Information Sciences 724 (2026) 122728

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

AI-driven semantic similarity-based job matching framework for recruitment systems

Mohammed-Hassan Ajjam, Hamed S. Al-Raweshidy *

Department of Electronic and Electrical Engineering (EEE), Brunel University of London, Uxbridge UB8 3PH London, United Kingdom

ARTICLE INFO

Keywords:
Semantic similarity
NLP
AI-driven recruitment efficiency
ML
Intelligent recruitment systems
Context-aware matching

ABSTRACT

This paper presents a real-time online recruitment application that integrates semantic similarity and artificial intelligence (AI) to improve job-candidate matching. It addresses the growing volume of job applications and the limitations of traditional keyword-based systems, which often fail to capture contextual meaning and complex semantic relationships in job-candidate alignment. The proposed system leverages natural language processing (NLP) techniques, specifically TF-IDF vectorization, cosine similarity scoring, and domain-specific keyword weighting, to interpret conceptual relevance between resumes and job descriptions, enabling more accurate and inclusive recruitment outcomes.

This research developed the system in Python and evaluated it using simulated and real-world recruitment datasets. Experimental results show that the semantic model consistently outperforms keyword-based matching across diverse job domains. For instance, in simulated tests, similarity scores reached 0.74 in the Software Engineer domain, compared to just 0.35 using keyword-based methods. Real-world evaluations further confirmed the model's effectiveness, with semantic scores of 0.83, 0.76, and 0.74 for the Hadoop, Data Science, and PMP domains, respectively. In contrast, the corresponding keyword-based scores remained below 0.17.

Additionally, the system performs well in aligning generalist and specialist profiles, achieving a score of 0.88 for Data analysis roles. These findings validate the system's robustness, scalability, and ability to interpret varied terminology across job sectors. The research presents a scalable, AI-driven framework that supports context-aware, fair, and accurate job matching, significantly advancing intelligent recruitment technology.

1. Introduction

AI-driven semantic similarity has emerged as a promising solution for improving job-candidate matching in recruitment systems. By integrating artificial intelligence with semantic technology, these systems overcome the limitations of traditional keyword-based hiring methods [1], enabling a more context-aware and semantically precise understanding of candidates' qualifications and job descriptions. As the recruitment landscape experiences a rapid increase in online applications and intensified talent competition [2], intelligent decision-support systems provide innovative solutions to enhance hiring accuracy and efficiency [3].Table 1.

Identifying suitable candidates aligns naturally with a natural language processing (NLP) task: computing semantic similarity between a query document (e.g., a job description) and a set of candidate profiles (e.g., resumes), followed by ranking these documents

* Corresponding author.
E-mail addresses: hassan.ajjam@brunel.ac.uk (M.-H. Ajjam), hamed.al-raweshidy@brunel.ac.uk (H.S. Al-Raweshidy).

https://doi.org/10.1016/j.ins.2025.122728
Received 11 June 2025; Received in revised form 23 September 2025; Accepted 23 September 2025
Available online 25 September 2025

based on similarity. Although conceptually straightforward, this task introduces several real-world challenges. First, the lack of ground truth—since final hiring decisions are typically undisclosed—complicates validation. Second, resume screening is subjective and varies widely across industries and evaluators. Third, the evolving nature of the job market—marked by new roles, technologies, and jargon—introduces data drift that can degrade model performance over time.

Traditional online recruitment platforms often suffer from fragmented information and ineffective candidate matching [4]. The semantic matching system introduced in this study addresses these challenges through a context-aware framework that goes beyond keyword overlap. It delivers personalized, real-time recommendations by analyzing contextual and conceptual relationships between resumes and job postings. This approach enhances job-matching precision and relevance by capturing intent and context that traditional methods often overlook. These benefits are validated through extensive testing on simulated and real-world recruitment datasets, confirming consistent performance improvements over baseline systems. Similar semantic strategies have proven effective in other domains involving user-generated content [5], which helps in understanding the broader applicability of this method within computational social systems.

In response to the growing need for fairness, accuracy, and adaptability in hiring, recent research has explored improvements in resume screening [6], bias mitigation, and advanced job-matching techniques [7]. Existing methods typically fall into three categories: keyword-based, semantic-based, and large language model (LLM)-based systems. Keyword-based methods remain popular due to their simplicity and transparency, but their reliance on exact string matches inherently limits their ability to recognize transferable skills or alternative terminology.

LLMs have recently demonstrated outstanding performance in several NLP tasks, but their application in recruitment remains limited [8]. It is due to two primary limitations: (1) the lack of labeled ground truth data and the high variability of recruitment jargons make fine-tuning difficult, and (2) LLMs have been shown to encode and perpetuate societal biases—including those related to gender, ethnicity, and disability—raising ethical concerns in high-stakes contexts such as hiring [8].

