Project Overview

This brief introduces our pipeline for converting unstructured clinical case reports into dynamic, queryable Directed Acyclic Graphs (DAGs), highlighting the data model and its attributes.

---

1. Background & Motivation

- **Initial Motivation**: To develop and validate our agentic tumor board AI—simulating multi-specialty clinical discussions—by leveraging the inherent complexity of published case reports. We have chosen thoracic oncology reports as our starting domain to mirror lung cancer tumor board workflows.
- **Challenge**: Case reports embed temporal and causal details in free text—difficult for algorithmic analysis or aggregation.
- **Approach**: Transform narratives into DAGs where **nodes** capture discrete patient states and **edges** capture their transitions, enabling:
  - Structured querying (e.g., "Which patients had lab elevation followed by med change?")
  - Trajectory similarity analyses and clustering
  - Downstream AI/predictive modeling on graph features

---

2. Pipeline Architecture (High Level)

1. **Text Extraction**: Extract text from PDF/XML via `extract_text_from_pdf`, with optional filtering of non-clinical sections.
2. **Chunking**: Use `ChunkingNodeModule` to respect LLM context limits (250–450 words per chunk).
3. **Node Generation**: Invoke `nodeConstruct` LLM signature to produce nodes.
4. **Edge Generation**: Invoke `edgeConstruct` with few-shot examples to produce edges.
5. **Clinical Data Extraction**: Use `nodeClinicalDataExtract` to structure labs, medications, vitals, etc.
6. **Branch Classification**: Use `branchClassify` to flag side branches.

---

3. Graph Data Model & Attribute Definitions

Nodes (Patient States)

Each node represents a snapshot of the patientś clinical state at a point in the narrative.

`node_id` **(str)** Unique alphabetical ID ("A", "B", ...).

`step_index` **(int)** Zero-based order in the main timeline.

`content` **(str)** Full narrative describing co-occurring findings, labs, imaging, interventions, and symptoms.

`timestamp` **(ISO 8601, optional)** When explicitly given in report.

`clinical_data` **(dict)** Structured, UMLS-aligned fields (present only if mapped):

- **medications**: List of tables with drug (CUI/string), dosage, frequency, modality, start/end dates, indication.
- **vitals**: Type (CUI/string), value, unit, timestamp.
- **labs**: Test (CUI/string), value, unit, flag, reference range, timestamp.
- **imaging**: Modality, body part, finding, impression, date.
- **procedures**: Name, approach, date, location, performed by, outcome.
- **HPI**: Summary, duration, onset, progression, associated/alleviating/exacerbating factors.
- **ROS**: System, findings.
- **functional_status**: Domain, description, score, scale.
- **mental_status**: Domain, finding, timestamp.
- **social_history**: Category, status, description.
- **allergies**: Substance, reaction, severity, date recorded.
- **diagnoses**: Code (ICD10/SNOMED/CUI), label, status, onset date.

Edges (State Transitions)

Edges encode clinical progression, side branches, or resolutions between nodes.

`edge_id` **(str)** Format `upstream_id_to_downstream_id`.

`branch_flag` **(bool)** `True` if initiating a side branch; otherwise `False`.

`content` **(str)** Narrative linking the two states (e.g., "CT showed mass progression").

`transition_event` **(dict, optional)** Structured change descriptors:

- Trigger type: `procedure | lab_change | medication_change | symptom_onset | interpretation | spontaneous`

- Trigger entities: List of UMLS CUIs for changed items.
- Change type: `addition | discontinuation | escalation | deescalation | reinterpretatio | resolution | progression | other`.
- Target domain: `medication | symptom | diagnosis | lab | imaging | procedure | functional_status | vital_sign`.
- Timestamp: ISO 8601 datetime if specified.
