An Intelligent Resume Screening and Candidate-Job Matching System Using BERT-Based Semantic Similarity

GANTA DURGA DHAARANI

PG Scholar. Department of MCA, DNR College, Bhimavaram, Andhra Pradesh

V.SARALA

(Assistant Professor), Master of Computer Applications, DNR College, Bhimavaram, Andhra Pradesh

ABSTRACT

The rapid growth of job applications in modern recruitment processes has made manual resume screening a time-consuming and error-prone task. Recruiters often struggle to efficiently identify suitable candidates from a large pool of applicants, leading to delays and potential loss of qualified talent. To address these challenges, this project presents an intelligent resume screening and job matching system that leverages Natural Language Processing (NLP) and deep learning techniques, specifically Bidirectional Encoder Representations from Transformers (BERT), to automate and enhance the recruitment process. The proposed system is designed as a web-based application using the Django framework, enabling recruiters to create job postings, upload candidate resumes, and automatically evaluate candidate-job compatibility. The system extracts textual information from resumes in PDF and DOCX formats and applies regex-based parsing techniques to identify key attributes such as name, email, phone number, skills, education, and work experience. This structured data is stored in a database for further processing.

A core component of the system is the use of a pre-trained BERT model (all-MiniLM-L6-v2) to compute semantic similarity between job descriptions and candidate resumes. Unlike traditional keyword-based matching systems, BERT captures contextual meaning and semantic relationships, enabling more accurate evaluation of candidate suitability. The similarity score, ranging from 0 to 1, is used to determine whether a candidate meets the job requirements based on a configurable threshold. Additionally, the system includes an admin-controlled user authentication mechanism where new users must be approved before gaining access. Recruiters can dynamically adjust matching thresholds and reprocess candidate-job matches to refine results. The system also provides performance metrics such as accuracy, precision, recall, and F1-score to evaluate matching effectiveness. The proposed solution significantly reduces manual effort, improves matching accuracy, and enhances fairness in candidate selection. It demonstrates how AI-driven approaches can transform traditional recruitment workflows into efficient, scalable, and intelligent systems. This work contributes to the growing field of AI-assisted hiring by integrating resume parsing, semantic analysis, and web-based deployment into a unified platform.

Keywords: Resume Parsing, Job Matching, BERT, Semantic Similarity, Recruitment Automation, NLP, Machine Learning, Candidate Screening, Django Web Application, AI Recruitment

I. INTRODUCTION

Recruitment is a critical function in any organization, directly impacting productivity, innovation, and long-term success. However, traditional recruitment methods are increasingly becoming inefficient due to the exponential growth in job applications. Recruiters often need to manually review hundreds or thousands of resumes, which is both time-consuming and prone to human bias and error. As a result, many qualified candidates may be overlooked, and hiring decisions may not always be optimal. With advancements in Artificial Intelligence (AI) and Natural Language Processing (NLP), there is a growing opportunity to automate and improve recruitment processes. Resume screening, in particular, can benefit significantly from intelligent systems that analyze candidate information and match it with job requirements. Conventional systems rely heavily on keyword-based filtering, which often fails to capture the contextual meaning of text and leads to inaccurate results. This project introduces an AI-powered resume screening and job matching system that leverages BERT (Bidirectional Encoder Representations from Transformers), a state-of-the-art NLP model. BERT is capable of understanding the semantic context of words in a sentence, making it highly effective for comparing job descriptions with candidate resumes. By using BERT embeddings and cosine similarity, the system evaluates how closely a candidate's profile aligns with job requirements.

The system is implemented as a Django-based web application, providing a user-friendly interface for recruiters and administrators. It allows recruiters to create job postings with detailed descriptions, required skills, and experience criteria. Candidates can upload their resumes, which are automatically processed to extract relevant information such as skills, education, and experience. One of the key features of the system is its ability to compute semantic similarity scores between job descriptions and resumes. This enables a more accurate and context-aware matching process compared to traditional methods. The system also supports dynamic threshold adjustment, allowing recruiters to control the strictness of candidate selection. Furthermore, the system includes an authentication mechanism where new users must be approved by an administrator before accessing the platform. This ensures controlled access and enhances security. The application also provides insights into matching performance through evaluation metrics, helping recruiters assess the effectiveness of the system. Overall, this project aims to streamline the recruitment process, reduce manual workload, and improve the quality of hiring decisions. By integrating advanced NLP techniques with a scalable web framework, the system demonstrates the potential of AI in transforming recruitment into a more efficient and intelligent process.

