International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

CerebroGuide: An AI-Powered Career Navigation System with Intelligent Resume Analysis and Domain-Tuned Transformer Models

Krishna Kumar Jha[1], Jay Sahu[2], Palak Agarwal[3], Harshvardhan Rathore[4]

Department of Computer Science, Oriental Institute of Science and Technology,
Bhopal, Madhya Pradesh, India
*Corresponding Author

ABSTRACT :

The global employment ecosystem is evolving faster than ever before under the influ- ence of artificial intelligence (AI), automation, and large-scale digitization. Job roles are emerging, mutating, and disappearing within short time frames, while organizations increasingly favor skill-based hiring over traditional degree-centric recruitment. In this landscape, students, early-career professionals, and even experienced employees struggle to make informed career decisions that align with both their competencies and real-time industry demands.

Traditional career guidance systems and widely deployed Applicant Tracking Systems (ATS) are no longer sufficient. Conventional tools are typically built on keyword match- ing, Boolean queries, and simple scoring rules. These approaches overlook semantic mean- ing, contextual relevance, and transferable skills expressed in varied natural language. For example, resumes that mention "ML", "Neural Networks" or "Deep Learning" may rep- resent the same underlying expertise, yet a keyword-based system often treats them as unrelated. As a result, capable candidates are rejected, hiring cycles are extended, and skill mismatches persist.

This paper presents **CerebroGuide**, an AI-powered career navigation and intelligent recruitment system that addresses these shortcomings through deep semantic analysis of resumes and job descriptions (JDs). The system integrates domain-tuned transformer models, Sentence-BERT (SBERT) based embeddings, and Named Entity Recognition (NER) to derive a rich representation of candidate profiles and job requirements. Cere- broGuide computes an interpretable job-fit score, detects skill gaps, and recommends targeted learning pathways to support continuous professional growth.

An experimental evaluation based on real-world resumes and job descriptions demon- strates that CerebroGuide significantly outperforms traditional ATS baselines in terms of accuracy, precision, recall, F1-score, and user satisfaction. Beyond raw performance, the system emphasizes transparency, user-centric feedback, and ethical handling of sensitive information. Together, these properties position CerebroGuide as a holistic framework for modern career guidance and recruitment intelligence.

Index Terms— Artificial Intelligence, Career Guidance, Resume Intelligence, Semantic Matching, Natural Language Processing, Skill Gap Analysis, Intelligent Recruitment, Transformers, BERT.

Introduction

The employment landscape of the twenty-first century is undergoing a profound transfor- mation driven by rapid advances in artificial intelligence, automation, cloud computing, and globalization. Industries such as software engineering, data science, fintech, health- care technology, and autonomous systems demand highly dynamic skill sets that evolve continuously with technological innovation. Static degree-based hiring processes are grad- ually being replaced by skill-based hiring, where demonstrable competencies, adaptability, and continuous learning are crucial.

For individuals, this shift has a dual effect. On one hand, there are more opportu- nities than ever before. On the other, the increasing complexity of job roles, tools, and technology stacks makes it difficult to understand where one truly fits. Students must de- cide which internships, projects, and certifications genuinely increase their employability. Mid-career professionals often wonder how to transition from one role to another, such as from manual testing to automation, or from software development to machine learning.

Organizations face a parallel challenge on the recruitment side. Modern companies routinely receive hundreds or thousands of resumes for a single open position. Manu- ally reading and evaluating each resume is not feasible at scale. This reality has led to widespread deployment of Applicant Tracking Systems (ATS), which attempt to auto- mate the screening phase. However, most legacy ATS solutions focus primarily on match- ing keyword frequency between resumes and job descriptions, often with some TF–IDF weighting or Boolean rules layered on top. While these methods are efficient, they are fundamentally shallow. They do not understand context, meaning, or transferable knowl- edge.

The consequences are serious for both sides. Qualified candidates are rejected due to vocabulary mismatches or unconventional formatting. Recruiters are presented with lists of candidates who appear relevant by keywords but are not suitable in practice. Skill gaps are rarely identified explicitly, and rejected applicants are seldom told what they need to improve. This creates a frustrating, opaque, and sometimes unfair ecosystem.

