Evaluating Retrieval Methods for Article→Topic Linking on CiNii Data Using Two Similarity Approaches

NII Internship Report

Vicente Lermanda (NII Research Intern)

1 Introduction

Topic labels in bibliographic databases make it much easier to search, filter, and analyze scholarly records, yet many large-scale resources lack such links. In particular, the CiNii bibliographic database[1] contains millions of articles without explicit connections to the research topics they address, and manual annotation at this scale is infeasible.

We pursue two aims, we evaluate methods for linking articles to topics so the literature can be explored by topic, and we outline a practical pipeline for creating those links and exporting them as RDF when needed. We compare two complementary similarity approaches—a *semantic* method using sentence-embedding retrieval and a *set-based* method using Jaccard/MinHash with LSH—and include a large language model (LLM) as a direct classification baseline.

Contributions. We provide (1) an empirical comparison of embedding-based and set-based retrieval for article→topic linking on a bibliographic dataset; (2) a simple, scalable pipeline design aligned with OECD FOS 2007 for writing back `(paper, dct:subject, fos:Topic)` triples; and (3) an LLM baseline that contextualizes the retrieval results.

2 Related Work

Text similarity, semantic and lexical, has been widely studied. On the semantic side, early neural embedding approaches such as word2vec [9] introduced dense representations that capture syntactic and semantic regularities; more recently, Sentence-BERT [13] extends this to sentence-level embeddings, enabling efficient semantic search and clustering. These embedding-based methods are the foundation of most modern semantic retrieval systems.

On the lexical side, efficient set similarity estimation has long relied on MinHash [3, 4], which preserves Jaccard similarity between sets through compact signatures. Combined with Locality-Sensitive Hashing [5], this allows sublinear-time retrieval of near-duplicates or highly overlapping documents. The textbook by Rajaraman & Ullman [12] provides a comprehensive treatment and serves as our primary theoretical reference for MinHash and LSH, establishing these methods as canonical tools for large-scale text mining.

Retrieval-assisted classification has been used both to create or refine labels and to improve decisions at inference. On the labeling side, neighbor-based methods retrieve similar examples in an embedding space and propagate or assign labels: graph/kNN label propagation and its variants [6, 14], as well as class-balanced kNN pseudo-labeling [2]. On the inference side, models retrieve external or exemplar evidence to boost predictions, e.g., retrieval-augmented classification for long-tail recognition [7] and retrieval-augmented zero-shot text classification [1]. These lines of work show that retrieval can supply supervision signals or decision-time context across domains. In our setting, we adopt the same principle but in a hybrid design:

[1] `https://cir.nii.ac.jp`

we pair semantic neighbor retrieval with Jaccard/set similarity via MinHash and LSH to scale candidate generation and provide complementary lexical evidence.

3 Methods

This section presents the retrieval methods used to support article→topic linking in a bibliographic dataset. We explore two complementary notions of similarity: (i) a semantic approach that ranks neighbors in a vector space, and (ii) a lexical approach that ranks neighbors by set overlap. The resulting ranked neighborhoods provide evidence for assigning topics. Implementation details are deferred to the Experimental Setup.

3.1 Embedding-based Similarity

Each document's text is represented by a fixed-length semantic embedding. Let $x \in \mathbb{R}^p$ denote the embedding of a query document and $X = \{x_i\}_{i=1}^N$ the corpus embeddings. We measure semantic proximity with cosine similarity

$$s_{\cos}(x, x_i) = \frac{x^\top x_i}{\|x\|_2 \, \|x_i\|_2}.$$

Nearest-neighbor retrieval ranks corpus items by decreasing $s_{\cos}$, forming a neighborhood of semantically related articles whose labels can be propagated or aggregated for topic assignment.

Indexing and retrieval For nearest-neighbor search over semantic embeddings, we use an approximate graph-based index, Hierarchical Navigable Small World (HNSW) [8]. At the scale considered here, the specific index has limited impact on results; we adopt HNSW as a widely used, state-of-the-art choice that offers a strong recall–latency trade-off. Queries return a ranked list of the k most similar documents under cosine similarity. HNSW has near-linear build time (excluding embedding build time), sublinear queries, and linear memory (vectors + a small number of links per item).