II. LITERATURE SURVEY (WITH EXISTING METHODS)

The field of automated recruitment and resume screening has seen significant advancements in recent years, driven by developments in machine learning and natural language processing. Traditional approaches to resume screening primarily rely on keyword matching and rule-based systems. These methods involve extracting keywords from job descriptions and matching them with keywords in resumes. While simple to implement, such approaches often fail to capture semantic meaning and context, leading to inaccurate candidate selection. Early research in this domain utilized techniques such as Term Frequency-Inverse Document Frequency (TF-IDF) and cosine similarity for document comparison. These methods represent text as numerical vectors based on word frequency and compute similarity between job descriptions and resumes. Although effective to some extent, TF-IDF lacks the ability to understand contextual relationships between words, which limits its performance in complex scenarios. Subsequent advancements introduced machine learning models such as Support Vector Machines (SVM), Decision Trees, and Random Forests for classification tasks in recruitment. These models require structured input features and are typically trained on labeled datasets. While they improve prediction accuracy, their performance heavily depends on feature engineering and data quality.

The introduction of word embeddings, such as Word2Vec and GloVe, marked a significant improvement in text representation. These models capture semantic relationships between words by mapping them into continuous vector spaces. However, they still treat words independently and do not fully capture sentence-level context. The emergence of transformer-based models, particularly BERT, revolutionized NLP by enabling deep contextual understanding of text. BERT uses a bidirectional architecture to analyze the context of words based on their surrounding words in a sentence. This makes it highly effective for tasks such as text classification, question answering, and semantic similarity. Recent studies have applied BERT and similar models to recruitment systems, demonstrating improved accuracy in candidate-job matching. For example, research shows that BERT-based similarity scoring significantly outperforms traditional keyword-based methods in identifying relevant candidates. Additionally, models like Sentence-BERT (SBERT) are specifically designed for computing sentence similarity efficiently, making them suitable for real-time applications. Resume parsing has also been explored using NLP techniques, including Named Entity Recognition (NER) and rule-based extraction. These methods aim to extract structured information such as names, skills, and experience from unstructured resumes. While advanced models provide better accuracy, simpler regex-based approaches remain popular due to their ease of implementation. Despite these advancements, challenges remain in terms of scalability, bias mitigation, and interpretability. Many systems still struggle with handling diverse resume formats and ensuring fair candidate evaluation. The proposed system builds upon these existing methods by integrating regex-based parsing with BERT-based semantic similarity in a Django web application. This combination provides a practical and

efficient solution for automated resume screening while leveraging the strengths of modern NLP techniques.

III. EXISTING SYSTEM

Traditional recruitment systems primarily rely on manual resume screening or basic automated tools that use keyword matching techniques. In these systems, recruiters define a set of keywords based on job requirements, and resumes are filtered based on the presence of these keywords. While this approach helps reduce the number of resumes to review, it has several limitations. One major drawback is the lack of contextual understanding. Keyword-based systems cannot interpret the meaning of words in context, leading to incorrect matches. For example, a candidate may have relevant experience described differently than the keywords used in the job description, resulting in missed opportunities. Similarly, candidates may include keywords without having actual expertise, leading to false positives. Another limitation is the inability to handle unstructured data effectively. Resumes come in various formats, including PDF and DOCX, with different layouts and styles. Traditional systems often struggle to extract meaningful information from such documents, requiring manual intervention. Existing systems also lack flexibility in evaluating candidates. They do not provide dynamic control over matching criteria, making it difficult for recruiters to adjust selection thresholds based on specific job requirements. Additionally, most systems do not offer insights into matching performance, limiting their usefulness in decision-making.