CerebroGuide is proposed as a response to this situation. It aims to move beyond superficial keyword matching and create a semantic, context-aware career guidance and recruitment system. Rather than simply counting occurrences of words like "Python", "Java", or "Machine Learning", CerebroGuide interprets resumes and job descriptions as rich documents that encode skills, experiences, domains, and learning trajectories. It combines transformer-based language models, semantic embeddings, NER-driven skill extraction, and explainable scoring functions to offer transparent recommendations that benefit job seekers, recruiters, and academic institutions alike.

The remainder of this paper elaborates on the underlying problem, existing literature, system design, algorithmic workflow, experimental evaluation, ethical considerations, and future extensions of CerebroGuide.

Problem Statement

Traditional career guidance and recruitment systems were designed for a relatively stable labor market, in which job roles changed slowly, and the primary objective was matching degrees to predefined positions. In the current era, this paradigm has broken down.

1. Limitations of Existing Career Guidance Approaches

Most legacy career counseling frameworks rely on:

- Static questionnaires and psychometric tests,
- One-time aptitude assessments,
- Manual counseling sessions based on counselor intuition,
- Generic suggestions that are weakly tied to real-time job market trends.

These methods have three critical limitations:

1. They do not adapt dynamically when industry skill requirements change.
2. They rarely integrate large-scale job market data or evolving technology stacks.
3. They offer limited personalization, often grouping many students into broad cate- gories like "software developer" or "data analyst" without nuanced differentiation.

2. Limitations of Keyword-Based ATS Systems

On the recruitment side, Applicant Tracking Systems were introduced to automate resume filtering. However, conventional ATS platforms rely heavily on keyword-based matching and Boolean queries (AND, OR, NOT). They may internally use TF–IDF or similar statistical measures, but the core limitation remains: they treat language as bags of words rather than structured, meaningful expressions.

Some typical failure scenarios include:

- A candidate writes "ML" instead of "Machine Learning", leading to missed matches.
- A resume states "I do *not* know Java", yet the keyword "Java" alone boosts the score.
- A role requires "Neural Networks" while the resume mentions "Deep Learning". Semantically they overlap, but keyword systems treat them as distinct.

This *semantic gap* leads to:

- Rejection of capable candidates due to vocabulary mismatch,
- Over-selection of candidates who have filled their resume with keywords without genuine expertise,
- Inefficient hiring cycles, longer time-to-fill, and reduced trust in ATS outcomes.

3. Lack of Skill Gap Feedback

Perhaps the most important missing element is honest, actionable feedback. Traditional systems almost never tell candidates:

"You are a 72% match for this job. You are strong in X and Y, but you are missing Z and W. Here are recommended resources to bridge those gaps."

Instead, applicants receive binary outcomes: shortlisted or rejected. This not only demotivates job seekers, it also slows down their skill development because they have no clear guidance on what to improve.

4. Core Problem

The core research problem addressed in this work is therefore:

How can we design an AI-powered system that understands resumes and job descrip- tions at a semantic level, produces interpretable job-fit scores, detects skill gaps, and supports continuous career development, while remaining scalable and ethically responsi- ble?

Objectives

In light of the above problem statement, the primary objectives of this research are:

- **Resume Intelligence:** To automatically extract structured information (skills, tools, experiences, domains, education) from unstructured resumes in formats such as PDF and DOCX.
- **Semantic Understanding:** To utilize transformer-based language models to un- derstand the *context* of skills and experiences rather than relying on raw keyword occurrence.
- **Job-Fit Scoring:** To compute a quantitative, interpretable job-fit score that re- flects semantic similarity, skill overlap, and experience alignment between a candi- date and a given job description.
- **Skill Gap Analysis:** To identify missing or weak skills relative to a target job and map them to specific learning resources, certifications, or courses.
- **User-Centric Feedback:** To offer transparent explanations so that candidates understand why they are or are not a good match for particular roles.
- **Scalability and Adaptability:** To design the system using modular, microservices- friendly components that can scale with data and evolve alongside new models.

CerebroGuide is conceptualized not just as a screening engine but as an intelligent *career companion* that accompanies students and professionals throughout their growth journey.

3 Literature Review

The development of CerebroGuide is grounded in several decades of research on infor- mation retrieval, machine learning, and natural language processing. This section briefly reviews the main paradigms and highlights the gaps that motivate our approach.

