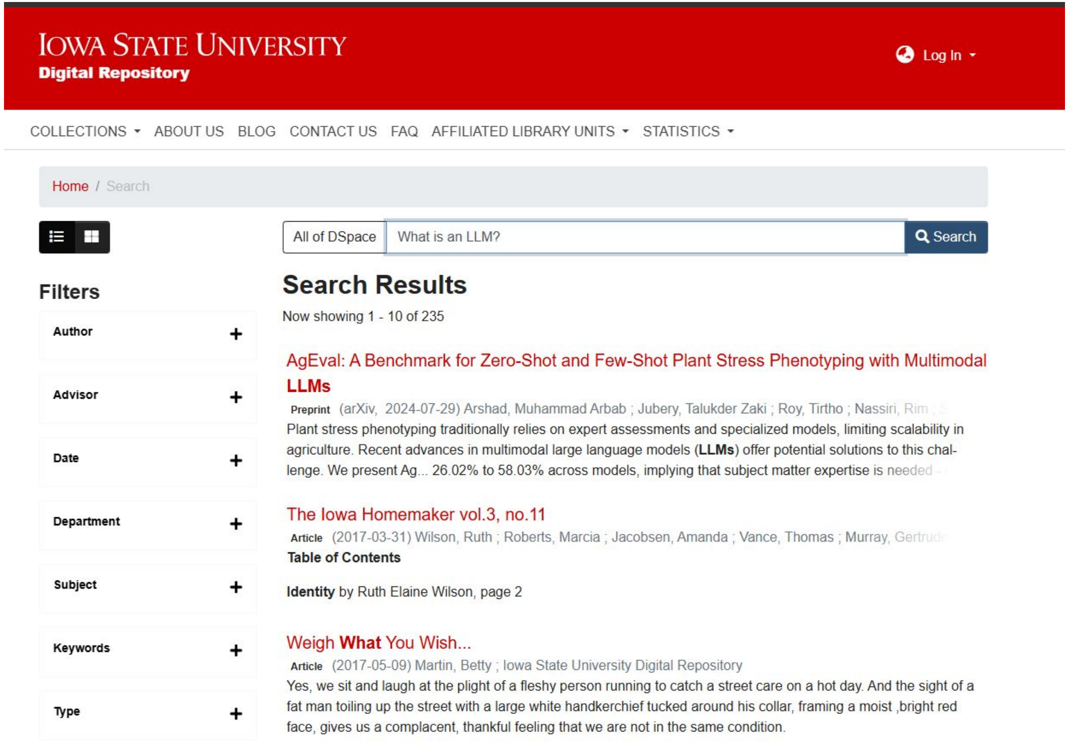
RAG Library Search

Team 01

Background



What is an LLM?

Search Results

Now showing 1 - 10 of 235

AgEval: A Benchmark for Zero-Shot and Few-Shot Plant Stress Phenotyping with Multimodal **LLMs**

Preprint (arXiv, 2024-07-29) Arshad, Muhammad Arbab ; Jubery, Talukder Zaki ; Roy, Tirtho ; Nassiri, Rim ;

Plant stress phenotyping traditionally relies on expert assessments and specialized models, limiting scalability in agriculture. Recent advances in multimodal large language models (**LLMs**) offer potential solutions to this challenge. We present Ag... 26.02% to 58.03% across models, implying that subject matter expertise is needed

The Iowa Homemaker vol.3, no.11

Article (2017-03-31) Wilson, Ruth ; Roberts, Marcia ; Jacobsen, Amanda ; Vance, Thomas ; Murray, Gertrud

Table of Contents

Identity by Ruth Elaine Wilson, page 2

Weigh **What** You Wish...

