TECHNICAL SPECIFICATION DOCUMENT

BioPAT

Biomedical Patent-to-Article Retrieval Benchmark

A Comprehensive Developer Handoff Document
for Building the First Dedicated Biomedical Patent-Literature Benchmark

Version 1.0
January 2026

CONFIDENTIAL

Table of Contents

Note: Update the Table of Contents after opening in Word by right-clicking and selecting 'Update Field'.

1. Executive Summary

1.1 Project Purpose

This document provides complete technical specifications for developing BioPAT (Biomedical Patent-to-Article Retrieval), a benchmark dataset for evaluating information retrieval systems that find relevant scientific literature given biomedical patent claims. This benchmark addresses a critical gap: while benchmarks exist for patent-to-patent retrieval (CLEF-IP, DAPFAM) and general literature retrieval (BEIR), no benchmark evaluates the cross-domain task of retrieving scientific papers that may anticipate or invalidate patent claims.

1.2 Business Value

Prior art search is a multi-billion dollar industry. Pharmaceutical companies conduct extensive literature searches before and during patent prosecution. Examiners at patent offices worldwide must identify relevant scientific publications. Legal teams need comprehensive prior art for litigation. Yet there is no standard benchmark to evaluate or compare systems for this task. BioPAT fills this gap, enabling: development of better retrieval systems, fair comparison of commercial and academic tools, and advancement of the field through reproducible research.

1.3 Scope and Deliverables

Phase	Primary Deliverable	Timeline
Phase 1	Minimum Viable Benchmark (2,000 queries, citation-based ground truth)	3-4 weeks
Phase 2	Enhanced Ground Truth (examiner-validated, graded relevance)	4-6 weeks
Phase 3	Full Benchmark with Comprehensive Baselines	4-6 weeks
Phase 4	Publication and Public Release	2-4 weeks

Total estimated timeline: 13-20 weeks (approximately 4-5 months) from project initiation to public release.

1.4 Key Design Decisions

API-First Architecture: Where possible, we use APIs rather than bulk downloads to reduce storage requirements, simplify infrastructure, and enable incremental updates. Bulk downloads are used only when APIs are insufficient (e.g., full claim text extraction).

Graded Relevance: Unlike binary benchmarks, BioPAT uses a 4-tier relevance scale (0-3) reflecting real-world prior art assessment: novelty-destroying references vs. background citations.

Claim-Level Granularity: Patent claims, not entire patents, are the unit of novelty. Each independent claim becomes a separate query, enabling fine-grained evaluation.

BEIR Compatibility: Output format follows BEIR conventions for immediate compatibility with existing retrieval evaluation frameworks.

Reproducibility: All data sources are public. All processing code is open-source. Random seeds are fixed. Results are deterministically reproducible.

2. Background and Motivation

2.1 The Patent Examination Process

When a patent application is filed, an examiner must determine whether the claimed invention is novel (35 U.S.C. §102) and non-obvious (35 U.S.C. §103) over prior art. Prior art includes any publicly available information before the filing date, encompassing both earlier patents AND scientific literature. The examiner searches databases to find relevant references, then issues Office Actions citing specific documents against specific claims.

This process is critical because: (1) Pharmaceutical patents can be worth billions of dollars, (2) Invalid patents harm competition and innovation, (3) Missed prior art leads to patents that should never have been granted, (4) The volume of scientific literature makes comprehensive manual search impossible.

2.2 The Benchmark Gap

Current information retrieval benchmarks fail to address the patent-to-literature retrieval task:

Benchmark	Query Type	Corpus Type	Limitation for Our Task
CLEF-IP	Full patent application	Patents only	No scientific literature
DAPFAM	Patent claims	Patents only	No scientific literature
PatentMatch	Patent claims	Patents only	No scientific literature
TREC-CHEM	Chemical queries	Mixed (dated)	Chemistry only, 2009-2011
BEIR	Natural questions	Various	No patent queries
BioPAT (ours)	Patent claims	Biomedical literature	Fills the gap

2.3 Why Biomedical?

The biomedical domain is ideal for this benchmark for multiple compelling reasons:

High Non-Patent Literature (NPL) Citation Rate: Biomedical patents cite scientific literature at 3-5x the rate of other domains. Marx and Fuegi (2020) found pharmaceutical patents average 8.2 NPL citations compared to 2.1 for mechanical patents. This provides rich ground truth data.

Clear Domain Boundaries: International Patent Classification (IPC) codes cleanly delineate the domain: A61 covers medical and veterinary science, C07 covers organic chemistry including pharmaceuticals, and C12 covers biochemistry and genetic engineering.

Structured Literature Corpus: PubMed contains over 36 million articles with rich metadata including MeSH terms, chemical indexing, and gene annotations. OpenAlex provides open access to this data with concept tagging.

Commercial Importance: Pharmaceutical patent litigation involves stakes of billions of dollars. Accurate prior art search directly impacts patent validity, licensing negotiations, and freedom-to-operate analyses.

Scientific Importance: The intersection of patents and scientific literature is understudied. This benchmark enables research into cross-domain retrieval, technical language understanding, and knowledge transfer between patents and papers.

2.4 Design Principles

BioPAT is designed according to the following principles, derived from analysis of existing benchmarks and consultation with patent professionals:

Principle 1 - Claim-Level Granularity: Patent novelty is assessed at the claim level, not the patent level. A patent may have 20 claims, each covering different aspects. A reference that anticipates claim 1 may be irrelevant to claim 15. Therefore, each independent claim should be a separate query. This follows PANORAMA and PatentMatch but differs from CLEF-IP which uses full patent documents.

Principle 2 - Graded Relevance: Binary relevance (relevant vs. not relevant) is insufficient for prior art. A paper that completely anticipates a claim (novelty-destroying) is categorically different from one that provides useful background. We adopt a 4-tier scale: 3 = novelty-destroying, 2 = highly relevant for obviousness, 1 = somewhat relevant, 0 = not relevant.