1. Statistical and Keyword-Based Approaches

The earliest generation of ATS and document retrieval systems relied on keyword-based logic and simple scoring. With the digitization of recruitment pipelines, methods such as Term Frequency–Inverse Document Frequency (TF–IDF) became popular. TF–IDF is typically defined as:

$$TF(t,d) = \frac{\text{count}(t,d)}{\sum_{t' \in d} \text{count}(t',d)} \quad (1)$$

where t is a term and d is a document. The intuition is that terms that appear frequently in a document but not across all documents carry discriminative power.

These approaches were attractive due to their simplicity and efficiency. However, they suffer from critical limitations:

- They ignore word order and sentence structure.
- They treat antonyms and negations badly (“I do not know Java” vs “I know Java”).
- They cannot capture higher-level concepts like “data visualization” vs “Matplotlib” or “Neural Networks” vs “Deep Learning”.

As a result, TF–IDF and basic keyword filters struggle with nuanced resume-job matching.

2. Machine Learning Classifiers

To overcome some of these limitations, researchers introduced supervised machine learn- ing models such as Naive Bayes, Support Vector Machines (SVM), and Logistic Regression for tasks like resume classification and candidate ranking. These models treat the prob- lem as a classification or scoring task based on engineered features (e.g., counts of specific skills, years of experience, education level).

While such models achieve higher accuracy than naive keyword matching, they depend heavily on:

- Reliable labeled datasets,

- Carefully engineered features,
- Domain-specific tuning.

Furthermore, generalization across domains is weak: a model trained on IT resumes may perform poorly on healthcare or mechanical engineering profiles. The models also struggle with the richness and variability of natural language.

3. The Transformer Revolution

The introduction of transformer-based architectures, especially BERT (Bidirectional En- coder Representations from Transformers), fundamentally changed NLP. BERT produces

contextual embeddings, meaning the representation of a word depends on its surround- ing sentence. This is crucial for phrases like "Java developer", "Java course", and "Java island", which carry different meanings.

Sentence-BERT (SBERT) builds upon BERT by optimizing embeddings at the sen- tence or document level specifically for semantic similarity tasks. It allows us to map resumes and job descriptions into a dense vector space where semantically similar texts are close, even when they use different vocabulary.

Recent domain-adapted models such as CareerBERT further fine-tune transformers on job market data, improving performance in job classification and recommendation tasks. The literature shows consistent gains in accuracy and ranking quality when semantic embeddings are used instead of pure keyword-based systems.

4. Research Gap

Despite these advancements, many real-world ATS deployments still rely on legacy key- word techniques due to integration challenges, computational costs, or lack of explain- ability. There is a clear need for a system that:

- Leverages transformer-based semantics,
- Incorporates explicit skill gap analysis,
- Provides interpretable job-fit explanations,
- Can be deployed as a scalable, modular service.

CerebroGuide is designed to occupy this gap by combining mature deep learning tools with practical recruitment and career guidance workflows.

Proposed System Methodology

CerebroGuide follows a structured, modular processing pipeline that converts raw user input (resumes and job descriptions) into actionable outputs (ranked job matches, skill gap reports, and learning suggestions).

1. High-Level System Architecture

At a high level, the system is organized into three main layers:

- **Presentation Layer:** A web-based interface built using React.js, where users upload resumes, browse recommended jobs, and view their skill analytics dashboard.
- **Backend Orchestration Layer:** A Node.js/Express server that manages authen- tication, API routing, job queueing, and communication between the database and the AI computation engine.
- **AI Computation Engine:** A Python-based microservice implemented using FastAPI and PyTorch, hosting SBERT models, NER pipelines, and scoring algorithms.

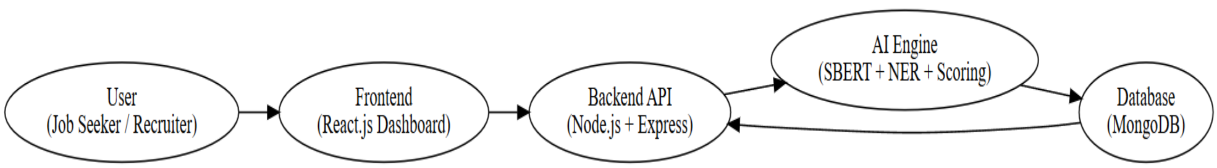

Figure 1: Overall System Architecture of CerebroGuide

The architecture is deliberately modular so that, for example, the transformer model can be swapped out or upgraded without affecting the user interface or the underlying database schema.

