCIT pharmacology fragment (3,989 gold pairs; target pool: 15,250 classes).

The BioML datasets differ from both the development datasets and the KG Track in three important respects. First, their gold standard sizes are an order of magnitude larger than any development dataset, making results here statistically robust and directly meaningful. Second, biomedical concept labels routinely differ across ontologies even when the underlying concept is equivalent: SNOMED CT favours fully-specified compositional names (e.g., “*Malignant neoplasm of body of stomach*”), while NCIT uses shorter, often acronym-heavy identifiers (e.g., “*Gastric Body Cancer*”). This creates a demanding test for all lexical retrieval methods. Third, the target pools are substantially larger than in the Conference or Anatomy tracks but substantially smaller than the KG Track instance pools, placing the BioML tasks in an intermediate difficulty regime for candidate generation.

4.4 Summary and Design Rationale

Table 1 consolidates all 34 datasets. Together they span fan-wiki, biomedical, environmental, plant science, anatomy, and conference domains, containing 34,866 gold-standard correspondences across class, predicate, and instance entity types. The two held-out tiers introduce matching challenges qualitatively absent from the development tracks: the KG Track adds multi-entity-type evaluation and large instance pools, while the BioML Track adds large-scale biomedical ontology matching with highly variable label conventions.

Table 1: Summary of all 34 datasets used across the three evaluation tiers. Gold count refers to the number of gold-standard correspondences used for evaluation after entity-type partitioning and filtering. Target pool size is the number of target entities available for retrieval. The candidate size is $\min(k_{\max}, |\mathcal{T}|)$ with $k_{\max} = 50$. Target pool sizes are not reported for development datasets as the candidate size equals 50 for all but a small number of Conference pairs where the target schema is smaller than 50. KG Track dataset names are abbreviated as follows: MCU–Marvel = marvelcinematicuniverse-marvel; MA–MB = memoryalpha-memorybeta; MA–STX = memoryalpha-stexpanded; SW–SWG = starwars-swg; SW–SWTOR = starwars-swtor.

Tier / Track	Dataset	Entity Type	Gold	Target Pool
<i>Development</i>				
Biodiversity	flopo-pto	Class	227	–
	envo-sweet	Class	606	–
Anatomy	mouse-human-suite	Class	1,497	–
Conference	21 pairs (see text)	Class	258	–
	<i>Development total</i>		2,588	–
<i>Primary Held-Out — Knowledge Graph Track</i>				
MCU–Marvel		Class	2	186
		Predicate	11	140
		Instance	1,653	532,781
MA–MB		Class	14	240
		Predicate	53	430
		Instance	9,295	124,443
MA–STX		Class	13	283
		Predicate	41	206
		Instance	1,725	29,506
SW–SWG		Class	5	67
		Predicate	20	149
		Instance	1,096	14,728
SW–SWTOR		Class	15	101
		Predicate	56	368
		Instance	1,358	7,587
	<i>KG Track total</i>		15,357	–
<i>Secondary Held-Out — BioML Track</i>				
	ncit-doid	Class	3,245	8,448
	omim-ordo	Class	2,595	8,838
	snomed.body-fma.body	Class	4,604	64,726
	snomed.neoplas-ncit.neoplas	Class	2,488	13,956
	snomed.pharm-ncit.pharm	Class	3,989	15,250
	<i>BioML Track total</i>		16,921	–
	Grand total		34,866	–

5 Results: Development Analysis

5.1 BM25 Hyperparameter Sensitivity

Figure 1 shows the 5×4 heatmap of average MRR across the 24 development datasets for all 20 BM25 configurations. The top-performing configuration ($k_1=2.4$, $b=0.75$) achieves an average MRR of 0.816, while the weakest configuration ($k_1=2.4$, $b=0.0$) reaches only 0.731, giving a total performance span of 0.084 across the entire grid. This narrow range is the central message of the BM25 sensitivity analysis: no configuration fails catastrophically, and the worst BM25 setting still substantially outperforms the weakest TF-IDF configurations described in Section 5.2.

Two structural patterns are clearly visible in the heatmap. First, the entire $b=0$ column (leftmost, pure IDF weighting with no length normalisation) is uniformly dark at 0.731, regardless of k_1 . This confirms that length normalisation is the dominant factor separating the low-performing region from the rest of the grid: without it, all k_1 values perform equally poorly. Second, once $b \geq 0.3$, the heatmap becomes uniformly bright, with the eight highest-performing configurations (all sharing $b \geq 0.75$) within 0.001 MRR of the maximum. The practical recommendation is therefore robust: any configuration with $b \geq 0.75$ and $k_1 \in \{0.6, 1.2, 1.8, 2.4\}$ can be adopted with confidence.

The λ -penalised selection policy (mean per-track MRR minus $0.5 \times$ track standard deviation, as defined in Section 3.4) selects ($k_1=2.4$, $b=0.75$) as the configuration with the best combination of high mean performance and cross-domain consistency. This configuration is carried forward to held-out evaluation.

The marginal gains in Recall across candidate list sizes reveal a characteristic pattern that informs the choice of $k=50$ as the standard reporting cutoff. For the selected BM25 configuration, the largest gain occurs at the $1 \rightarrow 5$ transition (mean $+0.084$ Recall), reflecting how quickly BM25 surfaces most true matches near the top of the ranked list. Gains then diminish substantially: $5 \rightarrow 10$ yields a mean gain of $+0.024$, and $10 \rightarrow 20$ only $+0.011$. Notably, the $20 \rightarrow 50$ transition rises again substantially to a mean gain of $+0.054$, indicating that a meaningful proportion of true matches are ranked between position 21 and 50 and would be missed by a $k=20$ cutoff. This non-monotonic profile, consistent across character n -gram and TF-IDF as well, confirms that $k=50$ captures substantially more recall than $k=20$ and is the appropriate default cutoff for the datasets studied here.