Principle 3 - Temporal Validity: Only documents published BEFORE the patent priority date are valid prior art. This constraint must be strictly enforced. Any paper published after the priority date, no matter how relevant, cannot be considered prior art and must be excluded from positive relevance judgments.

Principle 4 - Domain Stratification: Following DAPFAM, we partition evaluation by IPC subclass to measure cross-subdomain retrieval difficulty. IN-domain queries (where query and relevant docs share IPC codes) are typically easier than OUT-domain queries (no IPC overlap). This reveals whether systems generalize across subdomains.

Principle 5 - Reproducibility: All data sources must be publicly accessible without licensing fees. All processing code must be open source. All random seeds must be fixed and documented. Any researcher should be able to reproduce the benchmark from scratch.

3. Data Sources and Access Strategy

3.1 Overview of Data Architecture

The benchmark requires three types of data: (1) patent data including claims text and metadata, (2) scientific literature with titles, abstracts, and metadata, and (3) ground truth links between patents and papers. Our strategy prioritizes API access to minimize storage and enable incremental updates, falling back to bulk downloads only when necessary.

Data Type	Primary Source	Access Method	Rationale
Patent Metadata	PatentsView API	REST API	Free, fast filtering by IPC
Patent Claims Text	USPTO Bulk Data	Selective Download	API truncates; need full text
Patent-Paper Links	Reliance on Science	Bulk Download	Pre-computed, high quality
Examiner Citations	Office Action Dataset	Bulk Download	Gold standard relevance
Paper Metadata	OpenAlex API	REST API	Free, comprehensive, fast
MeSH Enrichment	PubMed E-utilities	REST API	Authoritative medical terms

3.2 Patent Data Sources

3.2.1 PatentsView API (Primary for Metadata)

PatentsView is a free API provided by the USPTO that enables programmatic access to patent data. It is the recommended primary source for patent metadata due to its speed, filtering capabilities, and zero cost.

Attribute	Details
Base URL	https://search.patentsview.org/api/v1/patent/
Rate Limit	45 requests/minute with API key; 10 without
Authentication	API key via X-Api-Key header (free registration)
Coverage	All US patents 1976-present; applications 2001-present
Key Fields	patent_id, patent_date, patent_title, patent_abstract, claims, ipc_current, cpc_current, cited_patents, application_citations
Limitations	Claims text may be truncated for long patents; NPL citations not fully parsed

Usage Pattern: Use PatentsView for initial patent discovery (filtering by IPC codes) and metadata retrieval. For patents selected for the benchmark, verify claim completeness and download full XML if truncated.

```
# Example API query for biomedical patents
POST https://search.patentsview.org/api/v1/patent/
Headers: X-Api-Key: YOUR_KEY
Body: {
  "q": {"_or": [
    {"_begins": {"ipc.ipc_class": "A61"}},
    {"_begins": {"ipc.ipc_class": "C07"}},
```

```
    {"_begins": {"ipc.ipc_class": "C12"}}
  ]},
  "f": ["patent_id", "patent_date", "patent_title", "claims"],
  "o": {"size": 100, "after": "cursor_token"}
}
```

3.2.2 USPTO Bulk Data (For Full Claims Text)

When PatentsView truncates claim text (common for patents with many claims), full text must be obtained from USPTO bulk data. The bulk data provides complete XML files for all patents.

Attribute	Details
Base URL	https://bulkdata.uspto.gov/data/patent/grant/redbook/fulltext/
Format	XML files, one per week, compressed (.zip)
Size	Approximately 50-100 MB per weekly file; full archive is several TB
Coverage	All US patents from 1976-present
Strategy	Download only files containing patents in benchmark; parse specific patents from XML

Implementation Note: Maintain a mapping of patent_id to bulk file. Download files on-demand when full claims are needed. Cache parsed results to avoid re-downloading.

3.2.3 USPTO Office Action Research Dataset (Gold Standard)

This dataset contains structured data from 4.4 million Office Actions issued by USPTO examiners between 2008 and 2017. It is the gold standard for relevance because it contains examiner judgments about which references anticipate or render obvious which claims.

Attribute	Details
URL	https://www.uspto.gov/ip-policy/economic-research/research-datasets/office-action-research-dataset-patents
Format	CSV and JSON files, approximately 15 GB compressed
Coverage	4.4 million Office Actions (2008-2017)
Key Fields	application_id, rejection_type (102/103/112), cited_ref_type (patent/npl), cited_ref_text, rejected_claims
Critical Value	Examiner-validated relevance: §102 citations are novelty-destroying (highest relevance), §103 citations support obviousness (high relevance)

This dataset is essential for Phase 2. A §102 rejection means the examiner determined that the cited reference anticipates the claim. This is the strongest possible relevance signal. A §103 rejection means the cited reference, alone or in combination, renders the claim obvious.

3.2.4 Reliance on Science Dataset (Patent-Paper Links)

The Reliance on Science (RoS) dataset provides pre-computed links between US patents and scientific papers. It was created by parsing NPL citations from patents and matching them to papers in PubMed and OpenAlex.

Attribute	Details

URL	https://zenodo.org/records/7996195
Format	CSV files, approximately 2 GB compressed
Coverage	Patent-to-paper citations through 2022
Key Fields	patent_id, openalex_id, pmid (where available), confidence (1-10), examiner_applicant flag
Quality Metrics	87.6% recall, 99.4% precision on linking accuracy (Marx & Fuegi 2020)
Citation	Marx, M., & Fuegi, A. (2020). Reliance on science: Worldwide front-page patent citations to scientific articles. Strategic Management Journal.