3.2 Set-based (Jaccard/MinHash) Similarity

Each document d is mapped to a set of *shingles* $S(d)$ derived from its text (same as n-grams). Lexical relatedness between two documents d and u is quantified by the Jaccard similarity

$$J\big(S(d), S(u)\big) = \frac{|S(d) \cap S(u)|}{|S(d) \cup S(u)|}.$$

To scale set similarity, we summarize each $S(d)$ with a compact *MinHash* signature, which preserves Jaccard similarity in expectation.

Scaling with LSH. Locality-Sensitive Hashing (LSH) indexes these signatures so that items with high estimated Jaccard are likely to co-locate in shared hash buckets. At query time, we probe the buckets indicated by the query's signature to obtain a small candidate pool of lexically similar documents. This procedure is fast and sublinear, but it is inherently threshold-oriented: it surfaces candidates rather than a global ranking.

Approximate top-k via LSH Forest. To produce an approximate top-k list, we adopt an LSH Forest, which organizes signatures into multiple prefix trees built from their hash digests. The forest explores paths that share longer prefixes with the query first (higher expected Jaccard), then broadens as needed until k candidates are accumulated, producing an approximate ranking. This approach has linear signature creation, near-linear index build, and sublinear queries that return a small candidate set; memory is linear in the number of items.

Other usages. For analysis and data quality control, we may also inspect high-cardinality buckets from a plain LSH (thresholded) index to surface clusters of near-duplicates or systematic artifacts (e.g., "Errata", empty fields, language mismatches).

3.3 LLM Baseline for Direct Topic Assignment

We include a direct text-classification baseline: given a document's title and abstract, a large language model predicts a single topic from a fixed schema. Unlike the retrieval methods that infer labels from nearest neighbors, this approach produces a label directly from the text.

4 Evaluation

4.1 Topic Scheme: OECD FOS and the Derived Ontology

We organize topics using the OECD *Fields of Science and Technology (FOS 2007)* scheme, which groups research into six major fields and finer 42 divisions ("subtopics") [10, 11]. Figure 1 shows a simplified graph of the six *major topics* as they appear in our vocabulary; Figure 2 zooms in on *Natural Sciences*, illustrating representative links to its subtopics. Both visualizations are pruned for readability while preserving the hierarchical relations used in our experiments.

To use FOS in our setting, we built a lightweight *ontology*: a single `skos:ConceptScheme` (`fos:FOS2007`) with `skos:Concept` nodes for each major field and subtopic, hierarchical links via `skos:broader`/`skos:narrower` (and `skos:topConceptOf`), and English labels via `skos:prefLabel`.

We assign URIs in the namespace `https://example.org/oecd-fos/2007/`. This ontology defines the closed set of labels used in evaluation (major topic and subtopic) and serves as the target for write-back triples `(paper, dct:subject, fos:Topic)` when enriching the dataset as RDF.

Example (abbrev.).

```
@prefix fos:  <https://example.org/oecd-fos/2007/> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .

fos:FOS2007 a skos:ConceptScheme .
  dct:title "OECD Fields of Science and Technology (FOS 2007)"@en ;
  dct:creator "Organisation for Economic Co-operation and Development"@en ;
  dct:description "Major fields and divisions from the OECD..."@en .


fos:Natural_Sciences a skos:Concept ;
  skos:inScheme fos:FOS2007 ;
  skos:topConceptOf fos:FOS2007 ;
  skos:prefLabel "Natural Sciences"@en ;
  skos:definition "Covers mathematics; computer and..."@en .


fos:Computer_and_Information_Sciences a skos:Concept ;
  skos:inScheme fos:FOS2007 ;
  skos:broader fos:Natural_Sciences ;
  skos:prefLabel "Computer and Information Sciences"@en ;
  skos:definition "Computer science, information science..."@en .
```