5.2 TF-IDF Hyperparameter Sensitivity

Figure 2 shows the MRR for all 16 TF-IDF configurations across the two `sublinear_tf` settings. The best-performing configurations use unigram vectors (`ngram_range=[1,1]`, `min_df=1`) and achieve an average MRR of 0.818 regardless of `sublinear_tf` setting. The two panels of Figure 2 are in fact numerically identical across all configurations: `sublinear_tf` has no measurable effect on aggregate MRR on these datasets. This is itself a finding: log-scaling of term frequencies does not change retrieval quality for the short, low-frequency label vocabularies characteristic

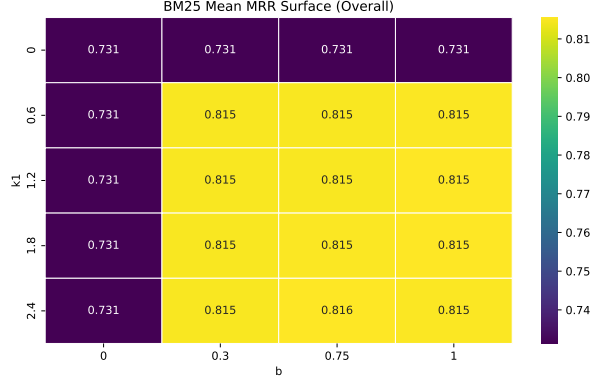


Fig. 1: Average MRR across 24 development datasets for all 20 BM25 configurations, with $k_1 \in \{0.0, 0.6, 1.2, 1.8, 2.4\}$ on the y-axis and $b \in \{0.0, 0.3, 0.75, 1.0\}$ on the x-axis. The entire $b=0$ column is uniformly at 0.731 (dark), regardless of k_1 , while all configurations with $b \geq 0.3$ and non-zero k_1 achieve MRR of 0.815 or above (bright). The total span across all configurations is only 0.084, confirming BM25’s robustness to hyperparameter choice in this setting.

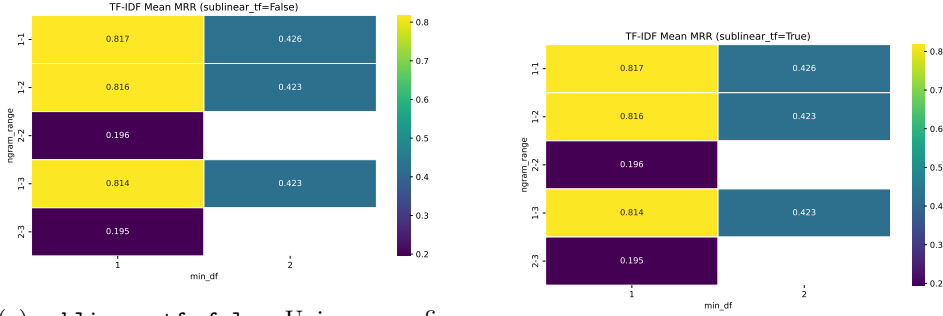
of ontology and KG matching tasks. The `sublinear_tf=false` variant is adopted as the selected configuration.

The most important finding in these results is not which configuration performs best, but how severely the worst configurations fail. Bigram-only (`ngram_range=[2, 2]`) and configurations starting from bigrams (`ngram_range=[2, 3]`) collapse to an average MRR of approximately 0.195—a drop of 0.622 from the best configuration on the same datasets. This failure is not a near-miss: it represents near-complete inability to rank true matches within the top 50 candidates. The mechanism is straightforward: entity labels in these datasets are predominantly single words or short phrases. A bigram-only feature space cannot represent any single-word label as a non-zero vector, causing those labels to score identically at zero against all queries and fall out of the ranking entirely. Note that the `ngram_range=[2, 2]` row has no entry for `min_df=2`: raising the minimum document frequency on an already sparse bigram vocabulary produces an empty feature space for most datasets, rendering retrieval undefined.

Raising `min_df` to 2 for configurations that already include unigrams (`ngram_range=[1, 1]` or `[1, 2]` or `[1, 3]`) causes a sharp drop from 0.817 to 0.426 MRR, because many entity labels in small ontologies are unique terms that appear in only one document and are eliminated by the frequency filter. This is a second failure mode distinct from the bigram collapse: it operates on unigram features but prunes precisely the rare, distinctive terms that would otherwise discriminate between candidates.

This catastrophic sensitivity is entirely absent from the BM25 grid. The contrast—a performance span of 0.084 for BM25 versus 0.622 for TF-IDF—is the strongest

empirical argument for preferring BM25 over word-level TF-IDF as a default candidate generation method in pipelines where per-dataset configuration validation is not feasible.



(a) `sublinear_tf=false`. Unigram configurations (`ngram_range=[1, 1]`) with `min_df=1` achieve $\text{MRR} = 0.818$. Bigram-only and trigram-range configurations collapse to $\text{MRR} \approx 0.196$. Raising `min_df` to 2 halves performance for unigram configurations ($0.817 \rightarrow 0.426$).

(b) `sublinear_tf=true`. All cell values are numerically identical to the left panel, confirming that log-scaling of term frequencies has no measurable effect on aggregate MRR for the short label vocabularies characteristic of this task.

Fig. 2: Average MRR across 24 development datasets for all 16 TF-IDF configurations. The y-axis shows `ngram_range` and the x-axis shows `min_df`. The colour scale is shared across both panels. The `ngram_range=[2, 2]` row has no entry for `min_df=2` (shown as blank): the combination produces an empty feature space for most datasets. The performance span of 0.622 between the best and worst TF-IDF configurations far exceeds the 0.084 span observed for BM25.

5.3 Per-Track Breakdown and Method Comparison

Table 2 summarises the best-configuration MRR for each method across the three development tracks. Character n -gram retrieval leads on all three tracks without any hyperparameter tuning, achieving MRR of 0.863 on Biodiversity, 0.857 on Anatomy, and 0.821 on Conference. The selected BM25 and TF-IDF configurations perform closely and trail character n -gram by 0.02–0.05 MRR depending on track. Exact match lags on all three tracks, most notably on Anatomy (0.619), where systematic naming variation between mouse and human anatomy ontologies prevents string equality from being effective.

Figure 3 plots the best-of-grid MRR for TF-IDF and BM25 side by side across all 24 individual development datasets, allowing the per-dataset relationship between the two methods to be inspected directly. The two methods are closely matched on most datasets (short connectors), but BM25 consistently avoids the low-performance outliers visible for TF-IDF on some Conference pairs. Figure 4 shows the distribution of best-of-grid MRR across the three tracks for the same two methods, confirming

that the Anatomy and Biodiversity tracks show less within-track variance than the Conference track, whose small gold counts per dataset produce higher individual-dataset noise.

The Conference track results should be interpreted with particular caution. With an average of only 12.3 gold correspondences per dataset, a single misclassified pair shifts an individual dataset’s Recall@ k by up to 0.25. The aggregate across all 21 pairs (258 total gold correspondences) is more reliable and is the basis for the track-level figures in Table 2.