To address these limitations, this study proposes a simple yet effective method that combines TF-IDF vectorization with cosine similarity to compute relevance scores between job descriptions and resumes. This approach offers domain-specific advantages:

- **Context Preservation**: TF-IDF is well-suited for lengthy documents, such as resumes and job postings, as it preserves the full textual content without truncation or loss of meaning.
- **Terminology Sensitivity**: The weighting mechanism emphasizes domain-specific terms (e.g., *ETL*, *Scrum*, *NLP*), improving alignment between job requirements and candidate qualifications.
- **Ethical and Interpretability Benefits**: Unlike pre-trained LLMs, which may embed opaque or biased patterns, TF-IDF offers transparency, interpretability, and reduced ethical risk.
- **Computational Efficiency**: The method is lightweight, easily scalable, and inexpensive to update, making it practical for real-time deployment.
- **Robust Similarity Measurement**: Cosine similarity, a widely used semantic metric, facilitates robust comparisons between high-dimensional TF-IDF vectors, especially in the context of unstructured text.

This work contributes a transparent, computationally efficient, and interpretable benchmark for job–resume matching, helping to fill a critical gap in recruitment-focused NLP research. It emphasizes the practical, fair, and scalable integration of AI into hiring systems, supporting the development of responsible computational tools for employment contexts [1]. In addition to its technical contributions, the system addresses broader social issues in the digital labor market—such as inequality, fragmentation, and exclusion—by promoting equity, efficiency, and inclusivity in recruitment. It supports more transparent hiring processes [9], reinforcing the role of semantic AI in building fairer employment ecosystems.

The remainder of this paper preceeds as follows: Section II reviews the related work and outlines the differences between our approach. Section III details the methodology, including the preparation process, simulation setup, and job-matching algorithm model. Section IV presents the experimental results and quantitative evaluation. Section V concludes the paper and discusses implications for future AI-driven recruitment systems.

2. Related work

As the job market becomes increasingly complex, the volume of job applications is also rising, necessitating more advanced

Table 1
Workflow comparison between keyword-based and semantic similarity model.

Steps	Keyword-Based Baseline	Semantic Similarity Model
Preprocessing	Tokenization, stopword removal, stemming, lemmatization	Tokenization, stopword removal, stemming, lemmatization
Representation	Exact keyword/token overlap (frequency counts)	TF-IDF weighted vectors with domain-specific reweighting
Similarity	Keyword overlap ratio	Cosine similarity over TF-IDF vectors
Ranking	Candidates ranked by the number of matching keywords	Candidates ranked by semantic similarity scores
Context Handling	None (exact matches only)	Captures synonyms, related skills, and context

The keyword-based baseline relies solely on exact token overlap, without considering semantic weighting, while the proposed semantic similarity model utilizes TF-IDF with cosine similarity.

recruitment systems. Traditional hiring processes rely heavily on keyword-based searches, which can overlook contextual meaning and detailed qualifications in job descriptions and candidate profiles [10]. Recruitment platforms have since evolved from basic text-matching methods to sophisticated AI-driven systems with enhanced semantic understanding, spurred by advances in artificial intelligence and semantic technology [11]. The integration of ontology-based structures and machine learning models enables personalized and context-sensitive job recommendations, while also reducing the cognitive load on recruiters [10].

2.1. Semantic matching and modern recruitment challenges

Recruitment technology has transitioned beyond exact keyword-matching methods, which historically identified only direct term overlaps between resumes and job descriptions. Such methods fail to capture the conceptual relevance of skills and experience, leading to poor job-candidate alignment and inefficiencies in hiring.

Semantic matching, by contrast, evaluates the meaning of job roles and the intent behind candidate qualifications, even in the absence of structural or lexical similarity, as shown in Wikipedia-based semantic approaches [10]. For instance, while a traditional model might require an exact mention of "Python" or "data visualization," a semantic model infers relevance through related terms, such as machine learning or statistical programming, offering a more holistic and context-aware approach. Classification techniques that incorporate Wikipedia-based or domain-specific embeddings support this semantic understanding [10] and have also been applied successfully in domains such as software requirements retrieval [12].

Despite these advances, modern recruitment continues to face persistent challenges. The growing volume of online applicants frequently leads to screening bottlenecks and delays in decision-making. Traditional matching systems continue to produce irrelevant or poorly ranked recommendations, which frustrates both employers and job seekers. Recruiters frequently struggle to identify suitable candidates at scale [13], while job seekers often receive inaccurate assessments when systems rely on keyword occurrences rather than actual capabilities. Conventional search mechanisms often overlook career progression, transferable skills, or evolving industry needs, resulting in longer hiring cycles, increased costs, and missed opportunities [14].

Semantic similarity techniques directly address these issues by reducing ambiguity, enabling more meaningful evaluations, and supporting inclusive job-candidate alignment.

2.2. AI-enabled recruitment portals

Modern job portals serve as centralized platforms for job seekers and employers; however, their effectiveness is contingent upon the quality of the underlying job-matching algorithms. Many still rely on basic keyword filters, which result in poor matches and inefficient candidate selection [6]. Recent proposals also include automated resume screening and job suggestions that enhance efficiency in screening job candidates [15]. AI-driven recruitment portals enhance matching accuracy by applying NLP, machine learning, and semantic technologies to extract contextual meaning from textual data [3].

