

## AI-Enhanced IT Helpdesk

### Overview

AI-Enhanced IT Helpdesk is an intelligent system that combines semantic retrieval with query refinement to provide precise answers to IT and networking questions. By leveraging advanced NLP techniques and large language models, the system transforms raw documentation into a smart knowledge base that understands the intent behind user queries.

### Key Features

- **Semantic Search:** Understand the meaning behind queries, not just keywords
- **Query Refinement:** Automatically improve user queries for better search results
- **Document Processing:** Extract structured information from Word documents
- **Retrieval-Augmented Generation (RAG):** Generate comprehensive answers using both retrieved context and LLM capabilities
- **Link Generation:** Create query-specific links to additional resources
- **Structured Output:** Provide answers with detailed information about OSI layers, algorithms, and solutions

### How It Works

The system processes information and queries through several steps:

1. **Document Processing:** Extract text and table data from Microsoft Word documents
2. **Semantic Chunking:** Split documents into meaningful chunks while preserving context
3. **Vector Embedding:** Convert text chunks into vector representations for semantic matching
4. **Query Refinement:** Use LLM to improve user queries before searching
5. **Context Retrieval:** Find the most relevant document chunks for a given query
6. **Answer Generation:** Generate comprehensive responses using both retrieved information and LLM knowledge

### Installation

#### Prerequisites

- Python 3.8 or higher

- Groq API key

## Setup

1. Clone this repository:
2. `git clone https://github.com/yourusername/ai-enhanced-it-helpdesk.git`
3. `cd ai-enhanced-it-helpdesk`
4. Create and activate a virtual environment (recommended):
5. `python -m venv venv`
6. `source venv/bin/activate # On Windows: venv\Scripts\activate`
7. Install the required dependencies:
8. `pip install -r requirements.txt`
9. Set up your Groq API key:
10. `export GROQ_API_KEY="your_api_key_here" # Linux/Mac`
11. # OR
12. `set GROQ_API_KEY=your_api_key_here # Windows`

## Usage

### Basic Usage

Run the main script and follow the prompts:

```
python it_helpdesk.py
```

The system will:

1. Load and process the specified Word document
2. Create vector embeddings of the content
3. Prompt you to enter an IT or networking query
4. Refine your query automatically
5. Retrieve relevant information
6. Generate a comprehensive answer

### Example

Enter your query about computer networking or IT problems: Why is my network slow?

Processing query...

Refined query: What factors might be causing network slowdown and how can I diagnose and fix them?

Searching for relevant information...

[Displays retrieved document information]

Generating final answer...

[Provides detailed answer with information about potential causes of network slowness, including hardware issues, congestion, configuration problems, along with diagnostic steps and solutions. The answer includes OSI layer information and relevant algorithms.]

Sources:

- LLM Model: Llama 3 (8B parameters)

- Websites:

1. [Networking

Expertise](<https://www.networkingexpertise.com/search?q=network+slowdown+diagnosis>)

2. [Pearson

Sample]([https://ptgmedia.pearsoncmg.com/images/9780789759818/samplepages/9780789759818\\_Sample.pdf#search=network+slowdown](https://ptgmedia.pearsoncmg.com/images/9780789759818/samplepages/9780789759818_Sample.pdf#search=network+slowdown))

## **Data Format**

The system expects Word documents with structured content. Ideally, documents should contain:

- Numbered entries (S.No)
- Topic headings in uppercase

- Detailed descriptions
- Tables with relevant information

## Configuration

The system can be configured by modifying the following parameters:

- chunk\_size: Size of text chunks for processing (default: 1000)
- chunk\_overlap: Overlap between consecutive chunks (default: 100)
- embedding\_model: Model used for generating embeddings (default: 'all-MiniLM-L6-v2')
- llm\_model: LLM used for query refinement and answer generation (default: 'llama3-8b-8192')

## Requirements

The project requires the following Python packages:

- pandas
- numpy
- sentence-transformers
- langchain
- faiss-cpu
- groq
- python-docx
- docx2txt

A complete list with versions is available in requirements.txt.

## Security & Safety

- The system validates file extensions before processing
- API keys are validated for basic format requirements
- Error handling ensures graceful failure in case of issues

## Contributing

Contributions are welcome! Please feel free to submit a Pull Request.

1. Fork the repository
2. Create your feature branch (git checkout -b feature/amazing-feature)
3. Commit your changes (git commit -m 'Add some amazing feature')
4. Push to the branch (git push origin feature/amazing-feature)
5. Open a Pull Request

## Acknowledgments

- [Sentence-Transformers](#) for the semantic embeddings
  - [LangChain](#) for document processing
  - [FAISS](#) for efficient similarity search
  - [Groq](#) for the LLM API
- 

Made by **[GANGI JATHIN]**