Table 2: Best-configuration MRR per method and development track. For BM25 and TF-IDF, the reported MRR corresponds to the λ -penalised selected configuration (Section 3.4). Character n -gram and exact match are fixed baselines with no configuration selection step.

Track	Exact Match	TF-IDF	BM25	Char n -gram
Biodiversity	0.644	0.838	0.838	0.863
Anatomy	0.619	0.791	0.809	0.857
Conference	0.583	0.817	0.814	0.821

6 Results: Held-Out Evaluation

6.1 KG Track — Schema-Level Results

Table 3 reports Recall@ k and MRR averaged across the five KG pairs for classes and predicates separately. Two patterns dominate.

For **classes**, character n -gram retrieval leads clearly, achieving Recall@50 of 0.969 and MRR of 0.917. TF-IDF (Recall@50 = 0.929, MRR = 0.824) and BM25 (Recall@50 = 0.844, MRR = 0.815) are competitive but trail by a meaningful margin. Exact match falls to MRR = 0.645, reflecting the lexical variation in class labels across KG versions, where morphological differences and naming conventions differ between wikis.

For **predicates**, all methods approach the performance ceiling. Character n -gram again leads (Recall@50 = 0.993, MRR = 0.981), but notably, exact match achieves MRR = 0.931—higher than both TF-IDF and BM25 for this entity type. This result reflects the relatively stable and concise nature of predicate labels in fan-wiki KGs: predicates such as **portrayedBy** or **homeworld** are short, distinctive, and consistent across related wikis, making exact normalised matching an effective strategy. The class–predicate performance gap therefore reflects a genuine difference in lexical drift between entity types rather than a limitation of any particular method.

The choice of aggregation method affects the relative ordering of BM25 and TF-IDF, and this is worth making explicit. Under *macro aggregation*—averaging across entity types first, then across datasets—TF-IDF edges ahead of BM25 (MRR: 0.851 vs 0.849). Under *micro aggregation*—weighting each dataset–entity-type unit by its gold

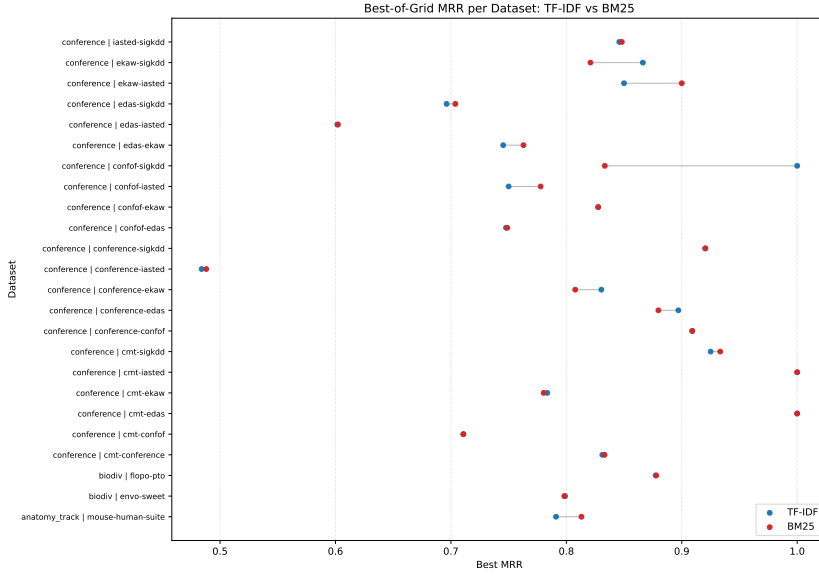


Fig. 3: Best-of-grid MRR for TF-IDF (blue) and BM25 (red) on each of the 24 development datasets. Each point marks the best MRR achieved by that method across its full hyperparameter grid; a connector line joins the two points for the same dataset. Short connectors indicate the two methods perform similarly; long connectors indicate one method outperforms the other on that dataset. BM25 avoids the low-performance outliers (e.g., conference-iaisted at MRR ≈ 0.48) that appear for TF-IDF, consistent with BM25’s narrower overall performance range.

Table 3: Mean Recall@ k and MRR for KG Track classes and predicates, averaged across the five held-out KG pairs. The selected BM25 and TF-IDF configurations from the development analysis are used throughout.

Method	R@1	R@5	R@10	R@20	R@50	MRR
<i>Classes</i>						
Exact match	0.645	0.645	0.645	0.645	0.645	0.645
TF-IDF	0.799	0.844	0.844	0.844	0.929	0.824
BM25	0.799	0.844	0.844	0.844	0.844	0.815
Char n -gram	0.895	0.940	0.955	0.955	0.969	0.917
<i>Predicates</i>						
Exact match	0.931	0.931	0.931	0.931	0.931	0.931
TF-IDF	0.905	0.909	0.913	0.938	0.949	0.910
BM25	0.905	0.905	0.928	0.928	0.937	0.908
Char n -gram	0.975	0.989	0.989	0.993	0.993	0.981

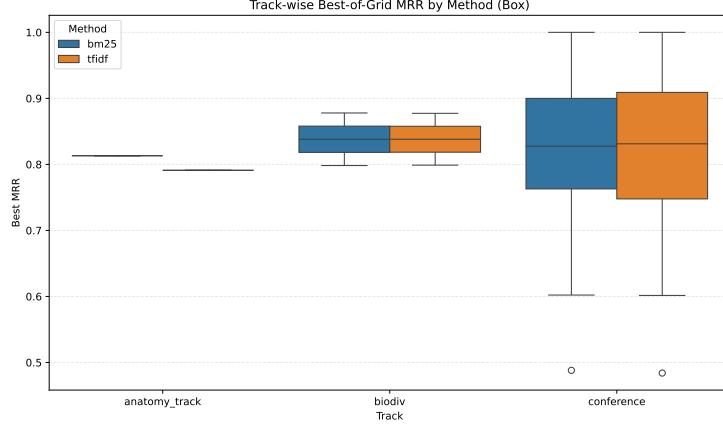


Fig. 4: Distribution of best-of-grid MRR for TF-IDF and BM25 across datasets within each development track. Each box summarises the interquartile range of per-dataset best MRR values within that track; the median is shown as a horizontal line. The Conference track shows higher within-track variance than Anatomy or Biodiversity, reflecting the noise introduced by small per-dataset gold counts. Character n -gram and exact match are fixed baselines and are not shown.

count, which causes the large instance pools to dominate—BM25 leads (MRR: 0.845 vs 0.836). The reversal reflects the entity-type composition of the gold standard: TF-IDF achieves slightly higher Recall@50 for schema-level classes (average 0.929 vs 0.844 for BM25), while BM25 performs slightly better on the large-instance tasks that dominate the micro-weighted aggregate. Neither difference is statistically significant ($p=0.67$ for macro MRR; $p=0.09$ for macro Recall@50). Readers comparing results across papers should therefore be attentive to which aggregation scheme is used, as it can reverse the apparent ordering of closely matched methods.