Additionally, real-time predictive analytics can adjust job recommendations in response to changing labor market trends, candidate behaviors, and organizational needs. These capabilities reduce the manual workload of recruiters and improve the overall candidate experience by personalizing job recommendations to align with individual profiles, interests, and career trajectories.

2.3. AI-driven systematic matching framework

AI frameworks streamline hiring through automation and intelligent ranking of job-candidate matches [16]. The proposed system combines semantic similarity models with machine learning techniques to build a scalable and context-sensitive job-matching system. Deep learning-based NLP models allow for efficient processing of a large volume of application data, while preserving the semantic integrity of resumes and job descriptions [3,16].

Employing Similar techniques in related fields, such as team formation using social semantics or matching software artifacts through ontological modeling. Semantic similarity scoring plays a pivotal role by computing job-candidate relevance beyond superficial word overlap, capturing intent, skills, and domain knowledge [7]. Real-time optimization further improves responsiveness by updating job-candidate recommendations in line with labor market trends [17]. The results are a streamlined hiring pipeline characterized by faster decisions, improved alignment, and a more robust digital hiring experience [18].

Limitations of LLM-based recruitment approaches

Large language models (LLMs) have demonstrated state-of-the-art performance in various NLP tasks, but their application in recruitment remains limited [8]. Most LLM-based recruitment studies rely on small, narrowly scoped datasets, which limit their generalizability across job sectors [8]. Moreover, the absence of definitive ground truth, due to confidentiality and subjectivity in hiring decisions, makes effective fine-tuning and validation difficult.

Additionally, pre-trained LLMs often encode societal and occupational biases, which raises ethical concerns in hiring decisions [19]. Their black-box nature, lack of interpretability, and high computational costs further constrain real-world deployment in recruitment, which demands transparency, efficiency, and fairness.

2.4. Distinction from prior work

Previous recruitment systems have focused mainly on isolated components, such as keyword search or AI-assisted parsing, without integrating them into cohesive, real-time platforms [16]. In contrast, this study proposes a comprehensive, modular system that

combines semantic similarity, context awareness, NLP, and real-time analytics.

Where earlier models often lacked adaptability and scalability, our approach supports dynamic job-candidate alignment by incorporating resume parsing, skill extraction, candidate ranking, and timely notifications. It addresses gaps in context insensitivity and fragmented system design [3], offering a unified solution optimized for accuracy, efficiency, and fairness in modern recruitment [20].

2.5. Gaps in prior work and bias in semantic matching

Although researchers have made significant progress in semantic matching, many existing models still overlook the risk of bias inherent in training data and language representations. Most systems prioritize performance metrics such as accuracy or relevance, while neglecting fairness and representational equity, particularly for underrepresented groups [21].

This oversight poses serious concerns, as demographic-specific language patterns may inadvertently influence similarity scoring and lead to discriminatory outcomes [20]. To address this, our system incorporates bias mitigation at multiple stages: (1) fairness-aware matching constraints (e.g., one-to-one candidate mapping); (2) exclusion of protected attributes from the similarity computation; and (3) normalization of domain-specific terminology and anonymization of data input. These design strategies represent a crucial step toward developing more inclusive, transparent, and responsible AI-driven recruitment systems. While not exhaustive, this

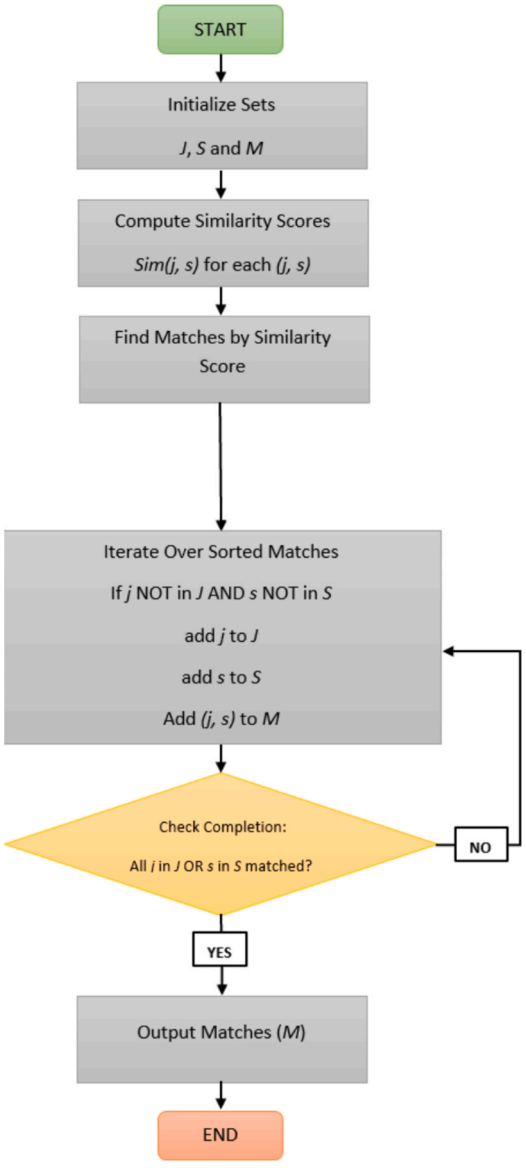

Fig. 1. Workflow of the semantic job-matching algorithm.

work makes a meaningful contribution to the ongoing conversation about ethics and equity in hiring.

