# **PaperIQ AI Powered Research Insight Analyze**

# **Abstract**

In the era of information overload, quickly extracting relevant information from academic papers, reports, and technical documents is a critical challenge. This project presents a Document Analyzer Chatbot capable of processing PDF and DOCX documents, extracting structured sections, and providing intelligent responses to user queries. The system leverages Natural Language Processing (NLP) techniques including tokenization, lemmatization, stop-word removal, and part-of-speech tagging to preprocess textual data. Query understanding is performed using regex-based topic extraction and TF-IDF similarity scoring for paragraph retrieval. Furthermore, the system incorporates a rule-based Named Entity Recognition (NER) module to extract tasks, methods, datasets, results, and conclusions from research documents. A user-friendly Streamlit interface provides interactive upload, chat, and summarization capabilities. This tool aims to reduce human effort in document analysis and enhance research efficiency.

## **Introduction**

With the exponential growth of research publications, technical reports, and structured documents, manually analyzing content and retrieving specific information has become a tedious task. Traditional keyword-based search methods often fail to consider context, relevance, and semantic similarity.

The Document Analyzer Chatbot addresses these challenges by:

1. Automatically extracting textual content from PDF and DOCX files.
2. Segmenting documents into structured sections and paragraphs.
3. Understanding user queries using regex-based topic extraction and semantic similarity scoring.
4. Delivering concise, context-aware responses relevant to user questions.
5. Performing entity recognition to identify key elements such as methods, datasets, results, and conclusions.

This system is particularly useful for researchers, students, and professionals who need quick access to document insights without manually reading entire texts.

## **3. Objectives**

The main objectives of the Document Analyzer Chatbot are:

1. To enable easy upload and processing of PDF and DOCX files.
2. To preprocess and clean textual content for efficient analysis.
3. To segment documents into meaningful sections and paragraphs.
4. To retrieve relevant paragraphs using a combination of regex patterns and TF-IDF similarity scoring.
5. To extract task, method, dataset, result, and conclusion entities using regex-based NER.
6. To provide interactive user responses via a web interface.
7. To allow document summarization based on TF-IDF sentence scoring.

## **4. Literature Review**

Existing approaches for document information retrieval and question answering (QA) can be categorized into:

1. Keyword-based search: Simple string matching, prone to low recall and poor semantic understanding.
2. Vector-based retrieval: Using embeddings (TF-IDF, Word2Vec, or transformer-based embeddings) to capture semantic similarity between queries and text.
3. Extractive QA systems: Models like BERT or SpaCy extract answers directly from paragraphs.
4. Rule-based extraction: Regex or pattern-based extraction for specific entities (e.g., method names, datasets, accuracy).

While modern pretrained LLM-based QA systems offer high accuracy, they are computationally heavy. This project implements a hybrid approach combining regex rules and TF-IDF-based retrieval, allowing lightweight, fast, and interpretable extraction from documents without large pretrained models.

## **5. Methodology**

The system is divided into multiple functional modules:

### **5.1 Document Ingestion**

The system supports PDF and DOCX files:

* PDF Extraction: Using PyMuPDF (fitz) to read text from pages, preserving paragraph structure.
* DOCX Extraction: Using python-docx to read paragraph-level text.

The extracted text is preprocessed to remove extra spaces, numbers, and irrelevant characters.

### **5.2 Text Preprocessing**

Preprocessing is crucial to ensure that queries and document text are comparable. Steps include:

1. Sentence Tokenization: Splitting paragraphs into sentences using nltk.sent\_tokenize().
2. Word Tokenization: Splitting sentences into words.
3. Stop-word Removal: Eliminating common English words that do not add semantic meaning.
4. Lemmatization: Converting words to their base form using POS-aware lemmatization.
5. Numeric Cleaning: Removing standalone numbers and digits in parentheses.

These steps reduce noise and improve semantic similarity calculations.

### **5.3 Section and Paragraph Extraction**

Documents often follow a structured format (abstract, introduction, methodology, etc.).

* The system identifies sections using:
  + All-uppercase headings (e.g., "INTRODUCTION")
  + Lines ending with a colon (:)
* Each section is cleaned, and text is split into paragraphs of roughly 200 characters, maintaining sentence integrity.
* This structured representation allows efficient retrieval for user queries.

### **5.4 Query Understanding**

To interpret user questions, the system employs:

1. Regex-based topic extraction: Removes common words like "what is", "define", and prepositions to identify the main keyword/topic of interest.
2. Query classification: Determines query type (definition, features, working, literature review, general) to adapt the response.