To assess whether the observed performance differences are consistent across datasets rather than driven by individual outliers, we apply pairwise signed-rank inference across the 15 held-out dataset-entity-type units (5 KG pairs $\times$ 3 entity types). The comparison is conducted at two levels of aggregation: across all 15 units individually, and across the 3 macro entity-type means (class, predicate, instance). At the 15-unit level, character n -gram’s MRR advantage over BM25 is statistically significant ($\Delta=-0.060$, 95% CI $[-0.097, -0.027]$, $p=0.0005$), as is its advantage over TF-IDF ($\Delta=-0.059$, 95% CI $[-0.095, -0.025]$, $p=0.001$). The difference between BM25 and TF-IDF is not significant ($\Delta=0.002$, 95% CI $[-0.003, 0.008]$, $p=0.67$), confirming that the two configurable methods are statistically indistinguishable in aggregate MRR on these held-out datasets. Both BM25 and TF-IDF significantly outperform exact match ($p=0.012$ in both cases). At the 3-unit macro level, the same directional patterns hold but none of the comparisons reach conventional significance thresholds, which is expected given the low statistical power of a three-observation signed-rank test. The entity-type breakdown reveals that character n -gram’s advantage is present across all three types: for classes ($\Delta=-0.102$ over BM25, $p=0.125$),

for predicates ($\Delta=-0.073$ over BM25, $p=0.125$), and for instances ($\Delta=-0.006$ over BM25, $p=0.125$), with the class-level advantage largest in absolute terms.

6.2 KG Track — Instance-Level Results

Instance matching presents a qualitatively different challenge from schema matching: target pools are orders of magnitude larger (up to 532,781 entities for the Marvel pair), yet all four methods maintain Recall@50 in the range 0.74–0.92. Table 4 reports the instance-level results. Character n -gram, TF-IDF, and BM25 cluster closely (Recall@50: 0.920, 0.911, 0.915; MRR: 0.830, 0.818, 0.824), while exact match falls to Recall@50 = 0.738 and MRR = 0.711. The fact that exact match still recovers 73.8% of instance gold pairs at Recall@50 indicates that many instance labels are already consistently named across related fan-wiki communities, even if some variation remains.

The most striking result at the instance level is not the recall values themselves but what they enable. Retrieving only the top 50 candidates from a pool of 532,781 entities reduces the search space by 99.99%, while retaining Recall@50 above 0.90 for all lexical methods. Across all five KG pairs, instance-level candidate reduction ratios range from 99.34% to 99.99% (Table 6). Downstream verification or neural scoring therefore operates on fewer than 0.01% of all possible entity pairs for the largest KG, making large-scale matching computationally tractable without meaningful sacrifice of recall.

Table 4: Mean Recall@ k and MRR for KG Track instances, averaged across the five held-out KG pairs.

Method	R@1	R@5	R@10	R@20	R@50	MRR
Exact match	0.684	0.711	0.723	0.730	0.738	0.711
TF-IDF	0.765	0.854	0.882	0.899	0.911	0.818
BM25	0.770	0.860	0.887	0.902	0.915	0.824
Char n -gram	0.778	0.866	0.894	0.908	0.920	0.830

6.3 BioML Track — Biomedical Class Matching

Table 5 reports unweighted average results across the five BioML datasets.

Character n -gram retrieval again leads, achieving an average Recall@50 of 0.907 and MRR of 0.761 across the five pairs. BM25 and TF-IDF perform comparably (Recall@50: 0.866 and 0.862; MRR: 0.721 for both), trailing character n -gram by approximately 0.04 MRR. Exact match collapses to Recall@50 = 0.329 and MRR = 0.329. The equality of these two values is not coincidental: exact match either retrieves the correct target at rank 1 or not at all, so Recall@50 equals Recall@1 for this method. This collapse demonstrates that biomedical concept labels cannot be matched by string equality alone: SNOMED CT and NCIT, for example, use systematically different naming conventions for the same concepts (e.g., “*Malignant neoplasm*”).

of *body of stomach*” versus “*Gastric Body Cancer*”), which no normalisation strategy can bridge without tolerance for variation.

The hardest individual task is **snomed.body-fma.body**, where exact match achieves $\text{Recall@1} = 0.070$ (fewer than 1 in 14 gold pairs recovered at the top position) and even BM25 reaches only $\text{Recall@50} = 0.875$. Character n -gram fares best with $\text{Recall@50} = 0.889$, but the overall performance level for all methods is lower than on other BioML pairs, reflecting the depth of vocabulary divergence between SNOMED CT’s compositional naming style and FMA’s systematic anatomical terminology. In contrast, **snomed.pharm-ncit.pharm** is the most tractable BioML task: character n -gram reaches $\text{Recall@50} = 0.968$ and $\text{MRR} = 0.920$, suggesting that pharmacological terminology is more consistently named across these two ontologies than anatomical or disease concepts.

Table 5: Unweighted Average Recall@ k and MRR for BioML Track class matching.

Method	R@1	R@5	R@10	R@20	R@50	MRR
Exact match	0.329	0.329	0.329	0.329	0.329	0.329
TF-IDF	0.678	0.815	0.842	0.854	0.862	0.721
BM25	0.671	0.818	0.846	0.858	0.866	0.721
Char n -gram	0.712	0.856	0.884	0.896	0.907	0.761

6.4 Candidate Reduction Analysis

Table 6 consolidates the candidate reduction ratios for all held-out datasets by entity type. Reduction ratios vary sharply and systematically by entity type, and understanding this variation is important for practical pipeline design.

For **schema elements** (classes and predicates), target pools are small. Class pools range from 67 to 283 entities across the five KG pairs; predicate pools from 140 to 430. A top-50 candidate list therefore prunes only 25%–88% of the target pool depending on pool size. The extreme case is the *starwars-swg* class pool of just 67 classes: a candidate list of 50 retains 74.6% of the entire target schema and offers almost no reduction benefit. For such small schemas, practitioners should use a smaller value of k proportional to the target pool size, rather than applying $k=50$ uniformly.

