AI & Machine Learning Internship – Applied AI for Talent Intelligence.

Internship Project Report

Submitted by: **Jagpreet Singh Mittu**

Submitted to: **Sedha Consulting Pte. LTd. Singapore**

Internship Organization: Sedha Consulting

Duration: Feb 9, 2026 to May 1, 2026

Acknowledgement

I would like to express my sincere gratitude to Sedha Consulting for providing me with the opportunity to work on a real-world problem in the field of recruitment and artificial intelligence.

I am especially thankful to my mentors and team members at the organization who guided me throughout the internship. Their practical insights helped me understand how industry systems actually work beyond theoretical knowledge.

I would also like to thank the faculty of the University of Toledo for their continuous support and encouragement during the course of this project.

Finally, I would like to thank my family for their support and motivation throughout this journey. This project has been a valuable learning experience, and it has helped me grow both technically and professionally.

Abstract

This project focuses on solving a major problem in the recruitment industry known as the "resume black hole," where qualified candidates are often missed due to keyword-based filtering systems. The aim of this internship project was to build an intelligent system that can understand resumes and job descriptions in a deeper way using artificial intelligence.

The system uses natural language processing, machine learning, and large language models to analyze and rank candidates based on their relevance to a job description. Unlike traditional systems, this approach uses semantic similarity instead of simple keyword matching.

A working dashboard was developed using Streamlit, where recruiters can upload job descriptions and instantly view ranked candidates along with AI-generated assessments. The system also highlights key strengths, gaps, and provides a hiring recommendation.

This project demonstrates how modern AI techniques can improve hiring efficiency, reduce manual effort, and provide better decision support for recruiters.

Contents

List of Figures

Chapter 1

Industry Problem and Context

In modern recruitment systems, companies receive a large number of resumes for every job opening. Most of these resumes are processed using Applicant Tracking Systems (ATS), which rely heavily on keyword matching. This approach creates a major issue where candidates who do not use exact keywords are often rejected, even if they are suitable for the role.

This problem is commonly referred to as the "resume black hole," where good candidates are lost due to poor filtering methods.

Sedha Consulting, the organization where this internship was conducted, faced similar challenges. Recruiters had to manually scan resumes, which was time-consuming and inefficient. There was a clear need for a smarter system that could understand the meaning of resumes instead of just matching keywords.

The goal of this project was to design and implement an intelligent system that can:

- Understand resumes using semantic analysis
- Rank candidates based on relevance
- Provide AI-based hiring insights

This project aims to bridge the gap between traditional hiring systems and modern AI-driven decision-making.

Chapter 2

System Architecture

The architecture of the AI Talent Intelligence System follows a modular and scalable design. Each component of the system is responsible for a specific task, which makes the overall system easier to understand, maintain, and extend.