3. Methodology

The key methodological contribution is the integration of fundamental theoretical principles with a semantic similarity model that utilizes contextual embeddings and feature weighting. This architecture enables modular, scalable recruitment pipelines that are suitable for real-world matching issues, thereby improving contextual understanding and semantic precision.

3.1. Job matching algorithm model

As built into this model, the job-matching algorithm maximizes the sum of similarity scores for all job matches. Each job seeker and each job description will match with a maximum of one other, ensuring a one-to-one correspondence to maintain fairness and avoid duplicate assignments.

1) **High-Level View of a Semantic Matching Model**

a) Feature Reweighting Mechanism: This component emphasizes the importance of domain-relevant features—such as technical skills, certifications, and job-specific terminology—within the matching process. It functions as a rule-based, domain-informed system. During preprocessing, the system applies curated keyword dictionaries and adjusts TF-IDF weights to boost the influence of semantically significant terms (e.g., *ETL*, *NLP*, *PhD*, *Scrum*). These adjustments strengthen the representation of critical features in the resulting document vectors. This transparent approach enhances interpretability and ensures consistency across technical domains, thereby avoiding the opacity or potential bias inherent in learned feature-weighting methods. Although the current weights remain static, future iterations may incorporate adaptive mechanisms based on recruiter feedback and hiring outcomes.

Example Weighting Rules and Derivation: To instantiate this mechanism, normalized weights in the range of 0 to 1 are assigned, indicating the relative importance of each feature. For instance, highly discriminative technical skills, such as "Python" and "ETL", are boosted by assigning high weights. In contrast, generic words, such as "project" or "management", are assigned much lower weights to avoid generic terms influencing similarity scores. Establishing the weighting rules through prioritizing industry-critical skills and down-weighting standard but less informative terms.

b) Similarity Scoring Engine: The engine computes semantic similarity between job and resume embeddings using cosine similarity, producing a normalized score that reflects contextual alignment. This score serves as the basis for ranking job-candidate pairs.

c) Optimization Constraint Engine: This component enforces one-to-one job-candidate pairing using a matching algorithm optimized for high-quality alignment.

Fig. 1 illustrates the logical flow of the algorithm, ensuring the system selects the best match while respecting pairing constraints. Fig 2.Fig 3.

The matching process employs a greedy algorithm that iteratively selects the highest remaining similarity score from the sorted list of job–resume pairs *(j, s)*, ensuring that each job *j* and candidate *s* are selected only once. At each step, the algorithm chooses the best available *(j, s)* pair based on their semantic similarity, enforcing a one-to-one constraint. This strategy prioritizes locally optimal decisions—making the best immediate match—without considering the globally optimal assignment across all pairs. Although this approach may not yield a globally optimal assignment, it significantly reduces computational complexity. This trade-off supports real-

```
for job in job_list:
    best_candidate = None
    best_score = 0
    for candidate in candidate_list:
        score = cosine_similarity(tfidf(job), tfidf(candidate))
        if score > best_score:
            best_score = score
            best_candidate = candidate
    if best_candidate:
        assign_match(job, best_candidate)
        remove(job, job_list)
        remove(best_candidate, candidate_list)
```

Fig. 2. A pseudocode to clarify the framework's inner workings.