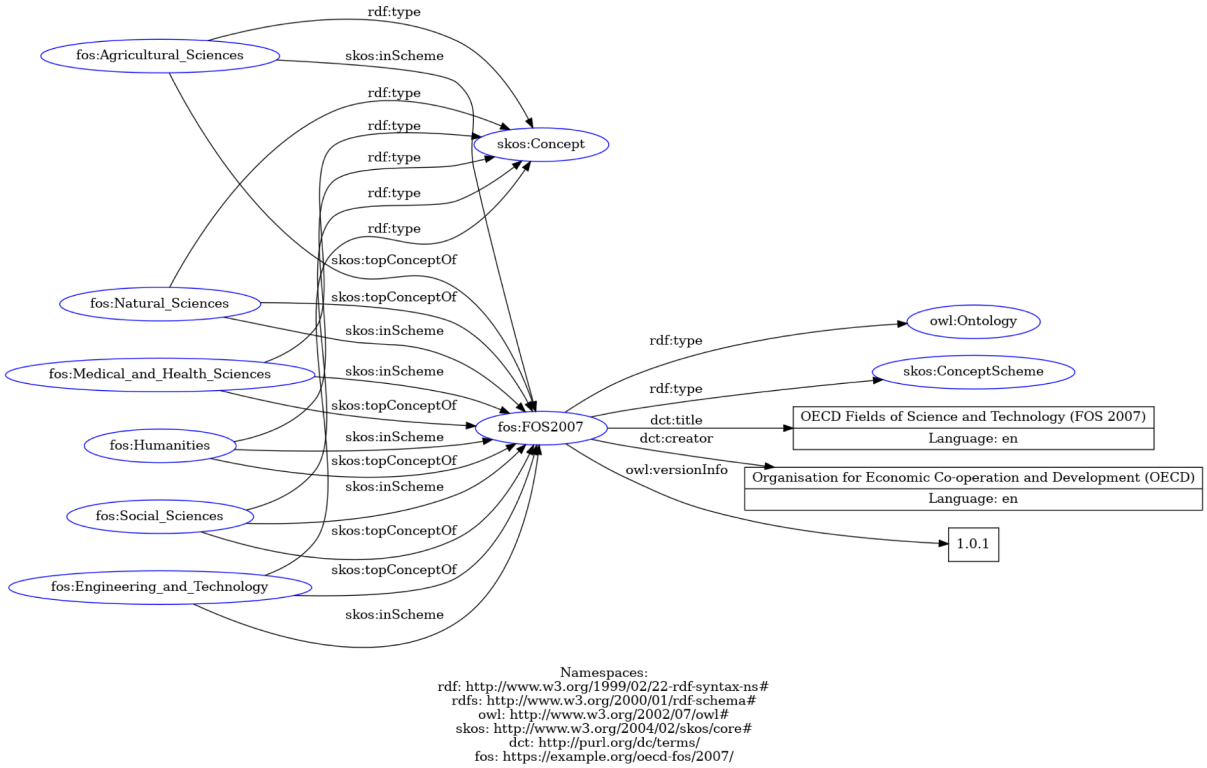

Figure 1: OECD FOS 2007 major topics (*simplified*). Nodes are the six top-level fields in our SKOS/OWL vocabulary; edges indicate hierarchical relations (e.g., `skos:topConceptOf`).

Data. We evaluate on an English scientific publications benchmark derived from CiNii. The benchmark file contains one row per document with fields

`uri`, `major_topic`, `subtopic`, `confidence`.

From the 174 total rows, we exclude entries that we were not able to label, marked with 'confidence=skip', resulting in $N = 150$ unique documents (one unique `uri` per row). Labels follow the OECD FOS 2007 scheme at two granularities: a *major topic* (6 distinct values in this set) and a *subtopic* (28 distinct values out of 42 possibles). Class frequencies are imbalanced toward *Natural Sciences* at the major-topic level (72/150), with the remaining documents distributed across *Engineering and Technology* (30), *Medical and Health Sciences* (16), *Agricultural Sciences* (15), *Social Sciences* (11), and *Humanities* (6). At the subtopic level, the most frequent labels include *Physical Sciences* (23), *Computer and Information Sciences* (13), and *Biological Sciences* (12).

Each document's text used for retrieval is the concatenation of its title and abstract obtained from CiNii for the given `uri`; the benchmark CSV itself stores labels and confidence only. Confidence scores (`low`, `medium`, `high`) reflect labeling certainty and are retained for analysis, but all experiments filter out rows tagged `skip`. The confidence distribution is approximately balanced (50 high / 51 medium / 49 low).

Preprocessing. We normalize the input texts—lowercasing, removing special symbols, and trimming extra whitespace—and then tokenize them, whether they encode it as dense embeddings or as MinHash signatures.