2.1 Architecture Overview

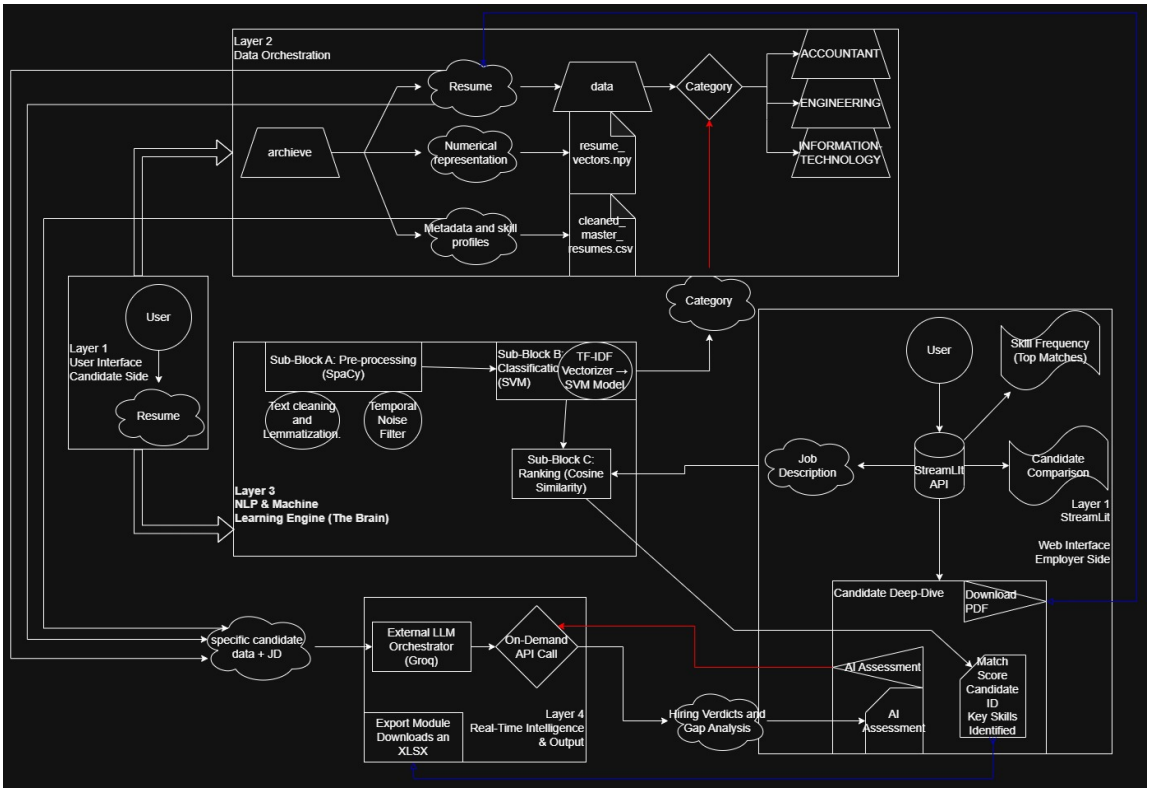

Figure 2.1: High-level architecture of the AI Talent Intelligence System

The above diagram shows the complete flow of data through the system, starting from resume input to final candidate evaluation. The system is divided into multiple layers,

each handling a specific part of the processing pipeline.

2.2 Architecture Interpretation

The architecture is divided into four main layers, each responsible for a specific part of the system.

Layer 1: User Interface (Candidate and Recruiter Side) This layer includes the interaction points where users upload resumes and recruiters provide job descriptions. The Streamlit-based dashboard also displays ranking results, candidate comparisons, and downloadable reports.

Layer 2: Data Orchestration This layer handles resume storage, preprocessing pipelines, and structured data generation. It includes resume parsing, metadata extraction, and conversion into numerical representations for further processing.

Layer 3: NLP and Machine Learning Engine This is the core of the system where text preprocessing, TF-IDF vectorization, and SVM classification are performed. Cosine similarity is used to rank candidates based on job description relevance.

Layer 4: AI Intelligence and Output This layer integrates the external large language model to generate candidate insights such as strengths, gaps, and hiring recommendations. It also supports exporting results and generating structured reports.

The layered design ensures modularity, allowing each component to be improved independently without affecting the overall system.

Chapter 3

User Interface and Dashboard

The system provides an interactive dashboard built using Streamlit. This interface allows recruiters to upload job descriptions and view ranked candidates in real time.