Article (2017-05-09) Martin, Betty ; Iowa State University Digital Repository

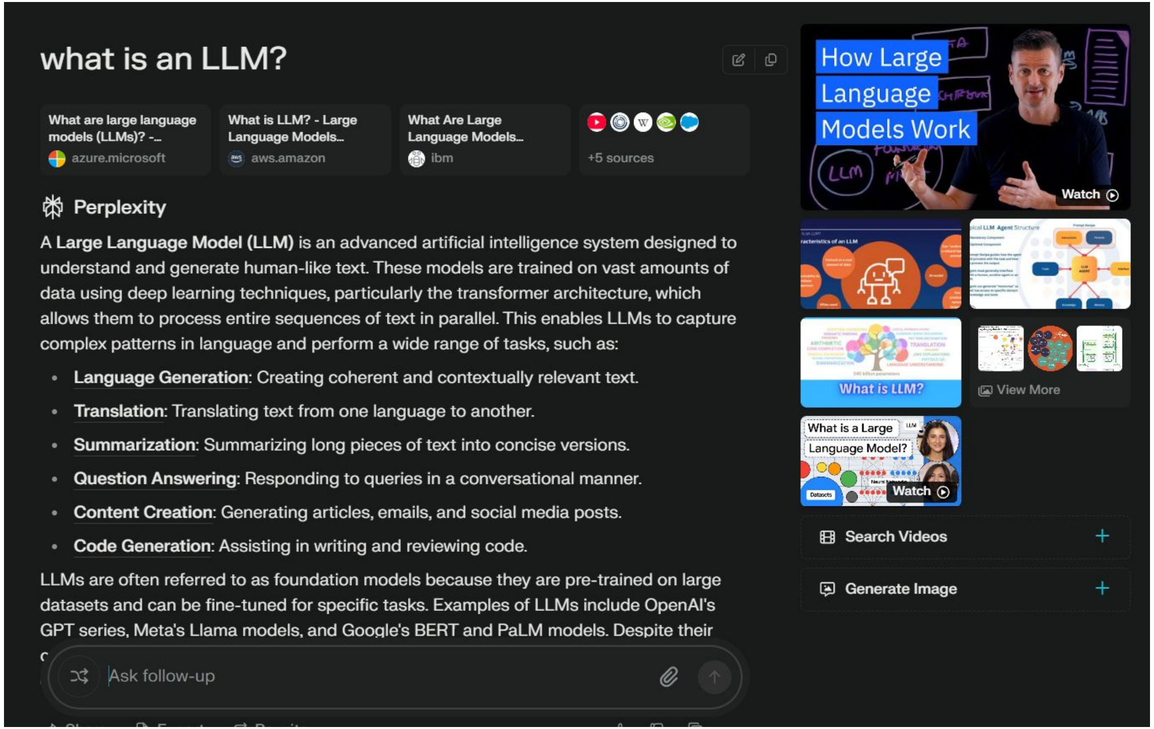
Yes, we sit and laugh at the plight of a fleshy person running to catch a street care on a hot day. And the sight of a fat man toiling up the street with a large white handkerchief tucked around his collar, framing a moist ,bright red face, gives us a complacent, thankful feeling that we are not in the same condition.

ISU Digital Repository

Find academic papers, projects, etc.

← keyword search flaw

Modern Search Platform



what is an LLM?

What are large language models (LLMs)? -...
azure.microsoft

What is LLM? - Large Language Models...
aws.amazon

What Are Large Language Models...
ibm

+5 sources

Perplexity

A **Large Language Model (LLM)** is an advanced artificial intelligence system designed to understand and generate human-like text. These models are trained on vast amounts of data using deep learning techniques, particularly the transformer architecture, which allows them to process entire sequences of text in parallel. This enables LLMs to capture complex patterns in language and perform a wide range of tasks, such as:

- **Language Generation**: Creating coherent and contextually relevant text.
- **Translation**: Translating text from one language to another.
- **Summarization**: Summarizing long pieces of text into concise versions.
- **Question Answering**: Responding to queries in a conversational manner.
- **Content Creation**: Generating articles, emails, and social media posts.
- **Code Generation**: Assisting in writing and reviewing code.

LLMs are often referred to as foundation models because they are pre-trained on large datasets and can be fine-tuned for specific tasks. Examples of LLMs include OpenAI's GPT series, Meta's Llama models, and Google's BERT and PaLM models. Despite their

Perplexity.ai

AI search with sources + other relevant info.

Allows conversation-style ← searching

Well-Defined Problem

ISU Digital Repository Search has:

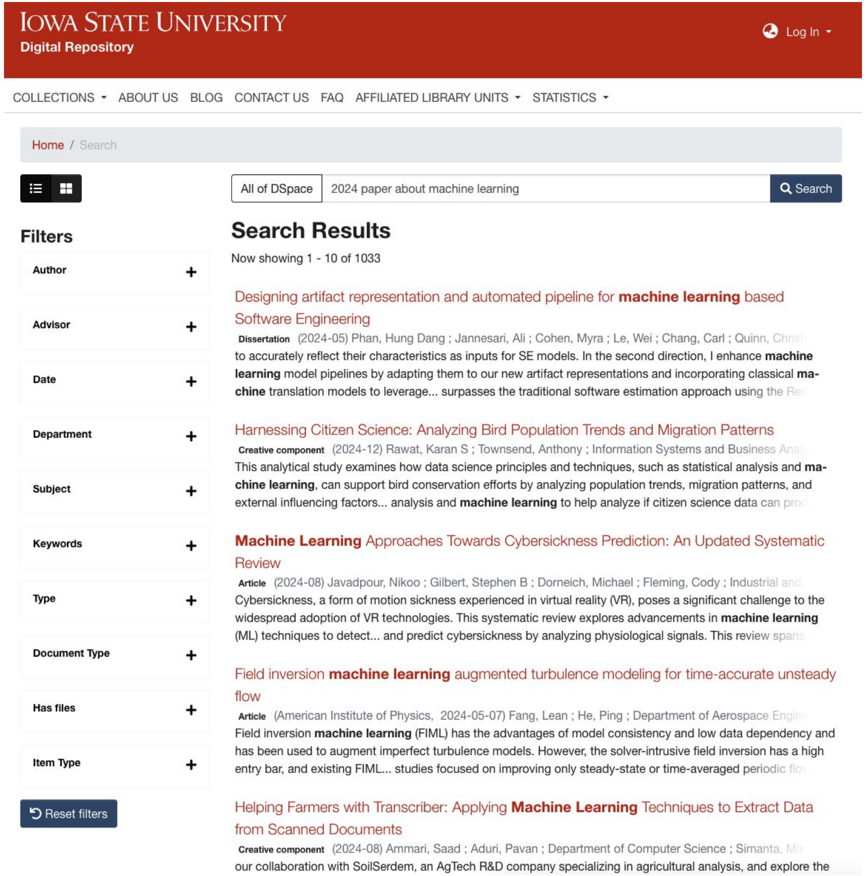
1. Inefficient Document Retrieval
    - Keyword based
2. Overwhelming Information Volume
    - Difficult to sift through
3. Lack of Intelligent Assistance
    - No dynamic results

Example

Vague results

User still needs to check each paper

User cannot be specific



2024 paper about machine learning

Search Results

Now showing 1 - 10 of 1033

Designing artifact representation and automated pipeline for **machine learning** based Software Engineering
Dissertation (2024-05) Phan, Hung Dang ; Jannesari, Ali ; Cohen, Myra ; Le, Wei ; Chang, Carl ; Quinn, Chris
to accurately reflect their characteristics as inputs for SE models. In the second direction, I enhance **machine learning** model pipelines by adapting them to our new artifact representations and incorporating classical **machine** translation models to leverage... surpasses the traditional software estimation approach using the Re

Harnessing Citizen Science: Analyzing Bird Population Trends and Migration Patterns
Creative component (2024-12) Rawat, Karan S ; Townsend, Anthony ; Information Systems and Business Ana
This analytical study examines how data science principles and techniques, such as statistical analysis and **machine learning**, can support bird conservation efforts by analyzing population trends, migration patterns, and external influencing factors... analysis and **machine learning** to help analyze if citizen science data can pro

Machine Learning Approaches Towards Cybersickness Prediction: An Updated Systematic Review
Article (2024-08) Javadpour, Nikoo ; Gilbert, Stephen B ; Dorneich, Michael ; Fleming, Cody ; Industrial and
Cybersickness, a form of motion sickness experienced in virtual reality (VR), poses a significant challenge to the widespread adoption of VR technologies. This systematic review explores advancements in **machine learning** (ML) techniques to detect... and predict cybersickness by analyzing physiological signals. This review spans

Field inversion **machine learning** augmented turbulence modeling for time-accurate unsteady flow
Article (American Institute of Physics, 2024-05-07) Fang, Lean ; He, Ping ; Department of Aerospace Engin
Field inversion **machine learning** (FIML) has the advantages of model consistency and low data dependency and has been used to augment imperfect turbulence models. However, the solver-intrusive field inversion has a high entry bar, and existing FIML... studies focused on improving only steady-state or time-averaged periodic flo

Helping Farmers with Transcriber: Applying **Machine Learning** Techniques to Extract Data from Scanned Documents
Creative component (2024-08) Ammari, Saad ; Aduri, Pavan ; Department of Computer Science ; Simanta, Mi
our collaboration with SoilSerdem, an AgTech R&D company specializing in agricultural analysis, and explore the

Project Objectives

1. **Build an Intelligent Retrieval System:**
    - semantic search
2. **Integrate an AI Chatbot:**
    - conversation and follow-up questions
3. **Enhance User Experience:**
    - improve research access
4. **Evaluate AI Models:**
    - compare cost and performance
5. **Lay the Groundwork for Future Expansion:**
    - scalable system and more complex functionalities (images/diagrams)

Solution - Core Features

- **Semantic Search** → meaning and context
- **AI Chatbot Interface** → answers user queries
- **Reference Tracking** → backed by references
- **Comparative Analysis of AI Models** → open-source models versus paid LLMs
- **Vision vs. Non-Vision Capabilities** → documents that contain images or diagrams.
- **Analytics and Metrics Collection** → tracks performance metrics
- **Research Topics discovery** → smart content mapping