For **instances**, the picture is entirely different. Target pools range from 7,587 to 532,781 entities, and a top-50 candidate list reduces the search space by 99.34% to 99.99%. Even the smallest reduction (99.34% for *starwars-swtor*) means downstream verification operates on fewer than 0.7% of all possible pairs. For the largest pool (Marvel instances at 532,781), the top-50 list is just 0.01% of the full target universe, delivering near-total compression of the search space while maintaining Recall@50 above 0.91 for all lexical methods.

For **BioML classes**, target pools range from 8,448 to 64,726, yielding reduction ratios of 99.41% to 99.92%—comparable to instance-level reductions. This confirms that even class-level matching in large biomedical ontologies benefits enormously from

a top-50 candidate list: the full cross-product of source and target classes is compressed by over 99% before any expensive scoring is applied.

Table 6: Candidate reduction ratios at $k=50$ for all held-out datasets, by entity type. A ratio of 1.0 would mean the entire target pool is excluded; a ratio of 0 means no pruning occurs. BioML datasets are class-only; KG Track datasets are reported across all three entity types. Dataset abbreviations follow Table 1.

Track	Dataset	Entity Type	Reduction Ratio
KG	MCU-Marvel	Class	0.731
		Predicate	0.643
		Instance	0.9999
	MA-MB	Class	0.792
		Predicate	0.884
		Instance	0.9996
	MA-STX	Class	0.823
		Predicate	0.757
		Instance	0.9983
	SW-SWG	Class	0.254
		Predicate	0.664
		Instance	0.9966
BioML	SW-SWTOR	Class	0.505
		Predicate	0.864
		Instance	0.9934
		Class	0.9941
		Class	0.9943
		Class	0.9992
		Class	0.9964
		Class	0.9967

7 Discussion

A key contribution of this study is the identification of reliability as a primary design criterion for candidate generation, alongside traditional effectiveness metrics.

7.1 BM25 as a Safe Default for Lexical Candidate Generation

The central practical message from this study is that BM25 is a safe default for lexical candidate generation in KG and ontology matching pipelines. Across all 24 development datasets and both held-out evaluations, no BM25 configuration failed catastrophically: the worst configuration ($b=0.0$, any k_1) still achieves $\text{MRR} = 0.731$, while the best ($b=0.75$, $k_1=2.4$) achieves 0.816—a span of only 0.084. The source of this robustness is structural. BM25’s probabilistic scoring function always produces

a non-zero similarity score for any query that shares at least one token with a document, regardless of parameter settings. A poor choice of k_1 or b changes the ranking order but never collapses the retrieval entirely. This is a fundamentally different failure mode from TF-IDF, where certain hyperparameter combinations prevent entire classes of labels from being represented at all.

A direct measure of cross-domain generalisation further strengthens the case for BM25. The transfer gap—the difference between a method’s development performance and its held-out macro MRR—reveals a striking asymmetry between the two configurable methods. BM25, selected with a λ -penalised MRR of 0.760 on development data ($\mu=0.860$, $\sigma=0.199$), transfers to held-out data with a macro MRR of 0.849, a drop of only 0.011. TF-IDF, which achieved a perfect normalised development rank of 1.0 across all three development tracks ($\mu=1.0$, $\sigma=0.0$), drops to a held-out macro MRR of 0.851—a transfer gap of 0.149, more than thirteen times larger than BM25’s. The explanation lies in TF-IDF’s development score: the unigram, `min_df=1` configuration ranked first on every development track, giving it a perfect score by construction, but zero variance across tracks—meaning the λ -penalty had nothing to penalise. This mathematical consistency on development data did not predict held-out behaviour, because the held-out domains introduced label distributions and schema characteristics absent from the development tracks. BM25’s development score, by contrast, was penalised for cross-track variance ($\sigma=0.199$) by the λ -penalty, and this penalty correctly anticipated its actual held-out behaviour. The transfer gap results confirm that BM25 is not merely competitive on held-out data—it is more reliably predictable from development evidence than TF-IDF.

7.2 TF-IDF is a High-Variance Option Requiring Explicit Validation

TF-IDF presents a sharply different risk profile. When correctly configured with unigrams (`ngram_range=[1, 1]`) and `min_df=1`, it matches BM25 at $\text{MRR} = 0.818$ and is a perfectly adequate candidate generator. When misconfigured, it fails completely. Two distinct failure modes were identified in this study.

The first is the *bigram collapse*: configurations that exclude unigrams (`ngram_range=[2, 2]` or `[2, 3]`) produce an empty or near-empty feature representation for any single-word label, causing those entities to score identically at zero against all queries and disappear from the ranking. The result is $\text{MRR} \approx 0.195$ on the same datasets where unigram TF-IDF achieves 0.817.

The second is the *minimum frequency trap*: raising `min_df` to 2 prunes rare terms from the vocabulary. For small ontologies and KG schemas, where many entity labels are unique concepts appearing in only one document, this filter eliminates precisely the distinctive terms that would otherwise discriminate between candidates, halving MRR from 0.817 to 0.426 even for unigram configurations.

Both failure modes are non-obvious to practitioners who have not examined their data distribution. A researcher who deploys TF-IDF with bigram features for NLP-inspired reasons, or raises `min_df` to reduce noise, will encounter severe performance degradation with no clear diagnostic signal beyond low end-to-end recall.

The recommendation is therefore explicit: if TF-IDF is used for candidate generation, its configuration must be validated on representative development data before deployment. In the absence of such validation, BM25 is the safer choice.

It is worth noting one context in which the selected TF-IDF configuration offers a practical advantage over BM25. For schema-level class matching on the KG Track held-out data, TF-IDF achieves a higher average Recall@50 (0.929) than BM25 (0.844). Across the five KG pairs, TF-IDF’s class Recall@50 equals or exceeds BM25’s on four of five datasets (margins of +0.071, +0.154, and +0.200 on three pairs; tied on two), with a paired delta of +0.085 (95% CI [0.014, 0.156], $p=0.25$). This advantage arises because unigram TF-IDF’s term weighting assigns high scores to rare, distinctive class labels that appear in only one or two documents, making it effective at surfacing correct matches that BM25 may rank lower due to its length normalisation. The practical implication is nuanced: for pipelines where the downstream matcher is capable of reranking the top 50 candidates effectively—such as neural classifiers or LLM-based verifiers—TF-IDF’s higher class Recall@50 is an argument in its favour for the schema matching stage, provided the configuration has been validated on development data. BM25 remains the safer choice when validation is not feasible, but practitioners who have confirmed a robust TF-IDF configuration should be aware that it may retrieve more true class matches within their candidate list.

