Journal of Computer Science

Original Research Paper

Smart Talent Sourcing Through Advanced Skill Profiling Technique

Amey Krishnanath Shet Tilve, Gaurang Sitaram Patkar, Vadiraj Gururaj Inamdar, Merwyn D'Souza and Janhavi Naik

Department of Computer Engineering, Don Bosco College of Engineering, Goa, India

Article history
Received: 12-07-2024
Revised: 16-09-2024
Accepted: 26-10-2024

Corresponding Author:
Amey Krishnanath Shet Tilve
Department of Computer Engineering, Don Bosco College of Engineering, Goa, India
Email: amey.tilve@dbcegoa.ac.in

Abstract: The hiring process often struggles with aligning job seekers' skills to employers' requirements, leading to inefficiencies and mismatches. To address this challenge, a dual-functionality system is proposed that leverages Natural Language Processing (NLP) techniques, including BERT for embedding textual information and cosine similarity to rank resumes according to their alignment with job descriptions and to recommend suitable candidates to employers and vice versa. The primary goal is to enhance the accuracy and efficiency of job-to-job-seeker matching by integrating these advanced methods as features within the model, alongside other relevant data points. The developed system effectively addresses challenges such as noisy data, heterogeneous sources and multilingualism, demonstrating its potential in improving the hiring process with increased accuracy and precision of the system. These findings suggest that the proposed method not only streamlines talent acquisition but also offers broader applications in talent management systems, ensuring more precise and efficient matching of candidates to job opportunities. The implications of this research extend to enhancing the recruitment process's overall effectiveness and providing a robust foundation for future advancements in AI-driven talent management.

Keywords: Cosine Similarity, BERT Embeddings, Resume Parsing, Recommendation System

Introduction

In the contemporary digital landscape, online job portals have become integral platforms for job seekers and recruiters alike. Despite their widespread use, the reliance on basic keyword-based searches within these portals has proven inefficient in accurately assessing candidates' skills and experiences (Sayyadian *et al*., 2007).

This shortcoming frequently results in a substantial disparity between qualifications job seekers hold and the precise demands of open positions. The conventional approaches used in online job portals often fall short in addressing the shortcomings between job seekers' qualifications and employers' requirements. This research addresses the critical issue of improving the accuracy and efficacy of the hiring process by developing an innovative method for online job hunting that incorporates advanced AI-driven skill profiling techniques (van Esch and Black, 2019). The proposed system aims to revolutionize the market by introducing a more comprehensive assessment of candidates' skills and enabling skill tagging at the source.

The conventional approaches used in online job portals often fall short in bridging the gap between job seekers' qualifications and employers' requirements. This research addresses the critical issue of improving the accuracy and efficiency of the hiring process by developing an innovative method for online job hunting that incorporates advanced AI-driven skill profiling techniques (van Esch and Black, 2019). The proposed system aims to revolutionize the job market by introducing a more comprehensive assessment of candidates' skills and enabling skill tagging at the source.

This research focuses on developing a dual-functionality system to rank resumes for the respective alignment with job descriptions. Implement a reverse matching process that suggests suitable candidates to employers via the enhanced skill profiles. Evaluate the proposed system's effectiveness in real-world recruitment scenarios, focusing on improving the alignment between job seekers' skills and employers' specific needs. A novel recruitment strategy, which overcomes the restrictiveness of standard keyword-based methods, is present. By

incorporating BERT embeddings for textual information and optimizing the resume matching process, the study offers a novel approach to improve results of candidate-job matching.

This research contributes to the existing literature by addressing the limitations of traditional keyword-based recruitment systems. By incorporating BERT embeddings for textual information and optimizing the resume matching process, the study offers a novel approach to enhancing the accuracy and precision of candidate-job matching. Furthermore, by facilitating skill tagging at the source, this research aims to significantly improve the visibility of recent graduates to potential employers, thereby creating a seamless and efficient recruitment experience for both job seekers and recruiters.

Literature Review

There has been a growing interest in creating job recommendation systems. Collaborative filtering is a widely used method that provides recommendations based on similarity within student profiles and requirements of job. However, this approach has limitations, such as requiring a large amount of user data and potentially being ineffective for users with unique preferences. Another approach, content-based filtering, involves analyzing student profiles and job descriptions to offer suggestions based on their similarities.

Agarwal and Senthilkumar (2022) proposed a novel approach to job recommendation by integrating collaborative filtering and content-based filtering techniques. Their system aims to address the challenge of finding optimal job positions amidst a vast array of postings. This research highlights the significance of combining various filtering methods.

Van Huynh *et al*. (2020) addresses the challenge of job prediction by various N-N models with appropriate positions. Their research explores the potency of TextCNN, Bi GRU LSTM CNN and Bi GRU CNN models in conjunction by applying pre-trained word-embeddings derived from an IT job corpus.