First steps

- Learn and understand the fundamentals
- LLM (Large Language Model)
- What is RAG?
- Experimenting with multiple LLM models
- Apps to run LLMs locally

LLM (Large Language Model)

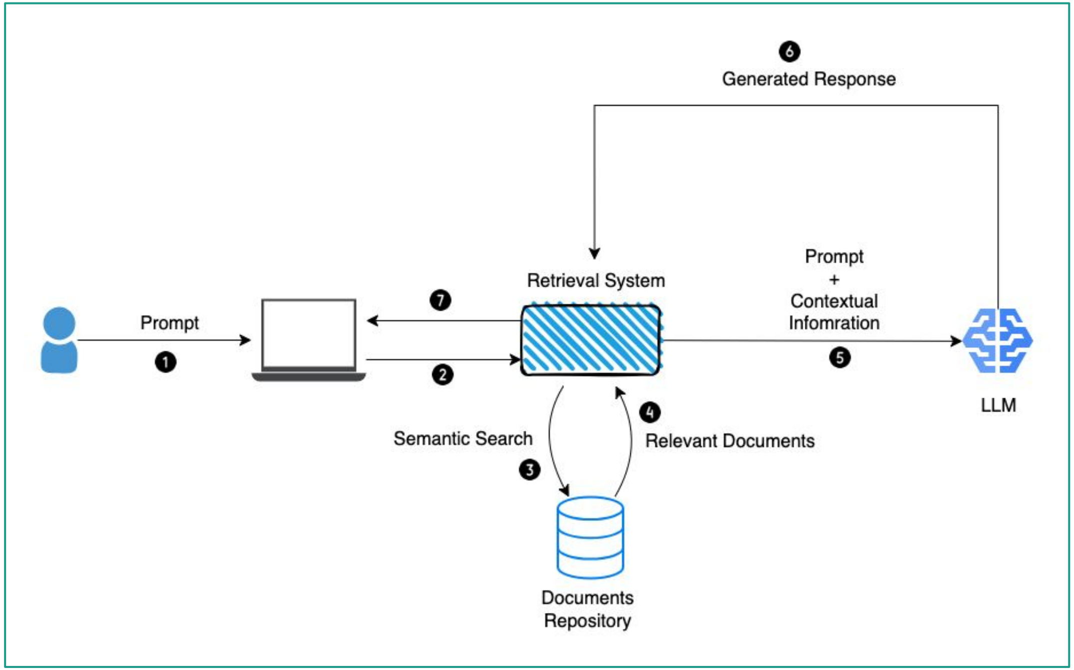
Large: extensive training data and modal size.

Language: focuses on natural language processing.

Model: makes responses based on learned data patterns

But, LLMs are static….

RAG (Retrieval Augmented Generation)



Why RAG?

- We need precision and credibility
- Research papers are published constantly
- User needs up-to-date answers

The experiment

- Select multiple papers from ISU digital repository
- Upload papers to LLMs in groups of three
- Ask general and specific questions
- Ask follow up questions
- Record results
- Compare models
- Decide most suited model for our application

Running LLMs locally vs cloud

AnythingLLM: provides a multi-user web interface, and supports document-based Q&A

LM Studio: models are easy to download, support many open-source models

OpenWebUI: open-source, could be hard to set up.

