Semantic Paper Analyzer

1. Title: Research Navigator: Semantic Paper Analyzer

---

2. Project Statement:

The exponential growth of digital documents and academic literature presents a major challenge for researchers and practitioners. Traditional analysis methods are slow, inefficient, and struggle with massive data volumes.

RESEARCH NAVIGATOR is an advanced, full-stack application that provides an intelligent, AI-powered system for rapid document analysis and academic insight extraction. It goes far beyond simple keyword searches, enabling users to analyze local files, utilize a search-grounded Gemini AI Assistant to synthesize multiple web sources, and retrieve visual, deep-level NLP metrics (Keywords, Named Entities, Readability) instantly. The entire history is secured and persisted in **MongoDB**.

---

3. Outcomes:

- **Automated Information Extraction:** Accurately extract key entities (e.g., research methods, datasets, theories, diseases, genes) and relationships from academic texts.
- **Semantic Search Capability:** Allow users to query research papers using natural language, retrieving results based on meaning rather than just keyword matches.
- **Intelligent Paper Recommendation:** Suggest related papers based on extracted semantic content, helping users discover relevant literature they might otherwise miss.

- **Conceptual Knowledge Overview:** Provide a structured (though not necessarily visual graph) overview of how concepts and methods are connected within a research domain.
- **Scalable Data Processing:** Designed to ingest and process large volumes of academic abstracts/papers from open-access repositories.
- **User-Friendly Interface:** An intuitive platform for searching, browsing, and analyzing research papers.

Outcome	Implementation Detail
Intelligent Information Extraction	**Deep NLP Pipeline:** Extracts Keywords (KeyBERT), Named Entities (spaCy), Extractive Summaries (TextRank), and Abstractive Summaries (BART-CNN).
Semantic Synthesis & Search	**Gemini AI Assistant:** Uses Google Search grounding to find external papers, extracts content, and synthesizes multi-source summaries (up to 5 papers at a time).
Document Compatibility	Supports analysis of local **PDF** and **DOCX** files, as well as text extracted from web URLs.
Advanced Reporting & Visualization	**Dashboard & Visuals Page:** Displays ROUGE metrics, Flesch-Kincaid Readability Grade, Entity Distribution charts, and Sentence Length histograms.
Persistent History	All analyses (local file uploads and AI synthesis) are permanently saved to a user-specific collection in **MongoDB** for secure historical access.
User-Friendly Interface (New)	Modern, fully responsive Flask interface featuring an interactive home page, dynamic greeting, visual feature cards, and a looping background video.

---

4. Modules:

The final project structure consists of the following tightly integrated modules, covering the full scope from user login to persistent data storage:

A. User Authentication & Persistence Module

- **Authentication:** Secure registration and login using **MongoDB** (replacing the local JSON file) with hashed passwords (werkzeug.security).
- **Verification:** Email verification via One-Time Passcode (OTP).
- **Data Storage:** User profiles and all analysis results are saved permanently in MongoDB.

B. Document & Preprocessing Module

- **Ingestion:** Handles local file uploads (PDF, DOCX) and secure file handling (werkzeug.utils). Also includes a web scraper simulation (requests) for the AI Assistant to fetch text from URLs.
- **Preprocessing:** Robust text cleaning, sentence segmentation (nltk), and initial tokenization.

C. Core NLP Analysis Module

- **Summarization:**
    - **Abstractive:** Uses the **BART-Large-CNN** model (transformers) to generate synthesized summaries.

    - **Extractive:** Uses **TextRank** (based on networkx and sentence-transformers) for key sentence extraction.
- **Keyword and Entity Extraction: KeyBERT** for relevant keywords and **spaCy** (en_core_web_sm) for Named Entity Recognition.
- **Readability:** Calculates the **Flesch-Kincaid Grade Level** for comprehension scoring.

D. AI Assistant & Synthesis Module (Gemini API)

- **API Integration:** Securely makes calls to the **Gemini 2.5 Flash API** using the user's query.
- **Search Grounding:** Leverages **Google Search** tools to find relevant academic papers based on the user's prompt.
- **Multi-Analysis:** Custom Python logic fetches content from up to 5 selected URLs, combines the text, and submits the combined content back to Gemini for a cohesive synthesis. The synthesis is then run through the full NLP pipeline.

E. Visualization & Reporting Module

- **Dashboard (results.html):** Displays side-by-side summaries, ROUGE scores, and key insights.
- **Visualizations (visuals.html):** Renders interactive charts using **Chart.js** for Entity Distribution, Keyword Relevance, and Sentence Length.
- **Analysis History (history.html):** Displays a personalized, filterable history of all user reports retrieved from MongoDB.

Technology Stack:

Category	**Tools & Libraries**
Backend Framework	Python 3, Flask, Werkzeug
Database	**MongoDB** (PyMongo)
AI/LLM	**Gemini 2.5 Flash API** (for search and synthesis)
NLP Models	**BART-Large-CNN** (HuggingFace Transformers), **KeyBERT**, **Sentence-Transformers**, **spaCy** (for NER), **NLTK** (for tokenization).
Data Science	NumPy, Scikit-learn (for cosine similarity), NetworkX (for TextRank).
Frontend	Jinja2 Templates, **Tailwind CSS**, JavaScript (for interactivity), **Chart.js** (for visualizations).
Document Handling	PyMuPDF (PDF), python-docx (DOCX)

---

5. Hardware and Software Requirements:

1. Hardware Requirements

- Processor (CPU): Intel i5 (8th Gen+) / AMD Ryzen 5 (2000 series+) or equivalent (i7/Ryzen 7 recommended for optimal performance).
- Memory (RAM): 8 GB Minimum (16 GB+ recommended for large NLP models and documents).
- Storage: 128 GB SSD Minimum (256 GB+ SSD recommended for speed).
- Network: Stable broadband internet connection (for API access, web documents, email verification).

2. Software Requirements

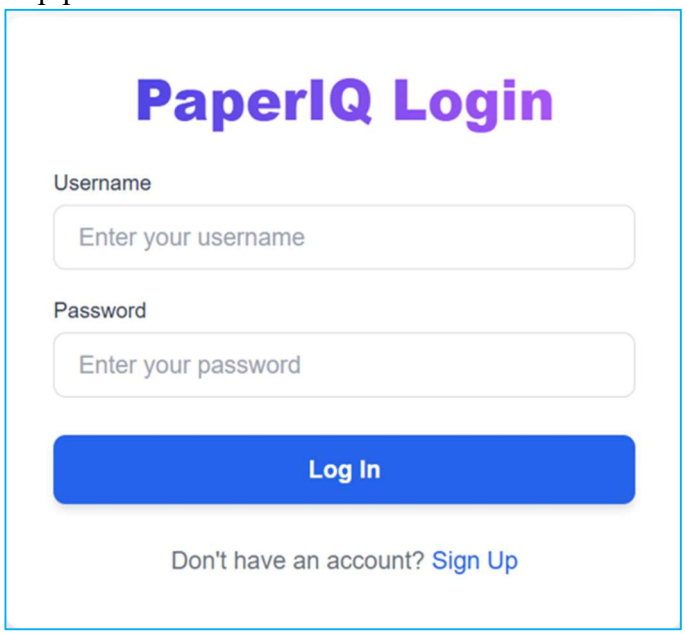
- Operating System: Windows 10/11 (64-bit), macOS, or Linux (Ubuntu 20.04+)
- Programming Language: Python 3.8 or higher.
- Web Browser: Latest version of Chrome, Firefox, Edge, or Safari.
- Database: MongoDB Server 4.4+ (or MongoDB Atlas cloud instance).
- Key Python Libraries:
    - Flask (Web framework)
    - pymongo (MongoDB driver)
    - google-generativeai (Gemini API)
    - transformers (BART model)
    - spacy (NER)
    - KeyBERT (Keyword Extraction)
    - nltk (Text processing)
    - python-docx, PyMuPDF (Document parsing)
    - requests, beautifulsoup4 (Web scraping)
    - werkzeug (Security), smtplib (Email OTP)
    - python-dotenv (Environment management)
- Environment Variables: FLASK_APP, MONGO_URI, GOOGLE_API_KEY, SENDER_EMAIL, SECRET_KEY, etc. (for API, DB, and security).

