SHL Generative AI Assignment

Assessment Recommendation System – Approach Document

Candidate: ***Tejpal Singh Rathore***

Role: AI Intern – Generative AI

Project: SHL Assessment Recommendation Engine

---

1. Problem Understanding

Hiring managers often struggle to identify the most appropriate assessments from a large assessment catalog based on specific hiring requirements. These requirements are typically expressed in natural language and may involve multiple dimensions such as technical skills, behavioral traits, duration constraints, and assessment formats.

The objective of this project is to design and implement an intelligent recommendation system that, given a hiring requirement as free-text input, recommends the most relevant SHL individual assessment solutions. The system must demonstrate semantic understanding rather than relying on simple keyword matching, must operate on real SHL catalog data obtained through web crawling, and must be evaluated quantitatively using Recall@10 on a provided labeled dataset.

---

2. Data Collection and Ingestion

2.1 SHL Catalog Crawling

The first and most critical step was to build a local assessment knowledge base by crawling the SHL product catalog website. A custom web scraper was implemented using requests and BeautifulSoup to extract assessment details directly from SHL catalog pages.

Special care was taken to:

- Crawl only **Individual Test Solutions**
- Exclude **pre-packaged job solutions**
- Handle pagination and multiple catalog sections
- Ensure reproducibility of the scraping process

The final dataset contained **423 individual assessments**, exceeding the minimum requirement of 377.

2.2 Extracted Fields

For each assessment, the following attributes were extracted and stored:

- Assessment name
- Assessment URL
- Description text
- Test type (technical, behavioral, or both)
- Duration (if available)
- Remote testing support
- Adaptive support

Raw data and cleaned data were stored locally in structured JSON format to ensure transparency and repeatability.

---

3. Data Cleaning and Preparation

The scraped data contained HTML artifacts, navigation text, and redundant content. A cleaning pipeline was applied to:

- Remove boilerplate and navigation text
- Normalize textual content
- Standardize categorical fields
- Prepare consolidated text representations suitable for semantic embedding

The cleaned dataset served as the single source of truth for all downstream components.

---

4. Baseline and Semantic Retrieval Approach

4.1 Baseline Motivation

To establish a reference point, an initial baseline retrieval strategy was considered using simple text similarity. This helped validate the dataset and provided a comparison point for later improvements.

4.2 Semantic Embeddings

To capture semantic meaning beyond keywords, assessment descriptions were converted into dense vector representations using a pre-trained Sentence-Transformer model (all-MiniLM-L6-v2). This model was chosen for its strong performance, efficiency, and suitability for semantic search tasks.

Each assessment was embedded once and stored in a vector index.

---

5. Vector Search with FAISS

FAISS (Facebook AI Similarity Search) was used to index the assessment embeddings and perform efficient similarity search at scale.

The workflow is as follows:

1. User query is embedded using the same embedding model
2. FAISS retrieves the top-N most similar assessment vectors
3. Corresponding assessment metadata is retrieved

This approach ensures fast and scalable semantic retrieval.

---

6. Reranking and Recommendation Balance Logic

Pure semantic similarity can sometimes over-favor one assessment type. To address this, a lightweight reranking and balance strategy was introduced:

- The query intent is analyzed for technical and behavioral signals
- When both are present, the final recommendations ensure a balanced mix of technical (K) and behavioral (P) assessments
- Results are trimmed to the top-10 recommendations

This step improves recommendation quality and aligns with real-world hiring expectations.

---

7. Evaluation Methodology

7.1 Training Dataset

A labeled training dataset containing human-annotated relevant assessments for multiple queries was provided. This dataset was used exclusively for evaluation and iteration.

7.2 Metric

The system was evaluated using **Mean Recall@10**, as specified in the assignment.

For each query:

- The top-10 recommended assessments were generated
- Recall@10 was computed based on overlap with labeled relevant assessments

The mean Recall@10 across valid queries was reported and used to guide iterative improvements in retrieval and reranking logic.

---

8. Test Set Prediction

After finalizing the system, predictions were generated for the unlabeled test dataset consisting of 9 queries.
For each query, the system produced the top-10 recommended assessment URLs.

The results were saved in the required CSV format:

Query,Assessment_url

This file is ready for submission and directly reflects the system's final behavior.

---

9. API and Frontend Implementation

A lightweight FastAPI backend was developed with the following endpoints:

- /health – system health check
- /recommend – returns assessment recommendations for a given query

A minimal frontend using HTML, CSS, and JavaScript was implemented to demonstrate end-to-end usability. The frontend communicates with the live API and displays recommendations in a user-friendly format.

---

10. Limitations and Future Improvements

While the system meets all assignment requirements, possible future improvements include:

- More advanced reranking using learning-to-rank techniques
- Explicit modeling of time and difficulty constraints
- Larger or domain-specific embedding models
- Personalized recommendations based on historical usage

---

11. Conclusion

This project demonstrates an end-to-end, production-ready assessment recommendation system built using real SHL catalog data, semantic retrieval techniques, quantitative evaluation, and clean API design. The solution emphasizes correctness, reproducibility, and practical relevance, aligning closely with real-world hiring scenarios.