For the embedding pipeline, we pass the normalized *title+abstract* directly to the encoder.

For the set-based pipeline, we had the option to tokenize based on k-shingles or words.

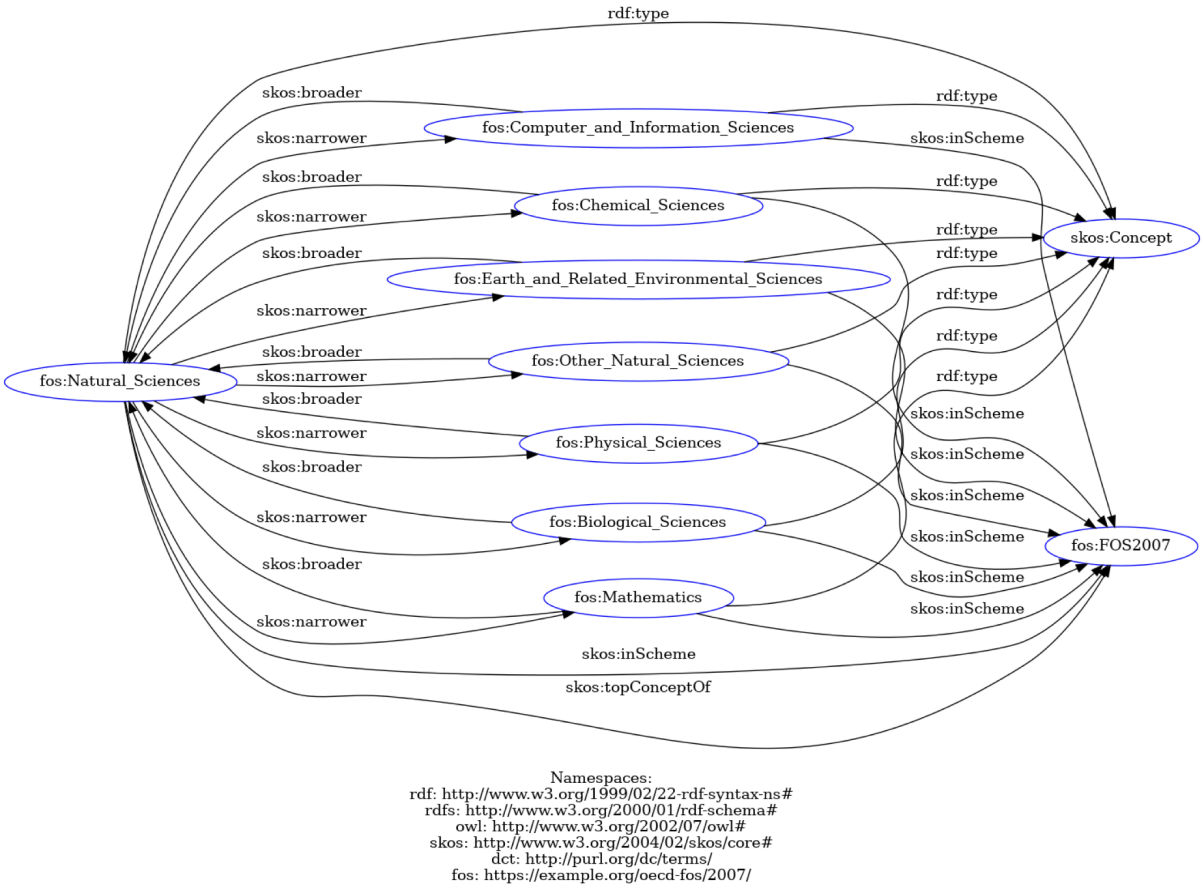

Figure 2: *Natural Sciences* focus view (*simplified*). Subset of subtopics and their hierarchical links under the Natural Sciences branch, shown for readability.

Nonetheless, the latter perform consistently worse in preliminary results so it ended up being discarded. We construct k-shingles with $k = 5$ over character-based tokens.

Methods. Embedding-based. We use `sentence-transformers` all-MiniLM-L6-v2 (sentence-BERT) to obtain d-dimensional vectors with $d = 384$. Similarity is cosine. Indexing is done with FAISS HNSWFlat, configured with arguments $m = 31$, `ef_construction` $= 200$, `ef_search` $= 50$ and top-k retrieval with $k = 5$.