The RoS dataset is the foundation for Phase 1 ground truth. The examiner_applicant field distinguishes between citations added by examiners (stronger relevance signal) vs. applicants (may be background). The confidence score reflects the quality of the citation match.

3.3 Literature Data Sources

3.3.1 OpenAlex API (Primary)

OpenAlex is a free, open catalog of the global research system including over 250 million works. It supersedes Microsoft Academic Graph and provides comprehensive coverage of biomedical literature.

Attribute	Details
Base URL	https://api.openalex.org/works
Rate Limit	100,000 requests/day (polite pool with email in User-Agent); higher with API key
Authentication	None required; email in User-Agent header recommended
Coverage	250M+ works including all PubMed, Crossref, ORCID
Key Fields	id, title, abstract_inverted_index, publication_date, doi, pmid, concepts, cited_by_count
Biomedical Filter	concepts.id: C71924100 (Medicine), C86803240 (Biology), C185592680 (Chemistry)

OpenAlex is the primary source for paper metadata. For papers linked via RoS, use the OpenAlex ID directly. The abstract is stored as an inverted index for space efficiency and must be reconstructed.

```
# Example: Fetch paper by OpenAlex ID
GET https://api.openalex.org/works/W2123456789
Headers: User-Agent: mailto:your@email.com

# Example: Batch fetch multiple papers
GET https://api.openalex.org/works?filter=openalex_id:W1|W2|W3

# Abstract reconstruction from inverted index:
abstract_inverted_index = {"The": [0], "study": [1], "shows": [2], ...}
# Sort by position, join words
```

3.3.2 PubMed E-utilities (MeSH Enrichment)

PubMed provides authoritative Medical Subject Heading (MeSH) terms for biomedical articles. MeSH terms enable sophisticated domain stratification and semantic analysis.

Attribute	Details
Base URL	https://eutils.ncbi.nlm.nih.gov/entrez/eutils/
Rate Limit	10 requests/second with API key; 3 without
Authentication	API key via api_key parameter (free registration at NCBI)
Key Endpoints	efetch.fcgi (retrieve records), esearch.fcgi (search)
Key Fields	MeshHeading (descriptor + qualifier), ChemicalList, KeywordList

Use E-utilities to enrich papers that have PMIDs with MeSH terms. This is optional for Phase 1 but valuable for Phase 3 analysis. MeSH provides controlled vocabulary that can improve domain stratification beyond IPC codes alone.

3.4 Data Access Strategy by Phase

Data Source	Phase 1	Phase 2	Phase 3
Reliance on Science	Required (bulk)	Required	Required
PatentsView API	Required	Required	Required
OpenAlex API	Required	Required	Required
Office Action Dataset	Not needed	Required (bulk)	Required
USPTO Bulk XML	On-demand	Required	Required
PubMed E-utilities	Optional	Optional	Recommended

4. Technical Architecture

4.1 System Overview

The BioPAT system consists of five major components: (1) Data Ingestion layer for downloading and caching source data, (2) Processing Pipeline for transforming and linking data, (3) Ground Truth Engine for computing relevance judgments, (4) Benchmark Assembly module for creating final outputs, and (5) Evaluation Framework for running baselines. These components are designed for modularity, allowing independent development, testing, and maintenance.

4.2 Technology Stack

Component	Technology	Rationale
Language	Python 3.11+	Ecosystem support for NLP, IR, data science; type hints for maintainability
HTTP Client	httpx (async)	Async support for parallel API calls; HTTP/2; connection pooling
Data Processing	Polars	Faster than pandas; lazy evaluation; better memory efficiency for large datasets
File Format	Parquet	Columnar storage; compression; fast queries; schema enforcement
Output Format	JSON Lines	BEIR compatibility; human readable; streaming processing
Database	DuckDB	Analytical queries on Parquet; SQL interface; embedded (no server)
Caching	diskcache	Persistent cache for API responses; TTL support; thread-safe
Configuration	YAML + Pydantic	Human-editable config; validation; type safety
Testing	pytest	Standard; fixtures; parametrization; coverage reporting
IR Evaluation	pytrec_eval	Standard TREC metrics; graded relevance; statistical tests
Embeddings	sentence-transformers	Pre-trained models; easy fine-tuning; GPU acceleration
Vector Search	FAISS	Fast similarity search; GPU support; billion-scale

4.3 Directory Structure

The project follows a standard Python package structure with clear separation of concerns:

```
biopat/
├── pyproject.toml          # Project metadata and dependencies
├── README.md               # Project documentation
├── configs/
│   ├── default.yaml        # Default configuration
│   ├── phase1.yaml         # Phase 1 specific settings
│   └── phase2.yaml         # Phase 2 specific settings
```

```
├── src/
│   └── biopat/
│       ├── __init__.py
│       ├── config.py           # Pydantic config models
│       ├── ingestion/          # Data download modules
│       │   ├── patentsview.py  # PatentsView API client
│       │   ├── openalex.py     # OpenAlex API client
│       │   ├── pubmed.py       # PubMed E-utilities client
│       │   ├── ros.py          # Reliance on Science loader
│       │   └── office_action.py# Office Action parser
│       ├── processing/         # Data transformation
│       │   ├── patents.py      # Patent filtering and claim extraction
│       │   ├── papers.py       # Paper metadata processing
│       │   └── linking.py      # Citation linking logic
│       ├── groundtruth/        # Relevance computation
│       │   ├── relevance.py    # Relevance tier assignment
│       │   └── temporal.py     # Temporal constraint validation
│       ├── benchmark/          # Output generation
│       │   ├── sampling.py     # Query and corpus sampling
│       │   ├── splits.py       # Train/dev/test splitting
│       │   └── beir_format.py  # BEIR output formatting
│       └── evaluation/         # Baseline runners
│           ├── bm25.py         # BM25 baseline
│           ├── dense.py        # Dense retrieval baselines
│           └── metrics.py      # Metric computation
├── data/                       # Data directory (git-ignored)
│   ├── raw/                    # Downloaded source data
│   ├── cache/                  # API response cache
│   ├── processed/              # Intermediate Parquet files
│   └── benchmark/              # Final BEIR-format output
├── scripts/                    # CLI entry points
│   ├── download.py         # Download all data sources
│   ├── process.py          # Run processing pipeline
│   ├── build_benchmark.py  # Build final benchmark
│   └── evaluate.py         # Run baseline evaluation
└── tests/                      # Test suite
    ├── test_ingestion.py
    ├── test_processing.py
    └── test_groundtruth.py
```

