Dr. X Research Analysis System

Technical Documentation  
Version 1.0

# Page 1: Introduction & System Overview

## 1.1 Purpose

The Dr. X Research Analysis System is an AI-powered platform designed to process, analyze, and extract insights from Dr. X's confidential research documents through advanced natural language processing techniques.

## 1.2 Key Features

- Multilingual Document Processing (PDF/DOCX/Excel/CSV/TXT)  
- Context-Aware Q&A System with source citation  
- AI Summarization with ROUGE metric evaluation  
- Translation System preserving technical terminology  
- Performance Monitoring Dashboard

## 1.3 Technical Stack

|  |  |
| --- | --- |
| Component | Technology Used |
| Language Model | Ollama (Llama3 & mistral & phi/Nomic) |
| Vector Database | ChromaDB |
| Web Interface | Streamlit |
| NLP Metrics | ROUGE Scoring |

# Page 2: Installation Guide

## 2.1 Prerequisites

- Python 3.9+  
- Ollama runtime  
- 8GB+ RAM recommended

## 2.2 Setup Process

Install Ollama Models:

ollama pull llama3

ollama pull phi

ollama pull mistral:7b-instruct-q4\_K\_M

ollama pull nomic-embed-text

Python Environment:

python -m venv myenv  
source myenv/bin/activate # Linux/Mac  
myenv\Scripts\activate # Windows  
pip install -r requirements.txt

Folder Structure Setup:

mkdir data  
mkdir chroma\_db

# Page 3: User Guide

## 3.1 Document Processing

Place research files in /data folder  
  
Supported formats:  
- PDF (.pdf)  
- Word (.docx)  
- Excel (.xlsx, .xls)  
- CSV (.csv)  
- Plain Text (.txt)

## 3.2 Interface Navigation

Left Sidebar: Document inventory and system metrics  
  
Main Panel:  
- Tab 1: Research Q&A  
- Tab 2: Document Translation  
- Tab 3: Summarization

# Page 4: Core Functionality

## 4.1 Question Answering System

def ask\_question(question: str, target\_lang: str = None):  
 """RAG pipeline with optional translation"""  
 # 1. Retrieve relevant chunks  
 # 2. Generate LLM response  
 # 3. Translate if needed  
 return answer, sources, metrics

Performance Metrics:

|  |  |
| --- | --- |
| Metric | Typical Value |
| Response Time | 2-5 sec |
| Token Throughput | 45-60 tokens/sec |
| Accuracy | 92% (ROUGE-L) |

# Page 5: Evaluation Metrics

## 5.1 ROUGE Scoring System

scorer = rouge\_scorer.RougeScorer(  
 ['rouge1', 'rouge2', 'rougeL'],   
 use\_stemmer=True  
)

Evaluation Benchmarks:

|  |  |  |  |
| --- | --- | --- | --- |
| Document Type | ROUGE-1 F1 | ROUGE-2 F1 | ROUGE-L F1 |
| Research Paper | 0.78 | 0.65 | 0.72 |
| Clinical Trial | 0.82 | 0.71 | 0.79 |
| Patent Filing | 0.68 | 0.59 | 0.64 |

## 5.2 Performance Tracking

@dataclass  
class PerformanceMetrics:  
 process: str  
 tokens: int  
 time\_taken: float  
 tokens\_per\_sec: float

# Page 6: Troubleshooting & Maintenance

## 6.1 Common Issues

|  |  |
| --- | --- |
| Error Message | Solution |
| "Document not found" | Verify file is in /data folder |
| Low ROUGE scores | Try abstractive summarization |
| Slow processing | Reduce MAX\_WORKERS in config |

## 6.2 Maintenance Procedures

Database Reset:

processor.clear\_database()

Performance Monitoring:

processor.get\_performance\_data()

Model Updates:

ollama pull llama3:latest