7.3 Character n -gram Retrieval as a Practical First Baseline

Character n -gram retrieval consistently outperforms both BM25 and word-level TF-IDF across all domains in this study, while requiring no hyperparameter tuning. On the KG Track held-out data it achieves Recall@50 of 0.969 for classes and 0.993 for predicates; on the BioML Track it leads with Recall@50 = 0.907 and MRR = 0.761, compared to 0.866 and 0.721 for both BM25 and TF-IDF.

The mechanism behind this advantage is the `char_wb` analyser’s ability to match subword patterns across label boundaries. Character n -grams of length 3–5 capture common prefixes, suffixes, and internal substrings shared between labels that differ in morphological form (e.g., “*neoplasm*” and “*neoplastic*”) or compositional structure (e.g., “*hair of trunk*” and “*trunk hair*”). Word-level methods treat these as entirely different tokens; character-level methods share substantial feature mass between them. This explains the particularly large advantage of `char_ngram` on biomedical datasets, where systematic variation in naming conventions is the norm rather than the exception.

The practical recommendation is to include character n -gram retrieval as the first baseline in any candidate generation evaluation. It is zero-configuration, available in standard libraries, and sets a performance ceiling that more complex methods must exceed to justify their additional complexity or tuning cost. The one open question is scalability: the largest target pool evaluated here contains 532,781 instance entities. At index sizes in the tens or hundreds of millions, the memory footprint of the character n -gram TF-IDF matrix may become a constraint, and investigating this regime is an important direction for future work.

7.4 Domain-Specific Observations and Lexical Regimes

The three domains studied in this paper represent three distinct *lexical regimes* that have practical implications for method selection.

KG predicates represent a high-consistency regime. Labels such as `portrayedBy` or `homeworld` are short, distinctive, and stable across related fan-wiki communities. In this regime, exact normalised matching achieves $\text{MRR} = 0.931$ —higher than both BM25 and TF-IDF—and the differences between all methods are small.

Fan-wiki classes and conference ontologies represent a moderate-variation regime. Labels share substantial lexical overlap but differ in morphological form, word ordering, or naming conventions. All lexical methods perform well here, but character n -gram holds a consistent advantage of 0.02–0.05 MRR over BM25. The Conference track results in this regime must be interpreted with caution given the small per-dataset gold counts (4–23 pairs), which amplify noise on individual datasets; aggregate results across all 21 pairs are more reliable.

Biomedical ontologies represent a high-variation regime. Labels for the same concept can differ substantially across ontologies due to systematic differences in naming philosophy (e.g., SNOMED CT’s compositional style versus NCIT’s abbreviated names). In this regime, the choice of method becomes critical: exact match collapses to $\text{Recall@50} = 0.329$, while character n -gram maintains 0.907. For practitioners working with biomedical ontologies specifically, character n -gram retrieval should be regarded as the minimum acceptable baseline, and exact match should not be used as a candidate generator without augmentation.

7.5 Implications for Matching Pipeline Design

The results of this study suggest several concrete recommendations for researchers and practitioners building multi-stage KG and ontology matching pipelines.

- **Always include character n -gram as a baseline.** It is zero-configuration, broadly available, and sets a performance ceiling against which other methods should be compared before additional complexity is introduced.
- **Use BM25 as the default tunable ranker.** With $b=0.75$ and $k_1 \in \{1.2, 1.8\}$, BM25 delivers robust near-optimal performance without per-dataset tuning. It is particularly appropriate when the target deployment domain is unknown or diverse.
- **Validate TF-IDF configuration explicitly.** If TF-IDF is used, test at minimum the unigram $([1, 1])$ and bigram-inclusive $([1, 2])$ configurations with `min_df= 1` on representative development data before deployment. Never assume a bigram or trigram configuration is safe without empirical evidence on the target label distribution.
- **Set k adaptively for schema elements.** For entity types with small target pools (fewer than 100 entities), $k=50$ provides little reduction benefit. In such cases, use $k \approx 0.3 \times |\mathcal{T}|$ or smaller to ensure the candidate list remains proportional to the pool. For large instance pools, $k=50$ is both effective and sufficient.
- **Report candidate reduction ratios as standard diagnostics.** $\text{Recall@}k$ and MRR alone do not capture the efficiency argument for candidate generation. The candidate reduction ratio makes explicit what fraction of the cross-product search

space is eliminated and should be reported alongside recall metrics as a standard diagnostic in system descriptions.

7.6 Limitations

Several limitations of this study should be considered when interpreting and generalising its findings.

Label-only text representation. All methods are evaluated on entity labels alone, with no use of textual definitions, synonyms, comments, or structural features such as class hierarchies or property domains. Including richer textual context—particularly for biomedical ontologies where synonyms and definitions are available—would likely improve all methods and may alter their relative ordering. The label-only design was chosen deliberately to isolate retrieval method effects, but it means the reported numbers should be treated as lower bounds on what is achievable with richer representations.

One-to-one alignment assumption. The evaluation assumes each source entity has exactly one gold-standard target. In practice, many-to-one, one-to-many, and complex mappings occur, particularly in biomedical ontologies where a SNOMED concept may subsume multiple NCIT concepts. The impact of candidate generation on these more complex alignment types is not assessed here.

Single library per method. BM25 is evaluated via BM25S and TF-IDF via scikit-learn. Minor scoring differences exist across implementations [20], and results may vary slightly with other backends such as Elasticsearch or Pyserini.

No dense retrieval baseline. Embedding-based dense retrieval using models such as Sentence-BERT [12] or domain-fine-tuned encoders represents an increasingly important class of candidate generation method [14, 17]. The absence of a dense retrieval baseline means the study cannot characterise the trade-off between lexical and semantic retrieval for this task. This is an important omission that future work should address.

Development-to-held-out domain proxy. The three development tracks (Biodiversity, Anatomy, Conference) are used as proxies for diverse matching domains. The degree to which they are representative of all practical KG and ontology matching scenarios—including enterprise knowledge graphs, cross-lingual settings, or dynamically constructed KGs—remains an open question.

Recall-only evaluation. All reported metrics measure candidate recall: whether the true match is present in the candidate list. The downstream impact of candidate generation quality on end-to-end pipeline precision and F1 is not evaluated. A pipeline that achieves high Recall@50 but admits many false positives into the candidate list imposes a burden on the scoring stage that is not captured here.

7.7 Implications for Real-World KG Matching Systems

The findings of this study have direct implications for the design and deployment of scalable Knowledge Graph and ontology matching systems. In practical settings, candidate generation must balance two competing objectives: maximising recall to

preserve true correspondences, and aggressively reducing the search space to ensure computational tractability.