3.1 Main Dashboard

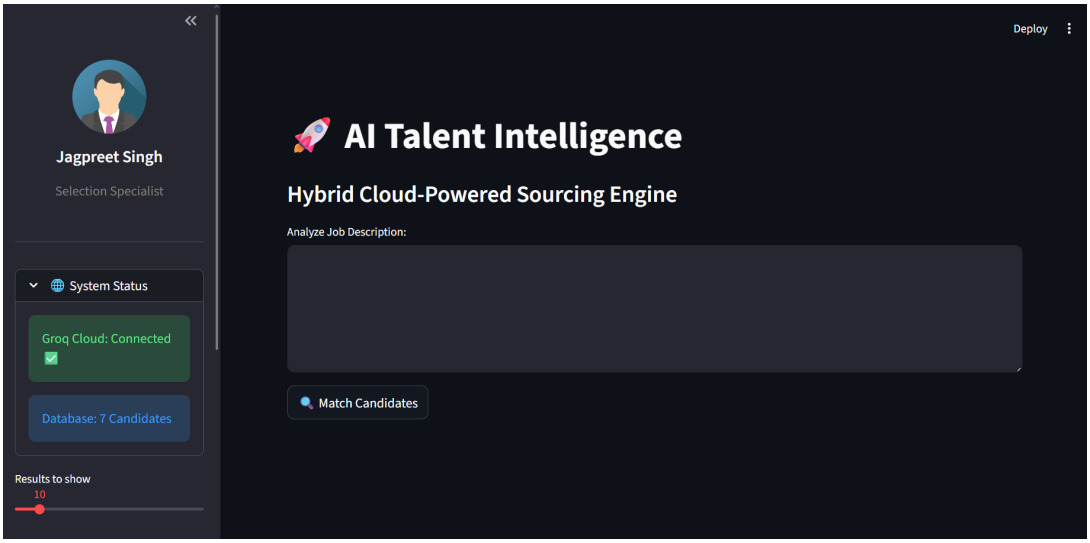

Figure 3.1: Main Dashboard Interface

The main dashboard includes:

- Job description input area
- Candidate ranking trigger button
- Adjustable filters for results

Figure 3.2: Candidate Ranking Results

3.2 Candidate Ranking Output

The system displays candidates ranked by match percentage. This helps recruiters quickly identify the most suitable candidates.

3.3 AI Assessment Panel

Figure 3.3: AI-Based Candidate Evaluation

The AI assessment provides:

- Top strengths of candidate

- Skill gaps
- Final hiring recommendation

3.4 Skill Frequency Chart

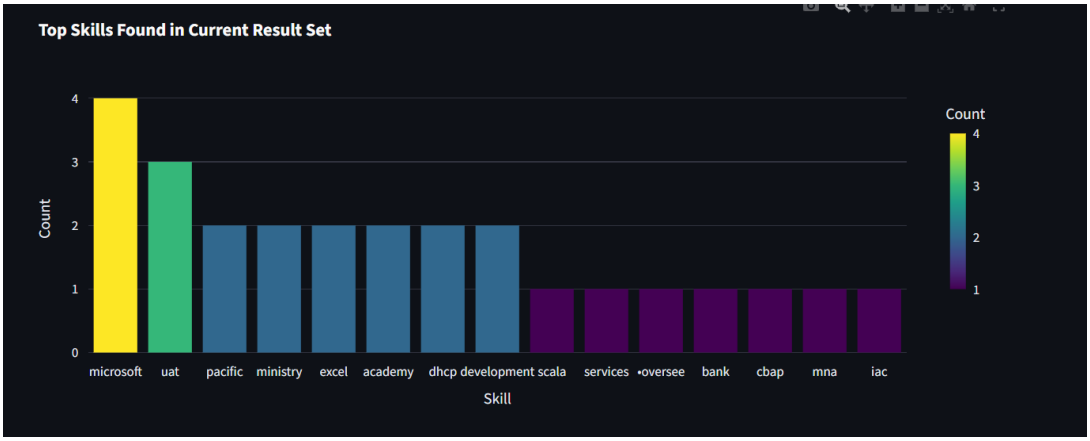

Figure 3.4: Skill Frequency Analysis

This visualization helps recruiters understand the distribution of skills in the candidate pool.

Chapter 4

Challenges and Solutions

Developing the AI Talent Intelligence System involved overcoming several technical hurdles, ranging from unstructured data processing to the optimization of real-time AI inferences.

4.1 Handling Unstructured Resume Data

Resumes are inherently diverse in their formatting, using various file types, layouts, and linguistic styles.

- **The Challenge:** Extracting clean, consistent text from PDFs and Word documents was difficult because traditional parsers often fail to preserve the logical flow of information.
- **The Solution:** I implemented a multi-stage preprocessing pipeline in Layer 2, utilizing `SpaCy` for text cleaning, lemmatization, and temporal noise filtering to ensure the machine learning models received standardized input.

4.2 Improving Matching Accuracy

The project's primary goal was to solve the "resume black hole" caused by keyword-based filtering.

- **The Challenge:** Initial keyword-matching methods yielded high false-negative rates, missing qualified candidates who used synonyms instead of exact terms.
- **The Solution:** I transitioned to semantic vector representations using TF-IDF and SVM classification. By calculating the Cosine Similarity between job descriptions and resume embeddings, the system significantly improved the relevance of its rankings.

4.3 API Optimization and Latency

Integrating the External LLM Orchestrator (Groq) introduced challenges related to real-time performance.

