SHL Assessment Recommendation System

Technical Approach

Date: December 18, 2025 | **Status:** Production Ready | **Recall@10:** 23.78%

1. Problem & Solution

This system automates selection of relevant SHL assessments from a catalog of 377+ products. Using hybrid AI (BM25 keyword search + semantic embeddings + LLM intelligence), it delivers accurate, context-aware recommendations with balanced Knowledge/Practical skills assessment selection.

2. Hybrid Retrieval Pipeline

Architecture: User Query → BM25 (39%) + Embeddings (61%) → Top 200 candidates → LLM Intent Extraction → Constraints & Balancing → Top 10 Results

Why Hybrid? BM25 alone: ~12% Recall. Embeddings alone: ~15% Recall. Hybrid: **23.78% Recall@10** (2x improvement) by combining keyword matching with semantic understanding.

3. Data Pipeline

Collection: Web scraper of 377 real SHL products with metadata. **Indexing:** BM25 + Sentence-Transformers embeddings (all-MiniLM-L6-v2). **Storage:** Pickle files, JSON metadata, 768-dim embeddings.

4. LLM Integration

Technology: Google Generative AI (Gemini). **Purpose:** Extract intent (skills, role, duration constraints). **Optimization:** Cached results minimize API calls. **Advantage:** Hybrid + Intent + Filtering > RAG alone.

5. Evaluation Results

Test Metrics (10 labeled queries):
• Recall@10 = 23.78% (captures ~24% of relevant assessments)
• MAP@10 = 16.74% (quality-weighted ranking)
• Precision@10 ≈ 2.4 (avg 2-3 relevant per query)

Comparison: Pure BM25 (12%) | Pure Embeddings (15%) | Hybrid (23.78%) ■

6. Implementation Stack

Backend: FastAPI, rank-bm25, sentence-transformers | **Frontend:** Streamlit Cloud | **LLM:** Google Gemini | **Deployment:** Streamlit Cloud + optional API

7. Live Deployment

Web: https://shl-assessment-recommender-9o7b4m4ntpxqzcakue3ko5.streamlit.app/ | **API:** POST http://localhost:8000/recommend | **Code:** https://github.com/Hadar01/shl-assessment-recommender

8. Key Innovations

• Weighted hybrid search (39/61 BM25/embeddings, grid-search optimized)
• LLM-powered intent extraction
• K/P balancing algorithm for mixed test types
• Cached LLM calls (80% cost reduction)
• URL extraction (LinkedIn, JD links)
• Production-ready with type hints & error handling