Set-based. We use `datasketch` to build MinHash signatures with `num_perm` $= 128$, an `LSHForest` with $l = 8$ prefix trees, and a Jaccard threshold of 0.5. For each query we retrieve up to $k = 5$ candidates directly from the forest.

LLM. We evaluate a topic-classification baseline using OpenAI API `gpt-5-mini` model with structured outputs constrained to the OECD FOS topic set.

Implementation details. Experiments ran on a MacBook Air M1 (8 GB RAM), macOS Ventura. Python 3.13 was used with the following main packages: faiss-cpu 1.11.0, `sentence-transformers` 4.1.0, `datasketch` 1.6.5, `pandas` 2.2.3, `openai` 1.81.0. Extra details (exact configs, commits) are provided in the repository [2].

[2] `https://github.com/vlermandac/cinii-topic-linking`

4.2 Evaluation Protocol

We evaluate as it follows:

Information retrieval (IR) protocol. For a given label space (either majors or subtopics), let $\ell \in \{\ell_{\text{maj}}, \ell_{\text{sub}}\}$ denote the label function for the current evaluation. For each document $d \in \mathcal{D}$ used as a query, we define its relevant set

$$\mathcal{R}(d) = \{\, u \in \mathcal{D} \setminus \{d\} \;:\; \ell(u) = \ell(d) \,\}.$$

Each method m returns a ranked list $L_m(d,k) = [u_1, \ldots, u_k]$ of the top-k neighbors for d; if d appears in its own result, we remove it. Queries with $|\mathcal{R}(d)| = 0$ are skipped to avoid undefined recall.

IR metrics. Given $L_m(d,k)$, define binary relevance $y_i = 1$ *iff* the retrieved item u_i has the same gold-standard label as d. We report the standard top-k metrics per query and then average over all evaluated queries:

Precision@k

$$\text{P@}k(d) = \frac{1}{k} \sum_{i=1}^{k} y_i.$$

Recall@k With $R(d) = |\mathcal{R}(d)|$,

$$\text{R@}k(d) = \begin{cases} \frac{1}{R(d)} \sum_{i=1}^{k} y_i, & R(d) > 0, \\ \text{undefined (skip)}, & R(d) = 0. \end{cases}$$

AP@k Let $\text{P@}i(d)$ be precision at i and $H = \sum_{i=1}^{k} y_i$,

$$\text{AP@}k(d) = \begin{cases} \frac{1}{\min(H,k)} \sum_{i=1}^{k} \text{P@}i(d)\, y_i, & H > 0, \\ 0, & H = 0. \end{cases}$$

nDCG@k With binary gains,

$$\text{DCG@}k(d) = \sum_{i=1}^{k} \frac{y_i}{\log_2(i+1)}, \quad \text{IDCG@}k(d) = \sum_{i=1}^{\min(k,R(d))} \frac{1}{\log_2(i+1)},$$

$$\text{nDCG@}k(d) = \begin{cases} \frac{\text{DCG@}k(d)}{\text{IDCG@}k(d)}, & R(d) > 0, \\ \text{undefined (skip)}, & R(d) = 0. \end{cases}$$

LLM evaluation We treat the LLM as a standard multiclass classifier and report *Accuracy*, *Macro-Precision*, *Macro-Recall*, and *Macro-F1*. These metrics are not directly comparable to the IR@k scores used for retrieval methods (they correspond roughly to an @1 view), but they provide a useful point of reference for our results.

5 Results and Discussion

This section presents the results of our experiments and discusses their implications.

Table 1: IR results at cutoff 5 for major topics ($\ell = \ell_{\text{maj}}$).

Method	P@5	R@5	mAP@5	nDCG@5
HNSW	0.560	0.111	0.716	0.576
LSH	0.332	0.041	0.452	0.325

Table 2: IR results at cutoff 5 for subtopics ($\ell = \ell_{\text{sub}}$).

Method	P@5	R@5	mAP@5	nDCG@5
HNSW	0.327	0.230	0.534	0.366
LSH	0.076	0.045	0.153	0.075

Table 3: GPT-5-mini classification results for major topics ($\ell = \ell_{\text{maj}}$) and subtopics ($\ell = \ell_{\text{sub}}$).