Ollama

Chatgpt.com

Claude.ai

Google Studio

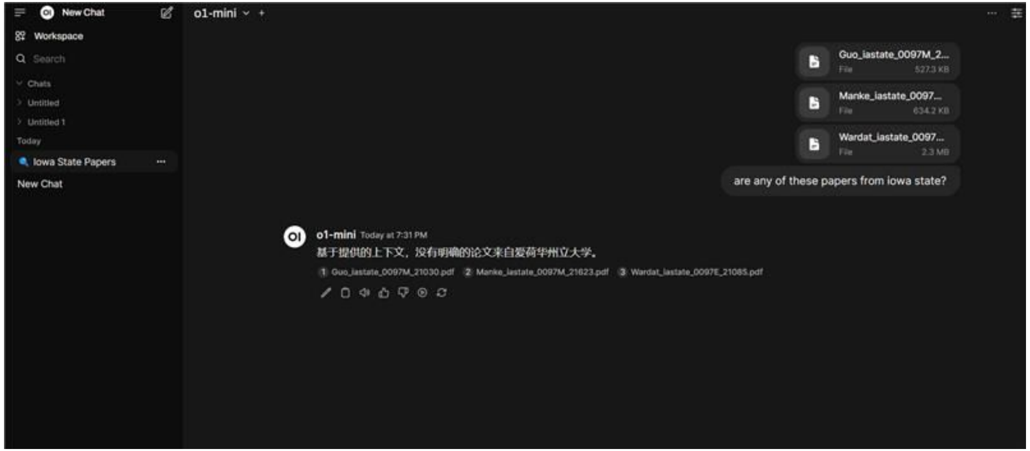
Experimenting with LLMs

ChatGPT

- GPT 4o
- o1
- o3 mini

Tested locally

Tested via cloud and API



Guo_iastate_0097M_2...
File 527.3 KB

Manke_iastate_0097...
File 634.2 KB

Wardat_iastate_0097...
File 2.3 MB

are any of these papers from iowa state?

o1-mini Today at 7:31 PM

基于提供的上下文，没有明确的论文来自爱荷华州立大学。

1 Guo_iastate_0097M_21030.pdf 2 Manke_iastate_0097M_21623.pdf 3 Wardat_iastate_0097E_21085.pdf

Claude 3.5 Sonnet

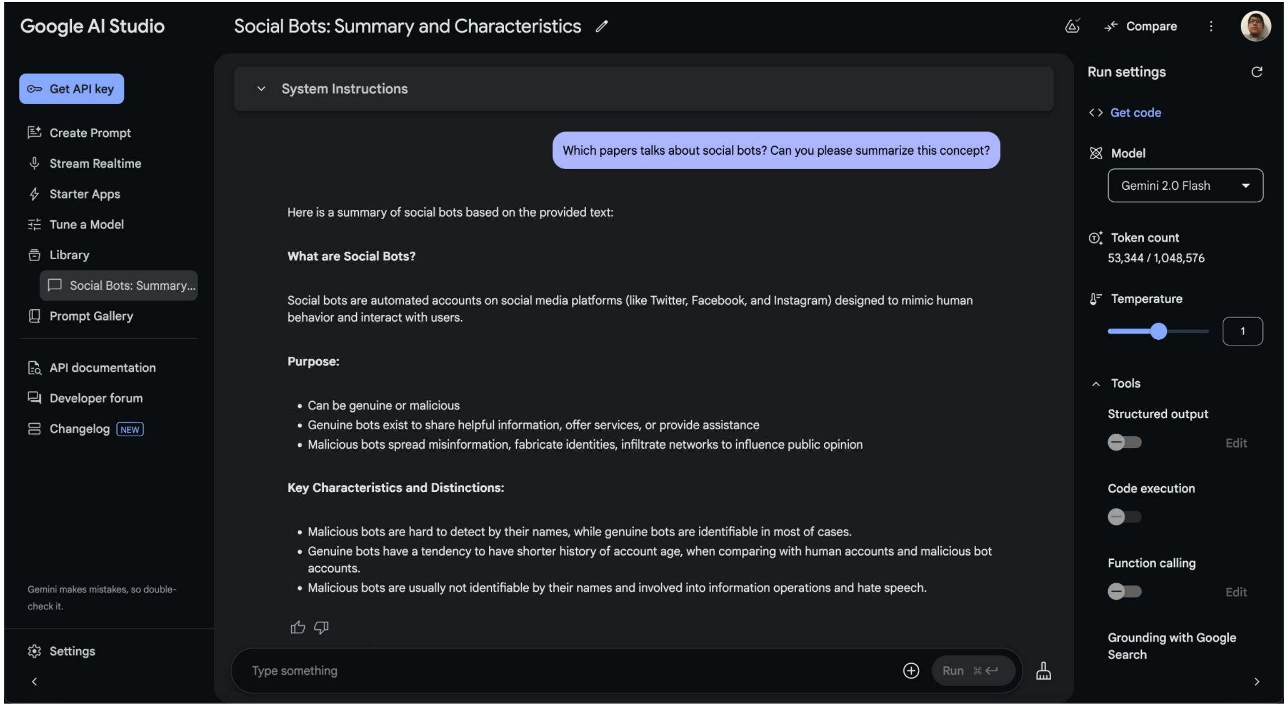
Long context but limited

Better at handling long documents

Could be expensive

Might get confused as documents become larger

Google's Gemini



Social Bots: Summary and Characteristics

Which papers talks about social bots? Can you please summarize this concept?

Here is a summary of social bots based on the provided text:

What are Social Bots?

Social bots are automated accounts on social media platforms (like Twitter, Facebook, and Instagram) designed to mimic human behavior and interact with users.

