

Thesis: Detailed Approach

Objective: To develop a system that identifies and ranks experts based on their suitability for a job description.

Approach: I've used AI techniques for data retrieval, processing, and explainable rankings

Step 1: Data Retrieval

Input Sources: Candidate descriptions and job descriptions are the primary data sources. Each candidate's description provides information on their background, skills, industry experience, and affiliations. The job description outlines the requirements for the role.

Data Representation: Candidate descriptions and job descriptions are summarized into keyword-based representations. This is achieved through prompt engineering and a Large Language Model (LLM) to extract keywords in five categories:

- **Background:** Educational and professional qualifications.
- **Industry:** Relevant industries the candidate has experience in.
- **Skills:** Technical and domain-specific skills.
- **Experience:** Roles, duration, and type of work performed.
- **Company Affiliations:** Past and current employers, partnerships, or collaborations.

Eg: Sample candidate descriptions summary with keywords

```
**Candidate Name:** candidate2

**Background:**

* Bachelor of Science
* Computer Science
* Second Year

**Industry:**

* Software Engineering
* Technology
* Computer Hardware/Software

**Skills (Potentially Inferred, More Information Needed):**

* Programming (Specific languages unknown, potentially Swift, Objective-C, C++, Python, Java)
* Software Development
* Debugging
* Version Control (e.g., Git)
* Problem-solving
* Analytical Skills
* Teamwork (likely, given corporate environment)

**Experience:**

* Software Engineer
* Internship (Potentially, given student status)

**Company Affiliations:**

* Apple
```

Step 2: Data Processing

Embedding Generation:

- Each summarized candidate description is converted into embeddings (numerical vector representations) using the *Google Generative AI Embeddings* model.
- This enables computational comparison of candidates with the job description. A vector database (*ChromaDB*) is used to store and retrieve embeddings efficiently.

Similarity Search:

- The job description embedding is compared with candidate embeddings using cosine similarity.
- This search retrieves the most relevant candidates (top 100) for the job.

Step 3: Scoring Mechanism

Score Generation & Evaluation Categories:

- The score for each retrieved candidate is generated using prompt engineering and a Large Language Model (LLM).
- The inputs for the prompt are the retrieved candidate summary in keywords, the job description summary in keywords, and the output format (a JSON object).
- Using the prompt, candidates' scores are generated across the following categories: ***Background, Industry, Skills, Experience, Company Affiliations***
- Each score is accompanied by a textual description explaining why the candidate received a particular score. (*Explainable AI*)

Scoring Method:

Each category is scored out of 100 based on the similarity of candidate and job description keywords. A weighted scoring system is applied to emphasize critical factors: (can change depending on the use case/requirements)

- **Experience:** 30%
- **Skills:** 20%
- **Background:** 20%
- **Industry:** 20%
- **Company Affiliations:** 10%

Step 4: Ranking Mechanism

Weighted Score Calculation: A weighted total score is computed for each candidate

- $\text{Weighted Score} = \sum (\text{Category Score} \times \text{Category Weight})$

Candidate Ranking:

- Candidates are ranked based on their weighted scores.
- The output includes detailed explanations for the rankings to ensure transparency and next steps.

Sample output of the tool:

```
Rank 1:
Name: Candidate3
Background:
  Score: 80
  Description: Candidate3's background in Computer Science is highly relevant. While still an undergraduate, their focus on AI/ML aligns perfectly with the
Industry:
  Score: 100
  Description: The candidate's interest and experience in Artificial Intelligence, Machine Learning, and Software Engineering are precisely what the job des
Skills:
  Score: 70
  Description: While the candidate likely possesses strong Python skills and experience with deep learning frameworks, the job description specifically ment
Experience:
  Score: 90
  Description: Despite being an undergraduate, the candidate's experience as an AI Engineer at both OpenAI and Apple is exceptional and surpasses the job de
Company Affiliations:
  Score: 100
  Description: Working at leading AI companies like OpenAI and Apple is a huge plus. This surpasses the desired startup experience and significantly elevat
Next Steps: Candidate3 is a very strong candidate. Schedule an interview to assess their proficiency in the specific technologies mentioned in the job descr
=====
Rank 2:
Name: Candidate4
Background:
  Score: 90
  Description: Candidate4 has a strong academic background with a Master's degree in Computer Science and Engineering from NYU Tandon School of Engineering
Industry:
  Score: 90
  Description: The candidate's experience in Technology, Software Development, Artificial Intelligence, Data Science, and Machine Learning directly aligns w
Skills:
  Score: 75
  Description: Candidate4 possesses several key skills required for the job, including Python and experience with AI/ML tools like PyTorch and TensorFlow.
=====
✓ 0s completed at 2:19 PM
```