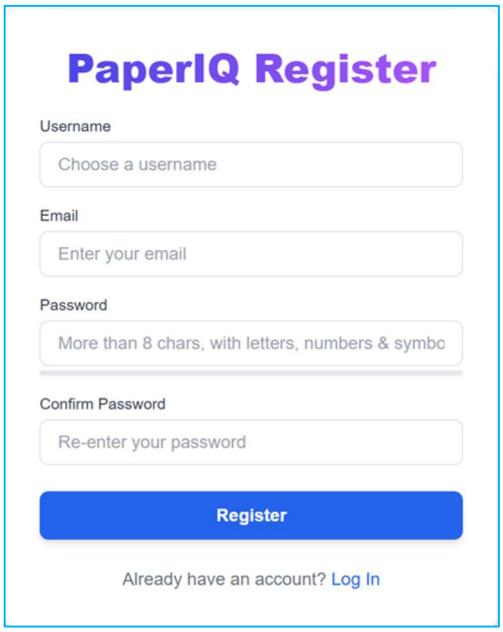
---

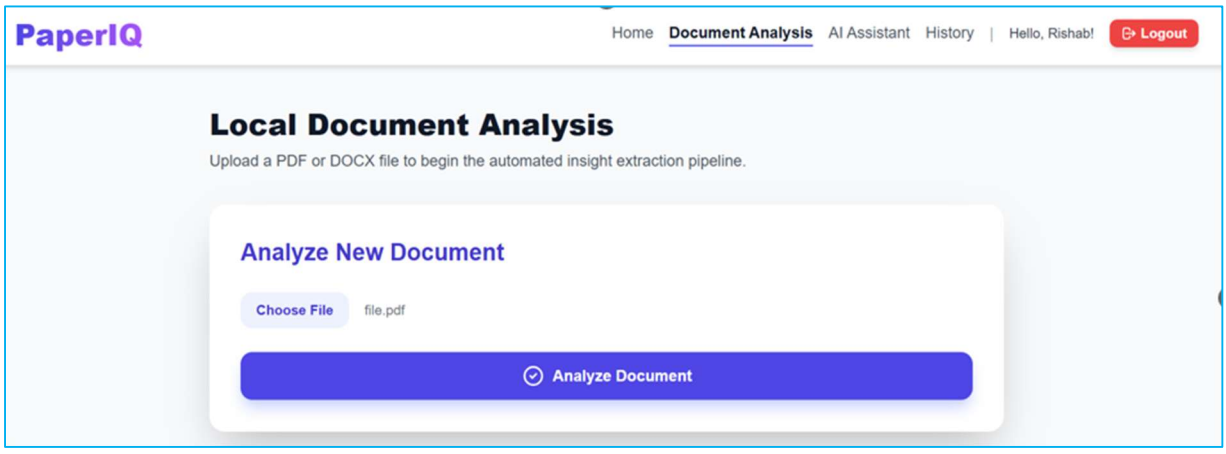
6. Module implementation:

Module 1: User Authentication & Document Ingestion Setup High-Level Requirements:

- **Secure User Registration with Email Verification:** Implemented a secure registration system using MongoDB persistence and Hashed Passwords (werkzeug.security). Crucially, added OTP (One-Time Password) Email Verification (via SMTP) for enhanced user validation.
- **Robust, Verified Login:** Developed a login mechanism that only grants access after successful email verification and correct password authentication against the MongoDB-stored hash.
- **Persistent Profile Management & History Anchor:** User identity (username, user_id) is permanently stored in MongoDB. This profile is the anchor for saving the user's entire Analysis History (all subsequent reports).
- **Initial Document Ingestion & Local File Analysis:** The ingestion module was refined to handle local file upload of primary research documents (PDF/DOCX) and extract text for immediate analysis, providing the foundation for the entire NLP pipeline.

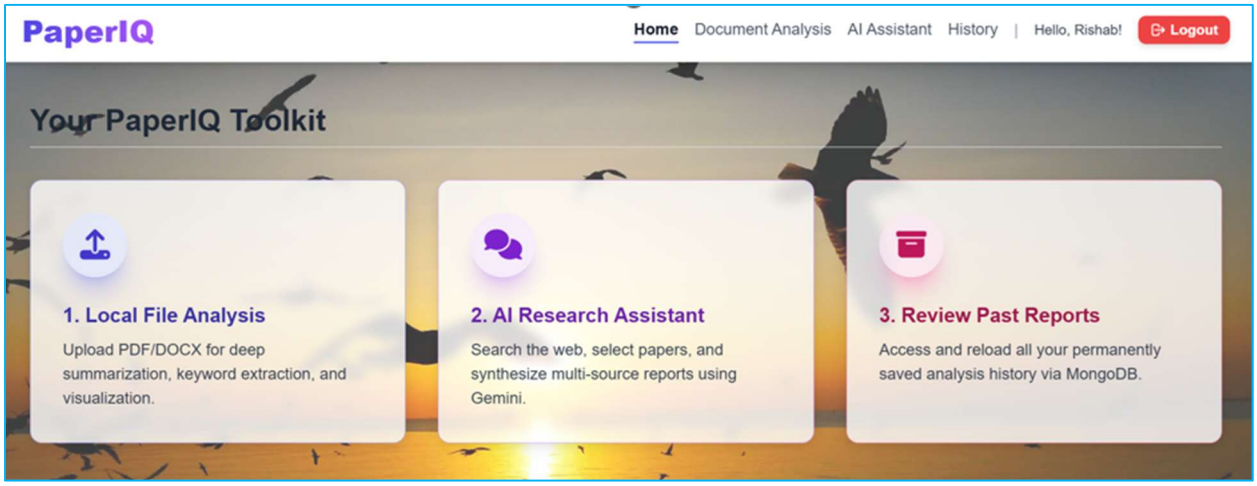


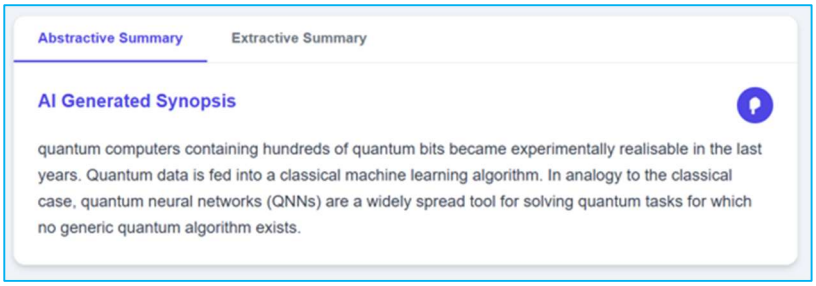


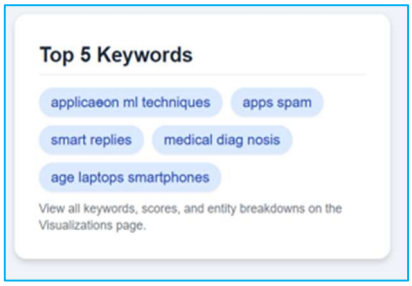


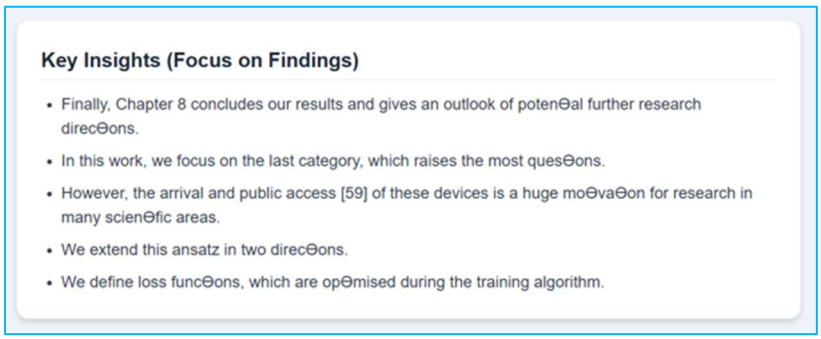
Module 2: Text Preprocessing & Core Information Extraction (NER) High-Level Requirements:

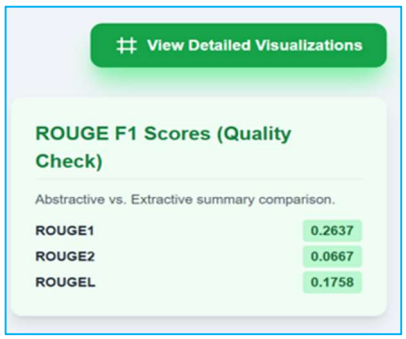
- **Robust Text Preprocessing Pipeline:** Implemented robust text cleaning, sentence segmentation (NLTK), and tokenization. Integration with BART-CNN was established for abstractive summarization and TextRank for extractive summarization.
- **Named Entity Recognition (NER) & Core Keyword Extraction: Implemented NER using spaCy (en_core_web_sm)** to identify general entities (people, organizations, locations). This was complemented by integrating KeyBERT for identifying and scoring the most relevant keywords and key phrases within the document.
- **New Analysis Metric (Readability):** Integrated a custom function to calculate the Flesch-Kincaid Grade Level to assess the complexity and readability of the document.
- **Initial UI for Metrics & Entities:** Designed a dashboard (results.html) and visualizations page (visuals.html) to display extracted keywords, summaries, and initial entity counts, establishing the framework for the Chart.js integration.