Purpose:

- Can be genuine or malicious
- Genuine bots exist to share helpful information, offer services, or provide assistance
- Malicious bots spread misinformation, fabricate identities, infiltrate networks to influence public opinion

Key Characteristics and Distinctions:

- Malicious bots are hard to detect by their names, while genuine bots are identifiable in most of cases.
- Genuine bots have a tendency to have shorter history of account age, when comparing with human accounts and malicious bot accounts.
- Malicious bots are usually not identifiable by their names and involved into information operations and hate speech.

Model: Gemini 2.0 Flash

Token count: 53,344 / 1,048,576

Temperature: 1

Structured output ○

Code execution ○

Function calling ○

Strong factual reasoning.

Supports RAG

Can interpret PDFs that involve images and charts.

Easy API integration

Experimenting with LLMs

Deepseek R1

- Strengths:
  - Shows how it thinks
- Problems:
  - Slow
  - Hard to escape context
  - Can give incorrect answers for the document is using

Experimenting with LLMs

Llama 3.2 Vision

- Strengths
  - Straight to the point
- Weaknesses
  - Not very accurate
  - Expensive

Results

ChatGPT o1 provides most accurate answers; o3-mini can be used for reasoning.

Claude 3.5 Sonnet is fast, and accurate most of the time. Could lose context after multiple follow-up questions.

Gemini 1.8 Flash provides fast token processing

Vectorization & Contextualization - Methodology

Technical Implementation:

- API Integration: **Google Gemini 1.5 Pro** LLM
- Vector Database: **ChromaDB** for document embedding and semantic retrieval
- Document Processing: PDF extraction via pdfplumber with hierarchical chunking
- Answers recorded in JSON files
- Error Handling: Exponential backoff for API calls (2-5-10s) with max_retries=3

Design Choices:

- ChromaDB for vector database for lightweight architecture and strong search capabilities
- pdfplumber allows for chunking at document and paragraph levels

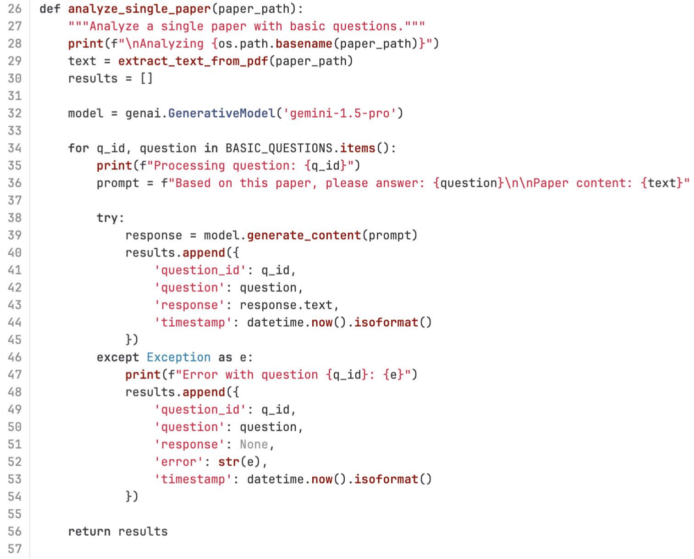
Vectorization & Contextualization - Basic Approach

Working:

- Direct document-to-query processing
- Full document context with no fragmentation
- Single-pass analysis without semantic indexing
- Implementation: direct API prompting

Technical Challenges:

- Context window limitations
- Inability to make connections between different ideas



```python
def analyze_single_paper(paper_path):
    """Analyze a single paper with basic questions."""
    print(f"\nAnalyzing {os.path.basename(paper_path)}")
    text = extract_text_from_pdf(paper_path)
    results = []

    model = genai.GenerativeModel('gemini-1.5-pro')

    for q_id, question in BASIC_QUESTIONS.items():
        print(f"Processing question: {q_id}")
        prompt = f"Based on this paper, please answer: {question}\n\nPaper content: {text}"

        try:
            response = model.generate_content(prompt)
            results.append({
                'question_id': q_id,
                'question': question,
                'response': response.text,
                'timestamp': datetime.now().isoformat()
            })
        except Exception as e:
            print(f"Error with question {q_id}: {e}")
            results.append({
                'question_id': q_id,
                'question': question,
                'response': None,
                'error': str(e),
                'timestamp': datetime.now().isoformat()
            })

    return results
```