Luo and Zhong (2024) investigated graduate employment prediction utilizing the CNN model. Their study 8 is based on employment information from a Guangdong university over the previous 5 years. The authors compare their CNN-based method with traditional approaches, including K-N-N, Naive-Bayes and S.V.M.

Rahman *et al*. (2023) introduced the Job Title Prediction and Recommendation System (JoTPaRS), a sophisticated tool designed to improvise job-matching for IT professionals. Personalized job title recommendations and salary range estimates are provided by this system based on user knowledge, job descriptions and experience through the use of sophisticated machine learning techniques. The system also incorporates experience-based salary ranges into its recommendations, addressing limitations of manual job search systems and aligning with the principles of Industry 5.0 and Sustainable Technology.

Chopra and Lal Saini (2023) explored application of various NN models to gauge employability skills of IT graduates. Their study compares multiple architectures like text CNN, Bi GRU LSTM CNN and Bi-GRU-CNN, using pre-trained word-embedding's. The authors also proposed an ensemble model that combines these neural networks, which 1 achieved the highest F1-score of 72.71%.

Materials

The research utilized a combination of datasets, tools, and algorithms to develop and evaluate the proposed dual-functionality system for talent sourcing and skill profiling. This section provides detailed information on the materials used in the study.

Data Sources

Job dataset:

- Source: A synthetic dataset of job postings sourced from Kaggle (Dataset)
- Content: Included job listings across multiple industries and roles
- Purpose: Served as the primary input for the system to evaluate job descriptions and match them with candidate profiles

Candidate dataset:

- Source: Derived from a ground-level survey involving 1,000 candidates
- Content: Included demographic information, skills, educational background, job preferences, and prior work experience
- Purpose: Enabled the development of candidate profiles and served as input for the recommendation algorithms

Tools and Technologies

Natural language processing framework:

- Tool: BERT (Bidirectional Encoder Representations from Transformers)
- Purpose: Used for extracting semantic embeddings from job descriptions and candidate resumes, enabling accurate matching based on textual similarity

Similarity metrics:

- Method: Cosine similarity
- Purpose: Employed to calculate the alignment between job descriptions and candidate profiles based on their vectorized representations

Resume parsing tool:

- Tool: Python-based resume parser
- Purpose: Automated the extraction of structured data (e.g., skills, experience, education) from unstructured candidate resumes

1. System Infrastructure

    - Hardware: The study was conducted on a MacBook Air M2
    - Processor: M2 Chip
    - RAM: 16 GB
    - Storage: 512 GB SSD
    - Software: Python 3.12, with key libraries: Transformers, NumPy, and Scikit-learn

2. Ethical Considerations

    - Data Privacy: Candidate data was anonymized to ensure privacy
    - Bias Mitigation: Efforts were made to reduce algorithmic biases through balanced data representation and unbiased scoring mechanisms

Methods

As a part of this research, there are distinct user classes, each personalized to specific roles and characteristics which emerge as:

a. Candidate

- Characteristic: Applicant who is looking for employment opportunities
- Action: Can create accounts by uploading their resumes and providing detailed information. The system, featuring resume auto-parsing, ensures a smooth onboarding process. Candidates can view and apply for relevant job listings, internships and apprenticeships. The system intelligently matches them with preferred jobs based on relevant factors

b. Recruiter

- Characteristic: Professional who leverages data to post job openings and assess candidate qualifications
- Action: Creates account, providing personal and company details. They can post 5 new job listings and the system automatically identifies and displays qualified candidates. Recruiters may also participate in job fairs hosted on the platform

Figure (1) represents a flowchart detailing a job and candidate recommendation system, focusing on the processes involved in job posting, resume submission and the subsequent recommendation generation. The system utilizes BERT embeddings for attributes extraction and cosine similarity for ranking candidates and jobs.

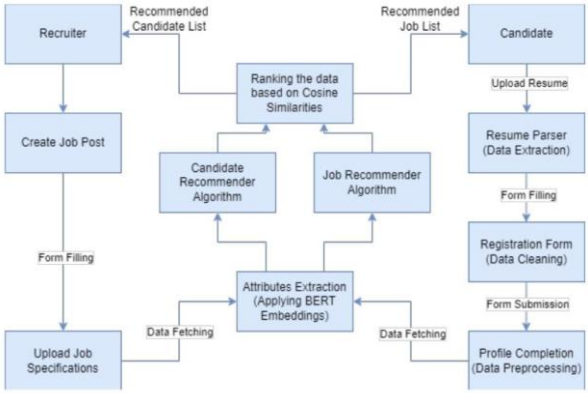

Fig. 1: System architecture

Key Components and Processes

a. Recruiter

- Create Job Post: The process starts when a recruiter creates a job post, which involves filling out necessary forms
- Upload Job Specifications: The recruiter then uploads the job specifications, which are essential for matching candidates to the job

