Semantic Resume Matching using LLMs, FAISS, and RAG

Manish Kanuri
Sanjana Bommegowda
Sushma Ramesh

1. Abstract

The manual screening of resumes represents a significant bottleneck in modern recruitment, often consuming excessive time and resources for recruiters. This inefficiency is largely attributable to the inherent limitations of traditional keyword-based Applicant Tracking Systems (ATS), which struggle to grasp the nuanced semantic relationships within candidate resumes and job descriptions (JDs). To overcome these challenges, we have engineered an advanced semantic matching pipeline. This pipeline leverages the power of Sentence Transformer embeddings to capture the contextual meaning of text, the efficiency of FAISS (Facebook AI Similarity Search) for rapid vector similarity search, and the interpretability of Retrieval-Augmented Generation (RAG) to surface the most relevant candidate resumes for a given JD. We meticulously constructed a filtered dataset of synthetic resumes tailored to early-career AI/ML roles and employed Large Language Model (LLM)-based explanations to provide transparent rationales for the generated resume-JD matches. Our empirical evaluation demonstrated that FAISS vector search, particularly when employing L2 distance, consistently yielded the most pertinent results with remarkable speed. Furthermore, our exploration of LLaMA and Mistral models for generating interpretable explanations showcased the potential to imbue the retrieval process with a crucial layer of transparency and understandability for users.

2. Introduction

The contemporary landscape of recruitment is often plagued by inefficiencies stemming from the widespread reliance on superficial keyword-matching algorithms embedded within most Applicant Tracking Systems (ATS). These legacy systems frequently fail to identify highly qualified candidates simply due to variations in vocabulary or differences in resume formatting. The central objective of this project was to conceive and develop a sophisticated semantic resume matching engine capable of significantly enhancing candidate retrieval accuracy by deeply understanding the underlying meaning and context of both resumes and job descriptions. Beyond improved accuracy, our system is designed to provide interpretable justifications for its recommendations, thereby fostering greater fairness and transparency throughout the hiring process.

The specific scope of our project focused on early-career opportunities within the dynamic fields of Artificial Intelligence and Machine Learning, with a particular emphasis on job descriptions related to Generative AI (GenAI). To further refine the candidate evaluation process, we integrated an experience-based pre-filtering step and incorporated language model-generated rationales to streamline the process and inject a degree of human-like understanding into the candidate assessment.

3. Background

Traditional Applicant Tracking Systems (ATS) predominantly rely on syntactic string matching, a method that often results in inaccurate and inefficient candidate filtering. This fundamental limitation arises from their inability to discern the semantic relationships between words and phrases. This critical gap can be effectively addressed through the adoption of semantic search methodologies, which involve encoding textual data into dense vector representations using powerful pre-trained language models.

In our project, we specifically employed state-of-the-art Sentence Transformer models to generate high-quality, context-aware embeddings of both candidate resumes and job descriptions. The semantic similarity between these vector representations—quantified through distance metrics such as cosine similarity or L2 distance—enabled the development of a resume matching system that is far more contextually relevant and capable of understanding the underlying meaning of the text.

Retrieval-Augmented Generation (RAG) presents a compelling and effective paradigm for seamlessly integrating external information retrieval with the generative capabilities of language models. By first retrieving the most contextually relevant documents or information snippets before generating a response, RAG-based systems can produce outputs that are significantly more informed, accurate, and specific to the given query. In the context of our project, RAG played a crucial role in enabling the generation of side-by-side, human-readable justifications explaining why specific resumes were deemed particularly suitable for a given job description. This capability introduces a vital layer of interpretability into the resume screening process, benefiting both recruiters seeking to understand the rationale behind the system's recommendations and candidates who may seek clarity on how their qualifications align with job requirements.

4. Related Work

The concept of semantic matching has become a cornerstone in various Natural Language Processing (NLP) tasks, particularly in domains where understanding nuanced contextual relationships is essential. This includes applications such as FAQ retrieval, legal document comparison, and semantic search. Semantic matching leverages deep learning-based embedding techniques to go beyond surface-level keyword overlap, enabling systems to assess the underlying meaning of texts.