4.4 Data Flow Pipeline

The pipeline processes data through five stages, each producing intermediate outputs that can be inspected and validated:

Stage 1: Data Ingestion

Download and cache all source data. API responses are cached to disk to enable reruns without re-downloading.

```
Input:  Configuration (IPC filters, date ranges)
Output: data/raw/ros.parquet        # Reliance on Science citations
```

```
        data/raw/office_actions/    # Office Action files (Phase 2)
        data/cache/patentsview/     # Cached API responses
        data/cache/openalex/        # Cached API responses
```

Stage 2: Patent Selection

Filter patents to biomedical domain, verify citation availability, and extract claim text.

```
Input:  ros.parquet, PatentsView API
Output: data/processed/patents.parquet
Fields: patent_id, priority_date, ipc_codes, claims[], title, abstract
```

Stage 3: Literature Assembly

Retrieve metadata for all papers linked via citations. Add negative samples.

```
Input:  Citation links, OpenAlex API
Output: data/processed/papers.parquet
Fields: paper_id, title, abstract, publication_date, concepts[], pmid
```

Stage 4: Ground Truth Construction

Compute relevance scores based on citation source, type, and confidence.

```
Input:  patents.parquet, papers.parquet, ros.parquet, office_actions/
Output: data/processed/qrels.parquet
Fields: query_id, doc_id, relevance (0-3), evidence_source
```

Stage 5: Benchmark Assembly

Create final BEIR-compatible output with train/dev/test splits.

```
Input:  patents.parquet, papers.parquet, qrels.parquet
Output: data/benchmark/
        ├── corpus.jsonl    # All papers
        ├── queries.jsonl   # Patent claims
        └── qrels/
            ├── train.tsv
            ├── dev.tsv
            └── test.tsv
```

5. Data Models and Schemas

5.1 Core Entity Schemas

All data is stored in strongly-typed Parquet files with defined schemas. This section specifies the exact fields, types, and semantics for each entity.

5.1.1 Patent Schema

```
{
  "patent_id": "US10123456B2",        // String: USPTO publication number
  "application_id": "US201615123456", // String: Application number
  "priority_date": "2016-03-15",      // Date: Earliest priority (for temporal filter)
  "filing_date": "2016-03-15",        // Date: US filing date
  "grant_date": "2018-11-06",         // Date: Grant date (null if application)
  "title": "Therapeutic antibody...", // String: Patent title
  "abstract": "The invention...",     // String: Patent abstract
  "claims": [                         // Array: All claims
    {
      "claim_number": 1,              // Int: Claim number
      "claim_text": "A method...",    // String: Full claim text
      "claim_type": "independent",    // Enum: independent | dependent
      "depends_on": null              // Int | null: Parent claim number
    }
  ],
  "ipc_codes": ["A61K39/395"],        // Array[String]: IPC classifications
  "cpc_codes": ["A61K39/39533"],      // Array[String]: CPC classifications
  "assignees": ["Pharma Inc."],       // Array[String]: Assignee names
  "inventors": ["John Doe"]           // Array[String]: Inventor names
}
```

5.1.2 Paper Schema

```
{
  "paper_id": "W2123456789",          // String: OpenAlex ID (primary key)
  "pmid": "12345678",                 // String | null: PubMed ID
  "doi": "10.1000/example",           // String | null: DOI
  "title": "Novel inhibitors...",     // String: Paper title
  "abstract": "We describe...",       // String: Abstract text
  "publication_date": "2015-06-15",   // Date: Publication date
  "journal": "J Med Chem",            // String | null: Journal name
  "authors": ["Smith J", "Doe J"],    // Array[String]: Author names
  "concepts": [                       // Array: OpenAlex concepts
    {
      "id": "C71924100",
      "name": "Medicine",
      "score": 0.85
    }
  ],
  "mesh_terms": [                     // Array | null: PubMed MeSH terms
    {
      "descriptor": "Protein Kinase Inhibitors",
```

```
      "qualifier": "pharmacology"
    }
  ],
  "cited_by_count": 156              // Int: Citation count
}
```

5.1.3 Citation Schema

```
{
  "citation_id": "US10123456B2_W2123456789", // String: Composite key
  "patent_id": "US10123456B2",               // String: FK to patent
  "paper_id": "W2123456789",                 // String: FK to paper
  "source": "examiner",                      // Enum: examiner | applicant | both
  "location": "front_page",                  // Enum: front_page | body | both
  "confidence": 9,                           // Int 1-10: RoS match confidence
  "rejection_type": "102",                   // String | null: 102 | 103 (from OA)
  "cited_claim_numbers": [1, 3, 5]           // Array[Int] | null: Claims citing ref
}
```

5.1.4 Query Schema (Benchmark Output)

```
{
  "query_id": "US10123456B2-c1",      // String: Patent + claim number
  "patent_id": "US10123456B2",        // String: Source patent
  "claim_number": 1,                  // Int: Claim number
  "text": "A method comprising...",   // String: Query text (claim)
  "priority_date": "2016-03-15",      // Date: For temporal filtering info
  "ipc_codes": ["A61K39/395"],        // Array: For domain analysis
  "domain": "A61K"                    // String: IPC3 for stratification
}
```