Metric	Major topics	Subtopics
Accuracy	0.7400	0.6467
Macro Precision	0.7881	0.4933
Macro Recall	0.8072	0.5174
Macro F1	0.7701	0.4788

HNSW over sentence embeddings clearly outperforms MinHash/LSH across all IR metrics for both major topics and subtopics (Tables 1–2), reflecting embedding's strength at capturing paraphrase/topical similarity beyond surface overlap. The lower R@5 for major topics vs. subtopics is a denominator effect: major classes have more relevant items, so five correct hits cover a smaller fraction of the relevant set.

The GPT-5-mini results (Table 3) are not directly comparable to retrieval metrics (classification Accuracy/Macro P/R/F1 vs. ranking at $k = 5$). Still, they indicate that a strong text classifier performs well without neighbors. However, full-corpus LLM labeling is often impractical at scale (privacy/compliance risks, high cost/latency) for big databases, in this case having >20M records.

Despite weaker ranking quality, MinHash/LSH remains valuable as fast, sublinear candidate generation and for deduplication/record linkage.

6 Proposed Pipeline for Article → Topic Linking

We propose a simple, scalable pipeline that combines fast lexical similarity with semantic retrieval, aligned to the OECD FOS 2007 ontology.

1. **Ingest & normalize.** Parse CiNii records, normalize encodings, and standardize titles/abstracts (lowercasing, whitespace, basic cleanup).

2. **MinHash/LSH checks and cleanup.** Build character k-shingles and MinHash signatures; index with an LSH Forest to surface clusters. Use these buckets to (i) flag and correct language–tag inconsistencies, (ii) identify empty/boilerplate/error patterns for removal, and (iii) spot near-duplicates—collapsing exact/near-exact cases and annotating candidate merges.

3. **Representations.**
   (a) *Golden exemplars.* Maintain a small set of high-quality, curated representative papers per topic. Use them as (i) additional kNN anchors during retrieval and (ii) a source of centroid embeddings to stabilize topic representations.
   (b) *Sentence embeddings (articles & topics).* Encode *title+abstract* with `all-MiniLM-L6-v2`. For topics, encode the English labels and short definitions to obtain topic vectors; optionally refine each topic vector by averaging embeddings of its "golden" representative papers.
   (c) *Lexical signatures (articles).* Retain the MinHash signatures from Step 2 for fast overlap retrieval.
4. **Indexes.**
   (a) *Semantic ANN.* Build a FAISS HNSWFlat index over article embeddings.
   (b) Keep the MinHash LSH over article signatures for sublinear lexical candidate search.
5. **Candidate generation.**
   (a) *Article→Topic (direct).* Using the article's embedding x, retrieve the top-k *topics* by cosine similarity against the precomputed topic vectors $\{t_j\}$. These are the initial candidates.
   (b) *Article→Article (neighbor-assisted).* Retrieve *article* neighbors for x: (i) semantic via HNSW over article embeddings; (ii) lexical via LSH over MinHash signatures. Collect the distinct topic labels that appear among these neighbors and add them to the candidate set.
6. **Scoring & Fusion.** For each candidate topic, compute:
   (a) *Semantic score:* max/mean of cosine similarities to topic vector and to top semantic neighbors with that topic.
   (b) *Lexical score:* max/mean Jaccard among LSH neighbors with that topic.
   (c) *Consensus signals:* how much the neighbors agree on the same topic (label agreement), how "pure" the top-k neighbors are for that topic, and how far the best topic is from the runner-up (score margin).

   Combine signals with a simple weighted sum to obtain a unified score.
7. **Decision Policy.** Emit top-k topics with a *confidence* derived from fused score, agreement, and margin. Use two thresholds:
   (a) τ_{auto}: auto-accept if confidence $\geq \tau_{\text{auto}}$.
   (b) τ_{review}: flag for review if $\tau_{\text{review}} \leq$ confidence $< \tau_{\text{auto}}$; otherwise discard.