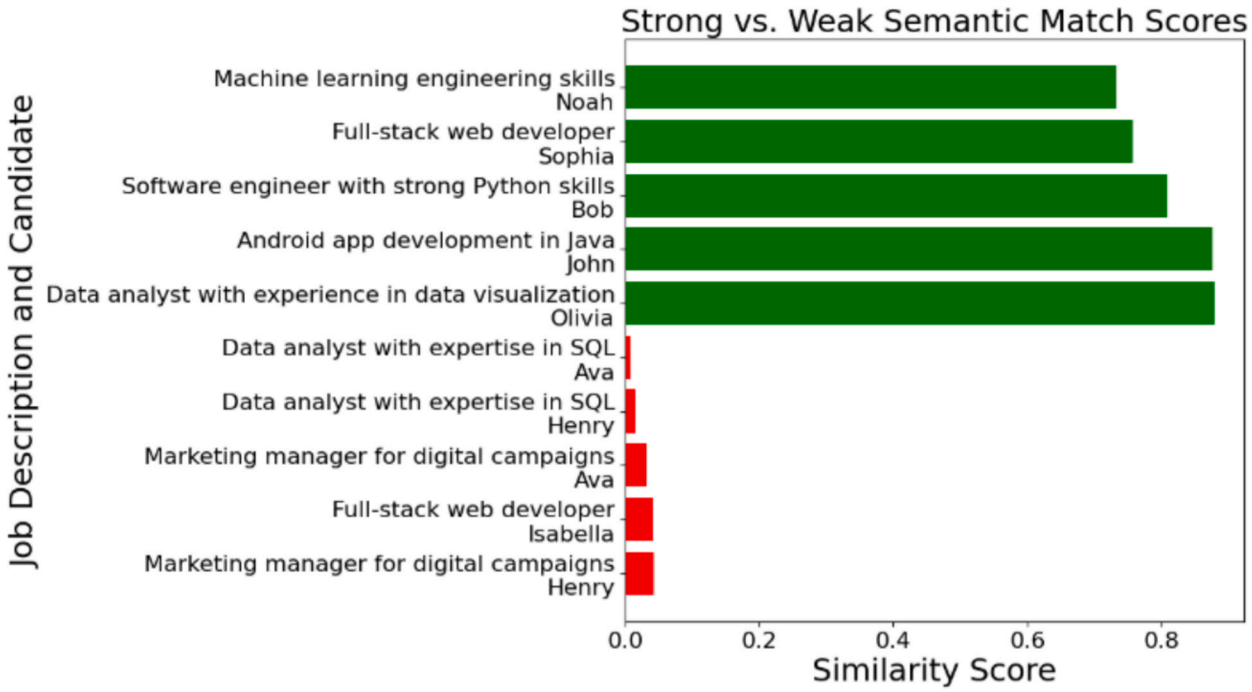

Fig. 3. Semantic similarity score distribution for selected job-candidate pairs. The top five pairs (green bars) represent strong matches (similarity $\geq$ 0.70), indicating strong contextual alignment. The bottom five pairs (red bars) represent weak matches (similarity $\leq$ 0.05), typically reflecting domain mismatch or limited skill overlap. Values correspond to individual job-candidate pairs; therefore, error bars are not applicable.

time performance and scalability, making it well-suited for practical recruitment systems that require fast and responsive candidate matching.

Let J be the set of job descriptions, and j element of J, where $j \in J$ represents a specific job description. Let S be the set of job seekers, and s element of S, where $s \in S$ represents a particular job seeker. Let M represent the set of matches and m element of M, where $m \in M$ represents job matches between job descriptions and job seekers, and M', a subset of M for the pair matches, is returned by $M' \subset M$. Let $Sim(j, s)$ be the variable that represents the semantic similarity score between job description j and job seeker s, and store the pairs (j, s) with the matching similarity scores in set M. Let $\mathscr{F}$ be a search function that takes two arguments, a query (q) and M, and returns the matching pair(s), where q *is* a job description or a job seeker that serves all users, covering different perspectives.

Equation (1): Where m represents the matching pair (j, s) based on the highest similarity score to ensure that each job description and each job seeker are matched with a maximum of one job description and one job seeker, respectively.

$$m = max(\{Sim(j,s) \mid \forall s \in S\}) \tag{1}$$

Equation (2): To return the matching pair(s), this guarantees that each job description and each job seeker will have the best match.

$$M' = \mathscr{F}(M, q) \tag{2}$$

B. Data Sourcing and Description

The datasets used in this study comprise a real-world job postings dataset, a diverse resumes dataset, and a simulated dataset for controlled experimentation.

1) Job Postings Dataset.

This study uses job postings from the Glassdoor "Data Science" dataset on Kaggle, comprising 660 unique entries in the Data Science domain [22]. Each record includes attributes such as job title, company name, location, salary estimate (where available), required qualifications, skills, and a detailed job description. The dataset's single-domain focus was intentional: Data Science is a rapidly growing and evolving field, making it an ideal test case for recruitment systems that must handle complex, skill-intensive roles. However, this domain-specific scope may introduce a bias toward data science–related terminology, skills, and career structures.

2) Resumes Dataset.

The resume dataset originates from the Kaggle "Resume Dataset," which comprises 962 candidate profiles [23]. These profiles span 25 domains, including data science, arts, civil engineering, marketing, and information technology, with both related and unrelated fields to the Data Science job postings. Each profile contains fields for candidate name (anonymized), domain classification, educational background, technical and soft skills, and work experience details. This cross-domain diversity introduces realistic recruitment challenges such as domain mismatch, hybrid skill sets, and varying levels of role relevance. The inclusion of unrelated domains allows

the system to be evaluated not only on direct skill alignment but also on its ability to identify transferable skills.

3) Simulated Dataset.

To address the scarcity of publicly available, labeled job–resume pairs and to enable robust evaluation, we generated a simulated dataset. The scenarios selected reflect real-world recruitment complexities, where transferable skills, atypical career paths, and changing job descriptions can significantly influence match quality. Incorporating such profiles enabled targeted testing of the algorithm's resilience, inclusivity, and fairness.

4) Representativeness and Bias Considerations.