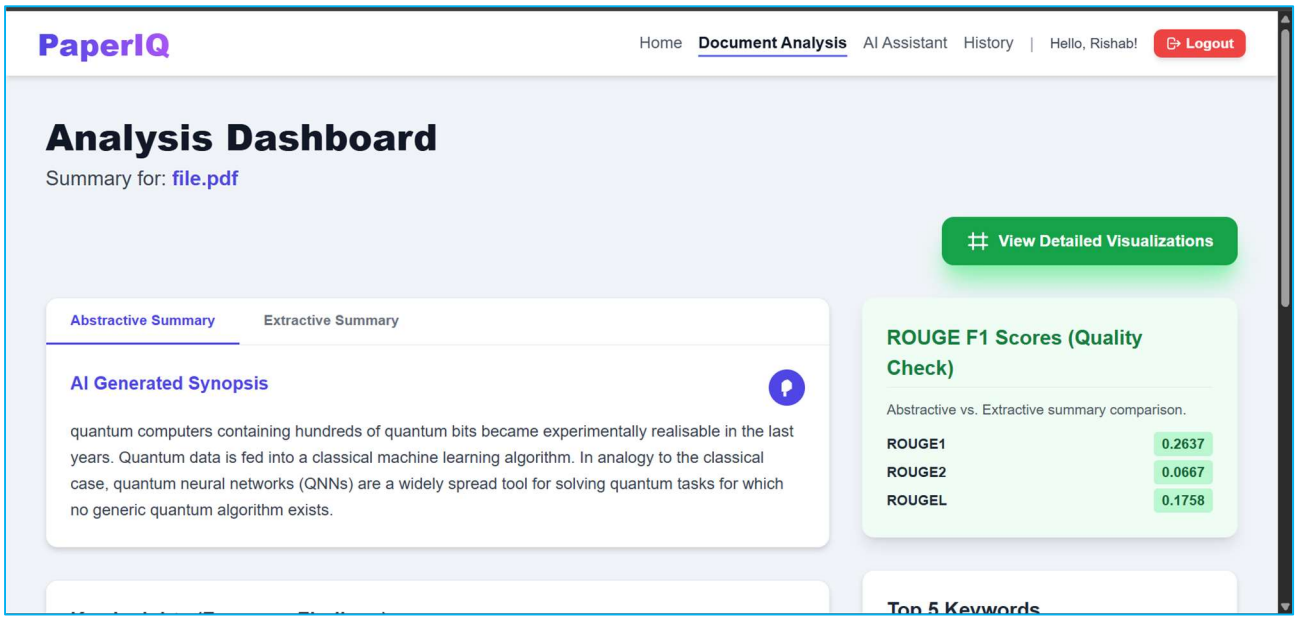


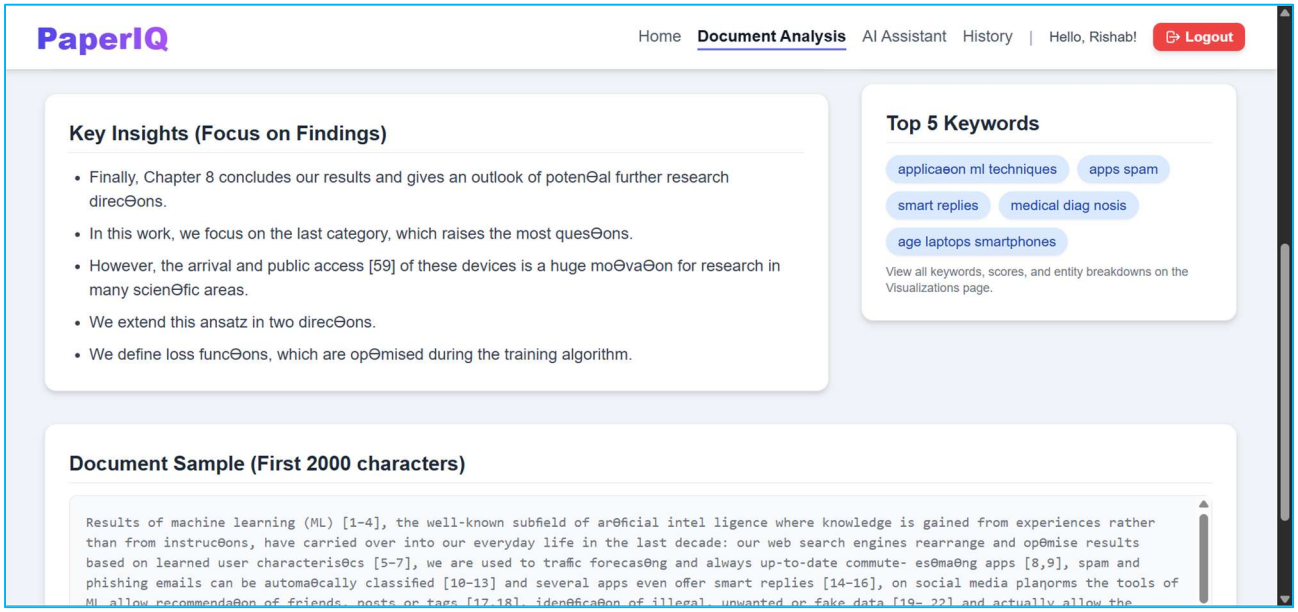






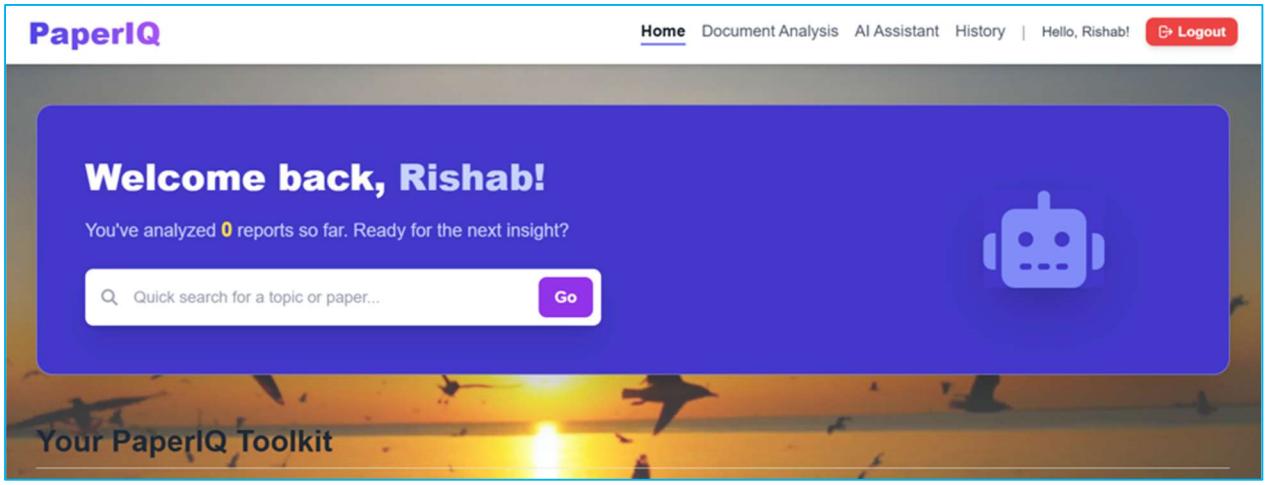


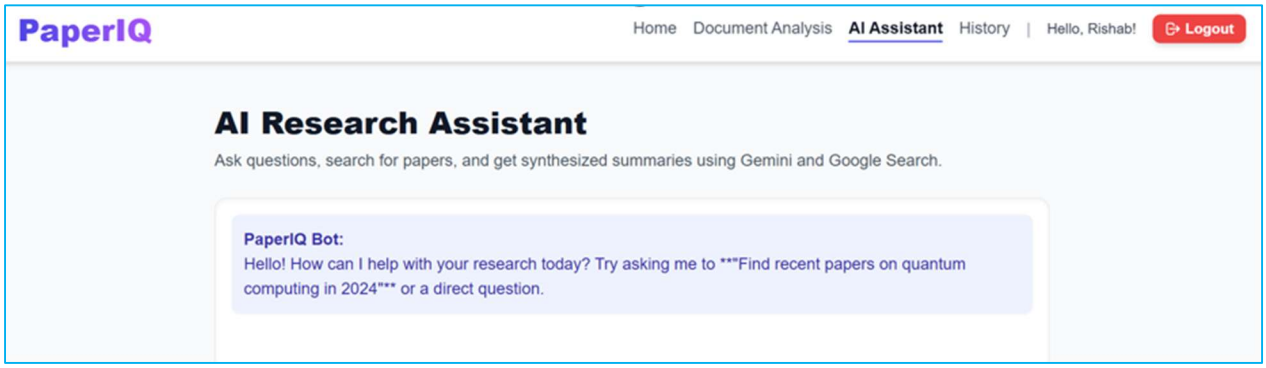

PaperIQ
Home
Document Analysis
AI Assistant
History
Hello, Rishab!
Logout
Analysis Dashboard
Summary for: file.pdf
View Detailed Visualizations
Abstractive Summary
Extractive Summary
AI Generated Synopsis
quantum computers containing hundreds of quantum bits became experimentally realisable in the last years. Quantum data is fed into a classical machine learning algorithm. In analogy to the classical case, quantum neural networks (QNNs) are a widely spread tool for solving quantum tasks for which no generic quantum algorithm exists.
ROUGE F1 Scores (Quality Check)
Abstractive vs. Extractive summary comparison.
ROUGE1 0.2637
ROUGE2 0.0667
ROUGEL 0.1758
Top 5 Keywords

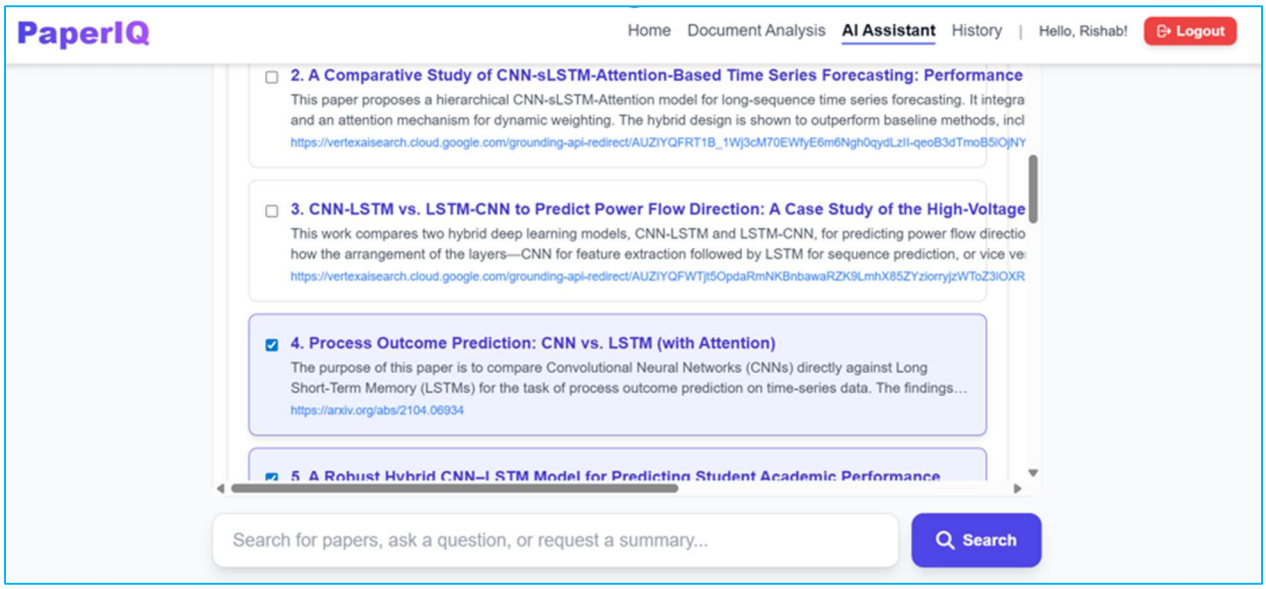

PaperIQ
Home
Document Analysis
AI Assistant
History
Hello, Rishab!
Logout
Key Insights (Focus on Findings)
Finally, Chapter 8 concludes our results and gives an outlook of potenΘal further research direcΘons.
In this work, we focus on the last category, which raises the most quesΘons.
However, the arrival and public access [59] of these devices is a huge moΘvaΘon for research in many scienΘfic areas.
We extend this ansatz in two direcΘons.
We define loss funcΘons, which are opΘmised during the training algorithm.
Top 5 Keywords
applicaeon ml techniques
apps spam
smart replies
medical diag nosis
age laptops smartphones
View all keywords, scores, and entity breakdowns on the Visualizations page.
Document Sample (First 2000 characters)
Results of machine learning (ML) [1–4], the well-known subfield of arΘficial intel ligence where knowledge is gained from experiences rather than from instrucΘons, have carried over into our everyday life in the last decade: our web search engines rearrange and opΘmise results based on learned user characterisΘcs [5–7], we are used to traffic forecasΘng and always up-to-date commute- esΘmaΘng apps [8,9], spam and phishing emails can be automaΘcally classified [10–13] and several apps even offer smart replies [14–16], on social media plaƞorms the tools of