- **The Challenge:** Large resumes can lead to token limit issues and increased latency during the AI Assessment phase.
- **The Solution:** I optimized the system by structuring prompts more efficiently and focusing the LLM analysis on extracted metadata and skill profiles rather than the raw, unprocessed text. Additionally, caching techniques were used to ensure the Streamlit dashboard remained responsive during candidate deep-dives.

Chapter 5

Responsible AI and Ethics

The integration of Artificial Intelligence into recruitment processes necessitates a rigorous ethical framework to ensure fairness, accountability, and transparency. The AI Talent Intelligence System is designed not to replace human judgment, but to augment it while actively mitigating the systemic biases often found in traditional Applicant Tracking Systems (ATS).

5.1 Algorithmic Transparency and Explainability

One of the primary ethical risks in AI is the "black box" nature of decision-making. To combat this, the system prioritizes interpretability:

- **AI Assessment Panel:** On top of providing a solitary numerical score, the system utilizes an External LLM Orchestrator to generate natural language justifications. This identifies specific matching strengths and critical gaps, allowing recruiters to understand the logic behind a candidate's ranking.
- **Visual Skill Frequency:** The dashboard includes a Skill Frequency Analysis chart. This transparency allows recruiters to see the distribution of skills across the entire candidate pool, ensuring that the system's output is grounded in the provided dataset.

5.2 Bias Reduction via Semantic Analysis

Traditional keyword-based filtering often penalizes qualified candidates who use non-standard terminology—a phenomenon known as the "resume black hole".

- **Beyond Keywords:** By implementing semantic vector representations (TF-IDF and SVM classification) in Layer 3, the system focuses on the *meaning* of a candidate's experience rather than exact string matching.

- **Standardized Evaluation:** The use of Cosine Similarity ensures that every candidate is measured against the job description using the same mathematical objective function, reducing the influence of subjective human snap-judgments during the initial screening phase.

5.3 Human Oversight and Agency

Maintaining "Human-in-the-loop" is a core pillar of this project's ethical design:

- **Recruiter-Centric Control:** The system is built as a decision-support tool. Recruiters retain the final authority to adjust filters, set minimum confidence thresholds, and trigger the AI ranking manually.
- **Hiring Verdicts as Recommendations:** The "Final Hiring Verdict" is explicitly presented as a recommendation (e.g., "Potential Match") rather than a definitive "Hire" or "Reject" command, encouraging further human technical assessment.

These measures collectively ensure that the AI Talent Intelligence System operates as a fair, responsible, and effective assistant in the modern recruitment workflow.

Chapter 6

Project Repository and Reproducibility

The complete implementation of the AI Talent Intelligence System is maintained in a version-controlled repository using **GitHub**. This ensures transparency, reproducibility, and ease of access for further development.

6.1 Repository Link

The project repository can be accessed at the following link:

`https://github.com/JAGPREET234/AI-Talent-Intelligence`

6.2 Repository Overview

The repository is organized in a modular way to separate different components of the system. This structure makes the project easier to understand and extend.

6.2.1 Folder Structure

- **data/** Contains sample resumes and job descriptions used for testing.
- **/week1-3.py** Includes scripts for text cleaning, normalization, and data preparation.
- **models/** Contains machine learning models such as TF-IDF vectorizer and SVM classifier.
- **/week5-7.py** Implements cosine similarity and candidate ranking logic.
- **groq module** Handles interaction with the large language model for generating candidate insights.

- **/week9.py** Contains the Streamlit dashboard code for the user interface.
- **requirements.txt** Lists all dependencies required to run the project.

6.3 Steps to Run the Project

The system can be reproduced locally by following these steps:

1. Clone the repository using Git:

   ```
   git clone https://github.com/JAGPREET234/AI-Talent-Intelligence
   ```

2. Navigate to the project directory:

   ```
   cd ai-talent-intelligence-system
   ```

3. Install required dependencies:

   ```
   pip install -r requirements.txt
   ```

4. Run the Streamlit application:

   ```
   python -m streamlit run [base_path]/internship/codes/talent_intelligence_app.py
   ```

5. The above prompt opens the application in a browser window, upload a job description to begin candidate analysis.

6.4 Reproducibility Notes

The project has been designed to ensure consistent results when executed in different environments. However, the following points should be considered:

- The results may vary slightly depending on the dataset used.
- API-based responses from the language model may differ due to dynamic generation.
- It is recommended to use the same Python version and dependency versions listed in the repository.