Data Ingestion and Preprocessing

Resumes and job descriptions arrive in diverse formats: DOCX, PDF, and plain text. Some resumes may even be scanned images exported as PDFs. To handle this diversity, CerebroGuide includes a robust data ingestion and preprocessing pipeline.

2. Text Extraction

For DOCX and text files, we use structured parsers to extract content while preserving section boundaries such as education, experience, and skills when possible. For PDF files, including scanned resumes, Optical Character Recognition (OCR) is applied to recover textual content.

3. Cleaning and Normalization

Once raw text is obtained, the system performs:

- **Tokenization:** Splitting text into words or sub-word tokens.
- **Lowercasing:** Converting all characters to lowercase to reduce sparsity.
- **Punctuation and Noise Removal:** Removing extraneous symbols that do not contribute meaning.
- **Stop-word Removal:** Discarding very common words such as “and”, “the”, and “is” which add little semantic weight.
- **Lemmatization:** Reducing words to their base forms (e.g., “developing” → “develop”) so that related morphological variants are recognized as the same concept.

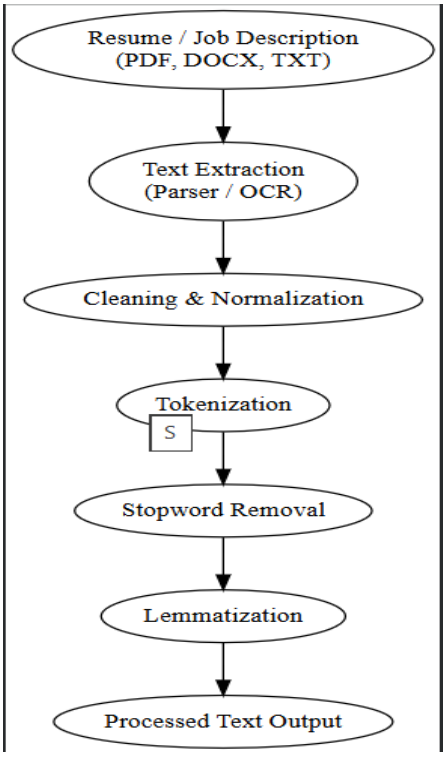

Figure 2: Data Ingestion and Preprocessing Workflow

The result of this stage is a set of clean, normalized text representations for each resume and job description, ready for semantic modeling.

Semantic Processing and Embedding

The core of CerebroGuide’s intelligence lies in its use of semantic embeddings. We employ a pre-trained SBERT model (such as *all-MiniLM-L6-v2*) fine-tuned for sentence-level similarity.

4. Resume and JD Embedding

Let T_r denote the normalized text of a resume and T_j denote the normalized text of a job description. The embedding step is defined as:

$$V_r = \mathrm{BERT}(T_r), \qquad V_j = \mathrm{BERT}(T_j) \tag{2}$$

Here, V_r and V_j are high-dimensional vectors (e.g., 384 dimensions) capturing the semantic content of the entire resume and job description respectively.

5. Cosine Similarity for Semantic Match

To compare these vectors, CerebroGuide computes cosine similarity:

$$\mathrm{Similarity}(V_r, V_j) = \frac{V_r \cdot V_j}{\|V_r\| \, \|V_j\|} \tag{3}$$

A higher similarity value indicates that the resume and job description discuss related topics, skills, and experiences, even if they use different words.

6. Skill Extraction via NER

Simultaneously, Named Entity Recognition (NER) is performed to extract explicit skills, tools, platforms, organizations, and role titles. A domain-adjusted NER model (for ex- ample, based on spaCy with custom training) is used to recognize entities like:

- Programming languages (e.g., Python, Java, C++)
- Frameworks (e.g., React, Django, TensorFlow)
- Cloud platforms (e.g., AWS, Azure, GCP)
- Roles (e.g., Data Analyst, DevOps Engineer)