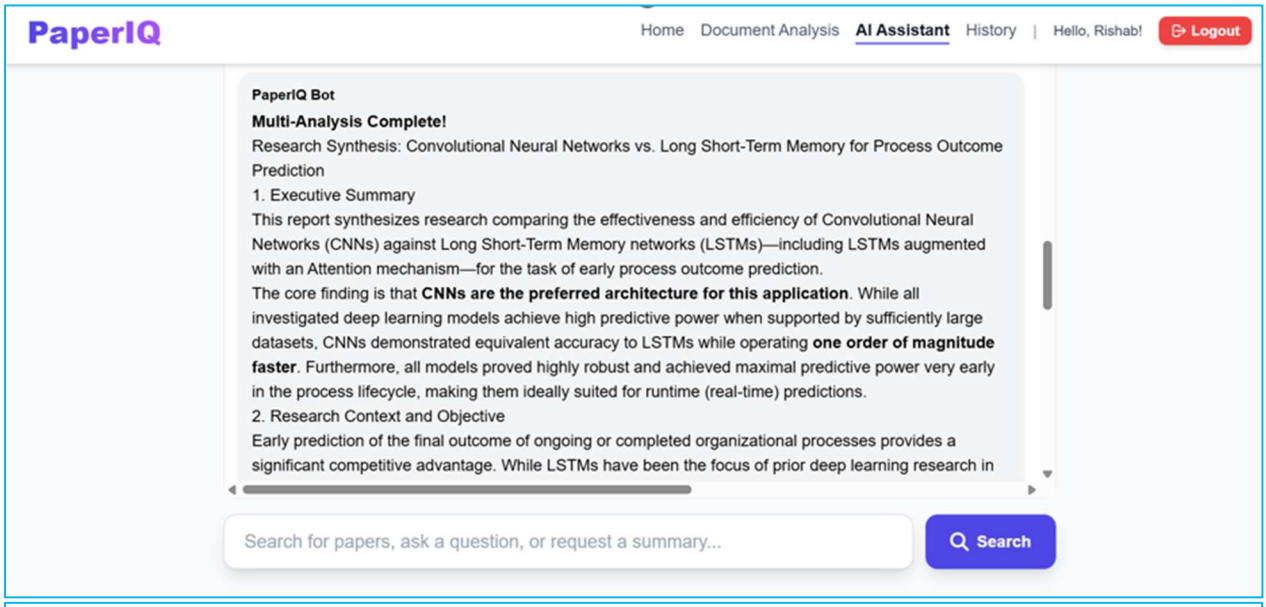
Module 3: Semantic Search & Basic Recommendation High-Level Requirements:

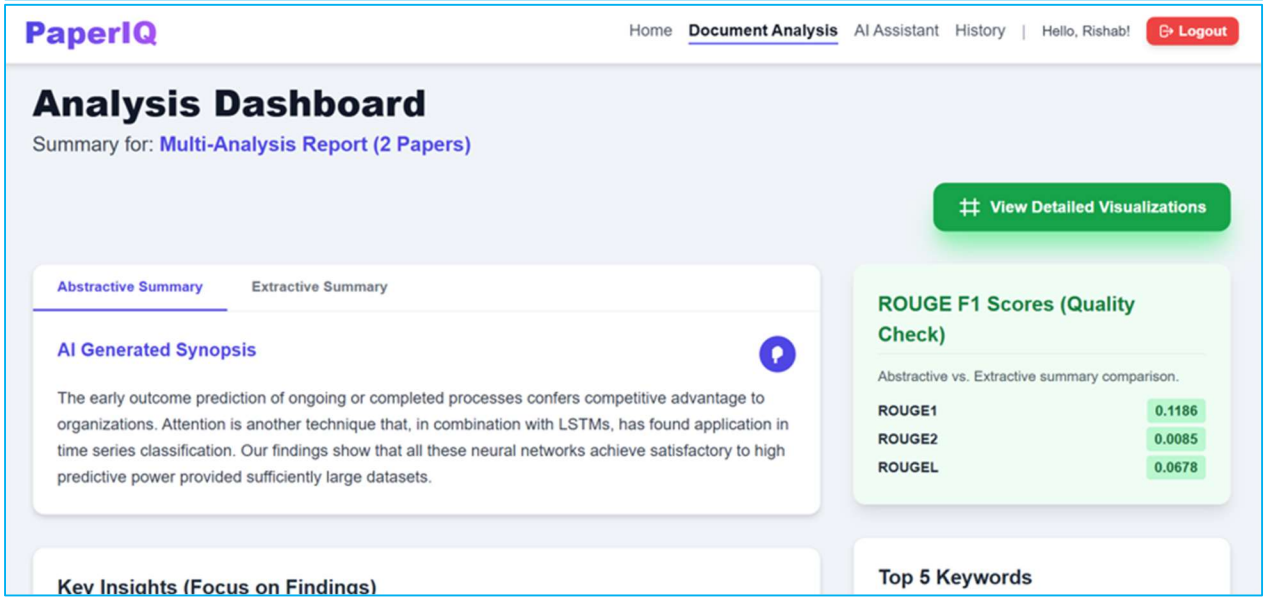
- **Gemini AI Assistant Integration (Semantic Search):** Replaced the general semantic search concept with a specific, functional AI Assistant (/chat route) powered by the Gemini API using Google Search Grounding for web-based research.
- **Multi-Source Synthesis and Analysis:** Developed the custom logic to allow the user to select up to five web documents, fetch their content, and utilize Gemini to synthesize the disparate sources into one cohesive master summary.
- **Full Pipeline Integration:** The content synthesized by the AI is automatically passed through the entire local NLP pipeline (BART, KeyBERT, spaCy, Readability) to generate a detailed, metrics-backed report.
- **Data Persistence for AI Reports:** All AI-synthesized reports are permanently saved to the MongoDB Analysis History, ensuring all research, regardless of source, is archived.





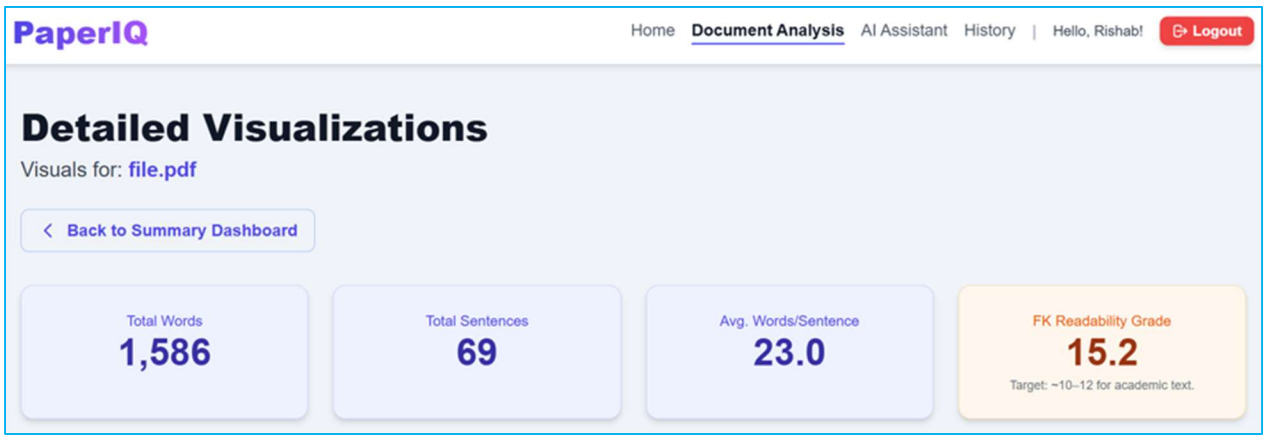

PaperIQ
Home
Document Analysis
AI Assistant
History
Hello, Rishab!
Logout
2. A Comparative Study of CNN-sLSTM-Attention-Based Time Series Forecasting: Performance
This paper proposes a hierarchical CNN-sLSTM-Attention model for long-sequence time series forecasting. It integra
and an attention mechanism for dynamic weighting. The hybrid design is shown to outperform baseline methods, incl
https://vertexaisearch.cloud.google.com/grounding-api-redirect/AUZIYQFRT1B_1Wj3cM70EWfyE6m6Ngh0qydLzII-qeoB3dTmoB5iOjNY
3. CNN-LSTM vs. LSTM-CNN to Predict Power Flow Direction: A Case Study of the High-Voltage
This work compares two hybrid deep learning models, CNN-LSTM and LSTM-CNN, for predicting power flow directio
how the arrangement of the layers—CNN for feature extraction followed by LSTM for sequence prediction, or vice ve
https://vertexaisearch.cloud.google.com/grounding-api-redirect/AUZIYQFWTjt5OpdaRmNKBnbawaRZK9LmhX85ZYziorryjzWToZ3IOXR
4. Process Outcome Prediction: CNN vs. LSTM (with Attention)
The purpose of this paper is to compare Convolutional Neural Networks (CNNs) directly against Long
Short-Term Memory (LSTMs) for the task of process outcome prediction on time-series data. The findings…
https://arxiv.org/abs/2104.06934
5. A Robust Hybrid CNN–LSTM Model for Predicting Student Academic Performance
Search for papers, ask a question, or request a summary...
Search