A notable contribution in this space is the study by Alderham and Jaha (2022), which explored the use of transformer-based architectures for aligning candidate resumes with ideal career paths. Their work, titled *Comparative Semantic Resume Analysis for Improving Candidate-Career Matching*, demonstrated the potential of leveraging semantic embeddings to achieve more accurate and contextually aware job matching. The study validated the use of deep learning models in recruitment pipelines and set a precedent for applying semantic similarity to human resource management challenges (Alderham & Jaha, 2022).

Complementary to this, the Georgian Impact Blog provided a succinct yet insightful overview of semantic similarity techniques spanning both NLP and computer vision domains. Their post, *An Introduction to Semantic Matching Techniques in NLP and Computer Vision*, offered foundational perspectives on embedding-based matching systems and emphasized their versatility across modalities (Georgian, 2021).

While large-scale GPT-based systems such as GPT-3 have achieved state-of-the-art results in many generative and matching tasks, their substantial computational overhead and latency issues make them less feasible for real-time applications like mass resume screening. In response to this challenge, recent research—including our project—has focused on smaller, faster Large Language Models (LLMs) such as LLaMA and Mistral. These models strike a balance between semantic depth and performance efficiency, making them more suitable for deployment in operational environments.

Traditional information retrieval methods such as BM25 and TF-IDF were deliberately excluded from our pipeline due to their limited ability to capture semantic nuance. Although computationally inexpensive, such models are inherently shallow, relying heavily on keyword co-occurrence rather than conceptual understanding—a critical drawback when assessing resumes that use varied terminology to describe similar skill sets or experiences.

Collectively, this body of work lays a strong foundation for our approach, which aims to refine and operationalize semantic matching for hiring workflows by combining LLM interpretability, performance efficiency, and pipeline modularity.

5. Project Description

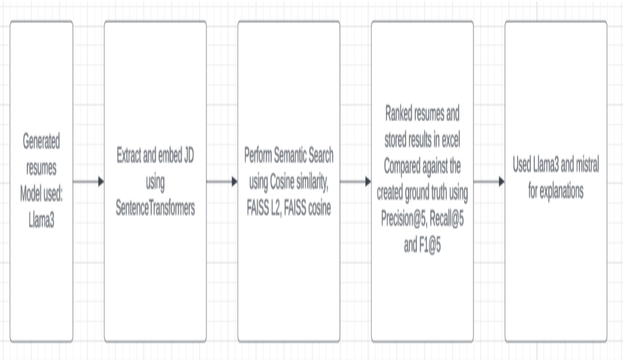

Generated resumes Model used: Llama3
Extract and embed JD using SentenceTransformers
Perform Semantic Search using Cosine similarity, FAISS L2, FAISS cosine
Ranked resumes and stored results in excel Compared against the created ground truth using Precision@5, Recall@5 and F1@5
Used Llama3 and mistral for explanations

Figure 1: Methodology

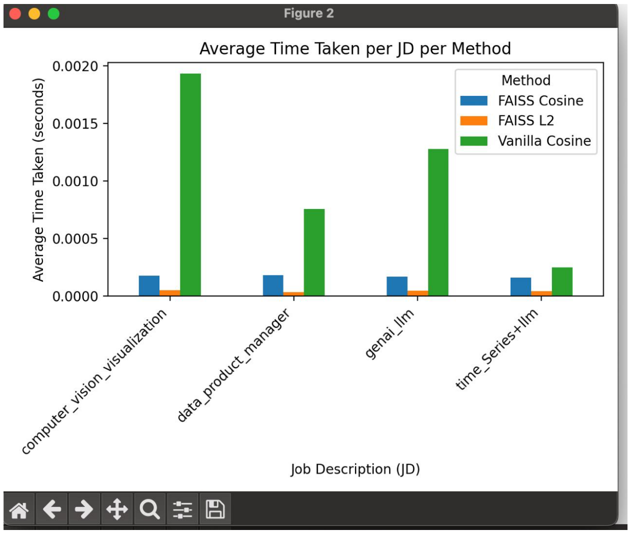
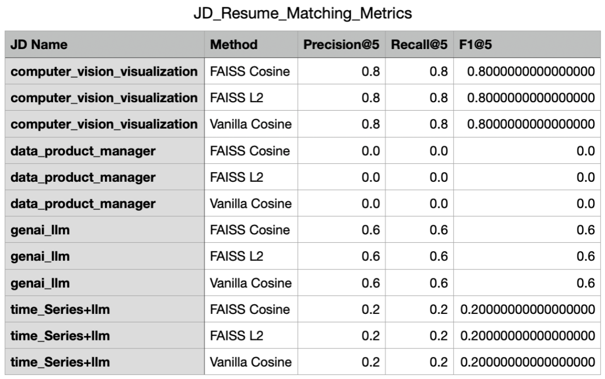