### **5.5 Paragraph Retrieval**

Paragraph retrieval uses a hybrid approach:

#### **A. Regex-based Matching**

* Predefined regex patterns identify paragraphs that explicitly contain the topic:  
  + Definition patterns: "topic is/are..."
  + Feature patterns: "key features of topic include..."
  + Section matches: entire paragraph sections labeled by headings.
* Matches are scored based on type:  
  + 100: Complete section match
  + 80: Definition match
  + 60: "is/are" pattern match
  + 40: Sentence start match
  + 20: General occurrence count

#### **B. TF-IDF Similarity Matching**

* Both the query and document sentences are preprocessed and vectorized using TfidfVectorizer.
* Cosine similarity is calculated between the query and each sentence.
* The highest-scoring sentence is selected as the candidate answer if it exceeds a similarity threshold (default 0.15).

#### **C. Response Formatting**

* Extracted paragraph sentences are selectively included based on query type:  
  + Definitions: 2-3 sentences
  + Features: up to 6 sentences
  + Process/working: up to 8 sentences
  + Literature review: up to 10 sentences
* Responses are then returned in natural, human-readable form.

### **5.6 Metadata Handling**

### The system tracks page numbers, section names, and paragraph indices for all extracted text. Key entities like METHOD, DATASET, RESULT, and CONCLUSION are also tagged using rule-based detection. This metadata enables section-aware retrieval, entity-based queries, and precise citation of source content.

### **5.7 Rule-based Named Entity Recognition (NER)**

The system identifies key entities using regex rules:

| Entity Type | Detection Patterns |
| --- | --- |
| TASK | "method", "approach", "technique", "framework” |
| METHOD | "method", "approach", "technique", "framework” |
| DATASET | Predefined popular datasets: MNIST, ImageNet, CIFAR, COCO |
| RESULT | "achieved", "accuracy of", "performance of" |
| CONCLUSION | "in conclusion", "we conclude", "to summarize” |
| ABSTRACT | Lines following the keyword "Abstract” |

This allows entity-based query answering, e.g., “What methods are used?” or “Which datasets are mentioned?”

### **6. Summarization**

* The system supports extractive summarization using a TF-IDF-based sentence scoring:  
  1. Sentences are tokenized and cleaned.
  2. Vocabulary IDF values are computed.
  3. Each sentence receives a TF-IDF weighted score.
  4. Top sentences (by score) are selected to form a summary (default 30% of total sentences).

This gives users a concise overview of long documents.

## 

## **7. Future Work**

While the current system provides reliable document analysis, several enhancements can further improve functionality and user experience:

1. Embedding-based semantic retrieval
   * Integrate word or sentence embeddings (e.g., Sentence-BERT) to capture deeper semantic meaning and improve similarity scoring beyond TF-IDF.
2. Multi-document support
   * Allow uploading and querying multiple documents simultaneously for cross-document search and comparisons.
3. Query-specific summarization
   * Implement extractive or abstractive summarization focused on user queries, not just general summaries.
4. Advanced entity extraction
   * Incorporate machine learning-based NER models to recognize custom entities beyond predefined patterns (e.g., novel methods, chemical compounds, financial metrics).
5. Interactive feedback loop
   * Allow users to rate answers or provide feedback to improve retrieval and ranking over time.
6. Multilingual support
   * Extend preprocessing, tokenization, and retrieval to other languages, allowing global applicability.

By implementing these enhancements, the Document Analyzer Chatbot can evolve into a comprehensive, intelligent research assistant, supporting academic, technical, and business document analysis at scale.

**8. Key achievements**

Key achievements of the project include:

1. Automated document ingestion: Supports PDFs and Word documents with accurate text extraction and paragraph segmentation.
2. Preprocessing for NLP: Incorporates tokenization, lemmatization, stop-word removal, and POS-aware processing to improve retrieval performance.
3. Robust query understanding: Uses topic extraction and query classification to handle diverse user queries such as definitions, features, workings, and literature review.
4. Hybrid paragraph retrieval: Combines regex-based matching with TF-IDF similarity scoring for accurate and interpretable answer retrieval.
5. Entity recognition: Identifies tasks, methods, datasets, results, and conclusions to enrich information retrieval.
6. User-friendly interface: Interactive Streamlit frontend enables file upload, chat-based querying, and summarization.

Overall, the system demonstrates how lightweight NLP techniques can be effectively combined to perform document-level question answering without relying on heavy pretrained models.

**9. Conclusion**

The Document Analyzer Chatbot provides an efficient and intelligent solution for extracting, understanding, and summarizing information from PDF and DOCX documents. By leveraging a hybrid approach that combines regex-based topic detection, TF-IDF semantic similarity, and rule-based entity extraction, the system is able to deliver context-aware responses to user queries.