5.1.5 Relevance Judgment Schema (qrel)

```
{
  "query_id": "US10123456B2-c1",      // String: Query identifier
  "doc_id": "W2123456789",            // String: Document identifier
  "relevance": 3,                     // Int 0-3: Graded relevance
  "label": "novelty_destroying",      // String: Human-readable label
  "evidence": {                       // Object: Provenance
    "source": "office_action",        // Enum: office_action | ros_examiner |
ros_applicant
    "rejection_type": "102",          // String | null
    "confidence": 9                   // Int | null
  }
}
```

5.2 Relevance Tier Definitions

The 4-tier relevance scale is central to BioPAT. Each tier has specific criteria:

Score	Label	Definition and Assignment Criteria

3	novelty_destroying	Examiner cited under §102 (anticipation). The paper alone discloses every element of the claim. This is the strongest relevance: the paper would invalidate the claim if predates it.
2	highly_relevant	Examiner cited under §103 (obviousness) OR RoS examiner citation with confidence >= 9. The paper is highly relevant and would contribute to an obviousness rejection.
1	relevant	RoS examiner citation with confidence 7-8 OR applicant citation with confidence >= 8. The paper is relevant as background or state of the art.
0	not_relevant	No citation link exists, OR confidence < 7, OR paper published after patent priority date. Default for unjudged documents.

Note: Documents with publication_date >= priority_date must ALWAYS receive relevance 0, regardless of citation presence. This is a hard constraint reflecting patent law.

5.3 BEIR Output Format

The final benchmark output follows BEIR format for compatibility with existing evaluation tools:

corpus.jsonl

```
{"_id": "W2123456789", "title": "Novel kinase inhibitors for cancer therapy", "text": "We describe the discovery and optimization of a series of kinase inhibitors..."}
{"_id": "W2234567890", "title": "Structure-activity relationships in BTK inhibitors", "text": "A systematic exploration of BTK inhibitor modifications..."}
```

queries.jsonl

```
{"_id": "US10123456B2-c1", "text": "A method of treating cancer in a subject comprising administering to the subject a therapeutically effective amount of a compound of Formula I..."}
{"_id": "US10123456B2-c5", "text": "The method of claim 1, wherein the cancer is selected from the group consisting of breast cancer, lung cancer, and colorectal cancer."}
```

qrels/test.tsv

```
query_id   doc_id score
US10123456B2-c1   W2123456789   3
US10123456B2-c1   W2234567890   2
US10123456B2-c1   W2345678901   1
```

6. Phase 1: Minimum Viable Benchmark

6.1 Phase 1 Objectives

Phase 1 delivers a functional benchmark using citation-based ground truth from the Reliance on Science dataset. The goal is speed to a working benchmark that can validate the overall approach before investing in more complex ground truth construction.

Attribute	Phase 1 Target
Query Patents	2,000 biomedical patents with high-confidence NPL citations
Queries	~5,000 independent claims from selected patents
Literature Corpus	~100,000 papers (cited + hard negatives)
Ground Truth	Binary relevance from RoS citations (confidence >= 8)
Splits	70/15/15 train/dev/test, stratified by IPC subclass
Baselines	BM25 only
Timeline	3-4 weeks

6.2 Phase 1 Implementation Tasks

Task 1.1: Environment Setup (2 days)

Create project structure, install dependencies, configure API credentials, set up development environment.

Deliverables: Working development environment, pyproject.toml with all dependencies, config files with API keys (gitignored), basic logging and error handling.

```
# Required dependencies
pip install httpx[http2] polars pyarrow duckdb pyyaml pydantic
pip install sentence-transformers pytrec-eval-terrier beir
pip install pytest pytest-cov diskcache tqdm

# API credentials needed
PATENTSVIEW_API_KEY=xxx    # Request at patentsview.org
NCBI_API_KEY=xxx           # Register at ncbi.nlm.nih.gov
```

Task 1.2: Download Reliance on Science Dataset (1 day)

Download and parse the RoS dataset from Zenodo. This provides pre-computed patent-to-paper links.

Deliverables: data/raw/ros.parquet with columns: patent_id, openalex_id, pmid, confidence, examiner_applicant

```
# Download
wget https://zenodo.org/records/7996195/files/_pcs_oa.csv.gz
```

```
# Parse and filter
import polars as pl
ros = pl.read_csv('_pcs_oa.csv.gz')
ros = ros.filter(pl.col('confidence') >= 8)  # High confidence only
ros.write_parquet('data/raw/ros.parquet')
```

Task 1.3: Identify Biomedical Patents (3 days)

Filter RoS patents to biomedical domain using PatentsView API. Verify citation availability.

Deliverables: List of ~10,000 candidate patent IDs with biomedical IPC codes and sufficient citations.

Algorithm:

```
1. Get unique patent IDs from RoS (examiner citations, confidence >= 8)
2. Query PatentsView for IPC codes
3. Filter to patents with IPC starting A61, C07, or C12
4. Filter to patents with >= 3 high-confidence paper citations
5. Sample 2,000 patents stratified by IPC subclass
```

Implementation Note: Use async HTTP client to parallelize PatentsView queries. Cache responses to avoid re-fetching. Implement exponential backoff for rate limiting.

Task 1.4: Extract Patent Claims (3 days)

Retrieve full claim text for selected patents. Handle truncation in API responses.

Deliverables: data/processed/patents.parquet with complete claim text for all selected patents.

Algorithm:

```
1. Query PatentsView for claims field
2. Check if claims appear truncated (heuristic: ends mid-word or mid-sentence)
3. For truncated patents, identify containing bulk XML file
4. Download bulk XML file and parse specific patents
5. Validate: each patent should have >= 1 independent claim
```

Implementation Note: Create a claim parser that handles both PatentsView JSON and USPTO XML formats. Identify claim type (independent vs. dependent) by parsing text for claim-back references.