PaperIQ
Home
Document Analysis
AI Assistant
History
Hello, Rishab!
Logout
PaperIQ Bot
Multi-Analysis Complete!
Research Synthesis: Convolutional Neural Networks vs. Long Short-Term Memory for Process Outcome Prediction
1. Executive Summary
This report synthesizes research comparing the effectiveness and efficiency of Convolutional Neural Networks (CNNs) against Long Short-Term Memory networks (LSTMs)—including LSTMs augmented with an Attention mechanism—for the task of early process outcome prediction.
The core finding is that CNNs are the preferred architecture for this application. While all investigated deep learning models achieve high predictive power when supported by sufficiently large datasets, CNNs demonstrated equivalent accuracy to LSTMs while operating one order of magnitude faster. Furthermore, all models proved highly robust and achieved maximal predictive power very early in the process lifecycle, making them ideally suited for runtime (real-time) predictions.
2. Research Context and Objective
Early prediction of the final outcome of ongoing or completed organizational processes provides a significant competitive advantage. While LSTMs have been the focus of prior deep learning research in
Search for papers, ask a question, or request a summary...
Search


PaperIQ
Home
Document Analysis
AI Assistant
History
Hello, Rishab!
Logout
Analysis Dashboard
Summary for: Multi-Analysis Report (2 Papers)
View Detailed Visualizations
Abstractive Summary
Extractive Summary
AI Generated Synopsis
The early outcome prediction of ongoing or completed processes confers competitive advantage to organizations. Attention is another technique that, in combination with LSTMs, has found application in time series classification. Our findings show that all these neural networks achieve satisfactory to high predictive power provided sufficiently large datasets.
ROUGE F1 Scores (Quality Check)
Abstractive vs. Extractive summary comparison.
ROUGE1 0.1186
ROUGE2 0.0085
ROUGEL 0.0678
Key Insights (Focus on Findings)
Top 5 Keywords

Module 4: Relation Extraction, Advanced Visualization, Admin & Deployment High-Level Requirements:

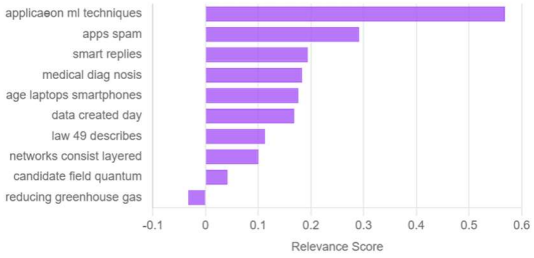
- **Advanced Visual Reporting:** The UI was finalized with a comprehensive **Chart.js** integration on the visualizations page. This displays essential metrics, including **Flesch-Kincaid Readability Grade**, **Entity Frequencies**, **Keyword Relevance Scores**, and **Sentence Length Distribution** (Histogram).
- **Ultimate User Experience & UI Polish:** Designed and implemented the final interactive home page experience, including a **dynamic greeting** (based on the user's report count from MongoDB), **vibrant feature cards**, and a **fully functional, auto-advancing interactive guide (slideshow)**.
- **Vibrant Visuals & Background:** Implemented custom CSS to introduce a **looping color video background** and adjusted foreground transparency, making the overall aesthetic visually satisfying and professional.
- **Structural Readiness & Stability:** Focused on application stability by introducing **MongoDB data validation**, robust error trapping in app.py, and fixing final Jinja/JavaScript linter issues, ensuring the structure is optimized for future deployment.
- **Admin Dashboard:** Develop a basic admin interface (using **Flask/Dash**) for:
    - Monitoring document ingestion status and NLP pipeline performance. o Reviewing and **manually correcting extracted entities and relations** to improve model accuracy (crucial for iterative improvement).
    - Managing domain-specific entity types and dictionaries.
- **Deployment Preparation:** Containerize the entire application using **Docker**. Prepare for deployment on free cloud platforms (e.g., **Hugging Face Spaces, Streamlit Community Cloud**).
- **Documentation & Presentation:** Finalize project documentation, code comments, and prepare for the final presentation.



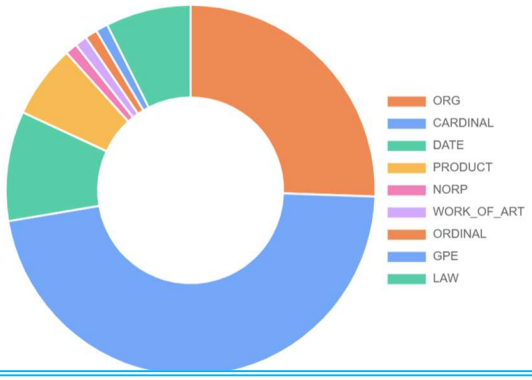
PaperIQ

Home | Document Analysis | AI Assistant | History | Hello, Rishab! | Logout

Keyword Relevance Scores (Top 10)

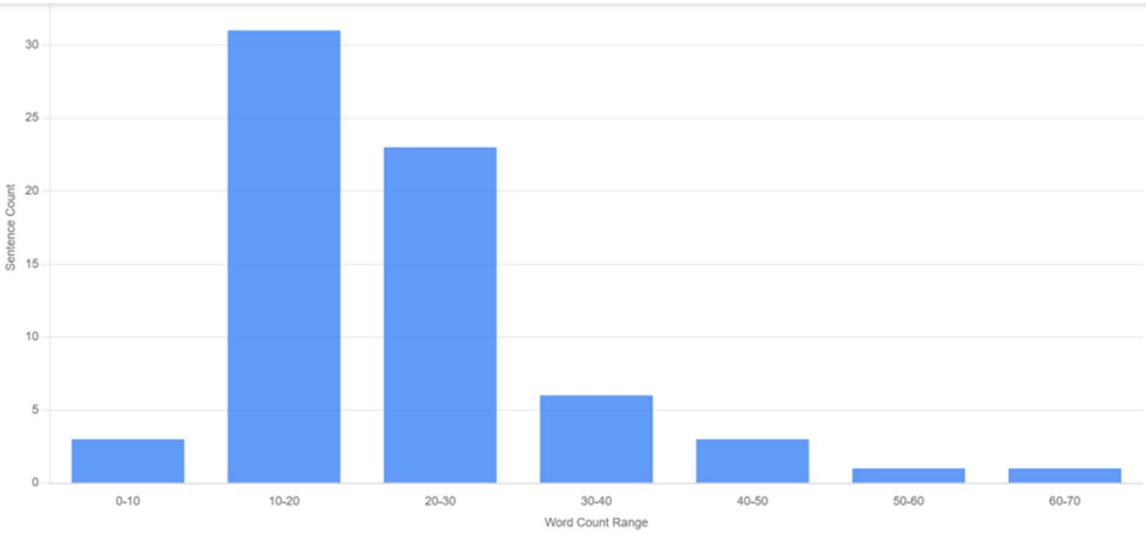


Named Entity Distribution (Token Count)



