Project

Presentation

Approach Document for SHL Assessment Recommendation System

Objective

The primary goal of this project is to develop an intelligent assessment recommender system that, given a Job Description (JD) or recruiter query, can suggest the most suitable SHL assessment(s). These assessments must match the skills, roles, and time constraints mentioned in the query.

Problem understanding

Recruiters often struggle to identify the right assessments that best evaluate candidates for a given role. The challenge is to:

- Accurately extract and understand the skills and requirements from the JD or query.
- Map these extracted keywords to a list of available assessments.
- Apply additional filters like maximum duration, assessment type (cognitive, technical, personality, etc.).
- Present the most relevant assessments in a user-friendly way.

Solution Architecture

A Data Preparation

- Loaded the shl_sample_assessments.csv which contains detailed metadata of SHL assessments including:
  - Assessment name
  - Description
  - Skills tested
  - Role-based tags
  - Time duration
- Preprocessed the dataset by:
  - Cleaning null values
  - Lowercasing and tokenizing keywords
  - Creating a searchable combined text column from all relevant fields

B Text Processing

- Preprocess the JD/query:
  - Lowercasing, punctuation removal
  - Stopword removal
  - Tokenization
- Applied TF-IDF vectorization using TfidfVectorizer from scikit-learn.
- Calculated cosine similarity between the user query and each assessment entry.
- Sorted the results based on similarity score.

c. Filtering

Included a time duration filter (user selects maximum duration).

Applied relevance threshold to recommend only top matches.

Filtered by specific needs like “cognitive” or “personality” if present in query.

d. Recommendation Output

Displayed top N matching assessments (e.g., top 5).

Showed detailed information: Assessment name, description, duration, category.

Optional API: Result also formatted as JSON for integration via REST.

User Interface

Built using Streamlit:

- Clean and intuitive web interface.
- Recruiter types the JD or query into a textbox.
- Dropdown for selecting maximum assessment duration.
- Recommended assessments are shown with details, neatly ranked.

Tools & Libraries Used

Tool/Library & Purpose

Python : : Core development language

Pandas : : Data loading and preprocessing

Scikit-learn : : TF-IDF and cosine similarity

NLTK / re : : Text preprocessing

Streamlit : : Web app UI

VS Code : : Development environment

Git & GitHub : : Version control and code sharing

Sample Supported Queries

Our system handles queries like:

- “Hiring for a backend developer with Java and Spring skills. Need assessments under 40 minutes.”
- “Looking for cognitive and behavioral tests for entry-level candidates within 30 minutes.”
- “Seeking an all-in-one test for Python, SQL, and JavaScript developers (max 1 hour).”

These demonstrate flexibility across tech stacks, personality traits, and durations.

Optional Evaluation Metrics

We explored but did not implement:

- Mean Recall@K – Measures how many relevant assessments were found in top K.
- MAP@K – Evaluates ranking precision for each query.

They can be integrated in future using labeled ground truth queries.

Outcome

- A working Streamlit-based recommender app.
- Successfully recommends SHL assessments based on any input JD/query.
- Supports duration filtering and skill relevance.
- Can be easily extended with an API and deployed for live use.

Conclusion

The SHL Assessment Recommendation System was developed to simplify the process of mapping job descriptions to the most relevant SHL assessments. By extracting key skills from a JD and matching them with the skills measured in SHL assessments, we enable HR teams to make informed decisions quickly and efficiently.

The project effectively demonstrates the power of text processing, skill-based matching, and data-driven recommendations. It is built with simplicity, scalability, and user experience in mind — offering both a web interface and API support.

This solution can be easily extended with advanced NLP techniques and relevance feedback mechanisms for improved accuracy and performance in real-world recruitment scenarios.

Thank You