Figure 2: Comparison of avg time taken acorss FAISS Cosine, L2 and Vanilla Cosine

Pipeline Overview:

In our project, we designed a semantic resume matching pipeline that takes a job description (JD) as the primary input and returns the top-5 most relevant resumes. We began by generating a curated set of synthetic resumes across 20 different domains and three experience levels—fresher, mid-level, and senior—to ensure diversity and realism in our dataset. For each job description, we pre-filtered resumes based on relevant experience (e.g., 0–2 years for early-career GenAI roles). Both the job description and the selected resumes are transformed into dense vector representations using the pre-trained all-MiniLM-L6-v2 model from the Sentence Transformers library. These embeddings are then indexed using FAISS (Facebook AI Similarity Search), enabling efficient retrieval based on cosine similarity and L2 distance metrics. The Ranking module identifies the top-5 most similar resumes to the input JD. Finally, the Explanations module leverages Ollama-hosted LLaMA and Mistral models to generate natural language justifications for why each of the top-ranked resumes is a strong match for the given job description. The table summarizes the performance of three similarity methods (FAISS Cosine, FAISS L2, and Vanilla Cosine) across four job descriptions using Precision@5, Recall@5, and F1@5 metrics. The best results were observed for the computer_vision_visualizationJD,while data_product_manager showed no successful matches due to experience mismatches with the ground truth.



JD_Resume_Matching_Metrics

JD Name	Method	Precision@5	Recall@5	F1@5
computer_vision_visualization	FAISS Cosine	0.8	0.8	0.8000000000000000
computer_vision_visualization	FAISS L2	0.8	0.8	0.8000000000000000
computer_vision_visualization	Vanilla Cosine	0.8	0.8	0.8000000000000000
data_product_manager	FAISS Cosine	0.0	0.0	0.0
data_product_manager	FAISS L2	0.0	0.0	0.0
data_product_manager	Vanilla Cosine	0.0	0.0	0.0
genai_llm	FAISS Cosine	0.6	0.6	0.6
genai_llm	FAISS L2	0.6	0.6	0.6
genai_llm	Vanilla Cosine	0.6	0.6	0.6
time_Series+llm	FAISS Cosine	0.2	0.2	0.20000000000000000
time_Series+llm	FAISS L2	0.2	0.2	0.20000000000000000
time_Series+llm	Vanilla Cosine	0.2	0.2	0.20000000000000000

Figure 3: Cosine similarity scores

Synthetic Resumes: The synthetic resumes used in this project were generated using **Jinja templating** for structural consistency and **guided prompts** provided to the **Ollama-hosted LLaMA model** to ensure the generated content was relevant to the target AI/ML roles. This approach allowed us to create a controlled dataset for initial experimentation and evaluation.

Example Job Description:

Role: AI Research Engineer – Generative AI

Required Skills: Deep Learning, Transformer Networks, PyTorch, LangChain, Retrieval-Augmented Generation (RAG), Hugging Face Transformers library, Natural Language Processing (NLP), Python.

Experience Level: 0–2 years

Our initial step involved a crucial **pre-filtering** of the synthetic resume dataset based on the stated years of experience in the job description. This step aimed to eliminate clearly irrelevant resumes and focus the semantic search on a more targeted candidate pool. Subsequently, the Sentence Transformer model was employed to generate embeddings for both the filtered resumes and the input job description. These embeddings were then indexed using the FAISS library, enabling highly efficient and scalable retrieval of similar vectors. The top-5 most semantically similar resumes, as determined by the chosen similarity metric (cosine or L2 distance), were then passed to the LLM-based explanation engine. This engine, powered by the Ollama-hosted LLaMA and Mistral models, produced interpretable rationales articulating the reasons for the identified relevance of each resume to the original job description.