b. Candidate

- Upload resume: Candidates begin by uploading their resumes
- Resume parser (data extraction): The uploaded resumes are processed by a resume parser, which extracts relevant data from the documents
- Registration form (data cleaning): After the data is extracted, candidates fill out a registration form where the data undergoes cleaning
- Profile completion (data preprocessing): The cleaned data is further preprocessed to complete the candidate's profile, making it ready for the matching process

c. Attributes Extraction

- Applying BERT embeddings: Both the job specifications and candidate profiles undergo attribute extraction using BERT embeddings. This step converts the textual information into numerical vectors that can be processed by the recommendation algorithms

d. Recommendation Algorithms

- Candidate Recommender Algorithm: This algorithm uses the extracted attributes to fetch data relevant to candidates and identifies those that match the job

specifications uploaded by recruiters
- Job Recommender Algorithm: Similarly, this algorithm fetches data relevant to jobs and identifies those that align with the candidates' profiles

e. Ranking

- Ranking Based on Cosine Similarities: The system ranks the data by comparing the cosine similarities between the BERT embeddings of job specifications and candidate profiles. This ranking determines the degree of match between candidates and jobs

f. Output

- Recommended Candidate List: For recruiters, the system outputs a list of recommended candidates based on the ranking
- Recommended Job List: For candidates, the system generates a list of job recommendations that best match their profiles

Figure (2) illustrates the flow of a job and candidate recommendation system, depicting interactions between candidates, recruiters, a web server, a matching algorithm, external APIs and a data store.

Key Components

- Candidate and Recruiter: Users of the system who request recommendations. Candidates request job recommendations and recruiters request candidate recommendations
- Web Server: The central component that handles incoming requests from candidates and recruiters, processes these requests and interfaces with the matching algorithm to generate recommendations

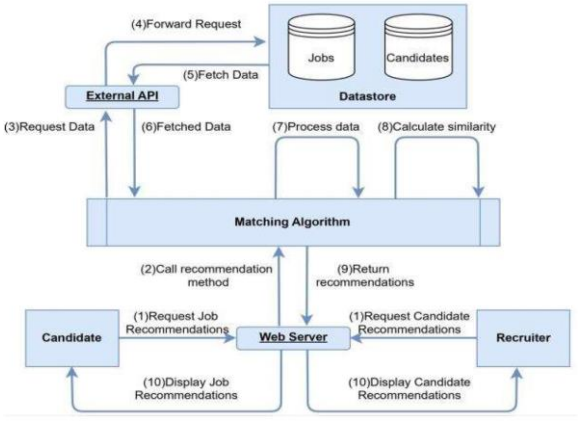

Fig. 2: Recommendation system

- Matching algorithm: The core component responsible for processing data and calculating the similarity between jobs and candidates. It retrieves data from the data store, processes it and returns recommendations
- Data store: Stores job and candidate data. It provides the data required by the matching algorithm to generate accurate recommendations
- External API: A component that fetches additional data required by the matching algorithm. This data is requested and processed alongside the data from the data store

System Flow

- Candidate/recruiter requests: The system begins when a candidate requests job recommendations or a recruiter requests candidate recommendations. These requests are sent to the web server
- Web server interaction: The web server receives the request and calls the appropriate recommendation method in the matching algorithm
- Data request to external API: If the recommendation process requires external data, the matching algorithm requests this data via an external API
- External API Response: The external API forwards the data request and retrieves the necessary information
- Data fetch from data store: The matching algorithm fetches data from the data store, including job and candidate data
- Data processing: The matching algorithm processes the fetched data, combining it with the external data, if any, to create a comprehensive data set
- Similarity Calculation: The matching algorithm then calculates the similarity between jobs and candidates using the processed data
- Returning recommendations: The calculated recommendations are returned to the web server
- Displaying recommendations: Finally, the web server displays the job recommendations to the candidate and the candidate recommendations to the recruiter

This flow ensures that candidates receive job recommendations tailored to their profiles and recruiters are provided with candidates matching their job postings. The system's modular architecture allows for scalable and efficient matching based on both internal and external data sources.

The preprocessing steps, as stated by Manning *et al*. (2008), are important for transforming raw candidate and job data into a structured format suitable for similarity metrics calculation. These steps ensure that the recommendation system can effectively match candidates to jobs based on skills, preferences,

education, languages and experience. Data formatting, as described by Mikolov *et al*. (2013), involves converting textual attributes into vectorized representations suitable for similarity calculations. The following pre-processing steps were applied:

a. Data cleaning and structuring: Preprocessing transforms raw candidate and job data into a structured format, which is essential for consistency and reliability in the recommendation process. For example, it ensures that job titles, skills and experience are uniformly formatted, removing ambiguities and discrepancies that could otherwise lead to inaccurate matches. Manning *et al*. (2008) emphasize that without proper preprocessing, raw data may contain noise, such as irrelevant information, inconsistencies, or duplications, which could distort the matching results. By cleaning the data, the system can focus on the most relevant features, improving the quality of the recommendations
b. Vectorization and feature extraction: Following the cleaning process, data formatting as discussed by Mikolov *et al*. (2013), involves converting textual attributes into vectorized representations. This step is crucial for enabling the recommendation system to calculate similarity metrics between candidates and jobs. The vectorized representations allow the system to quantify the relevance of various attributes (such as skills, education and experience) by representing them in a multidimensional space. This quantitative approach facilitates the use of algorithms like cosine similarity to measure the closeness between candidate profiles and job specifications
c. Handling complex and diverse data: Candidates and jobs may have complex and varied attributes, such as multilingual skills or diverse educational backgrounds. Preprocessing helps in normalizing these variations, making it easier for the system to handle and compare such diverse data. For instance, the system might use BERT embeddings, which are generated during preprocessing, to capture the contextual meanings of words and phrases in resumes and job descriptions, leading to more nuanced and accurate matching

The preprocessing steps are fundamental in ensuring that the raw data is transformed into a usable format that the recommendation system can work with. By structuring, cleaning and vectorizing the data, preprocessing enhances the system's ability to match candidates to jobs effectively, ensuring that the recommendations are both relevant and reliable. This ultimately leads to a better experience for both candidates and recruiters.

Key Terminologies

a. Embeddings: Mandelbaum and Shalev (2016) define embeddings as unsupervised word representations Pennington *et al*. (2014) made up of five vectors whose relative similarities correspond to semantic similarity. In computational linguistics, they are referred to as distributional semantic models or distributed representations
b. BERT model: Bidirectional Encoder Representations from Transformers (BERT) as per (Clark *et al*., 2019; Devlin *et al*., 2019) is a cutting-edge NLP model invented by Google that pre-trained on a huge corpus of text in a bidirectional way. Recent efforts have focused on integrating knowledge into BERT (Devlin *et al*., 2019). Zhang *et al*. (2019)) introduce ERNIE, a neural language model that integrates knowledge from entities into textual information from underlying layers. Wang *et al*. (2021) offer KEPLER, a model based on RoBERTa (Yinhan Liu *et al*., 2019) 1 that maps texts and entities onto the same semantic space using the same language model. It optimizes both knowledge embedding and masked language modeling
c. BERT Embeddings: BERT embeddings are dense vector representations generated by the BERT model. They capture the semantic meaning of words within their specific context. The process involves tokenizing input text into word pieces, feeding these tokens into BERT model and obtaining the contextualized embeddings from the final hidden states of the transformer layers. tokenizes the input text into a sequence of tokens $T = \{t_1, t_2,\ldots, tn\}$. For each token *ti*, BERT produces an embedding vector *ei*. These embeddings are processed by multiple layers of transformers to generate the final contextualized embeddings *hi*:

$$H = BERT\ (T) \tag{1}$$

where:

- $H = \{h_1, h_2,\ldots, hn\}$ represents the set of output embeddings from the last transformer layer
- $T = \{t_1, t_2,\ldots, tn\}$ represents the sequence of tokens

The mean of these embeddings is often used as the sentence embedding:

$$E_{mean} = \frac{1}{n}\sum_{i=1}^{n} h_i \tag{2}$$

where:

- E_{mean} is the mean of the embeddings from H
- n represents the total number of embeddings generated
- h_i represents the output embedding from the last transformer layer

d. Cosine similarity: Agarwal and Senthilkumar (2022) define cosine similarity as the measure of similarity between two vectors. It works by calculating the cosine angle between two vectorized texts Malgaonkar *et al*. (2016); Pazur Anicic *et al*. (2017). Genc *et al*. (2021) stated that angle between vectors determines whether they are pointing in the same or opposite directions. Vectors pointing in the same direction indicate similarities between texts. The closer vectors are to the axis, the more similar they are. The distance from the axis represents a decrease in similarity. To compare documents, each is plotted in space and cosine similarity is used to determine orientation. Jayakodi *et al*. (2016); Karajeh *et al*. (2016) conducted recent research to assess document orientation, while other techniques could be used to estimate document quantity

Given vectors A and B, the cosine similarity sim (*A*, *B*) is defined as:

$$cosineSimilarity\,(A,B) = \frac{A.B}{\|A\|*\|B\|} \quad (3)$$

where:

$$A \cdot B = \sum_{i=1}^{n} A_i \cdot B_i \quad (4)$$

where:

- *A* represents the job vector
- *B* represents the candidate vector
- *n* represents the length of the vector generated from the BERT

Magnitude of each vector: calculate magnitude of each vector:

$$\|A\| = \sqrt{\sum_{i=1}^{n} A_i^2}, \|B\| = \sqrt{\sum_{i=1}^{n} B_i^2} \quad (5)$$

e. *Performance Metrics*

- Accuracy: The proportion of correct predictions made by the model out of all the predictions. It shows how often the model is right:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$

