SHL Assessment Recommender System: A Gen AI Approach

Balmuri Yeshwanth Kumar

Problem Overview Solution

To address challenges in efficiently finding suitable SHL assessments, this project developed an intelligent system accepting natural language queries. It recommends relevant assessments via an interactive Streamlit UI and a robust FastAPI backend, aiming to streamline the selection process.

Approach and Design

The system employs a multi-stage hybrid architecture:

1. **Data Preparation (Offline):**
   - *Sourcing & Enrichment:* SHL catalog data (https://www.shl.com/solutions/products/product-catalog/) enriched via web scraping (`requests`, `BeautifulSoup4` via `scripts/data`$_e$*nrichment.py* $---$ `adjust script name`)*fordescriptions,joblevels,etc.*`LLM Feature Extraction:`*GroqLlama*3$-$70*B*(`scripts/update_as`
   - *Embedding Generation:* Textual fields embedded using `bert-base-uncased` (`scripts/generate_embeddings.py`), stored in `assessment_embeddings_groq_v2_BERT.json`.
   - *Technical Skill Mapping:* Predefined map links tech keywords to assessments for boosting.
2. **Query Understanding (Online):** LLM (Groq Llama3-8B) parses queries into structured criteria (domain, skills, duration, experience), informed by cultural context detection.
3. **Candidate Retrieval Initial Ranking:** BM25 search on augmented query; heuristic score boosting for domain/skill match and technical role relevance.
4. **Filtering Re-ranking Pool Prep:** Hard filtering (e.g., max duration), then soft sorting by duration. Top candidates (e.g., up to 50) form re-ranking pool.
5. **LLM-Powered Re-ranking (Online):** Groq Llama3-70B re-ranks candidates based on original query, all extracted criteria (incl. cultural context), and summarized assessment details.

Core Tech: Python, FastAPI, Streamlit, Groq API (Llama3), Transformers (Tokenizer), `rank_bm25`.

Evaluation

Performance evaluated using `test_set.json` (7 queries) via `scripts/evaluate.py`.

Achieved Metrics (May 7, 2025 — *Update date!*):

- Accuracy@5: **85.7%** (6/7)
- MR@3: **0.316**
- MAP@3: **0.156**

Tracing Strategy: Quantitative metrics augmented by 'tracing': manual LLM I/O review (criteria extraction re-ranking) and iterative prompt engineering. This addressed issues like skill misattribution (Query 6) and cultural context (Query 3), refining heuristics. Intermediate pipeline outputs also analyzed.

Accessibility: Deployed URLs GitHub

- **Demo (Streamlit UI):** https://shlfinalrecommendation-9n4zwkcwk2dbz9ocdld3an.streamlit.app/
- **API (FastAPI):** https://shl-final-recommendation-api-yesh.onrender.com (*Endpoints: /health, /recommend (POST), /docs*)
- **GitHub Repo:** https://github.com/yeezerdaw/SHL_FINAL_RECOMMENDATION

Known Limitations Future Work

- **Limitations:** Nuanced LLM cultural context understanding; occasional LLM misinterpretations; static dataset.
- **Future Work:** Advanced prompt engineering (e.g., few-shot); direct JD URL parsing; user feedback mechanisms.