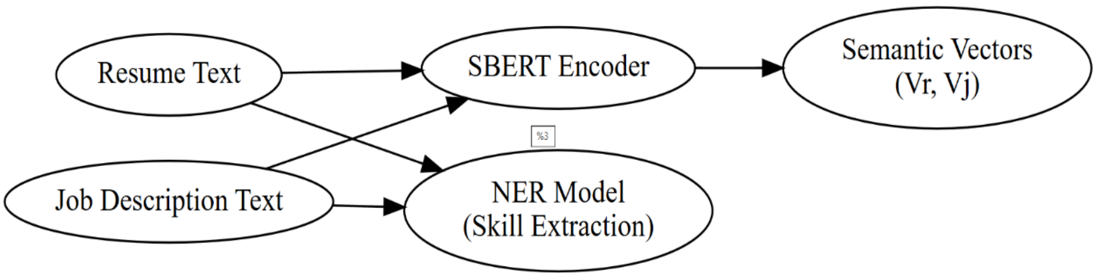

Figure 3: Semantic Embedding and Feature Extraction Process

The combination of dense semantic vectors and discrete skill entities provides a rich foundation for downstream scoring and gap analysis.

Matching Algorithm and Job-Fit Scoring

Although cosine similarity between V_r and V_j is a powerful signal, it is not sufficient on its own. CerebroGuide introduces a composite scoring function that combines semantic similarity, skill overlap, and experience alignment.

7. Composite Score

Let:

- S_{sem} be the semantic similarity between resume and JD,
- S_{skill} be a normalized measure of skill overlap between the sets of extracted skills,
- S_{exp} be an experience match score, reflecting years and relevance of experience.

The overall job-fit score is defined as:

$$\mathrm{Score} = W_1 \cdot S_{\text{sem}} + W_2 \cdot S_{\text{skill}} + W_3 \cdot S_{\text{exp}} \tag{4}$$

where W_1, W_2, and W_3 are tunable hyperparameters. In the current implementation,

a representative choice is:

$$W_1 = 0.5, \qquad W_2 = 0.3, \qquad W_3 = 0.2.$$

These weights can be adjusted based on recruiter preference or validated empirically against ground truth hiring data.

8.2 Skill Overlap and Normalization

Let S_r be the set of skills extracted from the resume, and S_j be the set of skills required by the job. The skill overlap score can be computed as:

$$S_{\text{skill}} = \frac{|S_r \cap S_j|}{|S_j|} \qquad (5)$$

This ratio measures how many of the required skills are present in the resume, relative to the total number of skills the role expects.

3. **Experience Match**

Experience match can be computed by comparing:

- The number of years of relevant experience,
- The domains in which the candidate has worked (e.g., finance, healthcare),
- The seniority level indicated in both resume and JD (e.g., junior, mid-level, senior). The experience match score S_{exp} is normalized to lie between 0 and 1.

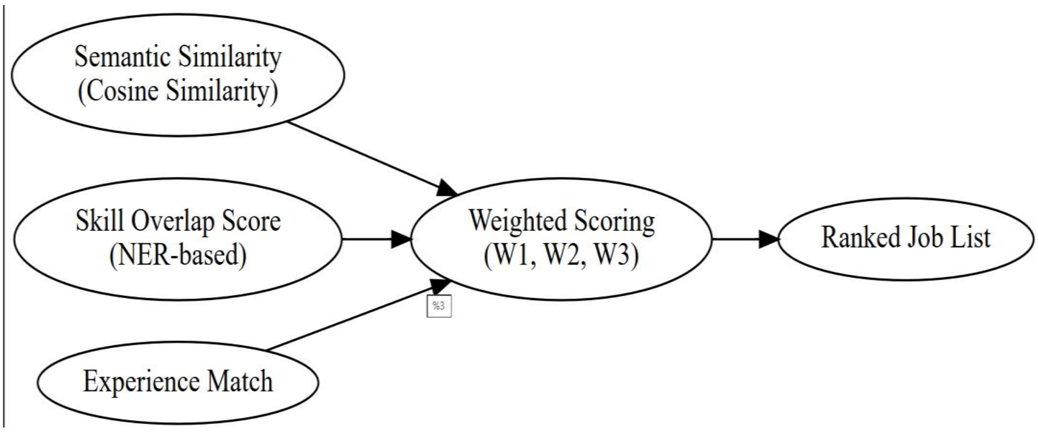

Figure 4: Job Matching and Scoring Architecture

This composite scoring formulation ensures that a candidate is not judged solely by keywords, but by a balanced integration of semantics, explicit skills, and practical experience.

Algorithmic Workflow

The overall algorithm that powers CerebroGuide's job recommendation and skill analysis can be summarized as follows.