- Precision: The proportion of true positive predictions out of all the positive predictions the model made. It indicates how many of the positive predictions are actually correct:

$$Precision = \frac{TP}{TP + FP} \quad (7)$$

- Recall: The proportion of true positive predictions out of all the actual positive cases. It measures how well the model identifies positive cases:

$$Recall = \frac{TP}{TP + FN} \quad (8)$$

- F1-Score: The harmonic mean of precision and recall. It provides a balanced measure of the model's performance, especially when there is an uneven class distribution:

$$F1\,Score = 2 * \frac{precision * Recall}{precision * Recall} \quad (9)$$

Job Recommendation to a Candidate

This approach is constructed for recommending eligible jobs to candidates on their abilities, preferences, education, language competence and previous experience. The following algorithm is implemented.

Algorithm

1. Create an empty list recommended_candidates to store the candidates that will be recommended
2. For each job in preprocessed_jobs_data: Calculate the similarity between the candidate's skills and the job's required skills
   Calculate the similarity between the candidate's, job preferences and the job's preferences
   Calculate the similarity between the candidate's education and the job's required highest education level
   Calculate the similarity between the languages the candidate knows and the languages required for the job
   Calculate the similarity between the candidate's previous job roles and the job's required experience
3. Compute the overall similarity score for the job using a weighted average of the individual similarity scores:
   40% weight for skill similarity
   20% weight for education similarity
   10% weight for language similarity
   10% weight for experience similarity
   Formula: overall_similarity = (0.40 * skill_similarity) + (0.20 * preference_similarity) + (0.20 * education_similarity) + (0.10 * language_similarity) + (0.10 * experience_similarity)
4. If overall_similarity is greater than the threshold, add the candidate to recommended_candidates with its similarity scores
5. Sort the recommended_jobs list in descending order based on the overall_similarity scores
6. Filter the recommended_jobs list to include only those jobs where overall_similarity is greater than the threshold
7. Return the list of recommended jobs in a formatted manner

Candidate Recommendation to Recruiter

This algorithm is designed to recommend eligible candidates to recruiters based on their abilities, preferences, education, language competence and previous experience. The following algorithm is implemented.

Algorithm

1. Create an empty list recommended_candidates to store the candidates that will be recommended
2. For each candidate in preprocessed_candidates: Check if the candidate has skills. If not, skip to the next candidate. Calculate skill similarity, preference similarity, education similarity, language similarity, experience similarity using respective methods
3. Compute Overall Similarity Formula: overall_similarity = (0.40 * skill_similarity) + (0.20 * preference_similarity) + (0.20 * education_similarity) + (0.10 * language_similarity) + (0.10 * experience_similarity)
4. If overall_similarity is greater than the threshold, add the candidate to recommended_candidates with its similarity scores
5. Sort recommended_candidates in descending order by overall_similarity
6. Filter recommended_candidates to keep only those with overall_similarity greater than the threshold
7. Return the filtered and sorted list of candidates in a formatted manner

Dataset

For this research, data-set is split into two components:

- Job dataset: A comprehensive collection of synthetic job postings sourced from Kaggle, designed to facilitate research and analysis in job market trends, Natural-Language Processing (NLP) and Machine-Learning (ML). This dataset, created for educational and research purposes, offers a variety of job listings across multiple industries and jobs
- Candidate data: Derived from a ground-level survey involving responses from 1000 candidates across various domains. This dataset provides valuable insights into candidate demographics, experiences and preferences, making it a rich resource for research in human resources, job market analysis and machine learning applications

Results

The recommendation system developed in this study was tested using a dataset of 1000 candidates and 100 jobs across various domains. The system's core mechanism involved converting candidate and job objects into BERT embeddings, followed by the application of cosine similarity to rank the results in descending order based on the overall similarity score. The system achieved an overall accuracy of 91% and a precision of 100%, indicating its effectiveness in accurately matching candidates with suitable job opportunities. The performance was evaluated across various threshold levels k, where both accuracy and precision metrics were used to determine the system's effectiveness.

At lower k values, the system demonstrated moderate success, with an initial accuracy of 53% and precision of 57%. Ask increased, both accuracy and precision improved, stabilizing at k = 45 with an accuracy of 62% and precision of 66%. However, further increases in k led to fluctuations in precision, indicating potential difficulties in maintaining high precision as the number of recommended jobs increased. This variation in result with the batch of jobs considered is represented in Table (1) as follows.

Figure 3 demonstrates the trends in accuracy and precision across a number of recommendations (k).