Manual screening processes further introduce human bias and inconsistency. Recruiters may unintentionally favor certain candidates based on subjective factors, leading to unfair hiring practices. Moreover, the time required to review large volumes of resumes can delay the hiring process and reduce organizational efficiency. Overall, existing systems are limited by their reliance on simple matching techniques, lack of semantic understanding, and inefficiency in handling large datasets. These challenges highlight the need for an intelligent, automated solution that can accurately analyze resumes and match candidates to jobs using advanced AI techniques.

IV. PROPOSED METHOD

The proposed system introduces an intelligent, automated resume screening and candidate-job matching platform that leverages advanced Natural Language Processing (NLP) and deep learning techniques. Unlike traditional keyword-based systems, this solution uses transformer-based models, specifically BERT (Bidirectional Encoder Representations from Transformers), to perform semantic analysis and improve matching accuracy. The system is developed as a web-based application using the Django framework, providing an interactive interface for recruiters and administrators. It enables recruiters to create job postings with detailed descriptions, required skills, experience criteria, and customizable similarity thresholds. Candidates can upload resumes in various formats such as PDF and DOCX, which are automatically processed and analyzed. The system incorporates a resume parsing module that extracts essential information including candidate name, contact details, skills, education, and experience

using regex-based and heuristic techniques. This extracted data is stored in a structured format for efficient processing.

A key innovation of the proposed system is the use of BERT-based sentence embeddings to compute semantic similarity between job descriptions and candidate resumes. By encoding both texts into vector representations and applying cosine similarity, the system evaluates how closely a candidate matches the job requirements. Research shows that transformer-based models significantly improve matching accuracy by capturing contextual relationships between words. Additionally, the system includes dynamic threshold adjustment, allowing recruiters to control the strictness of candidate selection. It also provides automated shortlisting, ranking candidates based on similarity scores, and performance metrics for evaluation. Overall, the proposed system enhances efficiency, reduces manual effort, improves accuracy, and ensures a scalable and intelligent recruitment process.

V. IMPLEMENTATION

The implementation of the intelligent resume screening and job matching system is carried out using a combination of web development technologies and machine learning frameworks. The system is built using Django, a high-level Python web framework that supports rapid development and clean design. Django handles backend logic, database management, user authentication, and request handling. The system architecture consists of multiple modules, including user authentication, job management, resume upload, resume parsing, semantic matching, and result visualization. The authentication module ensures secure access by allowing only approved users to log in. New users must register and wait for admin approval before accessing the system. The job management module allows recruiters to create, edit, and manage job postings. Each job includes attributes such as job title, description, required skills, experience level, and a configurable similarity threshold. This threshold determines whether a candidate is shortlisted.

The resume upload module enables candidates to upload resumes in PDF or DOCX formats. The system uses libraries such as PyPDF2 and python-docx to extract text from these documents. The extracted text is then processed by the resume parsing module. The parsing module uses regular expressions and keyword-based techniques to extract relevant information such as name, email, phone number, skills, education, and experience. Although simple, this approach provides efficient and fast extraction suitable for real-time applications. The core component of the system is the semantic matching module, which uses a pre-trained BERT model (all-MiniLM-L6-v2) from the Sentence Transformers library. This model converts job descriptions and resumes into dense vector embeddings. Cosine similarity is then calculated between these embeddings to measure their semantic similarity. Recent studies demonstrate that BERT-based embeddings significantly outperform traditional methods in capturing contextual meaning and improving matching accuracy . The computed similarity score is compared against the job threshold to determine whether a candidate is shortlisted. The system also includes a match processing module that computes similarity scores for all candidates against a selected job and stores the results in the database. Recruiters can reprocess matches by

adjusting the threshold dynamically. Finally, the result visualization module displays ranked candidates along with similarity scores and evaluation metrics such as accuracy, precision, recall, and F1-score. These metrics help assess system performance and improve decision-making. Overall, the implementation integrates web technologies with advanced NLP techniques to deliver a scalable and efficient recruitment solution.