8. **NeuroHire / CerebroGuide Matching Logic**

Input: Resume R, list of job descriptions $\{JD_1, JD_2, \ldots, JD_n\}$

Output: Ranked list of matching jobs and associated skill gaps

Steps:

1. Parse and preprocess the resume to obtain T_r.
2. Generate resume embedding V_r = BERT(T_r).
3. Extract resume skills S_r via NER.
4. For each job description JD_i:

   (a) Preprocess the job text to obtain T_{ji} .
   (b) Generate job embedding V_{ji} = BERT(T_{ji}).
   (c) Extract job skills S_{ji} via NER.
   (d) Compute semantic similarity $S_{\text{sem},i}$ between V_r and V_{ji} .
   (e) Compute skill overlap $S_{\text{skill},i}$ between S_r and S_{ji} .
   (f) Estimate experience match $S_{\text{exp},i}$.
   (g) Compute total score Score$_i$ using the weighted formula.
   (h) If Score$_i$ exceeds a configurable threshold, mark job JD_i as a candidate match.
   (i) Compute skill gap $S_{\text{gap},i} = S_{ji} \setminus S_r$.
   (j) Map each element of $S_{\text{gap},i}$ to available learning resources.

5. Sort all matching jobs in descending order of Score$_i$.
6. Return the ranked list of jobs along with skill gap reports and recommended learning items.

This workflow is implemented as a series of microservice calls, making it straight- forward to scale across multiple servers and to integrate with external job boards or institutional databases.

Skill Gap Analysis

Skill gap analysis is a central feature of CerebroGuide that distinguishes it from traditional ATS systems. Rather than treating rejection as an opaque outcome, the system explicitly identifies missing skills and provides guidance.

9. Skill Gap Computation

Given a set of required skills S_j for a target job and a set of possessed skills S_r extracted from the resume, the skill gap is defined as:

$$S_{\text{gap}} = S_j \setminus S_r \tag{6}$$

Each element of S_{gap} represents a competency that the candidate would likely need to acquire or strengthen to become a stronger match for that role.

10. Learning Path Recommendation

For each missing skill $s \in S_{\text{gap}}$, CerebroGuide consults a curated recommendation database that links skills to:

- Online courses (e.g., MOOCs on Coursera, Udemy),
- Certifications (e.g., cloud certifications),
- Books or official documentation,
- Project ideas for practical hands-on learning.

The system can then present a structured plan such as:
"To become a better fit for this Data Analyst role, you should learn: SQL (Beginner → Intermediate), Power BI dashboards, and basic statistics (hypothesis testing, confidence intervals). Here are three suggested courses and one sample project idea for each topic."

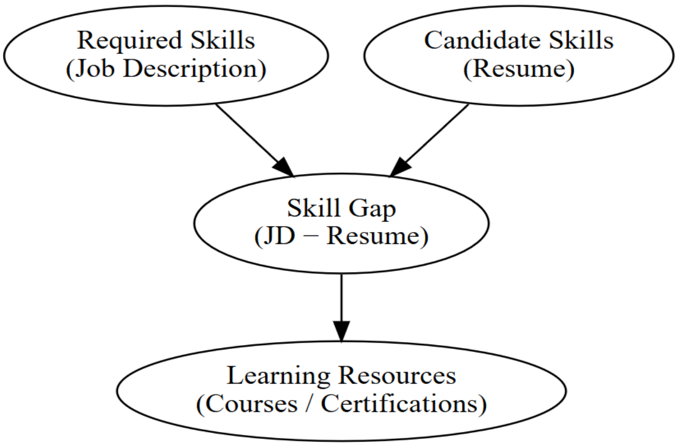

Figure 5: Skill Gap Detection and Recommendation Workflow

In this way, skill gap analysis transforms rejection into a constructive learning roadmap.

Implementation and Technical Framework

11. Technology Stack

CerebroGuide is implemented using a modern, open-source technology stack:

- **Frontend:** React.js with Tailwind CSS for responsive, user-friendly dashboards and form interfaces.
- **Backend Orchestration:** Node.js with Express.js for RESTful API routing, au- thentication (e.g., JWT-based), and session handling.
- **AI Service:** Python 3.9 with FastAPI, hosting PyTorch-based SBERT models, NER pipelines, and scoring logic.
- **Database:** MongoDB (NoSQL) to store user profiles, extracted features, job de- scriptions, recommendations, and logs. MongoDB's flexible document model is well-suited for semi-structured resume data.
- **Libraries:** HuggingFace Transformers, spaCy, NLTK, and scikit-learn for NLP and ML utilities.