Table 1: Performance metrics of recommendation system across various threshold levels

Threshold score	Accuracy	Precision	Batch of jobs considered
5	53	57	K = 8
10	53	57	k = 8
15	53	57	k = 8
20	53	57	k = 8
25	53	57	k = 8
30	53	57	k = 8
35	53	57	k = 8
40	53	57	k = 8
45	62	66	k = 7
50	65	62	k = 6
55	81	86	k = 4
60	91	100	k = 3

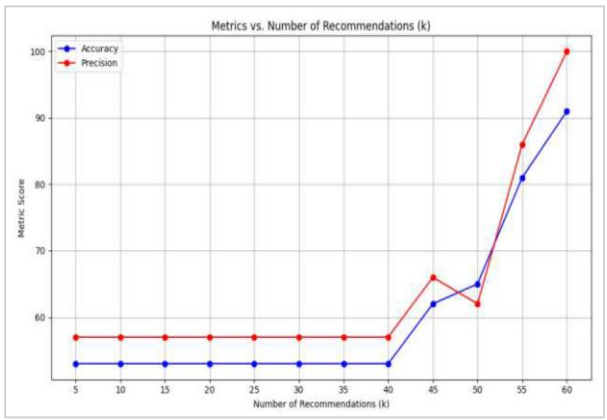

Fig. 3: Metrics vs. number of recommendations (k)

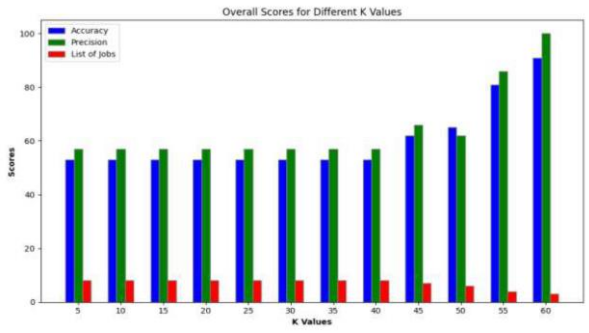

Fig. 4: Overall scores for different k values

Figure (4) depicts the performance of the system at different values of k, concentrating on criteria such as accuracy, precision and the number of jobs examined. The evaluation was carried out by utilizing a dataset of 1000 candidates and 100 jobs from various domains. This graph provides a full evaluation of system performance across a range of k values, making it valuable for extracting actionable insights and optimizing job recommendation systems. Considering the other side of the recommender, similar results were observed for recommending candidates for a given job.

Discussion

The results obtained from the recommendation system align with findings from recent studies in the literature. The hybrid approach of integrating collaborative filtering and content-based filtering, as proposed by Agarwal and Senthilkumar (2024), supports the effectiveness of combining multiple methods to enhance recommendation accuracy. Similarly, the use of deep neural network models for job recommendation, as explored by Van Huynh *et al*. (2020); Chopra and Lal Saini (2023), underscores the importance of advanced machine learning techniques in improving job matching systems. In particular, the high precision achieved by the proposed system is comparable to the results obtained by these studies, highlighting the robustness of BERT embeddings in capturing the semantic nuances of job descriptions and candidate profiles.

Proposed approach differs from traditional methods, such as those used by Luo and Zhong (2024); Rahman *et al.* (2023), which relied on models like CNN and ensemble techniques. While these methods have proven effective, use of BERT embeddings offers a more nuanced understanding of textual data, allowing for more accurate candidate-job matching. This is particularly evident in the high precision rates achieved by the system, even at varying threshold levels.

The fluctuations in precision at higher k values observed in study suggest that while expanding the scope of recommendations can improve accuracy, it also introduces challenges in maintaining precision. This finding aligns with the trade-offs discussed in the literature, where models like the ensemble approach proposed by Chopra and Lal Saini (2023)) achieved high F1- scores by balancing different model architectures.

To further evaluate the performance of the recommendation system, Fig. (5) along with Table (2) depicts the confusion matrix & performance metrics which provides deeper insights into the classification accuracy based on the confusion matrix that was built using the dataset. The different metrics as mentioned in the methodology section were calculated and are tabulated below. Compared to state-of-the-art research, the proposed method demonstrates competitive performance, particularly in terms of precision. The integration of BERT embeddings and cosine similarity has proven effective in achieving high accuracy and precision rates, similar to the results reported by Chopra and Lal Saini (2023); Van Huynh *et al*. (2020). Furthermore, this method provides a comprehensive evaluation across different threshold levels, which is often not extensively covered in existing literature.

However, there are areas where the given system could be further improved. For instance, the ensemble methods proposed by Chopra and Lal Saini (2023) could be integrated into the proposed approach to enhance performance further. Additionally, the incorporation of experience-based features, as suggested by Rahman *et al*. (2023), could provide a more personalized recommendation system.