   This way, we link high-confidence cases automatically and send the uncertain ones to a small review queue.
8. **Human/LLM Triage (optional).** Present borderline items in a minimal TUI: show title+abstract, top candidates, neighbor snippets, and scores. Allow quick confirm/edit. Optionally query an LLM for *just* the queued cases to suggest a tie-break.
9. **Write Back to the KG.** For accepted assignments, emit triples

   (`paper`, `dct:subject`, `fos:Topic`)

   plus metadata: `prov:wasDerivedFrom` (source signals), numeric confidence, timestamp, and pipeline version/params. Use `skos:Concept` URIs from the FOS 2007 scheme.

10. **Monitoring & Refresh.** Track acceptance/disagreement rate, drift by topic, and check precision. Periodically:

    (a) refresh embeddings (batch or streaming),

    (b) rebuild/merge HNSW layers,

    (c) re-tune LSH parameters (signature length, trees, probing depth),

    (d) re-calibrate fusion weights

7 Conclusions and Future Work

Our results suggest a clear split in roles. Semantic retrieval (HNSW over sentence embeddings) works best on its own for finding topic-wise similar papers. MinHash/LSH is lighter and faster, and it's very useful for finding likely candidates, catching near-duplicates, and keeping the search manageable on large datasets. Put simply: use lexical hashing to narrow the field, then use semantic methods to sort the best matches. The LLM baseline is helpful as a reference, but it is better used for spot checks or difficult cases, not for labeling the entire corpus, because of privacy, cost, and speed concerns.

Overall, the design is hybrid: use MinHash/LSH for fast, broad candidate generation and data checks, and use HNSW to retrieve strong semantic neighbors; then combine these signals in a simple fusion score to pick the final topic. An LLM is reserved for ambiguous cases only. Finally we save the results back as RDF: `(paper, dct:subject, fos:Topic)` triples with a confidence score so the links are easy to trace and update.

Looking ahead, we have to implement the full end-to-end pipeline at scale, extend the benchmark to Japanese titles/abstracts to test multilingual retrieval, and grow/curate the dataset by adding more labels, filtering by confidence, and analyzing failure cases. We will also need expert validation across fields to refine ambiguous instances and measure agreement, and we will fine-tune method settings and parameters to improve the accuracy–latency trade-off.

References

[1] Tassallah Abdullahi, Ritambhara Singh, and Carsten Eickhoff. Retrieval augmented zero-shot text classification. In *Proceedings of the 2024 ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR)*. ACM, 2024.

[2] Nicholas Botzer, David Vazquez, Tim Weninger, and Issam Laradji. TK-KNN: A balanced distance-based pseudo labeling approach for semi-supervised intent classification. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6472–6484, Singapore, December 2023. Association for Computational Linguistics.

[3] Andrei Z. Broder. On the resemblance and containment of documents. *Compression and Complexity of Sequences*, pages 21–29, 1997.

[4] Andrei Z. Broder, Moses Charikar, Alan Frieze, and Michael Mitzenmacher. Min-wise independent permutations. In *STOC*, pages 327–336, 2000.

[5] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *STOC*, pages 604–613, 1998.

[6] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5070–5079, June 2019.

[7] Alexander Long, Wei Yin, Thalaiyasingam Ajanthan, Vu Nguyen, Pulak Purkait, Ravi Garg, Alan Blair, Chunhua Shen, and Anton van den Hengel. Retrieval augmented classification for long-tail visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6959–6969, June 2022.

[8] Yu. A. Malkov and D. A. Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs, 2018.

[9] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at ICLR*, 2013.

[10] OECD. *Frascati Manual 2015: Guidelines for Collecting and Reporting Data on Research and Experimental Development*. OECD Publishing, Paris, 2015.

[11] Organisation for Economic Co-operation and Development. Revised field of science and technology (fos) classification in the frascati manual. Technical Report DSTI/EAS/STP/NESTI(2006)19/FINAL, OECD, 2007. Unclassified OECD document.

[12] Anand Rajaraman and Jeffrey D. Ullman. *Mining of Massive Datasets*. Cambridge University Press, 2011.

[13] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of EMNLP*, pages 3982–3992, 2019.

[14] Pau Rodríguez, Issam Laradji, Alexandre Drouin, and Alexandre Lacoste. Embedding propagation: Smoother manifold for few-shot classification. In *Computer Vision – ECCV 2020*, volume 12371 of *Lecture Notes in Computer Science*, pages 121–138, Cham, 2020. Springer.