First, the results demonstrate that character n -gram retrieval provides a strong and reliable zero-configuration baseline. Its consistent performance across domains and entity types makes it particularly suitable for systems where labelled development data or parameter tuning is not feasible. Practitioners can adopt character n -gram retrieval as a default candidate generation strategy with minimal risk.

Second, BM25 emerges as a robust and tunable alternative, offering stable performance across a wide range of parameter settings. The observed insensitivity to hyperparameter choices suggests that standard configurations can be used with confidence, making BM25 a safe choice for deployment in heterogeneous environments.

In contrast, word-level TF-IDF requires careful configuration and validation prior to deployment. The observed failure modes, particularly for higher-order n -gram settings and restrictive document frequency thresholds, highlight the risk of silent degradation in candidate recall. Systems that employ TF-IDF should therefore include explicit validation procedures to ensure that chosen configurations do not eliminate valid matches.

Finally, the analysis of candidate reduction ratios confirms that lexical retrieval methods can achieve substantial search-space compression while maintaining high recall, particularly for large instance-level matching tasks. This enables the integration of more computationally expensive downstream matchers, such as neural or structural approaches, without compromising scalability.

Overall, these findings support a design strategy in which robust lexical retrieval methods are used as a first-stage filter, providing a reliable and efficient foundation for multi-stage KG matching pipelines.

8 Conclusion and Future Work

Candidate generation is the step every KG and ontology matching pipeline depends on, yet it is the step that receives the least systematic attention. A true match absent from the candidate list is permanently lost to all downstream processing, however sophisticated. This paper has shown, across 34 datasets spanning five OAEI benchmark tracks and three qualitatively distinct domains, that the choice of lexical retrieval method and its configuration can be the difference between a pipeline that works reliably and one that fails silently.

Three findings stand out. First, character n -gram retrieval is the strongest single method in this study and requires no tuning: it achieves Recall@50 of 0.969 for schema-level classes and 0.993 for predicates on held-out Knowledge Graph data, and maintains Recall@50 above 0.90 on large-scale biomedical ontologies where exact label matching collapses to 0.33. It should be the default first baseline for any new matching pipeline. Second, BM25 is hyperparameter-robust: its performance varies by only 0.084 MRR across all 20 configurations tested, and the region $b \geq 0.75$, $k_1 \in [0.6, 2.4]$ consistently delivers near-optimal recall without per-dataset tuning. Third, word-level TF-IDF carries a hidden risk that this paper makes visible: bigram-only features or raised minimum document frequency collapse retrieval to near zero on

datasets where other methods work well, producing a performance span of 0.622 MRR across configurations—more than seven times the 0.084 span observed for BM25. A practitioner who deploys TF-IDF with a bigram configuration without validating it on development data may unknowingly cripple their pipeline’s recall.

The candidate reduction analysis adds a fourth, quantitative dimension to this picture. A top-50 candidate list compresses a target pool of up to 532,781 entities by over 99.99% while retaining Recall@50 above 0.91, making neural scoring and LLM-based verification tractable at scale. This compression is not a side effect of candidate generation—it is its primary purpose, and reporting it explicitly alongside recall metrics should become standard practice.

Three directions present the most promising extensions of this work. A direct comparison with dense retrieval methods—SBERT-based approximate nearest neighbour search and domain-fine-tuned biomedical encoders—would establish whether semantic embeddings can exceed the strong lexical baseline documented here, and at what additional cost. Extending the evaluation to richer text representations, including synonyms, definitions, and structural context, would establish how much the label-only design of this study underestimates the potential of each method. Finally, a downstream evaluation measuring end-to-end alignment F1 as a function of candidate generation quality would answer the question practitioners ultimately care about most: how much does the choice of first-stage retrieval method affect the final match set?

Data and Code Availability

The complete experimental code, configuration files, and result artefacts are publicly available at <https://github.com/swntsvt/kgcandgen> under the MIT License. All result files reported in this paper are additionally archived at <https://doi.org/10.5281/zenodo.19351110> to ensure long-term reproducibility. The OAEI benchmark datasets used in this study are publicly available through the OAEI website at <https://oaei.ontologymatching.org>.

Declarations

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Competing Interests

The authors declare that they have no competing interests.

Ethics Approval and Consent to Participate

Not applicable. This study does not involve human participants, human data, or animal subjects.

Consent for Publication

Not applicable.

Author Contributions

S.K.S. conceived the study, designed and implemented the experimental framework, conducted all experiments, analysed the results, and wrote the manuscript. S.G. and M.S. supervised the research, provided critical feedback on the experimental design, and reviewed and revised the manuscript. All authors read and approved the final manuscript.

References

- [1] Euzenat, J., Shvaiko, P.: *Ontology Matching: Second Edition*. Springer, Berlin, Heidelberg (2013). <https://doi.org/10.1007/978-3-642-38721-0>
- [2] Hertling, S., Paulheim, H.: The Knowledge Graph Track at OAEI – Gold Standards, Baselines, and the Golden Hammer Bias. In: *Extended Semantic Web Conference (ESWC)*. Lecture Notes in Computer Science, vol. 12123, pp. 343–359 (2020). https://doi.org/10.1007/978-3-030-49461-2_20 . Springer
- [3] Robertson, S., Zaragoza, H.: The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval* **3**(4), 333–389 (2009) <https://doi.org/10.1561/15000000019>
- [4] Totoian, M.-H., Mărginean, A., Blohm, P., Hussain, M.N.: HybridOM: Ontology Matching using Hybrid Search. In: *Proceedings of the 19th International Workshop on Ontology Matching Co-located with the 23rd International Semantic Web Conference*. CEUR Workshop Proceedings, vol. 3897, pp. 138–145 (2024). CEUR-WS.org. https://ceur-ws.org/Vol-3897/oaei2024_paper7.pdf
- [5] Hertling, S., Paulheim, H.: OLaLa: Ontology Matching with Large Language Models. In: *Proceedings of the 12th Knowledge Capture Conference 2023*, pp. 131–139 (2023). <https://doi.org/10.1145/3587259.3627571> . Association for Computing Machinery
- [6] Silva, M.C., Faria, D., Pesquita, C.: Matching Multiple Ontologies to Build a Knowledge Graph for Personalized Medicine. In: *The Semantic Web*, pp. 461–477 (2022). https://doi.org/10.1007/978-3-031-06981-9_27 . Springer International Publishing
- [7] Algergawy, A., Cheatham, M., Faria, D., Ferrara, A., Fundulaki, I., Gal, A., Groz, B., Harrow, I., Hertling, S., Jimenez-Ruiz, E., Karam, N., Köhler, J., Lewis, E., Meilicke, C., Molnár, G., Montanelli, S., Paulheim, H., Pesquita, C., Saveta, T., Shvaiko, P., Splendiani, A., Thiéblin, É., Trojahn, C., Hertling, S., Euzenat, J.: Results of the Ontology Alignment Evaluation Initiative 2018. In: *Proceedings of the 13th International Workshop on Ontology*