12. Hardware Configuration

Model inference and fine-tuning experiments were conducted on a machine with the following specifications:

- CPU: Intel Core i7 (12th Gen),
- RAM: 32 GB DDR4,
- GPU: NVIDIA RTX 3060 with 12 GB VRAM, used to accelerate SBERT embed- dings and transformer computations.

This configuration is sufficient for batch processing of resumes and real-time infer- ence for individual users. The system can also be deployed to cloud environments with autoscaling.

Results and Performance Evaluation

To evaluate CerebroGuide, we conducted experiments on a curated dataset consisting of 500 resumes and 1000 job descriptions spanning technology, engineering, and management domains. The goal was to compare our system against two baselines:

- A TF–IDF based keyword matching model,
- A standard ATS-like keyword scoring system (Solr-based).

Evaluation Metrics

We employed standard information retrieval and ranking metrics:

- **Accuracy:** Proportion of correct match predictions.

- **Precision:** Fraction of recommended jobs that are truly relevant.
- **Recall:** Fraction of all relevant jobs that are successfully retrieved.
- **F1-score:** Harmonic mean of precision and recall.
- **MRR (Mean Reciprocal Rank):** Evaluates the ranking quality by examining the position of the first relevant result.

12.2 Quantitative Results

Table 1 summarizes the performance comparison.

Table 1: Performance Comparison of CerebroGuide vs Traditional Methods

Metric	TF–IDF	Standard ATS	CerebroGuide
Accuracy	62.4%	65.1%	**75.2%**
Precision	68.2%	70.5%	**78.5%**
Recall	58.9%	61.2%	**73.1%**
F1-score	63.2%	65.5%	**75.7%**
Avg. Time / JD	0.2s	0.5s	1.2 s

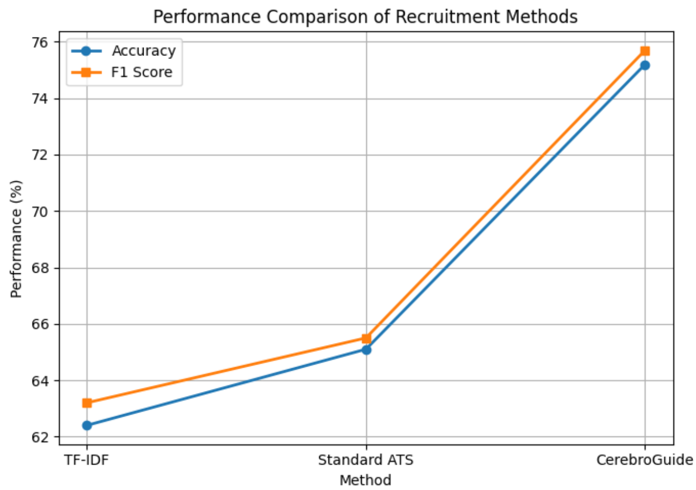

Figure 6: Accuracy and F1-Score Comparison Across Methods

While CerebroGuide has a slightly higher average processing time due to transformer computations, the gain in accuracy and F1-score is substantial and practically meaningful for recruitment decisions.

3. User Satisfaction Study

A beta group of 50 students used the system to upload their resumes and explore job rec- ommendations. After usage, they completed a feedback survey. The average satisfaction score was 4.3 out of 5.

Users particularly appreciated:

- The clarity of skill gap explanations,
- The concrete learning recommendations,
- The feeling that the system "understood" their resume beyond keywords.

This suggests that candidates value transparency and guidance more than opaque acceptance or rejection decisions.

Discussion

The experimental results and user feedback demonstrate that semantic understanding and skill gap analysis can substantially enhance both the technical accuracy and perceived usefulness of recruitment systems.

CerebroGuide's higher accuracy and F1-score indicate that transformer-based se- mantic embeddings capture nuance that keyword-based systems miss. Cases where re- sumes use synonyms, abbreviations, or unconventional phrasing are handled gracefully by SBERT embeddings. At the same time, the explicit use of skill sets and experience alignment ensures that the model's decisions are not purely "black box".

From a human perspective, the ability to see why a job is recommended—e.g., “strong match in Python, SQL, and cloud fundamentals, but missing advanced statistics”—builds trust. For students and job seekers, this turns the system into a mentor rather than a gatekeeper. For recruiters, the explainable job-fit score can assist in prioritizing candi- dates without replacing human judgment.

