NASA Vitalis Project

Modeling Expert Knowledge Structures in Scientific Texts

Selected Technical Evidence

This document presents representative technical evidence from an independent research project analyzing expert-authored scientific texts from the NASA research database. The excerpts and diagrams illustrate core methodological decisions used to model how expert knowledge is encoded, structured, and organized at scale.

1 Representative Code Excerpts

Code excerpts are lightly simplified for clarity and reflect the logic of the full analysis pipeline rather than complete production implementations.

1.1 Corpus Selection and Expert Text Filtering

Purpose: Operationalize expert knowledge using corpus-level selection criteria to ensure high-density domain-specific content.

```
def load_expert_corpus(metadata):
    # Filter for peer-reviewed, full-length technical papers
    expert_papers = metadata[
        (metadata["source"] == "NASA research database") &
        (metadata["document_type"] == "journal_article") &
        (metadata["word_count"] > 3000)
    ]

    texts = [load_pdf(paper["file_path"])
             for _, paper in expert_papers.iterrows()]
    return texts
```

Rationale:

- Establishes an operational definition of expert-authored material
- Reduces contamination from summaries or non-technical documents
- Supports large-scale structural analysis across hundreds of papers

1.2 Scientific Text Segmentation by Conceptual Units

Purpose: Preserve the rhetorical and conceptual structure of scientific writing rather than relying on fixed-length text chunks.

```
def segment_scientific_text(text):
    # Preserve rhetorical structure common in scientific writing
    sections = split_by_headers(
        text,
        headers=["Introduction", "Methods", "Results", "Discussion"]
    )

    # Further divide into semantically coherent paragraphs
    segments = [
        clean_and_normalize(p)
        for section in sections
        for p in paragraph_split(section)
    ]
    return segments
```

Rationale:

- Aligns segmentation with how scientific arguments are presented
- Avoids distortion introduced by arbitrary token windows
- Enables concept-level rather than sentence-level modeling

1.3 Concept Graph Construction and Structural Modeling

Purpose: Represent expert knowledge as an interconnected conceptual system rather than isolated textual units.

```
def build_concept_graph(segments, encoder, similarity_threshold=0.72):
    embeddings = encoder.encode(segments)
    graph = nx.Graph()

    for i, vec_i in enumerate(embeddings):
        for j, vec_j in enumerate(embeddings[i+1:], i+1):
            similarity = cosine_similarity(vec_i, vec_j)
            if similarity >= similarity_threshold:
                graph.add_edge(i, j, weight=similarity)
```

```
    return graph
```

Rationale:

- Translates semantic similarity into explicit structural relationships
- Emphasizes global organization over individual claim interpretation
- Enables analysis of hierarchy, clustering, and conceptual density

2 End-to-End Analysis Pipeline

This pipeline summarizes the sequence of methodological transformations applied to expert-authored texts, from corpus selection to structural comparison.

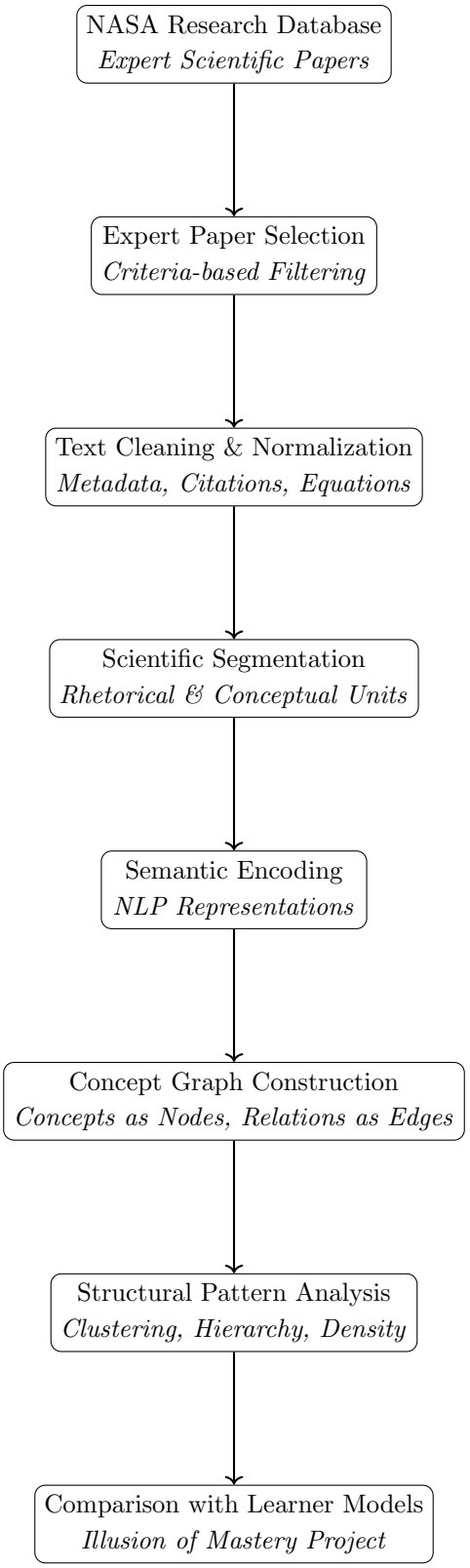

Figure 1: Visual overview of the analytical workflow used to model expert knowledge structures in scientific texts. Each stage reflects a methodological decision prioritizing conceptual organization over surface-level linguistic features in order to model how expert understanding is structured.

A detailed discussion of preprocessing choices and analytical methodology is provided in a separate technical methods document.