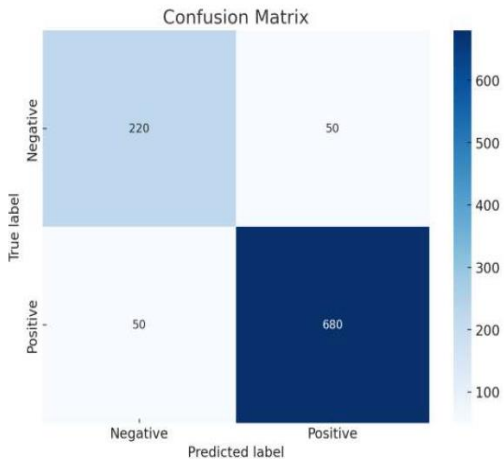

Fig. 5: Confusion matrix of the recommendation system

Table 2: Performance metrics of the system

Accuracy	Precision	Recall	F-1 Score
91%	93.14	93.14	93.14

Conclusion

The system was evaluated using a proper test set comprising 1000 candidates and 100 jobs from various fields and it has its limitations even though it shows some promising results. The system had an accuracy of 91% and a precision of 100% on a large scale. However, further analysis performed at different threshold levels (k values) proves that performance can change quite a lot, so this should be taken into account. As the number of recommended jobs (k) increased from 5 to 60, both accuracy and precision exhibited notable trends. At k = 5, the system demonstrated moderate success with 53% accuracy and 57% precision. Performance metrics stabilized until k = 45, where a significant improvement to 62% accuracy and 66% precision was observed. Beyond this threshold, accuracy continued to improve slightly, peaking at 60% for k = 60, whereas precision displayed fluctuations, highlighting challenges in maintaining high precision with an increasing number of recommendations.

These results point out the critical importance of selecting an optimal k value to balance accuracy and precision. While larger k values may upgrade precision encompassing more relevant job recommendations, the precision tends to suffer, suggesting potential difficulties in sustaining precise recommendations as the recommendation scope broadens. While the recommender platform looks promising, more exactness can be accomplished at cost of accuracy but balance between these two have to be very accurately set for optimizing the system. To guarantee the generally unwavering quality and accessibility of the framework meets down to earth needs, further iterations of the algorithm should focus on achieving higher precision while expanding the recommendation category. This balance will be vital for keeping up the system's down to earth appropriateness and viability within the real-world.

Limitations

Despite the promising results, the proposed method has several limitations. First, the effectiveness of the model is heavily dependent on the quality of the input data. In scenarios where the data is highly unstructured or contains significant noise, the model's performance may degrade, leading to less accurate skill profiling. Additionally, the reliance on certain preprocessing techniques, such as tokenization and vectorization, may introduce biases that affect the model's ability to generalize across different datasets.

Another limitation is the potential for algorithmic biases, particularly in the context of candidate selection. While the model is designed to minimize such biases, the inherent biases present in the training data may still influence the outcomes. Addressing this issue requires ongoing refinement of the model and careful consideration of ethical implications in AI-driven recruitment processes.

Lastly, the current framework primarily focuses on technical skills, with limited emphasis on soft skills or other non-technical attributes. Future research should explore ways to incorporate these elements into the profiling process, providing a more holistic approach to talent sourcing.

Future Scope

Building on the findings of this study, several avenues for future research can be explored. First, further investigation into more advanced preprocessing techniques, such as deep learning-based data augmentation, could help mitigate the challenges posed by unstructured or noisy data. This would enhance the model's robustness and improve its applicability across diverse datasets.

Secondly, there is a need for developing models that better account for non-technical skills, such as communication or teamwork, which are often critical in hiring decisions but difficult to quantify. Integrating sentiment analysis or psycholinguistic profiling techniques could provide a more comprehensive view of a candidate's qualifications.

Finally, addressing the ethical implications of AI in recruitment remains a crucial area of study. Future research should focus on designing transparent and explainable AI models that ensure fairness and reduce biases in the talent sourcing process. This could involve developing frameworks for bias detection and mitigation, as well as exploring the long-term impact of AI-driven hiring decisions on organizational diversity and inclusion.

Acknowledgment

The authors would like to express their gratitude to the Department of Computer Engineering, Don Bosco College of Engineering, Goa, for providing the resources and support required to carry out this research. The authors also acknowledge the valuable feedback and encouragement received from peers and colleagues throughout the development of this manuscript.

Funding Information

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Author’s Contributions

Amey Krishnanath Shet Tilve: Conceptualization, methodology, supervision.

Gaurang Sitaram Patkar: Data curation, software development, formal analysis.

Vadiraj Gururaj Inamdar: Validation, writing original draft, review and edited, project administration.

Merwyn D'Souza: Software development, data analysis, visualization.

Janhavi Naik: Literature review, writing review and edited, data collection.

Ethics

The authors confirm that this study complies with all relevant ethical guidelines. The datasets used in this study were sourced from publicly available platforms and anonymized survey data, ensuring no identifiable personal information was utilized. The research did not involve human or animal subjects, and as such, no specific ethical approvals were required. The authors remain committed to addressing any ethical concerns that may arise post-publication.

