Installation Guide

This guide will walk you through setting up the RAG System step-by-step.

Quick Start (Linux/macOS)

```
# 1. Navigate to project directory
cd /home/ubuntu/rag_system

# 2. Run setup script
chmod +x setup.sh
./setup.sh

# 3. Download Llama 3 model
source venv/bin/activate
huggingface-cli download TheBloke/Llama-3-8B-Instruct-GGUF llama-3-8b-in-
struct.Q4_K_M.gguf --local-dir models --local-dir-use-symlinks False

# 4. Start the server
./run.sh
```

Detailed Installation

1. System Requirements

- **OS**: Linux (Ubuntu 20.04+), macOS 10.15+, or Windows with WSL2
- **Python**: 3.11 or higher
- **RAM**: 16GB minimum (32GB recommended)
- **Storage**: 10GB free space
- **GPU** (Optional): NVIDIA GPU with 8GB+ VRAM

2. Install Python 3.11+

Ubuntu/Debian:

```
sudo apt update
sudo apt install python3.11 python3.11-venv python3-pip
```

macOS:

```
brew install python@3.11
```

Windows (WSL2):

```
sudo apt update
sudo apt install python3.11 python3.11-venv python3-pip
```

3. Clone or Navigate to Project

```
cd /home/ubuntu/rag_system
```

4. Create Virtual Environment

```
python3.11 -m venv venv
source venv/bin/activate  # On Windows WSL: source venv/bin/activate
```

5. Install Python Dependencies

```
pip install --upgrade pip
pip install -r requirements.txt
```

If installation fails, try installing packages individually:

```
pip install fastapi uvicorn[standard]
pip install pypdf2 pdfplumber beautifulsoup4 requests
pip install langchain langchain-community
pip install sentence-transformers chromadb
pip install llama-cpp-python
```

6. Install GPU Support (Optional)

If you have an NVIDIA GPU with CUDA:

```
# Install CUDA-enabled llama-cpp-python
CMAKE_ARGS="-DLLAMA_CUBLAS=on" pip install llama-cpp-python --force-reinstall --no-
cache-dir
```

7. Download Llama 3 Model

Method 1: Using Hugging Face CLI (Recommended)

```
# Install huggingface-hub if not already installed
pip install huggingface-hub[cli]

# Create models directory
mkdir -p models

# Download the model (choose one):

# Option A: Llama 3 8B Q4_K_M (~4.9GB) - Recommended
huggingface-cli download TheBloke/Llama-3-8B-Instruct-GGUF llama-3-8b-in-
struct.Q4_K_M.gguf --local-dir models --local-dir-use-symlinks False

# Option B: Llama 3 8B Q5_K_M (~5.9GB) - Better quality
huggingface-cli download TheBloke/Llama-3-8B-Instruct-GGUF llama-3-8b-in-
struct.Q5_K_M.gguf --local-dir models --local-dir-use-symlinks False

# Option C: Llama 3 8B Q8_0 (~8.5GB) - Best quality
huggingface-cli download TheBloke/Llama-3-8B-Instruct-GGUF llama-3-8b-in-
struct.Q8_0.gguf --local-dir models --local-dir-use-symlinks False
```

Method 2: Manual Download

1. Visit: https://huggingface.co/TheBloke/Llama-3-8B-Instruct-GGUF
2. Click on “Files and versions”
3. Download your chosen .gguf file
4. Move it to the `models/` directory

8. Configure the System

Edit `config.yaml` to match your model:

```yaml
llm:
  model_path: "./models/llama-3-8b-instruct.Q4_K_M.gguf"  # Update if using different model
  n_gpu_layers: 35  # Set to 0 if no GPU
  n_threads: 8      # Set to your CPU core count
```

9. Start the Server

```
# Make run script executable
chmod +x run.sh

# Start the server
./run.sh
```

Or manually:

```
python -m uvicorn backend.main:app --host 0.0.0.0 --port 8000
```

10. Access the Application

Open your browser and navigate to:
- **Web Interface**: http://localhost:8000
- **API Documentation**: http://localhost:8000/docs

Verification

Test your installation:

1. **Check Health**: Visit http://localhost:8000/api/health
2. **Upload a Test PDF**: Use the web interface to upload a PDF
3. **Ask a Question**: Try querying the uploaded document

Platform-Specific Notes

macOS (Apple Silicon M1/M2)

```
# llama-cpp-python has native Metal support
CMAKE_ARGS="-DLLAMA_METAL=on" pip install llama-cpp-python --force-reinstall --no-
cache-dir

# In config.yaml, adjust:
llm:
  n_gpu_layers: 1  # Metal will handle GPU offloading
```

Windows (Native)

1. Install Python from python.org
2. Open PowerShell as Administrator
3. Navigate to project directory
4. Create virtual environment:
   `powershell`
   `python -m venv venv`
   `.\venv\Scripts\Activate.ps1`
5. Install dependencies:
   `powershell`
   `pip install -r requirements.txt`
6. Download model using same commands as Linux
7. Start server:
   `powershell`
   `python -m uvicorn backend.main:app --host 0.0.0.0 --port 8000`

Troubleshooting Installation

Issue: `pip install` fails with compiler errors

Solution: Install build tools:

```
# Ubuntu/Debian
sudo apt install build-essential python3-dev

# macOS
xcode-select --install
```

Issue: ChromaDB installation fails

Solution: Install SQLite development files:

```
# Ubuntu/Debian
sudo apt install libsqlite3-dev

# macOS (usually not needed)
brew install sqlite
```

Issue: Out of disk space

Solution:
- Clear pip cache: `pip cache purge`
- Remove unused packages: `pip autoremove`
- Ensure at least 10GB free space

Issue: Model download is slow

Solution:
- Use a download manager
- Download from mirror sites
- Use torrent links if available

Issue: Permission denied on scripts

Solution:

```
chmod +x setup.sh run.sh
```

Next Steps

After successful installation:

1. Read the README.md (README.md) for usage instructions
2. Try the example queries
3. Configure settings in `config.yaml` for your needs
4. Check out the API documentation at `/docs`

Getting Help

If you encounter issues:

1. Check logs: `tail -f logs/rag_system.log`
2. Review the Troubleshooting (README.md#troubleshooting) section
3. Ensure all prerequisites are met
4. Verify model file exists: `ls -lh models/`

Happy Installing! 🚀