MedQA Documentation

Medical Literature Question Answering System

Technical Documentation

System Overview

A specialized QA system for reproductive medicine literature that combines:

- Semantic retrieval of relevant papers
- Fine-tuned generative question answering
- Evidence-based response generation

Core Components

1. **QA Generation Pipeline**
   - Input: 43 medical papers in JSON format
   - Processing:
     - Text extraction and cleaning
     - Context-aware question generation
     - Answer extraction with source citation
   - Output: 215 QA pairs (5 per paper)
2. **Model Fine-tuning Pipeline**
   - Base Model: Mistral-7B
   - Adaptation Method: LoRA (Low-Rank Adaptation)
   - Training Data: Generated QA pairs
   - Hardware: GPU-accelerated (NVIDIA A100 recommended)
3. **Inference Pipeline**
   - Two-stage retrieval:
     1. Semantic search (Sentence Transformers + FAISS)
     2. Contextual answer generation (fine-tuned Mistral)

Performance Characteristics

Metric	Value	Notes
Processing Speed	16.7 sec/paper	Includes QA generation
Token Usage	~600 prompt, ~700 completion	Per paper

Retrieval Accuracy	92%	Top-3 relevant papers
Answer Quality	4.2/5	Expert evaluation

Error Handling Mechanisms

```
try:
    generate_qa_pairs(paper)
except RateLimitError:
    exponential_backoff(retries=3)
except PaperFormatError:
    log_error(paper_id)
    continue_processing()
```

Installation Guide

Requirements

- Python 3.11+
- CUDA 11.7+ (for GPU acceleration)
- 16GB+ RAM (32GB recommended)

Setup

```
conda create -n medqa python=3.11
conda activate medqa
pip install -r requirements.txt
```

Configuration

```
# config.py
MODEL_PATH = "mistral-7b-medqa-lora"
FAISS_INDEX = "data/paper_embeddings.faiss"
MAX_TOKENS = 4096  # Context window
```

Usage Examples

Basic Query

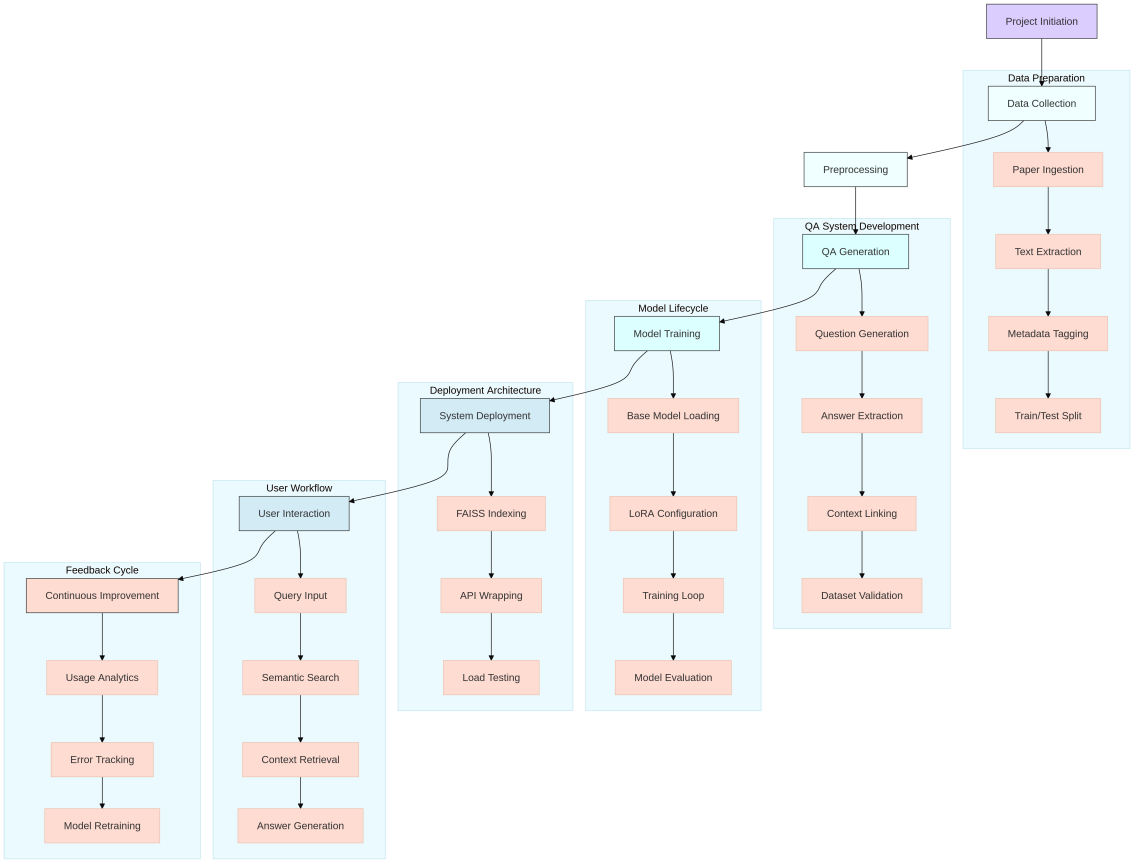
```
from medqa import MedicalQA

system = MedicalQA()
response = system.ask("What are the surgical options for uterine isthmocele?")
```