While the Data Science job postings dataset provides depth in a single, high-demand field, it may not fully represent other industries or broader labor market trends. This concentration could bias the system toward vocabulary, skills, and job structures typical of data science. The resumes dataset, by contrast, offers cross-domain coverage that partially mitigates this bias by evaluating generalization across unrelated fields. The simulated dataset enhances representativeness by incorporating diverse and edge-case scenarios; however, its synthetic nature limits its ability to capture all subtleties of real-world hiring, such as cultural fit, employer preferences, or fine-grained skill evaluations. Acknowledging these limitations ensures transparency in the assessment of our semantic matching framework.

3.2. Text processing

Text processing is a critical stage for enabling accurate semantic similarity computations and extracting meaningful insights from textual recruitment data. It involves cleaning and standardizing raw job descriptions and resumes to ensure consistency and reduce noise [15]. The preprocessing pipeline includes duplicate removal, tokenization, stopword removal, stemming, and lemmatization, ensuring normalization of text for subsequent analysis.

To remove stop words, apply the English stopword list. Lemmatize the tokens, which reduces words to their canonical form while preserving context meaning. Conducting both preprocessing steps using NLTK [Ref]. Following preprocessing, TF-IDF (Term Frequency–Inverse Document Frequency) converts the textual data into numerical representations. In our implementation, configure TF-IDF using scikit-learn's TfidfVectorizer with unigrams (single words, such as "Python", "Excel","PhD") and L2-normalized (ensuring cosine similarity reflects content similarity rather than document length). This configuration captures single words while filtering out overly frequent or rare terms. Implement cosine similarity using scikit-learn's cosine_similarity function on the normalized TF-IDF vectors, producing similarity scores in the interval [-1, 1]. This method reduces the influence of common but less informative words. It represents job descriptions and resumes as high-dimensional vectors that accurately capture technical vocabulary (jargon) while maintaining contextual integrity during similarity computations. This method reduces the influence of common but less informative words. It represents job descriptions and resumes as high-dimensional vectors that accurately capture technical vocabulary (jargon) while maintaining contextual integrity during similarity computations.

To assess the similarity between a job description and a resume, the system computes cosine similarity, which measures the cosine of the angle between their vector representations. Let A be the vector for the job description and B be the vector for the resume:

$$Cosine\,Similarity = \frac{\vec{A} . \vec{B}}{\|\vec{A}\| * \|\vec{B}\|} \tag{3}$$

A higher cosine similarity score indicates a stronger match. The system ranks candidates according to their similarity scores. Then it selects a consistent set of top matches while enforcing a one-to-one matching constraint, ensuring that each job seeker is paired uniquely with a job description.

This comprehensive text processing pipeline preserves the semantic relevance of the data while minimizing redundancy and irrelevant variation, providing a robust foundation for the job–resume similarity scoring process.

3.3. Algorithm principles analysis

The proposed job–resume matching algorithm combines optimization strategies with semantic analysis to efficiently match candidates to job postings in a fair and interpretable manner. Five key principles guide the process:

1) **Maximizing Similarity Scores:** The algorithm prioritizes pairing candidates and job postings that achieve the highest similarity scores, ensuring it matches the most qualified applicants with the most relevant positions.
2) **One-to-One Matching Constraint:** The system matches each job seeker to at most one job and each job posting to at most one candidate. This constraint eliminates duplicate assignments, promotes fairness, and ensures consideration for every candidate and job.
3) **Semantic Similarity:** Rather than relying solely on keyword overlaps, the algorithm evaluates the meaning and context of text in both job descriptions and resumes to determine semantic similarity. It enables it to identify relevant matches even when people use different wording for similar skills or experiences.
4) **Scoring Mechanism:** A Semantic Similarity Score between 0 and 1 is assigned to each job-candidate pair, representing the degree of match. This score can be directly integrated into recruitment portals to help employers quickly identify top candidates.
5) **Search Function Flexibility:** The search function $\mathscr{F}$ enables both job seekers and recruiters to query the system according to their specific needs, ensuring a user-centric design and enhancing overall usability.

6) **Greedy Matching Implementation (Pseudocode)** – The greedy one-to-one matching strategy iteratively pairs the highest-scoring job–candidate combination, removing both from the pool to prevent duplicate matches. It ensures efficiency and adherence to the one-to-one constraint:

To balance semantic accuracy with computational efficiency, the system uses **TF-IDF vectorization** rather than large pre-trained language models such as BERT or SBERT. This choice ensures transparency in term weighting, reduces computational overhead, and avoids introducing societal biases embedded in large-scale models. The TF-IDF vocabulary is constructed after preprocessing steps, resulting in high-dimensional, dense vectors suitable for cosine similarity scoring. Because the model is not pre-trained, it requires no fine-tuning, remains fully interpretable, and can adapt quickly to evolving domain-specific terminology—making it well-suited for dynamic and fair job–resume matching tasks.

3.4. Comparison to keyword-based approach

The keyword-based comparison serves as the baseline for evaluating the effectiveness of the proposed semantic similarity approach. This baseline reflects current practice in many commercial recruitment platforms, where systems compare resumes and job descriptions using exact token overlap after basic preprocessing. The method computes similarity as the raw overlap ratio, without TF-IDF weighting, contextual embeddings, or semantic enrichment. While simple, this approach is fast and interpretable, but it cannot capture synonyms, transferable skills, or contextual meanings, which limits its ability to identify suitable candidates when terminology differs.