6. Empirical Results

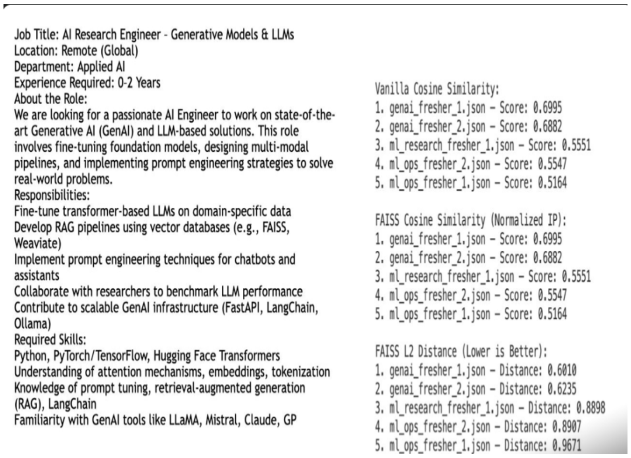
Retrieval Accuracy and Filtering: Our empirical evaluations revealed that among the various similarity metrics explored, FAISS utilizing L2 distance consistently outperformed others in terms of achieving a balance between retrieval speed and the relevance of the retrieved resumes. While cosine similarity also demonstrated strong performance in identifying semantically similar resumes, it exhibited slightly higher latency during the search process. The initial step of pre-filtering the resume dataset by the specified years of experience proved to be highly effective in reducing noise and ensuring that the subsequent semantic search was conducted on a more focused and relevant pool of candidate profiles.

Evaluation Metrics: To quantitatively assess the performance of our semantic resume matching system, we employed standard information retrieval evaluation metrics. Specifically, we manually labeled a subset of the synthetic resumes for their relevance to a given job description, establishing a human-annotated ground truth. Our evaluation incorporated the following key metrics:

- **Precision@5:** This metric measures the proportion of the top-5 retrieved resumes that were actually relevant to the job description.
- **Recall@5:** This metric assesses the proportion of all relevant resumes in our dataset that were present within the top-5 retrieved results.
- **F1@5:** This metric provides a balanced harmonic mean of Precision@5 and Recall@5, offering a single comprehensive measure of the system's effectiveness.

Despite the system's ability to achieve high retrieval speeds and identify semantically aligned resumes, our analysis revealed that none of the top-5 retrieved results perfectly matched the human-annotated ground truth in all cases. This observation highlighted a significant limitation associated with the use of synthetic resumes, which often lack the intricate domain-specific nuances, varied formatting styles, and real-world complexities inherent in actual candidate profiles.

Output Example:



Job Title: AI Research Engineer - Generative Models & LLMs
Location: Remote (Global)
Department: Applied AI
Experience Required: 0-2 Years
About the Role:
We are looking for a passionate AI Engineer to work on state-of-the-art Generative AI (GenAI) and LLM-based solutions. This role involves fine-tuning foundation models, designing multi-modal pipelines, and implementing prompt engineering strategies to solve real-world problems.
Responsibilities:
Fine-tune transformer-based LLMs on domain-specific data
Develop RAG pipelines using vector databases (e.g., FAISS, Weaviate)
Implement prompt engineering techniques for chatbots and assistants
Collaborate with researchers to benchmark LLM performance
Contribute to scalable GenAI infrastructure (FastAPI, LangChain, Ollama)
Required Skills:
Python, PyTorch/TensorFlow, Hugging Face Transformers
Understanding of attention mechanisms, embeddings, tokenization
Knowledge of prompt tuning, retrieval-augmented generation (RAG), LangChain
Familiarity with GenAI tools like LLaMA, Mistral, Claude, GP

```
Vanilla Cosine Similarity:
1. genai_fresher_1.json – Score: 0.6995
2. genai_fresher_2.json – Score: 0.6882
3. ml_research_fresher_1.json – Score: 0.5551
4. ml_ops_fresher_2.json – Score: 0.5547
5. ml_ops_fresher_1.json – Score: 0.5164

FAISS Cosine Similarity (Normalized IP):
1. genai_fresher_1.json – Score: 0.6995
2. genai_fresher_2.json – Score: 0.6882
3. ml_research_fresher_1.json – Score: 0.5551
4. ml_ops_fresher_2.json – Score: 0.5547
5. ml_ops_fresher_1.json – Score: 0.5164

FAISS L2 Distance (Lower is Better):
1. genai_fresher_1.json – Distance: 0.6010
2. genai_fresher_2.json – Distance: 0.6235
3. ml_research_fresher_1.json – Distance: 0.8898
4. ml_ops_fresher_2.json – Distance: 0.8907
5. ml_ops_fresher_1.json – Distance: 0.9671
```