There are, however, trade-offs. Transformer models are computationally more ex- pensive than keyword filters. In large-scale enterprise deployments, careful engineering, batching, and caching become important to maintain responsiveness. Nonetheless, the experiments suggest that the added latency is acceptable compared to the quality bene- fits.

Limitations and Ethical Considerations

Any AI-based recruitment system must be designed with strong awareness of ethical risks and limitations.

Data Bias

Models learn patterns from historical data. If past hiring data or training corpora contain biases (e.g., underrepresentation of certain genders, regions, or colleges), the model may inadvertently reproduce or amplify these biases.

To partially mitigate this:

- Personally identifying attributes such as name, gender-specific titles, and phone numbers are removed before feeding text into the AI engine.
- Explicit bias checks can be applied to outputs, monitoring whether certain groups are systematically under-recommended.

Still, bias cannot be eliminated entirely, and CerebroGuide should be used as a decision-support tool, not a fully autonomous hiring authority.

Privacy and Security

Resumes contain sensitive personal information. CerebroGuide must therefore:

- Anonymize PII (names, contact details, addresses) during processing,
- Encrypt data at rest and in transit,
- Offer deletion mechanisms to comply with data protection norms.

User consent and clear privacy policies are essential for responsible deployment.

Explainability

Although transformer models are powerful, their internal workings are not easily inter- pretable. CerebroGuide addresses this by:

- Exposing intermediate artifacts such as skill overlaps,
- Decomposing the final score into semantic, skill, and experience contributions. This does not fully “open the black box” of neural networks but provides useful, human-readable signals.

Future Scope

The current version of CerebroGuide provides a strong foundation for AI-driven career navigation, but several extensions are envisioned:

- **AI Interview Simulation:** Integrating a generative AI module to conduct mock technical or behavioral interviews, tailored to the user’s resume and target job.
- **Predictive Career Trajectory:** Using historical career data to forecast plausible career paths over 3–5 years and suggest intermediate roles or skills to pursue.
- **LinkedIn and Job Portal Integration:** Importing profiles and application his- tories directly from platforms such as LinkedIn and major job boards.
- **Multilingual Support:** Extending resume analysis to multiple languages to serve global or regional markets.
- **Blockchain-Based Credential Verification:** Leveraging blockchain to store and verify certifications, degrees, and employment histories, reducing the risk of creden- tial fraud.

These directions would further strengthen CerebroGuide as a comprehensive platform for both individuals and organizations.

Conclusion

CerebroGuide demonstrates how modern AI techniques—particularly transformer-based semantic modeling and skill-aware analysis—can transform the way career guidance and recruitment systems operate. By moving beyond keyword-centric filtering and embracing a holistic view of candidate capabilities, the system offers more accurate matches, clearer feedback, and actionable paths for professional growth.

The experimental results show that CerebroGuide significantly outperforms tradi- tional ATS-like approaches in accuracy, precision, recall, and F1-score, albeit at a modest computational cost. More importantly, user feedback highlights the value of being told not just "yes" or "no", but *why* a match exists and *what* could be done to improve.

While important challenges related to bias, privacy, and explainability remain, this work illustrates a concrete, implementable step toward more intelligent, fair, and human- centered recruitment ecosystems.

Acknowledgment

The authors express their sincere gratitude to the Department of Computer Science, Oriental Institute of Science and Technology, Bhopal, for continuous academic guidance, infrastructure, and encouragement throughout the development of this work.

REFERENCES

1. [1] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirec- tional Transformers for Language Understanding," *arXiv preprint arXiv:1810.04805*, 2019.
2. N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in *Proc. EMNLP*, 2019.
3. M. Rosenberger et al., "CareerBERT: A Transformer-based Model for Job Classifica- tion," 2023.
4. C. C. Aggarwal, *Machine Learning for Text*. Springer International Publishing, 2018.
5. S. Gupta and A. Verma, "Semantic Resume Screening Using NLP," *IEEE Access*, vol. 9, pp. 11234–11245, 2021.
6. R. Patel and D. Mehta, "AI-Based Applicant Tracking Systems: A Comparative Study," *IEEE Transactions on Computational Social Systems*, 2021.
7. Vaswani et al., "Attention is All You Need," in *Advances in Neural Information Processing Systems*, 2017.