**Figure Caption Extraction**

**Overview**

This system extracts structured metadata from scientific papers in PubMed Central (PMC), focusing on:

* Title
* Abstract
* Figure captions
* Figure image URLs (if any)
* Named entities (e.g., genes) from captions using BERN2

It supports both batch processing (via CLI or files) and API-based access, and is deployed using Docker for portability.

**Purpose**

To design and implement a production-ready, extensible system that:

* Ingests a list of PMCIDs (optionally PMIDs in future).
* Extracts metadata including title, abstract, and all figure captions from scientific papers.
* Identifies biomedical named entities (e.g., genes, diseases) from figure captions using **BERN2**.
* Stores this data in a configurable backend (default: DuckDB).
* Makes data accessible through a secure API and CLI.

**High-Level Architecture**

[CLI / API]

│

▼

[Input Handler] ── Reads PMCID list (file or API payload)

│

▼

[PMC Fetcher] ── Uses BioC PMC API to get article structure & figure captions

│

▼

[Entity Extractor] ── Uses local BERN2 model to extract biomedical entities from captions

│

▼

[Data Formatter] ── Converts extracted data to JSON and/or CSV

│

▼

[Storage Engine] ── Stores structured data (default: DuckDB)

│

▼

[REST API Server] ── Serves queried data, password/API key protected

**1. Input Layer**

* Accepts a list of PMC IDs via:
  + Command-line (e.g., --ids ids.txt)
  + API upload (to be extended)

**2. Extraction Layer**

* Uses the [BioC-PMC API](https://www.ncbi.nlm.nih.gov/research/bionlp/APIs/BioC-PMC/) to retrieve structured XML for each PMC ID.
* Extracts:
  + Paper title
  + Abstract
  + List of figure captions
  + Figure URLs (if any)

**3. Entity Recognition Layer**

* Uses a **local BERN2 instance** to extract biomedical named entities from each figure caption.
* This happens locally with no dependency on PubTator.

**4. Storage Layer**

* Extracted metadata and entities are saved into:
  + **DuckDB** database (default)
  + **JSON** or **CSV** files for output and inspection

**5. API Layer**

* Optional Flask/FastAPI layer (planned or present) to:
  + Accept new PMC ID submissions
  + Serve filtered results
  + Provide download endpoints (CSV/JSON)
  + Protect access with password or API key

**6. Configuration**

* Managed via CLI flags or environment variables:
  + --file, --format
  + Database backend
  + Logging verbosity
  + Entity extraction method (default: BERN2)

**Key Components**

| **Components** | **Description** |
| --- | --- |
| CLI Runner | Accepts PMCID list as input file, runs extraction, saves output. |
| PMC Fetcher | Queries [BioC PMC API](https://www.ncbi.nlm.nih.gov/research/bionlp/APIs/BioC-PMC/) for title, abstract, and figure captions. |
| BERN2 Entity Extractor | Uses a local Python script that integrates with BERN2 (via subprocess or local HTTP server) to identify entities in captions. |
| Storage Backend | Saves all extracted data into DuckDB by default (modular for future backends like PostgreSQL or cloud DBs). |
| REST API (FastAPI) | Exposes endpoints for uploading IDs, querying stored data, and downloading JSON/CSV. |
| Dockerized Deployment | System is fully containerized and deployable with Docker. Optional DNS override included for PubMed API reachability. |

**Deployment Diagram**

┌──────────────────────────────┐

│ Client (User/Admin) │

└────────────┬─────────────────┘

│ REST/CLI

▼

┌───────────────┐

│ Ingestion │

└───────────────┘

│

▼

┌────────────────┐

│ PMC API Fetch │ <─ External BioC API

└────────────────┘

│

▼

┌──────────────────┐

│ BERN2 NER Model │ <─ Local Script or Server

└──────────────────┘

│

▼

┌────────────────┐

│ DuckDB Engine │

└────────────────┘

│

▼

┌────────────────────┐

│ FastAPI Server │

└────────────────────┘

**Build image**:

docker build -t pubmed-parser .

**Run batch job**:

docker run --rm -v $(pwd):/app pubmed-parser

**Optional: Use Google DNS if needed**:

docker run --rm --dns=8.8.8.8 -v $(pwd):/app pubmed-parser

**Dependencies and Justifications**

| **Dependency** | **Reason** |
| --- | --- |
| Python 3.11 | Modern language features and compatibility. |
| DuckDB | Lightweight, embedded SQL engine ideal for tabular scientific data. |
| Requests | To call external APIs. |
| BERN2 | Biomedical NER model to identify genes, diseases, etc., from captions. |
| FastAPI | High-performance REST API framework with automatic docs. |
| Docker | Enables reproducible deployment across environments. |
| JSON / CSV | Output data formatting options. |

**Security**

* API key/password protection using environment variables or .env.
* Docker container isolates the processing environment.
* Logs include timestamped INFO and DEBUG entries for traceability.

**Future-Proofing & Extensibility**

* Modular architecture allows adding support for PMIDs by mapping them to PMCIDs using eutils.
* New data sources (e.g., arXiv, bioRxiv) can be integrated via the Fetcher interface.
* Pluggable storage backend (PostgreSQL, S3, etc.).
* Logging, format, and authentication are all configurable.