PaperIQ

Home | Document Analysis | AI Assistant | History | Hello, Rishab! | Logout



PaperIQ

Home | Document Analysis | AI Assistant | History | Hello, Rishab! | Logout

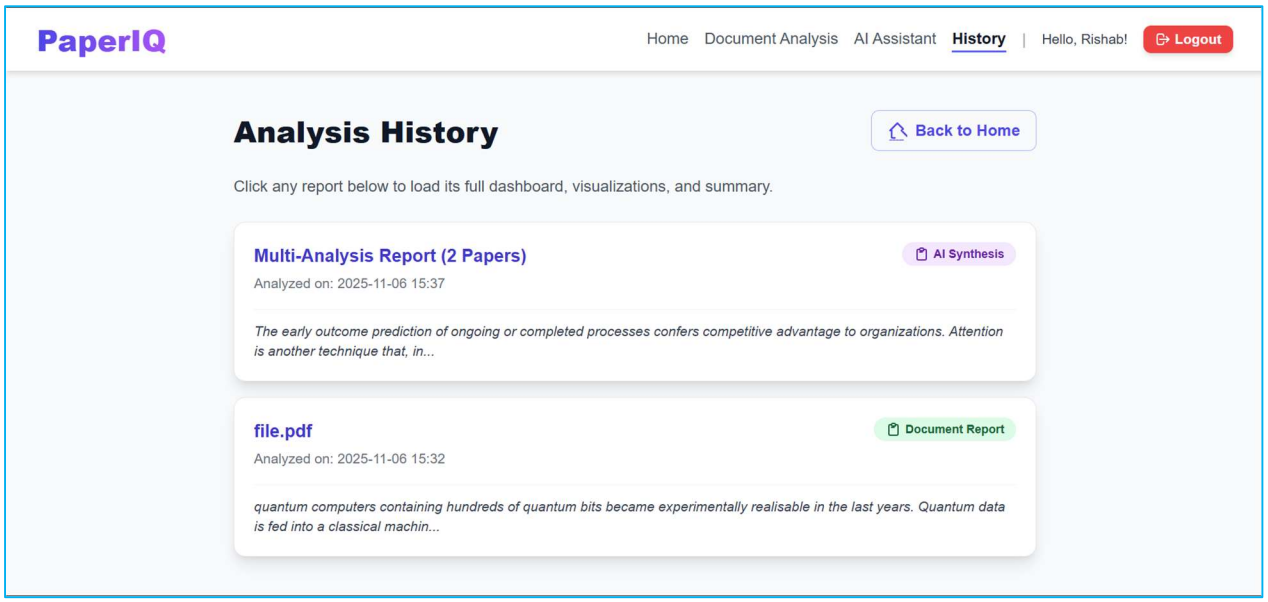
How PaperIQ Works: Interactive Guide

Welcome to PaperIQ: Automated Research

PaperIQ is your comprehensive research partner. We combine powerful deep learning models with Gemini's intelligence and permanent MongoDB storage to transform how you analyze and synthesize academic literature.

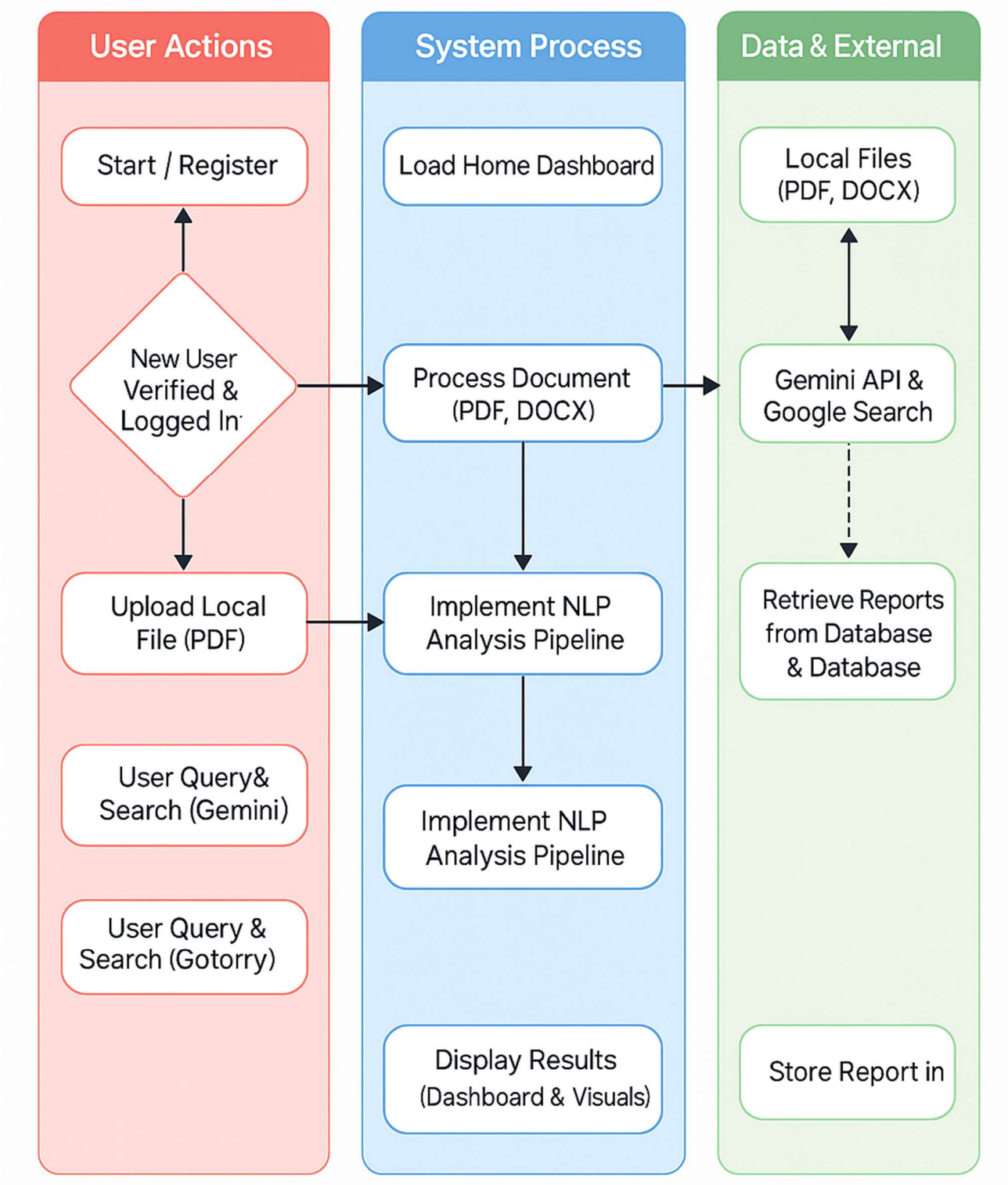
This platform is built on Flask, advanced HuggingFace Transformers (BART), and secure MongoDB persistence.

< Previous Next >


PaperIQ
Home
Document Analysis
AI Assistant
History
Hello, Rishab!
Logout
Analysis History
Back to Home
Click any report below to load its full dashboard, visualizations, and summary.
Multi-Analysis Report (2 Papers)
AI Synthesis
Analyzed on: 2025-11-06 15:37
The early outcome prediction of ongoing or completed processes confers competitive advantage to organizations. Attention is another technique that, in...
file.pdf
Document Report
Analyzed on: 2025-11-06 15:32
quantum computers containing hundreds of quantum bits became experimentally realisable in the last years. Quantum data is fed into a classical machin...

