# GenAI Project Plan – Literature Assistant

Group 22
Cem Ashbaugh (12433756)
Noah Weidenhaupt (12327501)
Jonas Kruse (12434740)
Leon Baumgärtner (12434736)
Tim Krüger (12438588)

16.11.2025

## 5.1 Project Plan

### 5.1.1 Users

Our target users are **students and researchers** who frequently work with scientific literature in the context of research projects, theses, or academic publications. These users routinely collect, organise, and analyse publications from multiple sources and must integrate them into their own academic writing.

#### User goals

- **Learning & understanding:** Gain a clear overview of relevant research and understand how individual publications relate to their own project.
- **Productivity:** Reduce time spent manually organising metadata, summarising papers, and determining relevance.
- **Decision-making:** Identify the most relevant publications for each research question or section of their paper.
- **Synthesis:** Combine scattered pieces of literature into coherent insights for use in drafts or reports.

#### Workflows

Typical workflows include:

- Searching for new publications across platforms (e.g., Google Scholar, Semantic Scholar, arXiv, IEEE).
- Importing publications into tools like Zotero and manually filling in missing metadata.
- Reading papers, taking notes, and writing informal summaries about relevance.

- Connecting publications to the draft of a thesis or paper.

- Identifying gaps in the literature and refining research questions.

These workflows are time-consuming, error-prone, and only partially supported by existing tools, which is where the Literature Assistant intervenes.

### 5.1.2 Data

Our system works on several forms of user-provided and externally retrieved data.

#### Types of information

1. **Structured metadata**
   Title, authors, publication venue, keywords, year. These fields are often incomplete or inconsistent across sources.

2. **Semi-structured / unstructured PDF content**
   Full papers, abstracts, and extracted text, as well as notes describing why a publication is relevant for the user's project.

3. **User's publication draft**
   Early versions of papers, chapter outlines, or sections that need literature support.

4. **External data**
   Search results from public sources such as Semantic Scholar or arXiv.

#### Information organisation characteristics

**Granularity:** Mixed. Some publications have rich metadata and summaries, while others only have raw PDFs. Notes on relevance vary in length and quality.

**Connections:** Largely **implicit**. Users know why a publication is relevant but rarely link papers to each other or to specific sections of their draft. Relationships must often be inferred from text similarity.

**Completeness:** Highly **fragmented**. Many publications lack summaries, relevance tags, consistent keywords, or explanations of how they connect to the research draft.

**Context:** Limited metadata on why a publication is relevant, when it was added, or how it fits into the research narrative.

**Heterogeneity:** Data comes from multiple formats (PDFs, notes, draft documents, external APIs), and structure varies significantly.

#### Dataset justification

We will assemble a representative dataset of:

- approximately 20–40 PDFs,

- approximately 200–400 metadata fields,
- approximately 5–10 draft sections.

This size enables meaningful retrieval, enrichment, and summarisation tests without requiring excessive data collection time.

### 5.1.3 The Problem

Researchers face three core challenges when managing literature:

1. **Metadata incompleteness**
   Publications are imported with missing fields or missing summaries. Users must manually extract key information from PDFs.
2. **Retrieval difficulties when searching for new publications**
   Users lack tools to query across external databases using contextual, project-specific criteria. Queries such as "Find studies using methodology X with results similar to publication Z" cannot be answered effectively by traditional keyword-based tools.
3. **Lack of connection and synthesis within their literature library**
   Existing publications are often poorly described and not connected to each other or to the user's own draft. Users struggle to understand how each publication contributes to the research project.

These issues slow down academic writing, reduce the quality of literature reviews, and burden users with tedious manual work.

### 5.1.4 The Solution

#### Concept

The Literature Assistant addresses the above problems through three core tasks:

1. **Understand & enrich**
   Parse PDFs, extract key concepts, and automatically populate a predefined metadata schema (authors, keywords, relevance summary, potential usage in the paper). Add contextual explanations such as "Why this paper is useful for your project".
2. **Retrieve & recommend**
   Allow users to issue natural-language queries for new literature (e.g., "Find papers using methodology X with results similar to publication Z"). Retrieve top relevant publications using external APIs and re-rank them using semantic similarity.
3. **Connect & synthesise**
   Analyse existing publications, find conceptual overlaps, and link them to relevant sections of the user's draft. Produce suggestions such as:
   - "This paper may support the methodology section."
   - "Publication A and B address similar hypotheses."

#### User interface

We will use a Jupyter Notebook as interface (will be extended to a web application if the workload allows it), and may additionally integrate the Zotero API for reference management.

**Justification.** Researchers already use web-based tools (Zotero, Notion, Overleaf, Google Scholar). A web application:

- allows easy integration of PDF upload, metadata display, and external API search,
- supports interactive views such as literature lists, relevance summaries, and RAG-based recommendations,
- is accessible across devices without installation.

#### Technical approach

We focus on three key technical challenges:

1. **PDF parsing & metadata extraction**
   Extract text reliably from diverse PDF formats and use LLM function calling to fill a structured metadata schema (e.g., title, authors, keywords, short relevance summary).
2. **Semantic retrieval pipeline**
   Implement query expansion, external search API calls, and retrieval-augmented generation (RAG). Combine lexical search (from external APIs) with embedding-based re-ranking to surface the most relevant results for a user's query.
3. **Embedding-based connection discovery**
   Chunk and embed PDFs, store embeddings in a vector database, and use similarity clustering to detect thematic relationships. This will be used to propose links between publications and relevant sections of the user's draft.

### 5.1.7 Evaluation

#### Definition of success

Our system is successful if it:

1. correctly extracts and fills metadata fields from PDFs,
2. retrieves relevant new publications for user queries with high relevance,
3. provides useful connection insights between existing publications and the user's draft.

#### Quantitative metrics

Quantitative evaluation requires labels produced by humens, which might be beyond the scope of the project. In this case, we might resort to simpler methods.

1. **Metadata extraction accuracy.**

- Measure correctness of extracted fields (title, authors, keywords, relevance summary).
- Compare against manually curated ground truth.
- Metrics: field-level accuracy and F1 score for keywords.

2. **Retrieval quality.**

- Evaluate with a set of curated search queries and a human-labelled relevance baseline.
- Metrics: precision@k, recall@k, MRR@k for different values of $k$ (e.g., $k = 5, 10$).

3. **Connection suggestions relevance.**

- For a sample of publications, test whether the system's suggestions match human-labelled relevant draft sections.
- Metrics: top-$k$ overlap (how often relevant sections appear in the top suggestions) and ranking correlation between system ranking and human ranking.