SHL Assessment Recommendation Engine:
Development Journey

SHL Intern Assignment

Project Overview

This document chronicles my journey developing a Retrieval-Augmented Generation (RAG) system for recommending SHL talent assessments. Starting with a comprehensive literature review of recommendation architectures and vector search mechanisms, I designed a solution integrating data acquisition, semantic search, and LLM-powered generation. My research-driven approach enabled me to identify the most effective techniques for each system component, resulting in a highly responsive recommendation engine capable of understanding complex HR requirements.

1 Data Acquisition & Processing

The project began with a thorough analysis of SHL's assessment catalog structure, revealing no accessible public API. Drawing on my background in web information retrieval, I engineered a sophisticated web scraper with adaptive mechanisms to handle several complex challenges:

- Two distinct catalogs with different pagination mechanisms requiring separate traversal strategies
- Non-standard pagination that demanded custom HTTP header configurations and intelligent retry logic
- Widely varying HTML structures necessitating a cascade of context-aware selector strategies
- Heterogeneous data representation across different assessment types

The extraction of consistent assessment descriptions presented particularly intricate challenges. Through iterative refinement and pattern analysis of the DOM structures, I developed a recursive traversal algorithm that employed contextual heuristics to distinguish primary content from peripheral elements. This approach successfully identified and extracted meaningful descriptions even across substantially different page structures, demonstrating my ability to recognize patterns in unstructured data.

2 RAG Pipeline Implementation

The recommendation engine architecture emerged from methodical research into current RAG approaches:

2.1 Vector Database Evaluation

After conducting a systematic comparison of vector database solutions against requirements for search performance, filtering capabilities, and integration complexity, I selected PostgreSQL with pgvector as the optimal foundation. This led to the development of:

- Custom SQL function incorporating both semantic matching and structured filtering in a single query
- Carefully tuned HNSW indexing parameters based on empirical testing with the assessment corpus
- Dynamic SQL construction techniques that push filter constraints to the database layer

2.2 Embedding Strategy

My research into semantic representation approaches informed several critical design decisions:

- Systematic benchmarking of embedding models with different dimensionality and context window characteristics
- Development of composite embedding templates that capture both categorical and descriptive assessment attributes
- Implementation of database triggers that ensure embedding consistency during assessment updates

2.3 Recommendation Pipeline Architecture

The core pipeline incorporates several innovative elements derived from both academic research and practical experimentation:

- Asynchronous query processing with dynamically adjustable retrieval parameters based on query complexity
- Multi-stage filtering that preserves semantic relevance while enforcing organizational constraints
- Carefully engineered LLM prompting strategies that produce contextually grounded explanations
- Adaptive score normalization that balances vector similarity with semantic relevance

Through extensive experimentation with different pipeline configurations, I discovered that batched processing combined with strategically calibrated retrieval multipliers significantly enhanced response efficiency while maintaining recommendation quality. This process of methodical experimentation demonstrates my commitment to evidence-based optimization.

3 Evaluation Framework Design

To ensure recommendation quality and enable data-driven refinement, I developed a comprehensive evaluation system reflecting best practices in information retrieval:

- Creation of a diverse ground truth dataset covering various job roles and assessment types
- Implementation of industry-standard metrics that measure different aspects of recommendation quality
- Design of a persistent evaluation service that tracks performance across system iterations
- Development of RESTful evaluation endpoints enabling continuous quality monitoring

This framework allowed me to systematically investigate the impact of different parameter configurations, leading to the identification of optimal threshold values and retrieval strategies. The ability to quantify system performance across iterations proved invaluable for guiding further enhancements, highlighting my methodical approach to system optimization.

4 Technical Challenges & Problem-Solving Approaches

- **HTML Structure Variability:** When confronted with wildly inconsistent HTML patterns, I developed a machine learning-inspired approach that combined multiple selector strategies with statistical validation of extracted content. This adaptive parsing system could effectively identify relevant content even when page structures deviated significantly from expected patterns.
- **Semantic Representation Challenges:** Upon discovering degradation in query understanding for specialized HR terminology, I researched embedding techniques specifically suited for domain-specific language. This led to the development of composite embedding templates that combine structured metadata with narrative descriptions, preserving contextual relationships between assessment attributes.
- **Query Performance Bottlenecks:** Through systematic profiling of query execution patterns, I identified that post-retrieval filtering was creating significant latency for complex constraints. By reformulating the filtering approach to leverage database-side query optimization, I was able to dramatically reduce processing time for multi-constraint scenarios.
- **LLM Hallucination:** When detecting instances of the LLM generating non-existent assessments, I studied recent research on prompt engineering for grounded generation. This in-

formed the development of robust instruction formats with explicit grounding requirements that substantially reduced fabrication issues.

- **System Responsiveness:** After investigating request patterns under load, I implemented a strategic caching architecture for embeddings and intermediate query results, significantly improving system throughput during peak usage scenarios.

5 Frontend Integration

Drawing inspiration from user experience research in conversational interfaces, I designed a Streamlit-based interface that balances sophistication with accessibility:

- Natural language query interface supplemented with intuitive filtering options for precise requirement specification
- Information-rich result cards that contextualize recommendations with relevant explanations
- Responsive design principles that adapt the interface based on device capabilities
- Comprehensive error handling with graceful degradation pathways for component failures

The system is deployed and accessible at:

- Main Frontend: https://shl-assessment-frontend.onrender.com
- Evaluation Frontend: https://shl-evaluation-frontend.onrender.com
- Backend API endpoints Swagger Documentaion: https://shl-assessment-backend-aiou.onrender.com/docs

6 Future Research Directions

- **Domain-Specific Representation Learning:** Investigate techniques for fine-tuning embedding models on HR-specific corpora to capture specialized semantic relationships within talent assessment terminology.
- **Natural Language Understanding for Constraints:** Explore advanced entity recognition approaches that could automatically extract structured constraints (job levels, duration requirements, test types) from natural language queries.
- **Continuous Learning Architecture:** Design an MLOps framework incorporating automated A/B testing and parameter optimization based on evolving usage patterns.
- **Feedback-Driven Refinement:** Develop mechanisms for capturing both explicit and implicit user feedback to continuously enhance recommendation relevance through reinforcement learning techniques.

Conclusion

This project demonstrates how systematic research and methodical problem-solving can produce an intelligent recommendation system that truly understands complex requirements. By drawing on diverse disciplines—from information retrieval and natural language processing to database optimization and user experience design—I created a solution that bridges structured filtering with semantic understanding. The resulting system delivers contextually relevant recommendations that surpass conventional keyword-based approaches in both relevance and usability, significantly streamlining the assessment selection process for HR professionals.