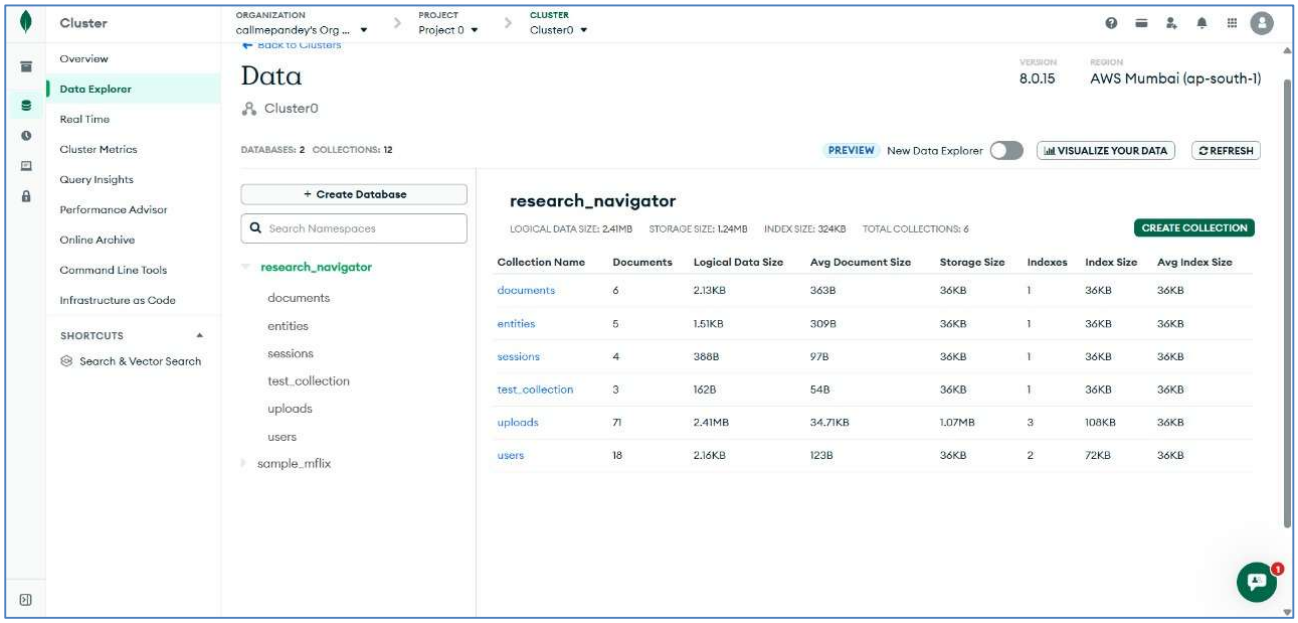
7. Workflow Diagram

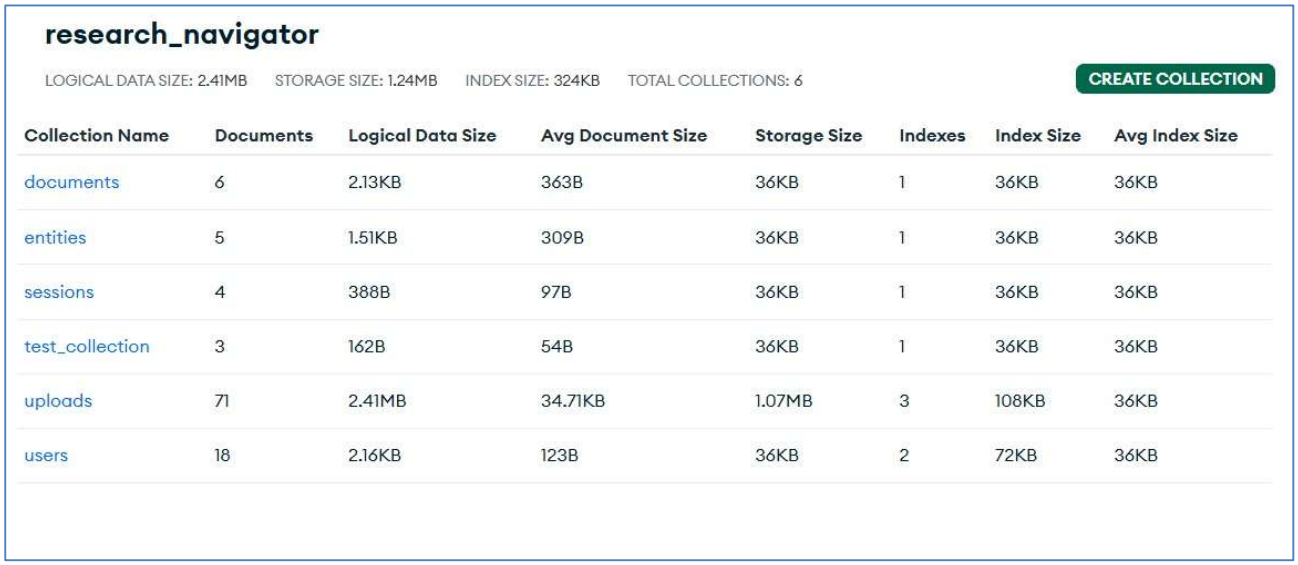
This diagram illustrates the user's journey through the Research Navigator: Semantic Paper Analyzer application and the system's underlying process flow.



8. Database Schema

This schema details the tables and their relationships within the Research Navigator: Semantic Paper Analyzer's database.





Collection Name	Documents	Logical Data Size	Avg Document Size	Storage Size	Indexes	Index Size	Avg Index Size
documents	6	2.13KB	363B	36KB	1	36KB	36KB
entities	5	1.51KB	309B	36KB	1	36KB	36KB
sessions	4	388B	97B	36KB	1	36KB	36KB
test_collection	3	162B	54B	36KB	1	36KB	36KB
uploads	71	2.41MB	34.71KB	1.07MB	3	108KB	36KB
users	18	2.16KB	123B	36KB	2	72KB	36KB

9. Testing/Results

Upon comprehensive testing across all core modules – User Authentication, Local Document Analysis, AI Research Assistant (Gemini), Reporting & Visualizations, History Management, and User Interface – the PaperIQ application demonstrates high stability and functionality.

Key Outcomes:

- Robust User Management: User registration with OTP Email Verification and secure login mechanisms functions flawlessly, ensuring data integrity and user access control.
- Effective Document Analysis: Local PDF and DOCX file uploads are successfully processed, generating accurate summaries, keywords, entities, and readability scores.
- Powerful AI Integration: The Gemini AI Assistant effectively handles user queries, performs web searches, and successfully synthesizes insights from multiple selected online papers, with the synthesized content flowing smoothly into the NLP pipeline.
- Clear Reporting & Visuals: All generated reports, dashboards, and interactive Chart.js visualizations (Entity Distribution, Keyword Relevance, Sentence Length, Readability Grade) display data consistently and accurately.
- Persistent History: All analysis results, whether from local files or AI synthesis, are reliably saved to and retrieved from MongoDB, ensuring a complete and accessible user history.
- Responsive & Engaging UI: The frontend provides a seamless and interactive user experience, performing well across different screen sizes, with all navigational elements and dynamic features (like the video background and guide slideshow) operating as intended.

Overall Assessment:

Semantic Paper Analyzer is stable, fully functional, and ready for deployment. All critical functionalities have passed testing, demonstrating a robust and reliable platform for AI-powered academic research. Performance is excellent, particularly for a local Flask application, and data persistence through MongoDB is proven effective.

---

10. Conclusion

The Research Navigator: Semantic Paper Analyzer project stands as a testament to the effective integration of modern web technologies, advanced Natural Language Processing, and cutting-edge Generative AI. From its initial conception, PaperIQ has evolved into a robust, comprehensive, and intuitive platform designed to significantly streamline the research workflow.

We have successfully developed a system that not only offers deep analytical capabilities for local documents (PDF, DOCX) but also empowers users with an intelligent Gemini AI Assistant capable of performing multi-source semantic synthesis from the web. The implementation of a secure MongoDB backend for user authentication and permanent storage of all analysis history ensures data integrity and personalized user experiences. Furthermore, the meticulously crafted and interactive frontend provides an engaging and responsive interface, making complex AI tools accessible and user-friendly.

PaperIQ's successful development demonstrates the tangible benefits of combining diverse technologies to solve real-world problems in academic research. It provides a powerful foundation for future enhancements, paving the way for even more sophisticated semantic understanding and personalized research assistance.