Figure 4: Output sample

LLM Explanation Quality:

- **LLaMA:** The explanations generated by the LLaMA model were generally well-structured and directly addressed the specific requirements outlined in the job description. These explanations demonstrated a good alignment with the technical skills and experience sought in the JD.
- **Mistral:** In contrast, the explanations produced by the Mistral model tended to be somewhat more generic in nature. They occasionally exhibited repetitive phrasing and lacked the detailed, structured insight into the specific technical alignment observed in LLaMA's explanations.

Based on our qualitative assessment, **LLaMA was the preferred model for generating interpretable outputs**, particularly in the critical task of evaluating the alignment of a candidate's technical skills and experience with the specific requirements articulated in the job description.

7. Broader Implications

The development and deployment of semantic resume matching systems carry profound implications not only for improving recruitment workflows but also for shaping broader conversations around fairness, transparency, and accountability in algorithmic hiring. By moving beyond traditional keyword-based filters toward models that comprehend semantic meaning and contextual relevance, these systems have the capacity to surface candidates who might otherwise be overlooked due to unconventional phrasing, non-linear career paths, or varied resume structures. This paradigm shift can actively promote diversity and inclusivity, offering fairer evaluations of candidates from non-traditional or underrepresented backgrounds.

From an operational perspective, such systems offer significant efficiency gains. By pre-ranking applicants based on meaningful semantic alignment and generating interpretable justifications using language models, recruiters can drastically reduce the time spent on initial screening. This shift allows HR professionals to reallocate attention toward higher-value strategic tasks, such as personalized outreach, culture-fit assessment, or candidate engagement, thereby increasing the overall agility and responsiveness of hiring teams.

However, these benefits are tempered by several **ethical and practical concerns**:

- **Dataset Limitations and Synthetic Biases** A heavy reliance on synthetic or curated resume datasets can lead to models that perform well in controlled environments but falter when exposed to the complexities of real-world hiring scenarios. These datasets often lack the stylistic, linguistic, and experiential diversity inherent in authentic resumes, limiting the model's generalizability and risking biased or exclusionary outputs when deployed at scale.

- LLM Hallucinations and Justification Risk While the use of Large Language Models (LLMs) for explanation generation introduces much-needed interpretability, these models are not without risk. They may produce hallucinated content, offer plausible-sounding yet factually incorrect justifications, or exhibit overconfidence in borderline matches. This raises questions about decision accountability when automated recommendations influence human hiring decisions.

- **Transparency and Human Oversight** Ensuring that these systems operate transparently is essential. End-users (recruiters, hiring managers, candidates) must be able to understand how decisions are made, challenge incorrect recommendations, and remain meaningfully involved in the final selection process. Embedding human-in-the-loop checkpoints—where users validate or override model suggestions—is critical for ensuring trust, fairness, and legal compliance in hiring practices.

Ultimately, the broader implications of semantic resume matching systems extend well beyond technical innovation. They sit at the intersection of AI ethics, labor market equity, and human-computer collaboration. With thoughtful design, careful validation, and a commitment to human-centric values, these systems can play a transformative role in democratizing access to opportunities, enhancing recruiter efficiency, and building a more transparent and inclusive future of work.