1) **Objectives**

a) Accuracy: Provide a clear baseline for assessing job-candidate alignment by measuring direct keyword overlap between job descriptions and resumes.

b) Efficiency: Delivers a fast and lightweight comparison method suitable for large datasets, ensuring reproducible and interpretable baseline results.

3.5. Privacy and security analysis

Ensuring the privacy and security of candidate and employer data is essential in any AI-driven recruitment system. Without robust safeguards, sensitive information such as personal identifiers, job descriptions, and application histories may be vulnerable to unauthorized access, breaches, or misuse. This section outlines the key principles and measures implemented to protect data integrity and confidentiality.

1) **Encryption and Data Anonymous:** All personal identifiers, including candidate names, email addresses, and phone numbers, are removed to maintain referential integrity without exposing personal data. Scanning free-text resume fields for identifying references (e.g., specific companies or universities) and generalizing or removing them. Generating synthetic profiles by combining non-identifiable attributes (e.g., skill sets, education levels, and job histories) to support testing without referencing real individuals. Data is encrypted both in transit and at rest using industry-standard protocols, ensuring that intercepted data remains unreadable to unauthorized parties. This layered approach complies with GDPR and CCPA requirements while preserving the semantic fidelity needed for model training and evaluation.
2) **Access Control:** Role-Based Access Control (RBAC) can be secured inside the data center or between servers [24]. This system establishes permissions for accessing specific data, allowing only authorized individuals with bona fide reasons to do so. Moreover, it is best practice to maintain audit logs to track data access and search for unauthorized access or anomalies in data handling.
3) **Data Minimization:** One possible way to mitigate data exposure is to collect only the necessary details required for the job-matching process, avoiding the retention of any unnecessary sensitive information. Retention policies outline how long organizations retain information and how they securely delete it.
4) **Legal Compliance:** Adherence to data protection laws, such as the General Data Protection Regulation (GDPR) in Europe or the California Consumer Privacy Act (CCPA), is vital [25]. These laws require applications to obtain explicit consent before collecting or processing their data, as well as all rights to access and correct the information. At the same time, improving efficiency in recruitment processes through automation-driven solutions such as resume screening and job suggestion systems [26].
5) **Ethical Considerations:** Candidates must feel confident that the organization handles their data responsibly, making complete transparency essential. One fundamental way to achieve this is by informing candidates how organizations will use their data, who will have access to it, and how they will protect it. Additionally, it is important to periodically audit the algorithm used in the analysis to ensure bias avoidance and fair treatment of all candidates.
6) **Security Measures:** Maintaining application security by adhering to secure development practices, including eliminating vulnerabilities to prevent cyberattacks from exploiting them. Security audits and penetration testing are often required to identify potential vulnerabilities that could be exploited [25].

By combining these measures, the system provides a privacy- and security-conscious environment for intelligent recruitment. These safeguards ensure that responsible parties handle sensitive information appropriately, meet compliance obligations, and maintain stakeholder trust. For the operational privacy workflow, see Appendix A.

4. Experimental results

4.1. Simulated data results

The findings in Table 2 below provide a comprehensive view of the simulation results for the first eight applicants from the perspective of job matching. These results demonstrate the utility of a real-time online recruitment application portal built using a semantic technology and AI-based approach for enhanced job-candidate matching. The data highlights the system's ability to generate relevant matches by analyzing contextual alignment between job requirements and applicant profiles with precision and responsiveness.

In terms of the Data Analyst and Software Engineering job postings, the two top matches in the results table indicate Olivia as the best fit for the Data Analyst role, which requires expertise in data visualization, and Bob as the strongest candidate for the Software Engineer position, which requires strong Python expertise. Olivia had the highest overall similarity score (0.88) because she has substantial experience in SQL, Tableau, and Power BI. Bob had a high alignment score (0.81), with backed development experience and cloud deployment experience. These results suggest that the semantic system can interpret technical capability and context, rather than merely relying on keyword matching.

John and Sophia stood out as the highest scorers in the Android App Development and Full-stack Web Developer positions. John's score of 0.88 shows that he has an adequate background in Java and Android Studio, which is part of the job requirements. Sophia has a proficiency rating of 0.76 in full-stack development, with expertise in React, Node.js, and PostgreSQL. Some essential matches highlight the system's ability to distinguish actual, practical development experience in specific programming domains.

Regarding the Machine Learning Engineering role, Noah was the best match, with a similarity score of 0.73. His experience with TensorFlow and Python, combined with the real-time interface, makes him a good fit for the job requirements. It still achieves a high semantic match, even though it is lower than some of the other scores in these first eight candidates, especially considering the generalist nature of the task and the critical importance of specialist roles to assess.