6.5 Version Control and Updates

GitHub was used to track all changes made during the development process. This includes feature additions, bug fixes, and improvements in system performance.

Version control helped in:

- Maintaining code history
- Managing different development stages
- Ensuring stability of the final system

This structured approach makes the project suitable for further development and real-world deployment.

Chapter 7

Future Scope

While the current version of the AI Talent Intelligence System successfully demonstrates semantic candidate ranking, there are several avenues to enhance its scalability and transition it into a production-grade enterprise tool.

7.1 Cloud Infrastructure and Scalability

The current prototype runs as a local Streamlit application. To support a global recruitment workflow, the following deployment strategies are proposed:

- **Containerization with Docker:** Packaging the application, its specific Python dependencies, and the `requirements.txt` library into Docker containers will ensure the system runs consistently across different environments.
- **Cloud-Native Hosting:** Transitioning the system to cloud platforms such as AWS, Azure, or Google Cloud would allow for auto-scaling capabilities. This ensures the system can handle a high volume of concurrent resume uploads without performance degradation.
- **Vector Database Integration:** Replacing local `.npy` vector storage with a dedicated vector database (e.g., Pinecone) would allow for real-time indexing and searching of millions of candidate profiles with minimal latency.

7.2 Advanced AI and Model Optimization

The core intelligence of the system can be further refined to provide deeper insights:

- **Fine-tuned LLM Models:** Moving from general-purpose models to those fine-tuned specifically on recruitment and technical domain data could improve the accuracy of the "Hiring Verdicts".

- **Multilingual Support:** Implementing translation layers or multi-language embeddings would allow the system to process resumes and job descriptions in various languages, supporting global talent acquisition.

7.3 Enterprise Integration

To maximize industrial utility, the system must bridge the gap with existing professional tools:

- **ATS Platform Integration:** Developing APIs to allow the system to sit on top of existing Applicant Tracking Systems (ATS) would allow recruiters to use semantic ranking within their current software ecosystem.
- **Automated Data Pipelines:** Automating the Data Orchestration layer (Layer 2) to pull directly from job boards like LinkedIn or Indeed would streamline the candidate sourcing process.

By implementing these enhancements, the AI Talent Intelligence System can move from a functional prototype to a robust, scalable solution capable of eliminating the "resume black hole" at an industrial scale.

Chapter 8

Learning Outcomes

During this internship, I gained practical experience in working with real-world datasets and building end-to-end machine learning systems.

I developed a better understanding of natural language processing, vector-based similarity techniques, and how large language models can be integrated into applications.

This project also helped me improve my problem-solving skills, especially when dealing with unstructured data and system performance challenges.

Chapter 9

Conclusion

During this internship, I worked on building an AI-based system to improve the way candidates are screened for job roles. The main goal was to move beyond traditional keyword-based systems and create something that understands the actual meaning behind resumes and job descriptions.

Through this project, I learned how different components like machine learning models, NLP techniques, and large language models can be combined to solve a real-world problem. One of the most interesting parts was seeing how semantic similarity can produce better results compared to simple keyword matching.

The system that was developed is able to rank candidates, highlight important skills, and even provide a basic hiring recommendation. While there is still scope for improvement, the current version shows how such systems can be used in real recruitment workflows.

Overall, this internship helped me gain hands-on experience and gave me a better understanding of how AI is applied in industry-level applications.

Bibliography

[1] Targeting the "Resume Black Hole": Understanding Applicant Tracking Systems (ATS) and Keyword-based Filtering in Modern Recruitment.

[2] Honnibal, M., & Montani, I. (2017). *spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.*

[3] Pedregosa, F., et al. (2011). *Scikit-learn: Machine Learning in Python.* (Used for TF-IDF Vectorization and SVM Classification).

[4] Streamlit Inc. (2026). *Streamlit Documentation: Building interactive data applications in Python.*

[5] Groq Cloud. (2026). *LPU Inference Engine: Real-time LLM Orchestration and API Documentation.*

[6] Han, J., Kamber, M., & Pei, J. (2012). *Data Mining: Concepts and Techniques.* (Standardizing Candidate Ranking via Cosine Similarity).

[7] Meta AI. (2024). *The Llama 3 Herd of Models.* (Foundation for AI Assessment and Hiring Verdicts).