- Matching Co-located with the 17th International Semantic Web Conference. CEUR Workshop Proceedings, vol. 2288, pp. 73–129 (2018). CEUR-WS.org. https://ceur-ws.org/Vol-2288/oaei18_paper0.pdf
- [8] Abd Nikooie Pour, M., Algergawy, A., Blomqvist, E., Buche, P., Chen, J., Cotovio, P., Coulet, A., Cufi, J., Dong, H., Faria, D., Ferraz, L., Hertling, S., He, Y., Horrocks, I., Ibanescu, L., Jain, S., Jiménez-Ruiz, E., Karam, N., Kraus, F., Zhou, L.: Results of the Ontology Alignment Evaluation Initiative 2024. In: Proceedings of the 20th International Workshop on Ontology Matching Co-located with the 24th International Semantic Web Conference. CEUR Workshop Proceedings, vol. 4144 (2025). CEUR-WS.org. <https://ceur-ws.org/Vol-4144/>
 - [9] Faria, D., Balasubramani, B.S., Silva, M.C., Santos, E., Couto, F.M., Pesquita, C.: AgreementMakerLight. Semantic Web **16**(2), 413–436 (2025) <https://doi.org/10.3233/SW-233304>
 - [10] Jiménez-Ruiz, E., Cuenca Grau, B.: LogMap: Logic-Based and Scalable Ontology Matching. In: The Semantic Web – ISWC 2011. Lecture Notes in Computer Science, vol. 7031, pp. 273–288 (2011). https://doi.org/10.1007/978-3-642-25073-6_18 . Springer
 - [11] Christen, P.: A Survey of Indexing Techniques for Scalable Record Linkage and Deduplication. IEEE Transactions on Knowledge and Data Engineering **24**(9), 1537–1555 (2012) <https://doi.org/10.1109/TKDE.2011.127>
 - [12] Reimers, N., Gurevych, I.: Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, pp. 3982–3992 (2019). <https://doi.org/10.48550/arXiv.1908.10084> . Association for Computational Linguistics
 - [13] Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., Yih, W.-t.: Dense Passage Retrieval for Open-Domain Question Answering. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 6769–6781 (2020). <https://doi.org/10.18653/v1/2020.emnlp-main.550>
 - [14] He, Y., Chen, J., Antonyrajah, D., Horrocks, I.: BERTMap: A BERT-Based Ontology Alignment System. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 5684–5691 (2022). <https://doi.org/10.1609/aaai.v36i5.20527>
 - [15] Chen, J., Hertling, S., Zhapa-Camacho, F., Hohenecker, P., Paulheim, H., Horrocks, I.: BioML Challenge 2023: Biomedical Ontology Matching. In: Proceedings of the 18th International Workshop on Ontology Matching Co-located with the 22nd International Semantic Web Conference. CEUR Workshop Proceedings, vol. 3591, pp. 1–13 (2023). CEUR-WS.org. https://ceur-ws.org/Vol-3591/oaei2023_paper1.pdf

- [16] Babaei Giglou, H., D’Souza, J., Engel, F., Auer, S.: LLMs4OM: Matching Ontologies with Large Language Models. In: The Semantic Web: ESWC 2024, pp. 22–38 (2024). https://doi.org/10.1007/978-3-031-60635-9_2 . Springer
- [17] Taboada, M., Martínez, D., Arideh, M., Mosquera-Losada, M.R.: Ontology Matching with Large Language Models and Prioritized Depth-First Search. *Information Fusion* **114**, 102908 (2025) <https://doi.org/10.1016/j.inffus.2025.102908>
- [18] Song, Y., Chen, J., Schmidt, R.A.: GenOM: Ontology Matching with Description Generation and Large Language Model. In: Proceedings of the 20th International Workshop on Ontology Matching Co-located with the 24th International Semantic Web Conference. CEUR Workshop Proceedings, vol. 4144, pp. 26–38 (2025). CEUR-WS.org. <https://ceur-ws.org/Vol-4144/>
- [19] Nikolaev, F., Kotov, A.: Joint Word and Entity Embeddings for Entity Retrieval from a Knowledge Graph. In: Advances in Information Retrieval, pp. 141–155 (2020). https://doi.org/10.1007/978-3-030-45439-5_10 . Springer International Publishing
- [20] Kamphuis, C., Vries, A.P., Boytsov, L., Lin, J.: Which BM25 Do You Mean? A Large-Scale Reproducibility Study of Scoring Variants. In: Advances in Information Retrieval. Lecture Notes in Computer Science, vol. 12036, pp. 28–34 (2020). https://doi.org/10.1007/978-3-030-45442-5_4 . Springer International Publishing
- [21] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, É.: Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* **12**(85), 2825–2830 (2011)
- [22] Dokoozli, M.R.: BM25S: Fast and Flexible BM25 for Python. *Journal of Open Source Software* **8**(82), 4872 (2023) <https://doi.org/10.21105/joss.04872>
- [23] Hertling, S., Paulheim, H.: DBkWik: A Consolidated Knowledge Graph from Thousands of Wikis. In: 2018 IEEE International Conference on Big Knowledge (ICBK), pp. 17–24 (2018). <https://doi.org/10.1109/ICBK.2018.00011> . IEEE
- [24] Hofmann, A., Perchani, S., Portisch, J., Hertling, S., Paulheim, H.: DBkWik: Towards Knowledge Graph Creation from Thousands of Wikis. In: International Semantic Web Conference (Posters, Demos & Industry Tracks). CEUR Workshop Proceedings, vol. 1963, p. 2 (2017). CEUR-WS.org. <https://ceur-ws.org/Vol-1963/paper540.pdf>

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [bm25grid.pdf](#)
- [tfidfmrheatmapsublineartrue.pdf](#)
- [tfidfmrheatmapsublinearfalse.pdf](#)
- [bestmrrtrackbox.pdf](#)
- [bestmrrdumbbell.pdf](#)