Optimizing Techniques for Business RAG QA Bot

This report details innovative techniques for optimizing the Retrieval Augmented Generation (RAG) model developed for a business question-answering bot. The following optimizations aim to enhance the performance, efficiency, and output quality of the RAG system.

1. Utilizing Firecrawl for Efficient Data Collection

Overview

Firecrawl is an API service that crawls websites and converts content into clean, structured data suitable for LLM processing.

Implementation

- Integrated Firecrawl's Python API to crawl the business website
- Processed 67 distinct documents from various directories and subdirectories

Benefits

- Eliminates manual data scraping, reducing human error and bias
- Provides comprehensive, up-to-date information from the entire website
- Ensures data consistency and relevance to the business domain

Impact on RAG Performance

- Improved data quality and coverage, leading to more accurate and comprehensive responses
- Reduced time and effort in data preparation, allowing for faster model updates

2. Optimizing Chunk Size for Effective Embeddings

Overview

Chunk size plays a crucial role in creating meaningful embeddings for vector database storage and retrieval.

Experimental Process

- Tested chunk sizes: 128, 256, 512, and 1024 tokens
- Evaluated impact on embedding quality and retrieval relevance

Optimal Configuration

- Selected chunk size: 512 tokens

Rationale

- 128 and 256 tokens: Too granular, resulting in disconnected information
- 1024 tokens: Excessive data per chunk, potentially exceeding LLM context limits
- 512 tokens: Balanced approach, preserving context while remaining processable

Impact on RAG Performance

- Enhanced retrieval accuracy by maintaining semantic coherence within chunks
- Improved LLM processing efficiency by optimizing input size

3. Leveraging Long Context LLMs

Model Selection

Chosen model: Gemini-1.5-Flash by Google

Key Features

- 2 million token context length
- Fast inference times
- State-of-the-art performance

Advantages

1. Handles large data chunks efficiently
2. Provides rapid response times
3. Cost-effective (free tier available)
4. Includes a high-quality embedding model (text-embedding-004)

Impact on RAG Performance

- Expanded knowledge integration capabilities
- Reduced latency in generating responses
- Improved overall system coherence and context understanding

4. Utilizing High-Speed Inference Providers

Selected Provider

Groq LPU™ AI inference technology

Key Benefits

- Exceptionally fast token generation (700-1200 tokens/second)
- Access to various open-source LLMs (e.g., LLaMA, Mixtral, Gemma)
- Free API access for testing and development

Impact on RAG Performance

- Near-instantaneous response generation
- Flexibility to experiment with different LLMs for optimal performance
- Scalability for high-volume business applications

Conclusion

The implementation of these four optimization techniques significantly enhances the capabilities of our Business RAG QA Bot. By leveraging efficient data collection, optimized data chunking, advanced LLM models, and high-speed inference, we have created a system that delivers accurate, comprehensive, and rapid responses to business queries. These improvements position our QA bot as a valuable tool for enhancing customer service, internal knowledge management, and decision-making processes within the organization.