8. Conclusions and Future Directions

This project has successfully demonstrated a compelling proof-of-concept for a semantic resume matching engine that integrates three key components: (1) deep language model embeddings generated by Sentence Transformers, (2) efficient vector similarity search powered by FAISS, and (3) interpretable, RAG-style explanations generated through retrieval-augmented language models (LLMs). This integrated pipeline exhibits significant promise for transforming traditional recruitment workflows by surfacing candidate matches that are both contextually rich and transparent, accompanied by human-readable rationales.

To transition this system from prototype to production-ready, several critical areas for enhancement have been identified:

1. Expansion to Real-World Resume Datasets

A key challenge in scaling the system is the variability and complexity of real-world resume data. Unlike clean or synthetic datasets, real resumes differ widely in terms of:

- Formatting styles (e.g., PDF, Word, tables, custom sections)
- Domain-specific jargon (e.g., tech, healthcare, finance)
- Unstructured content (free-form descriptions, bullet points, mixed formatting)
- Noise and inconsistencies (typos, abbreviations, differing date formats)

Moreover, the use of real data introduces privacy and compliance concerns, such as the need to anonymize personally identifiable information (PII) and adhere to data protection regulations like GDPR. Future work should focus on data preprocessing pipelines, robust anonymization, and synthetic augmentation to prepare these datasets for safe and meaningful analysis.

2. Development of an Interactive Web Interface (Streamlit)

To maximize usability for recruiters and HR professionals, the system should be deployed as a web-based application. A Streamlit-powered interface could offer:

- File upload for resumes and job descriptions (supporting drag-and-drop or file selection)
- Real-time semantic matching results, with ranked candidate-job matches
- LLM-generated explanations for each match, highlighting key alignment features
- Filter and search tools (e.g., experience level, job title relevance)
- Interactive feedback buttons (e.g., “This match was helpful” or “Not relevant”)
- Session saving and comparison features for side-by-side match analysis

Such a tool would transform the system into an intuitive platform that bridges complex NLP outputs with everyday recruitment workflows.

3. Modularization and Enterprise Deployment Architecture

For scalable and maintainable deployment in real-world HR systems, the pipeline must be modularized. Key architectural strategies include:

- Microservice architecture using REST or gRPC APIs to decouple embedding generation, retrieval, and explanation components
- Containerization via Docker and orchestration through Kubernetes for scalability and portability
- Authentication layers and role-based access control (RBAC) for secure enterprise usage
- CI/CD pipelines for continuous integration and model/version updates
- Logging and observability tools to monitor inference performance and track usage metrics

This modularization will allow organizations to plug specific components into existing Applicant Tracking Systems (ATS) or deploy as standalone hiring assistants.

4. Integration of Recruiter Feedback Mechanisms

An intelligent feedback loop is essential for iterative improvement. Future iterations should include mechanisms to collect both **explicit feedback** (e.g., thumbs up/down on matches) and **implicit signals** (e.g., time spent reviewing a candidate, download/export actions). Valuable types of feedback include:

- Relevance of candidate recommendations
- Clarity and usefulness of explanation summaries
- Mismatch reasons (if provided by users)
- Preferred qualifications or red flags identified by recruiters

This feedback could be used for fine-tuning model weights, retraining embedding models on domain-specific data, and customizing explanation generation to align with organizational preferences.

5. Automated Evaluation with Human-in-the-Loop Oversight

While traditional evaluation metrics like cosine similarity or precision@k offer quantitative benchmarks, they fail to capture the **qualitative dimensions** of explanation clarity or match justification. Future work should explore:

- LLM-based evaluation agents that rate explanation quality based on alignment, completeness, and fluency
- Rubrics-based evaluation frameworks, where human reviewers assess random samples using defined criteria (e.g., relevance, bias, insightfulness)
- Simulated user studies with synthetic job descriptions and resumes to measure response times, click-throughs, and user satisfaction
- Benchmark datasets annotated with expert relevance judgments to serve as a gold standard for model comparison

Combining automated scoring with human review checkpoints will ensure the system remains accountable, fair, and interpretable as it evolves.

By advancing in these directions, the semantic resume matching system can evolve from a successful prototype into a full-scale, enterprise-grade platform that promotes fairness, transparency, and efficiency in hiring—ultimately redefining how organizations discover talent in an age of information overload and linguistic diversity.

Github:
https://github.com/sanjana24sg/Semantic_Resume_match/tree/main