

Experimentation and Solution Analysis for Policy QA Service

Introduction

The Policy QA Service stands as a cornerstone in providing employees with quick and precise answers to inquiries concerning organizational policies. This document explains the various approaches explored to construct this service, elucidating the rationale behind each approach. Furthermore, it dives into the advantages and drawbacks of each solution, accompanied by an evaluation table for comparative analysis.

Approaches Experimented

Two primary approaches were experimented with:

ElasticSearch + Langchain:

- Leveraged ElasticSearch for efficient storage and retrieval of policy documents.
- Employed Langchain for document preprocessing, encompassing text segmentation and question-answering capabilities.

Knowledge Graph + Neo4j:

- Entailed the construction of a knowledge graph using Neo4j to represent policy-related concepts.
- Integration with the ChatGPT API facilitated question-answering tasks.

Thought Process and Rationale

ElasticSearch + Langchain:

- Efficiency: ElasticSearch emerged as the backbone, offering robust and easy document retrieval mechanisms, opting for managing extensive policy document repositories efficiently.
- Integration: Langchain's seamless integration with ElasticSearch streamlined document preprocessing tasks, such as document loader and text chunking, augmenting the service's overall efficacy.
- Scalability: Dockerization facilitated the deployment and scalability of the application, ensuring consistent performance across diverse environments and accommodating increased workload demands effortlessly.

Knowledge Graph + Neo4j:

- Semantic Understanding: Leveraging Neo4j for constructing a knowledge graph enabled a structured representation of policy-related concepts, fostering semantic understanding and facilitating nuanced query interpretations.
- Flexible Querying: Neo4j's graph-based querying capabilities empowered flexible exploration of policy information, facilitating complex query traversals and yielding comprehensive responses tailored to user inquiries.
- Contextual Answers: Integration with the ChatGPT API gives the service the capability to generate contextually relevant responses, leveraging the pre-trained language model's contextual understanding to provide comprehensive answers.

Pros and Cons of Each Solution

Approach	Pros	Cons
ElasticSearch + Langchain	<ul style="list-style-type: none">- Ease document retrieval- Seamless integration with ElasticSearch and ChatGPT API- Scalable deployment using Docker	<ul style="list-style-type: none">- Additional configuration and maintenance overhead- Learning curve for Langchain functionalities- Docker setup complexity
Knowledge Graph + Neo4j	<ul style="list-style-type: none">- Structured representation of policy-related concepts- Flexible querying capabilities with Neo4j- Contextually relevant responses with ChatGPT API	<ul style="list-style-type: none">- Substantial effort required for knowledge graph construction- Learning curve for Neo4j setup and querying- Integration complexity with ChatGPT API

Evaluation Table

Criteria	ElasticSearch + Langchain	Knowledge Graph + Neo4j
Efficiency	High	Moderate
Integration	Seamless	Moderate
Scalability	High	Moderate
Semantic Understanding	Limited	High
Flexible Querying	Limited	High
Contextual Answers	Limited	High

Conclusion

In conclusion, both approaches offer unique advantages and challenges. While the ElasticSearch + Langchain approach excels in easy document retrieval and seamless integration, it demands additional maintenance and configuration overhead. Conversely, the Knowledge Graph + Neo4j approach, with its structured representation and flexible querying, requires substantial upfront effort but offers comprehensive semantic understanding. The choice between the two hinges on project-specific requirements and constraints, with each approach catering to different aspects of the Policy QA Service's functionality and scalability.