Task 1.5: Build Literature Corpus (3 days)

Retrieve paper metadata from OpenAlex for all cited papers. Add hard negative samples.

Deliverables: data/processed/papers.parquet with ~100K papers.

Algorithm:

```
1. Get all paper IDs from RoS citations to selected patents
2. Query OpenAlex API for metadata (batch endpoint, 50 IDs per request)
3. Reconstruct abstracts from inverted index
4. Add hard negatives:
   a. For each cited paper, find 10 papers from same journal, different year
```

```
   b. Add papers with similar concepts but not cited
5. Validate: paper_date < patent_priority_date for positives
```

Implementation Note: Abstract reconstruction requires sorting the inverted index by position. Cache OpenAlex responses. For missing PMIDs, query PubMed by DOI.

Task 1.6: Construct Ground Truth (2 days)

Create query-document relevance judgments. Apply temporal constraint.

Deliverables: data/processed/qrels.parquet with binary relevance judgments.

Algorithm:

```
1. For each patent, create one query per independent claim
2. For each query-paper pair:
   a. If paper_date >= patent_priority_date: relevance = 0 (HARD RULE)
   b. Else if RoS confidence >= 8: relevance = 1
   c. Else: relevance = 0
3. Validate: no temporal violations in positive judgments
```

Note: Phase 1 uses binary relevance (0/1). Phase 2 will upgrade to graded relevance (0-3) using Office Action data.

Task 1.7: Create Splits and Format Output (2 days)

Split data into train/dev/test. Format as BEIR-compatible JSON Lines.

Deliverables: data/benchmark/ directory with corpus.jsonl, queries.jsonl, qrels/{train,dev,test}.tsv

Algorithm:

```
1. Stratified split by IPC3 (first 4 chars of IPC code)
   - Train: 70%, Dev: 15%, Test: 15%
2. Ensure no query appears in multiple splits
3. Corpus includes all papers (no split-specific corpora)
4. Format corpus.jsonl: {_id, title, text}
5. Format queries.jsonl: {_id, text}
6. Format qrels: TSV with query_id, doc_id, score
```

Task 1.8: Run BM25 Baseline (2 days)

Implement and evaluate BM25 retrieval as sanity check and baseline.

Deliverables: Baseline results report with NDCG@10, Recall@100 on dev and test sets.

```
from beir import util
from beir.datasets.data_loader import GenericDataLoader
from beir.retrieval.search.lexical import BM25Search as BM25
from beir.retrieval.evaluation import EvaluateRetrieval

# Load benchmark
corpus, queries, qrels = GenericDataLoader('data/benchmark/').load(split='test')
```

```
# Index and search
model = BM25(index_name='biopat')
retriever = EvaluateRetrieval(model)
results = retriever.retrieve(corpus, queries)

# Evaluate
ndcg, _map, recall, precision = retriever.evaluate(qrels, results, [10, 100])
```

6.3 Phase 1 Acceptance Criteria

Phase 1 is complete when all the following criteria are met:

1. Benchmark contains >= 2,000 query patents and >= 4,000 queries (claims)
2. Corpus contains >= 50,000 documents with abstracts
3. Zero temporal violations in positive relevance judgments
4. BM25 achieves Recall@100 > 0.3 (sanity check)
5. Pipeline is reproducible with single command
6. All code is tested with >= 80% coverage
7. Documentation includes data statistics and known limitations

7. Phase 2: Examiner-Grade Ground Truth

7.1 Phase 2 Objectives

Phase 2 enhances ground truth quality using USPTO Office Action data. Examiner citations under §102 (anticipation) and §103 (obviousness) provide validated relevance judgments. This phase also introduces graded relevance and claim-level mapping.

Attribute	Phase 1	Phase 2
Ground Truth Source	RoS citations only	RoS + Office Actions
Relevance Scale	Binary (0/1)	Graded (0/1/2/3)
Rejection Type Info	Not available	§102/§103 distinguished
Claim Mapping	Patent-level	Claim-level from OA
Query Count	~5,000	8,000-12,000
Timeline	-	4-6 weeks

7.2 Phase 2 Implementation Tasks

Task 2.1: Download Office Action Dataset (2 days)

Download and parse the USPTO Office Action Research Dataset. This is approximately 15GB compressed.

Deliverables: data/raw/office_actions/ directory with parsed rejection data.

```
# Key tables to extract:
# - office_actions: OA metadata (application_id, mailing_date)
# - rejections: Rejection details (rejection_type, claims)
# - citations: Cited references (ref_type, ref_text)
```

Task 2.2: Parse NPL Citations from Office Actions (5 days)

Extract and link non-patent literature citations to papers in the corpus. This is the most complex task due to citation format variability.

Deliverables: Linked NPL citations with >75% linking accuracy.

Algorithm:

```
NPL citations appear in multiple formats:
- Structured: 'PMID: 12345678' or 'DOI: 10.1021/...'
- Semi-structured: 'Smith et al., J Med Chem, 2015, 58, 1234-1240'
- Unstructured: Free-text bibliographic references

Linking strategy:
1. Regex for explicit PMID/DOI
2. Parse with GROBID for bibliographic fields
3. Query OpenAlex by title + authors
```

```
4. Fuzzy match on normalized title
5. Manual review sample for accuracy estimation
```

Implementation Note: Use multiple parsing strategies in cascade. Log unlinked citations for analysis. Prioritize precision over recall for the final linking.

Task 2.3: Map Rejections to Claims (3 days)

Parse claim numbers from rejection text. Create claim-level queries.