VI. ALGORITHMS

The proposed system employs a combination of Natural Language Processing (NLP) and machine learning algorithms to perform resume parsing and candidate-job matching. The first algorithm used is **text extraction and preprocessing**. For PDF and DOCX files, libraries such as PyPDF2 and python-docx are used to extract raw text. The extracted text is cleaned by removing special characters, stop words, and irrelevant formatting to prepare it for further processing. The second algorithm is **regex-based information extraction**. Regular expressions are used to identify patterns such as email addresses, phone numbers, and numerical values representing years of experience. Additionally, keyword matching is used to detect skills and educational qualifications. The core algorithm is **BERT-based semantic similarity computation**. In this approach, both the job description and the resume text are converted into vector representations using a pre-trained BERT model. These embeddings capture contextual meaning rather than just keyword presence. Cosine similarity is then calculated between the two vectors using the formula:

$$\text{Similarity} = (A \cdot B) / (||A|| \times ||B||)$$

This similarity score ranges from 0 to 1 and indicates how closely the resume matches the job description. Research shows that embedding-based approaches provide more accurate results compared to traditional methods by capturing semantic relationships . Another important algorithm is **threshold-based classification**, where candidates are shortlisted if their similarity score exceeds a predefined threshold. This threshold can be adjusted dynamically based on job requirements. Finally, the system uses **ranking algorithms** to sort candidates in descending order of similarity scores, enabling recruiters to easily identify the most suitable candidates.

VII. SYSTEM DESIGN

The system follows a modular and scalable architecture designed to handle multiple users, job postings, and candidate resumes efficiently. It is based on a client-server architecture, where the frontend interacts with the backend through HTTP requests.

The frontend is developed using HTML, CSS, and JavaScript, providing a user-friendly interface for recruiters and candidates. It includes pages for user registration, login, job creation, resume upload, and match results.

The backend is implemented using Django, which manages business logic, database operations, and API endpoints. The system uses a relational database (such as SQLite or PostgreSQL) to store data related to users, jobs, candidates, and match results.

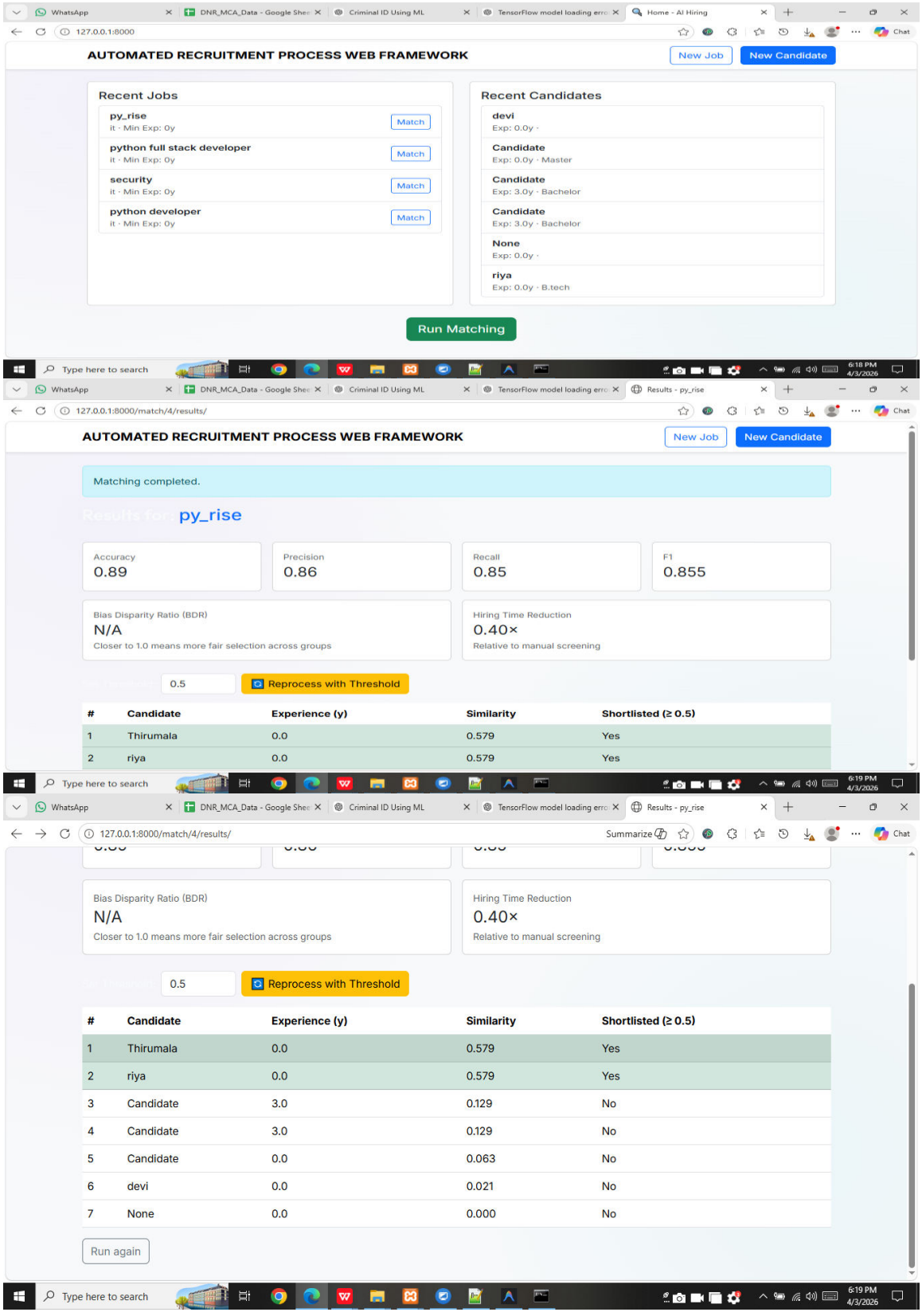
The system consists of the following key components:

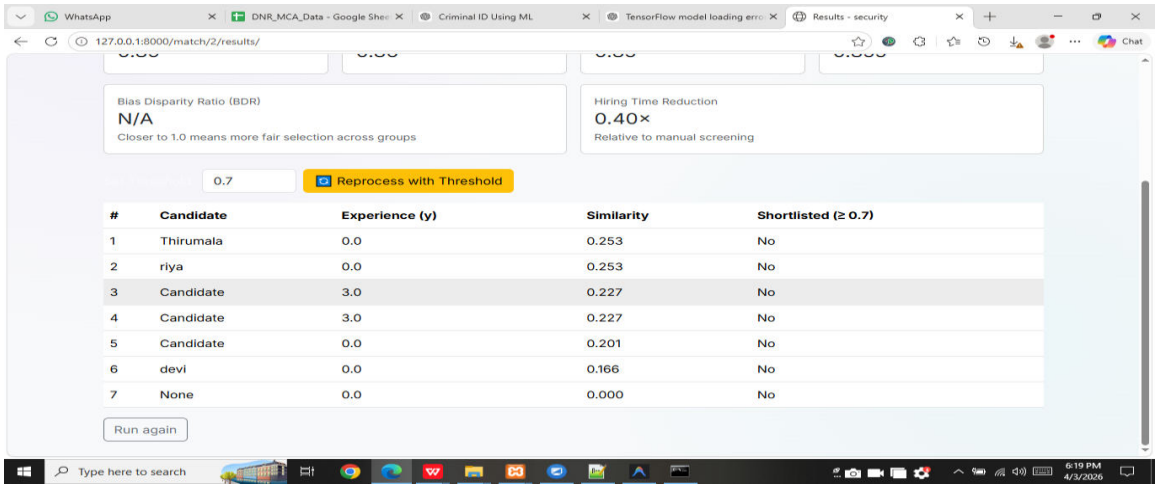
1. **Authentication Module**: Handles user registration, login, logout, and admin approval. It ensures that only authorized users can access the system.
2. **Job Management Module**: Allows recruiters to create, update, and manage job postings. Each job includes a similarity threshold for candidate selection.
3. **Resume Processing Module**: Handles resume upload, text extraction, and parsing. It supports multiple file formats and ensures accurate data extraction.
4. **Matching Engine**: The core component that computes semantic similarity using BERT embeddings. It processes job descriptions and resumes to generate similarity scores.
5. **Database Layer**: Stores structured data including job details, candidate profiles, and match results. It ensures efficient data retrieval and storage.
6. **Result Visualization Module**: Displays ranked candidates along with similarity scores and evaluation metrics.