References

Agarwal, A., & Senthilkumar, D. (2022). Resume Recommendation System Using Cosine Similarity. *International Research Journal of Modernization in Engineering Technology and Science*, *4*(4), 158–162.

Chopra, A., & Lal Saini, M. (2023). Comparison Study of Different Neural Network Models for Assessing Employability Skills of IT Graduates. *2023 International Conference on Sustainable Communication Networks and Application (ICSCNA)*, 189–194. https://doi.org/10.1109/icscna58489.2023.10368605

Clark, K., Khandelwal, U., Levy, O., & Manning, C. D. (2019). What Does BERT Look at? An Analysis of BERT's Attention. *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 276–286. https://doi.org/10.18653/v1/w19-4828

Devlin, J., Chang, M.-W., Lee, Kenton, & Toutanova, K. (2019). BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4171–4186. https://doi.org/https://doi.org/10.18653/v1/N19-1423

Genc, Z., Babieva, N. S., Zarembo, G. V., Lobanova, E. V., & Malakhova, V. Y. (2021). The Views of Special Education Department Students on the Use of Assistive Technologies in Special Education. *International Journal of Emerging Technologies in Learning (IJET)*, *16*(19), 69–80. https://doi.org/10.3991/ijet.v16i19.26025

Van Huynh, T., Van Nguyen, K., Nguyen, N. L.-T., & Nguyen, A. G.-T. (2020). Job Prediction: From Deep Neural Network Models to Applications. *2020 RIVF International Conference on Computing and Communication Technologies (RIVF)*, 1–6. https://doi.org/10.1109/rivf48685.2020.9140760

Jayakodi, K., Bandara, M., Perera, I., & Meedeniya, D. (2016). WordNet and Cosine Similarity based Classifier of Exam Questions using Bloom's Taxonomy. *International Journal of Emerging Technologies in Learning (IJET)*, *11*(04), 142–149. https://doi.org/10.3991/ijet.v11i04.5654

Karajeh, W., Hamtini, T. M., & Hamdi, M. (2016). Designing and Implementing an Effective Courseware for the Enhancement of e-Learning. *International Journal of Emerging Technologies in Learning (IJET)*, *11*(04), 70–76. https://doi.org/10.3991/ijet.v11i04.5384

Luo, J., & Zhong, W. (2024). Graduates Employment Prediction Based on Deep Learning. *ICDEL '24: Proceedings of the 2024 9th International Conference on Distance Education and Learning*, 162–166. https://doi.org/https://doi.org/10.1145/3675812.3675826

Malgaonkar, S., Soral, S., Sumeet, S., & Parekhji, T. (2016). Study on Big Data Analytics Research Domains. *2016 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, 200–206. https://doi.org/10.1109/icrito.2016.7784952

Mandelbaum, A., & Shalev, Adi. (2016). Word Embeddings and Their Use in Sentence Classification Tasks. *Hebrew University of Jerusalem. URL*. https://doi.org/10.48550/arXiv.1610.08229

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient Estimation of Word Representations in Vector Space*. https://doi.org/https://doi.org/10.48550/arXiv.1301.3781

Pazur Anicic, K., Divjak, B., & Arbanas, K. (2017). Preparing ICT Graduates for Real-World Challenges: Results of a Meta-Analysis. *IEEE Transactions on Education*, *60*(3), 191–197. https://doi.org/10.1109/te.2016.2633959

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543. https://doi.org/10.3115/v1/d14-1162

Rahman, S., Habiba, K., Roy, S., & Nur, F. N. (2023). Job Title Prediction and Recommendation System for IT Professionals. *2023 5th International Conference on Sustainable Technologies for Industry 5.0 (STI)*, 1–6. https://doi.org/10.1109/sti59863.2023.10464457

Sayyadian, M., LeKhac, H., Doan, A., & Gravano, L. (2007). Efficient Keyword Search Across Heterogeneous Relational Databases. *2007 IEEE 23rd International Conference on Data Engineering*, 346–355. https://doi.org/10.1109/icde.2007.367880

van Esch, P., & Black, J. S. (2019). Factors that Influence New Generation Candidates to Engage with and Complete Digital, AI-Enabled Recruiting. *Business Horizons*, *62*(6), 729–739. https://doi.org/10.1016/j.bushor.2019.07.004

Wang, X., Gao, T., Zhu, Z., Zhang, Z., Liu, Z., Li, J., & Tang, J. (2021). KEPLER: A Unified Model for Knowledge Embedding and Pre-Trained Language Representation. *Transactions of the Association for Computational Linguistics*, *9*, 176–194. https://doi.org/10.1162/tacl_a_00360

Yinhan Liu, M. O., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. *ArXiv:1907.11692*. http://arxiv.org/abs/1907.11692

Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., & Liu, Q. (2019). ERNIE: Enhanced Language Representation with Informative Entities. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 1441–1451. https://doi.org/10.18653/v1/p19-1139