Deliverables: Claim-citation mapping for all linked NPL rejections.

```
Office Actions specify rejected claims:
- 'Claims 1-5 and 12 are rejected under 35 USC 102(a)(1)'
- 'Claim 1 is rejected under 35 USC 103'

Parsing requirements:
1. Extract claim numbers (handle ranges: '1-5', lists: '1, 3, and 5')
2. Extract rejection type (102/103)
3. Link to cited references
4. Create one ground truth entry per (claim, paper) pair
```

Task 2.4: Implement Graded Relevance (2 days)

Assign 4-tier relevance scores based on citation source and type.

Deliverables: Updated qrels with graded relevance (0-3).

```
def assign_relevance(citation):
    if citation.paper_date >= citation.patent_priority_date:
        return 0  # Temporal violation (HARD RULE)
    elif citation.rejection_type == '102':
        return 3  # Novelty-destroying (examiner validated)
    elif citation.rejection_type == '103':
        return 2  # Obviousness (examiner validated)
    elif citation.source == 'examiner' and citation.confidence >= 9:
        return 2  # High-confidence examiner citation
    elif citation.source == 'examiner' and citation.confidence >= 7:
        return 1  # Medium-confidence examiner
    elif citation.source == 'applicant' and citation.confidence >= 8:
        return 1  # High-confidence applicant
    else:
        return 0  # Low confidence or no link
```

Task 2.5: Expand and Rebalance Corpus (3 days)

Add papers from Office Actions not in Phase 1 corpus. Improve hard negative sampling.

Deliverables: Expanded corpus with ~200K papers including improved hard negatives.

Hard Negative Strategies:

```
1. BM25 negatives: Papers with high BM25 score but not relevant
2. Concept negatives: Papers with same OpenAlex concepts, not cited
```

```
3. Journal negatives: Papers from same journals as positives
4. Temporal negatives: Relevant papers published after priority date
   (useful for training models to respect temporal constraint)
```

Task 2.6: Domain Stratification Analysis (2 days)

Implement IN-domain vs OUT-domain evaluation following DAPFAM methodology.

Deliverables: Split qrels by domain type; analysis of performance by IPC subclass.

```
def classify_domain(query_ipc3, doc_concepts):
    '''
    IN-domain: Query and relevant doc share IPC3
    OUT-domain: No IPC3 overlap
    '''
    # Map OpenAlex concepts to IPC3 codes
    doc_ipc3 = map_concepts_to_ipc(doc_concepts)
    if query_ipc3 & doc_ipc3:
        return 'IN'
    else:
        return 'OUT'
```

7.3 Phase 2 Acceptance Criteria

8. Graded relevance implemented with clear tier definitions
9. NPL citation linking achieves >75% accuracy on manual sample
10. At least 500 queries have §102 (novelty-destroying) ground truth
11. IN-domain and OUT-domain splits created and documented
12. Updated BM25 baseline with NDCG metrics (graded relevance)

8. Phase 3: Comprehensive Baselines

8.1 Phase 3 Objectives

Phase 3 runs comprehensive baseline evaluations, performs ablation studies, and conducts error analysis. The goal is to characterize benchmark difficulty and establish reference performance levels.

8.2 Baseline Models

Model	Type	Rationale for Inclusion
BM25	Lexical	Strong sparse baseline; often competitive in technical domains
Contriever	Dense	State-of-art general dense retriever; unsupervised pretraining
SPECTER2	Dense	Scientific paper embeddings; citation-trained
PubMedBERT	Dense	Biomedical domain-specific; trained on PubMed
PaECTER	Dense	Patent-specific embeddings; if available
GTR-T5-XL	Dense	Large-scale dense retriever; good zero-shot
BM25 + CE	Hybrid	BM25 retrieval + cross-encoder reranking
ColBERT	Late Interaction	Token-level matching; good for long documents

8.3 Phase 3 Implementation Tasks

Task 3.1: Embedding Infrastructure (3 days)

Set up embedding generation and vector search for dense models.

```
from sentence_transformers import SentenceTransformer
import faiss

# Generate embeddings
model = SentenceTransformer('allenai/specter2')
doc_embeddings = model.encode(corpus_texts, show_progress_bar=True)

# Build FAISS index
index = faiss.IndexFlatIP(768)  # Inner product for cosine
faiss.normalize_L2(doc_embeddings)  # Normalize for cosine
index.add(doc_embeddings)
```

Task 3.2: Run All Baselines (5 days)

Execute retrieval for all models. Save results in TREC format.

Deliverables: results/ directory with metrics and run files for all baselines.

Task 3.3: Hybrid and Reranking Methods (3 days)

Implement BM25 + dense fusion and cross-encoder reranking.

```
# Reciprocal Rank Fusion
def rrf(results_list, k=60):
    scores = defaultdict(float)
    for results in results_list:
        for rank, (doc_id, _) in enumerate(results):
            scores[doc_id] += 1.0 / (k + rank + 1)
    return sorted(scores.items(), key=lambda x: -x[1])

# Cross-encoder reranking
from sentence_transformers import CrossEncoder
reranker = CrossEncoder('cross-encoder/ms-marco-MiniLM-L-12-v2')
# Rerank top-100 BM25 results
```

Task 3.4: Ablation Studies (3 days)

Analyze performance across different conditions.

Ablation Dimensions:

```
1. Query representation: claim only vs claim + patent abstract
2. Document representation: title + abstract vs full text
3. Domain: IN-domain vs OUT-domain
4. Temporal: Recent patents (2015+) vs older
5. IPC subclass: A61K vs C12N vs A61B performance
```

Task 3.5: Error Analysis (3 days)

Manual analysis of retrieval failures to characterize benchmark difficulty.