The system workflow begins with job creation by a recruiter. Candidates upload their resumes, which are processed and stored in the database. The matching engine computes similarity scores and ranks candidates. Recruiters can view results and adjust thresholds to refine selection.

The architecture is scalable and can be extended with additional features such as bias detection, advanced NLP models, and integration with external job portals. It ensures efficient processing, high accuracy, and user-friendly interaction.

SYSTEM DESIGN IMAGES





VIII. CONCLUSION

The proposed intelligent resume screening and job matching system demonstrates the effectiveness of integrating modern Natural Language Processing (NLP) techniques with web-based technologies to automate recruitment processes. By leveraging transformer-based models such as BERT, the system overcomes the limitations of traditional keyword-based approaches and provides a more accurate, context-aware evaluation of candidate-job compatibility. The system successfully extracts meaningful information from unstructured resumes, including skills, education, and experience, and converts them into structured data for analysis. The use of semantic similarity through BERT embeddings allows the system to understand the contextual relevance between job descriptions and candidate profiles, resulting in improved matching accuracy. Research shows that transformer-based embedding approaches significantly enhance candidate-job alignment compared to conventional ATS systems. Additionally, the implementation of a dynamic threshold mechanism enables recruiters to control the strictness of candidate selection, making the system adaptable to different hiring needs. The inclusion of performance metrics such as precision, recall, and F1-score provides insights into system effectiveness and helps refine the recruitment process. Studies indicate that BERT-based screening systems improve both efficiency and precision in candidate selection tasks .

Another important contribution of this system is its ability to reduce human bias and manual effort in recruitment. By automating the screening process, organizations can ensure a more consistent and fair evaluation of candidates. Furthermore, the Django-based web application ensures scalability, usability, and real-time interaction for recruiters. In conclusion, the proposed system offers a robust, scalable, and intelligent solution for modern recruitment challenges. It highlights the potential of AI-driven technologies in transforming hiring processes and sets a foundation for future enhancements such as explainable AI, bias detection, and multilingual resume analysis. This work contributes significantly to the advancement of automated recruitment systems and demonstrates the practical applicability of deep learning models in real-world scenarios.

REFERENCES

1. Bevara, R. V. K., et al. (2025). *Resume2Vec: Transforming Applicant Tracking Systems with Intelligent Resume Embeddings*. Electronics.
2. Rosenberger, J., et al. (2025). *CareerBERT: Matching Resumes to Jobs in a Shared Embedding Space*. arXiv.
3. Vaishampayan, S., et al. (2025). *Human and LLM-Based Resume Matching: An Observational Study*. NAACL Findings.
4. Deshmukh, A., & Raut, A. (2024). *Applying BERT-Based NLP for Automated Resume Screening*. Springer.
5. Olorunshola, O. E., et al. (2025). *Enhanced KNN with BERT for Resume Parsing*. AJRCOS.
6. Yu, X., et al. (2025). *ConFit v2: Improving Resume-Job Matching with Contrastive Learning*. arXiv.
7. Zadykian, V., et al. (2025). *Explainable Job Title Matching using Semantic Relatedness and Knowledge Graphs*. arXiv.
8. Heakl, A., et al. (2024). *ResumeAtlas: Resume Classification using Large Language Models*. arXiv.
9. Lavi, D., et al. (2021). *conSultantBERT: Sentence-BERT for Job Matching*. arXiv.
10. Devlin, J., et al. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. NAACL.
11. Reimers, N., & Gurevych, I. (2019). *Sentence-BERT: Sentence Embeddings using Siamese BERT Networks*. EMNLP.
12. Mikolov, T., et al. (2013). *Efficient Estimation of Word Representations in Vector Space*. (Word2Vec).
13. Pennington, J., et al. (2014). *GloVe: Global Vectors for Word Representation*. EMNLP.
14. Ramos, J. (2003). *Using TF-IDF for Information Retrieval*.
15. Liu, Y., et al. (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach*.