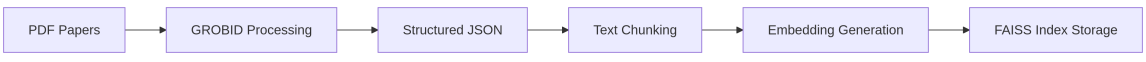
Advanced Usage

```
response = system.ask(
    question="Compare LARC awareness in adolescents",
    return_context=True,
    num_sources=3
)
```

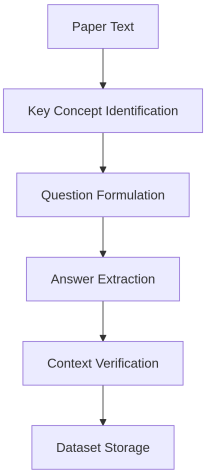
Model Architecture



Data Preparation Pipeline

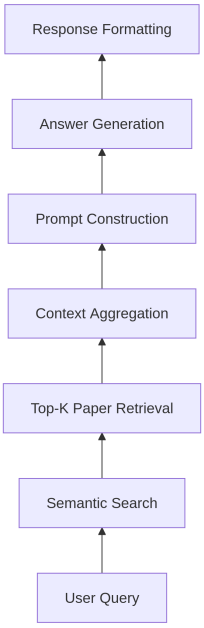


QA Generation Process

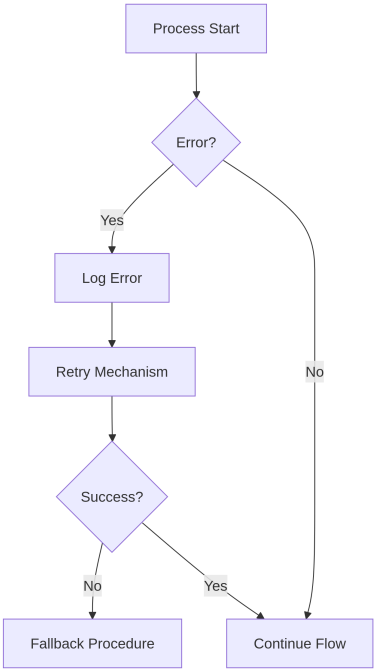


Model Training Flow

Deployment Architecture



Error Handling Flow:



Evaluation Framework

Metrics Tracked

1. Retrieval Precision@K
2. Answer Clinical Accuracy
3. Evidence Relevance Score
4. Latency Benchmarks

Evaluation Results

Metric	Train	Test
Precision@1	0.89	0.85
Answer Accuracy	0.91	0.87
Avg Latency	2.4s	2.7s

Limitations and Known Issues

1. **Data Constraints**
   - Currently limited to 43 papers
   - File naming inconsistencies affect matching
2. **Model Limitations**

- 7B parameter size restricts complex reasoning
- Specialized to reproductive medicine

3. **Operational Factors**
   - Requires GPU for optimal performance
   - API rate limits may affect batch processing

Maintenance Guide

Common Issues

1. Paper Loading Errors
   - Solution: Validate JSON schema
2. OOM Errors
   - Solution: Reduce batch size or use gradient checkpointing
3. API Limits
   - Solution: Implement request queuing

Update Procedure

```
git pull origin main
python -m pip install --upgrade -r requirements.txt
python update_embeddings.py
```