Deliverables: Error analysis report categorizing failure modes.

```
Failure categories to identify:
- Vocabulary mismatch (patent jargon vs scientific terminology)
- Abstraction level (claim too specific/general)
- Cross-domain (relevant paper in different subfield)
- Semantic gap (same concept, different phrasing)
- False negative (missing ground truth)
```

8.4 Evaluation Metrics

Metric	Definition	Interpretation
NDCG@k	Normalized Discounted Cumulative Gain at rank k	Primary metric; rewards graded relevance and ranking
Recall@k	Proportion of relevant docs in top k	Coverage metric; critical for legal applications
MAP	Mean Average Precision	Overall ranking quality across recall levels
MRR	Mean Reciprocal Rank	Position of first relevant document
P@k	Precision at rank k	Proportion of top k that are relevant

9. Phase 4: Publication and Release

9.1 Phase 4 Objectives

Phase 4 prepares the benchmark for public release and academic publication. This includes documentation, licensing, hosting, and paper writing.

9.2 Release Components

9.2.1 HuggingFace Dataset

Host the benchmark on HuggingFace Hub for easy access via the datasets library.

```
# Usage after release:
from datasets import load_dataset
biopat = load_dataset('org/biopat')

# Structure:
biopat/
├── corpus/
├── queries/
└── qrels/
```

9.2.2 Zenodo Archive

Archive on Zenodo for DOI and long-term preservation.

Contents: Complete dataset, processing code, baseline results, documentation.

License: CC-BY-NC-SA 4.0 (allows research use, requires attribution and share-alike).

9.2.3 GitHub Repository

Open-source code repository with evaluation scripts and baselines.

Contents: Processing pipeline, evaluation framework, baseline implementations, issue tracker.

9.2.4 Documentation

Comprehensive documentation following Datasheets for Datasets framework:

```
1. Motivation: Why was the dataset created?
2. Composition: What does it contain?
3. Collection: How was data collected?
4. Preprocessing: What preprocessing was applied?
5. Uses: Intended uses and limitations
6. Distribution: How is it distributed?
7. Maintenance: Who maintains it?
```

9.3 Research Paper

Draft paper for submission to relevant venue (SIGIR, EMNLP, ACL, or NAACL).

Target Length: 8 pages + references (ACL format).

Outline:

```
1. Introduction - Gap in benchmarks, contributions
2. Related Work - Patent IR, cross-domain retrieval, biomedical IR
3. Dataset Construction - Sources, methodology, quality control
4. Benchmark Design - Tasks, metrics, splits
5. Experiments - Baselines, results tables, analysis
6. Analysis - Ablations, error analysis, difficulty characterization
7. Conclusion - Summary, limitations, future work
```

10. Risk Assessment and Mitigation

10.1 Technical Risks

Risk	Likelihood	Impact	Mitigation
API rate limits	Medium	Schedule delay	Caching, parallel accounts, bulk fallback
NPL parsing fails	High	Reduced ground truth	Multiple parsers, accept lower recall
Insufficient §102 citations	Medium	Weak gold standard	Expand IPC scope, use §103 as silver
Claims truncated	High	Incomplete queries	Bulk XML download fallback
Temporal violations	Medium	Invalid benchmark	Strict validation, automated tests

10.2 Contingency Plans

If Phase 1 exceeds 4 weeks: Reduce query count to 1,000; defer hard negative mining to Phase 2.

If Office Action parsing proves too difficult: Proceed with RoS-only ground truth; document limitation.

If API access becomes restricted: Pivot fully to bulk downloads; add 2 weeks for infrastructure.

If baseline results are anomalous: Manual inspection of sample queries; validate ground truth quality.

11. Appendices

Appendix A: IPC Code Reference

Code	Description	Examples
A61K	Preparations for medical, dental, or toilet purposes	Drug formulations, vaccines
A61P	Specific therapeutic activity of compounds	Anticancer, antibacterial
A61B	Diagnosis; surgery; identification	Medical devices, imaging
C07D	Heterocyclic compounds	Small molecule drugs
C07K	Peptides	Antibodies, therapeutic proteins
C12N	Microorganisms; enzymes; genetic engineering	Cell therapy, gene editing
C12Q	Measuring or testing involving enzymes	Diagnostic assays, PCR
G01N	Investigating materials by their properties	Clinical diagnostics

Appendix B: API Quick Reference

PatentsView API

```
Base: https://search.patentsview.org/api/v1/
Auth: X-Api-Key header
Rate: 45 req/min (with key)
Docs: https://patentsview.org/apis/api-endpoints
```

OpenAlex API

```
Base: https://api.openalex.org/
Auth: None (email in User-Agent recommended)
Rate: 100K/day polite pool
Docs: https://docs.openalex.org/
```

PubMed E-utilities

```
Base: https://eutils.ncbi.nlm.nih.gov/entrez/eutils/
Auth: api_key parameter
Rate: 10 req/sec (with key)
Docs: https://www.ncbi.nlm.nih.gov/books/NBK25501/
```

Appendix C: Glossary

IPC: International Patent Classification. Hierarchical system for classifying patents by technology.

NPL: Non-Patent Literature. Scientific papers, books, and other publications cited in patents.

Prior Art: Any publicly available information before a patent's priority date that is relevant to the claimed invention.

Priority Date: The earliest filing date that establishes when the invention was disclosed; determines the temporal cutoff for prior art.

§102: US patent law section covering novelty. A §102 rejection means the cited reference anticipates (fully discloses) the claim.

§103: US patent law section covering obviousness. A §103 rejection means the claim would have been obvious given the cited references.

Office Action: Official communication from a patent examiner during prosecution, typically containing rejections and cited references.

BEIR: Benchmarking IR. Standard format and framework for information retrieval evaluation.

NDCG: Normalized Discounted Cumulative Gain. Metric that accounts for graded relevance and position in ranking.

qrels: Query relevance judgments. File mapping query IDs to relevant document IDs with relevance scores.

END OF DOCUMENT

BioPAT Technical Specification v1.0

For questions or clarifications, contact the project lead.