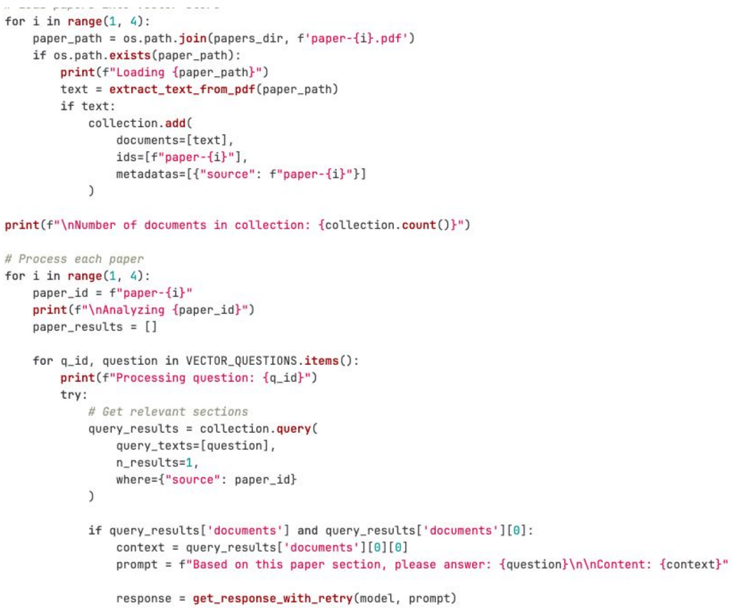
Vectorization & Contextualization - Vectorized Approach

Working:

- Document-level dense vector representations
- Semantic similarity search with n_results=1
- ChromaDB collection with document-level indexing

Problems & Challenges:

- Loss of document structure
- Finding optimality between comprehension and response accuracy is tricky
- Loss of broader context



```python
for i in range(1, 4):
    paper_path = os.path.join(papers_dir, f'paper-{i}.pdf')
    if os.path.exists(paper_path):
        print(f"Loading {paper_path}")
        text = extract_text_from_pdf(paper_path)
        if text:
            collection.add(
                documents=[text],
                ids=[f"paper-{i}"],
                metadatas=[{"source": f"paper-{i}"}]
            )

print(f"\nNumber of documents in collection: {collection.count()}")

# Process each paper
for i in range(1, 4):
    paper_id = f"paper-{i}"
    print(f"\nAnalyzing {paper_id}")
    paper_results = []

    for q_id, question in VECTOR_QUESTIONS.items():
        print(f"Processing question: {q_id}")
        try:
            # Get relevant sections
            query_results = collection.query(
                query_texts=[question],
                n_results=1,
                where={"source": paper_id}
            )

            if query_results['documents'] and query_results['documents'][0]:
                context = query_results['documents'][0][0]
                prompt = f"Based on this paper section, please answer: {question}\n\nContent: {context}"

                response = get_response_with_retry(model, prompt)
```

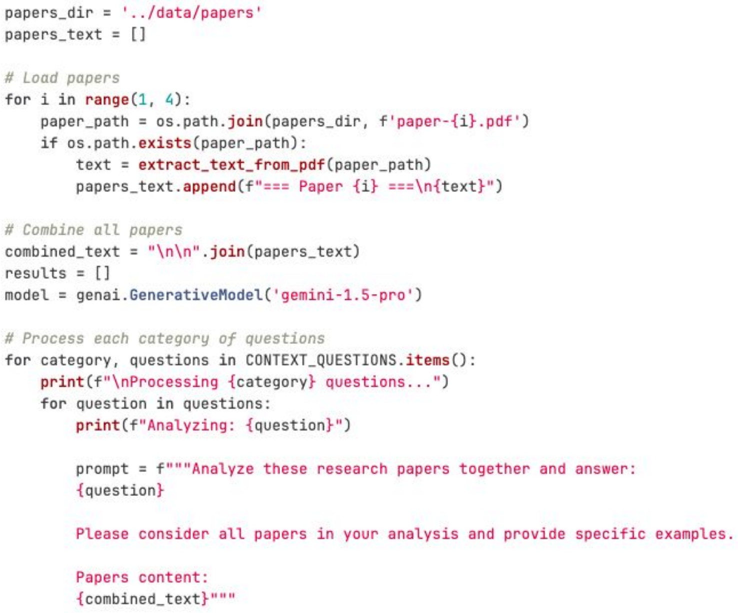
Vectorization & Contextualization - Context QA

Working:

- Multi-document aggregation with boundary markers
- Cross-document thematic analysis
- Category-based question hierarchy (comparative, thematic, synthesis)
- Combined context with explicit document demarcation

Challenges:

- Context window limitations for larger applications
- Combined context window might exceed token limits for models
- Processing entire documents leads to processing overhead
- Treats all documents with equal importance



```
papers_dir = '../data/papers'
papers_text = []

# Load papers
for i in range(1, 4):
    paper_path = os.path.join(papers_dir, f'paper-{i}.pdf')
    if os.path.exists(paper_path):
        text = extract_text_from_pdf(paper_path)
        papers_text.append(f"=== Paper {i} ===\n{text}")

# Combine all papers
combined_text = "\n\n".join(papers_text)
results = []
model = genai.GenerativeModel('gemini-1.5-pro')

# Process each category of questions
for category, questions in CONTEXT_QUESTIONS.items():
    print(f"\nProcessing {category} questions...")
    for question in questions:
        print(f"Analyzing: {question}")

        prompt = f"""Analyze these research papers together and answer:
        {question}

        Please consider all papers in your analysis and provide specific examples.

        Papers content:
        {combined_text}"""
```

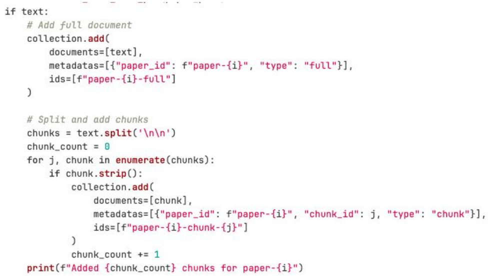
Vectorization & Contextualization - Vectorized Context Window

Working

- Hybrid chunking strategy with full-document and paragraph-level embeddings
- n_results=5 with cross-document relevance ranking
- Metadata-based retrieval with source tracking
- ChromaDB query with contextual recombination of top chunks

Problem:

- Rate Limiting implementation is required
- Potential vector database costs
- Data scraping



```
if text:
    # Add full document
    collection.add(
        documents=[text],
        metadatas=[{"paper_id": f"paper-{i}", "type": "full"}],
        ids=[f"paper-{i}-full"]
    )

    # Split and add chunks
    chunks = text.split('\n\n')
    chunk_count = 0
    for j, chunk in enumerate(chunks):
        if chunk.strip():
            collection.add(
                documents=[chunk],
                metadatas=[{"paper_id": f"paper-{i}", "chunk_id": j, "type": "chunk"}],
                ids=[f"paper-{i}-chunk-{j}"]
            )
            chunk_count += 1
    print(f"Added {chunk_count} chunks for paper-{i}")
```

Vectorization & Contextualization - Results

Basic Question and Answer

Depth of Understanding

Identifies main findings, results, limitations, etc. Answers are mostly surface level.

Contextual Understanding

Connects ideas across different section. Understands relationships between methods, results, and their significance

Quality of Response

Organizes information logically in it's response

&

presents information in a coherent hierarchy. Sometimes lacks depth.

Vectorization & Contextualization - Results

Vectorized Question and Answer

Depth of Understanding

Much more detailed technical explanations.

Contextual Understanding

Stronger linkage between the different sections of the paper. Clearer relationships between methods & their purpose.

Quality of Response

More precise and technically accurate. Better organized responses with a clearer structure.

Vectorization & Contextualization - Results

Contextualized Question and Answer

Depth of Understanding

Very deep understanding when comparing the different papers. Stronger grasp of theory and practical implications of these papers.

Contextual Understanding

Exceptional ability to draw inferences across the 3 papers. Understands how first paper theory lays the framework for the second and third papers.

Quality of Response

Clear use of examples to support points. Strong analytical answers - maintains balance between all 3 papers and individual concepts.

Vectorization & Contextualization - Results

Vectorized Contextual Question and Answer

Depth of Understanding

Shows significantly enhanced understanding of how papers interconnect and build upon each other.

Contextual Understanding

Superior ability to trace progression across papers. Better at explaining relationships between theory and practical notions.

Quality of Response

Better organization with clear hierarchical presentation. Stronger supporting examples and evidence

Vectorized Contextual Q&A provides the most sophisticated and nuanced understanding of relationships between papers.

Progress & Challenges

- Experimenting with tools and technologies

Remaining Work

1. **Unit 1 (Weeks 3–6):**
    - Requirements gathering, initial design, and early demos.
2. **Unit 2 (Weeks 7–11):**
    - Development of data ingestion, embedding integration, and LLM-based chatbot features.
3. **Unit 3 (Weeks 12–13):**
    - Final integration, advanced features, and performance analytics.
4. **Weeks 14–15:**
    - Final system polishing, comprehensive testing, and preparation for the final presentation.

Demo

- Semantic Search
  - Embeddings: text-embedding-ada-002
  - Vector DB: FAISS
- Retrieval Augmented Generation
  - Gpt-4o api

Next Steps

- Implement ChromaDB, Google Gemini 1.5 Pro LLM
- Create Web Based UI
  - Chatbot
  - Image and Diagram Capabilities