By contrast, some of the top 8 candidates achieved moderate similarity scores of 0.65–0.70 (e.g., Mia, David, and James), yet they were still the best-scoring candidates for those roles. That reflects the system's ability to identify relatively best-fit candidates in severely limited, perfectly aligned cases. These cases illustrate the model's ability to handle sparse or vague resume job description matches, where keyword systems might fail.

In summary, the simulation results for the first eight candidates demonstrate the system's ability to generate precise and meaningful job-candidate pairings. The high similarity scores across job descriptions indicate a practical evaluation of qualifications, skills, and expertise. Additionally, these results highlight the versatility of candidates with multiple skills and the system's capacity to account for broader competencies. These findings support the research's assertion that the proposed real-time job application portal can enhance precision and significantly transform the recruitment industry.

4.2. Real-world data results

This subsection presents the practical results derived from real-world recruitment data using the proposed semantic similarity model. Table 3 shows 10 job-resume pairs, where each row represents a single pair and its corresponding semantic similarity score. Selected these pairs to illustrate both clear and less intuitive matches: the first five reflect obvious semantic alignment. At the same time, the latter five demonstrate the model's capacity to identify cross-domain relevance.

Similarity scores range from 0.665 to 0.856, indicating varying but meaningful degrees of conceptual alignment between job requirements and candidate profiles. The top-ranking pairs involve traditional data-centric roles—such as *Senior Data Analyst*, *Data Engineer*, and *Data Scientist*—paired with resumes from domains like *Hadoop* and *Data Science*. These high scores reflect strong semantic cohesion, where skills and terminology closely align with job expectations.

Tables S1 and S2 present clear examples highlighting the key phrases contributing most to the semantic similarity scores. In Table S1, for the Senior Data Analyst role (Job ID 392), the phrases "massive scale data stores/data processing" and "Experience with All-Source data analysis to perform technical targeting analytic support in the Intelligence Community" are semantically aligned with the resume content "Develop MapReduce coding that works seamlessly on Hadoop clusters" and "APACHE HADOOP MAPREDUCE-Exprience-37," demonstrating strong relevance despite different wording. Similarly, Table S2 shows that the Data Scientist job (Job ID

Table 2
Semantic similarity simulated data results.

Candidate	Job Description	Similarity Score
Olivia	Data analyst with experience in data visualization	0.88
John	Android app development in Java	0.88
Bob	Software engineer with strong Python skills	0.81
Sophia	Full-stack web developer	76
Noah	Machine learning engineering skills	0.73
James	Android app development in Java	0.70
David	Graphic designer skilled in Adobe Creative skills	0.68
Mia	Machine learning engineering skills	0.65

The simulated table summarizes the first 8 best-matching candidates, sorted by semantic similarity scores.

Table 3
Semantic similarity real data results.

Job ID	Job Title	Resume ID	Resume Domain	Similarity Score	Key Matched Phrases
392	Senior Data Analyst	748	Hadoop	0.856	Hadoop ↔ MapReduce; data stores ↔ data processing
81	Data Engineer	745	Hadoop	0.849	data pipelines ↔ Hadoop clusters; ETL ↔ data integration
428	Data Scientist – Machine Learning	746	Hadoop	0.845	machine learning models ↔ Hadoop MapReduce; model deployment ↔ distributed system
433	Data Scientist	4	Data Science	0.842	predictive models ↔ machine learning; data visualization ↔ analytical models
615	Data Engineer	2	Data Science	0.828	data warehouses ↔ SQL/ETL pipelines; big data processing ↔ Python scripts
107	RFP Data Analyst	296	Civil Engineer	0.711	analytical reports ↔ QA/QC documents; data documentation ↔ forecast flow
458	Data Scientist	515	Operations Manager	0.710	process optimization ↔ workflow efficiency; data reporting ↔ "operational dashboards
657	Data Scientist	893	Testing	0.701	test automation ↔ data validation; quality assurance ↔ analytical evaluation
485	Data Scientist	85	Advocate	0.685	data analysis ↔ case documentation; evidence synthesis ↔ "report generation
150	Sr Data Scientist	657	Network Security Engineer	0.665	threat detection ↔ anomaly detection models; network logs ↔ data monitoring

The table shows the semantic similarity scores for 10 job-resume pairs, illustrating both high-confidence (top 5) and cross-domain (bottom 5) semantic matches derived from real-world recruitment data. The Key Matched Phrases column provides qualitative examples of semantically aligned terms, such as "Hadoop ↔ MapReduce", that illustrate why specific job-resume pairs achieved their similarity scores. Extended phrase-level alignment is available in Supplementary Tables S1-S3.

433) includes phrases such as "build predictive models," "automate collection processes," and "data visualization techniques," which align with the candidate's experience in "Machine learning," "Extract data from client systems," and "Develop and deploy analytical models," confirming high contextual overlap with Resume ID 4.

In contrast, the latter pairs involve candidates from diverse backgrounds such as Civil Engineering, Operations Management, Testing, and Network Security Engineering. While less directly aligned by title, these resumes share underlying competencies—such as analytical thinking, process optimization, systems integration, and familiarity with technical workflow—that support their relevance to data-focused roles. It highlights the model's ability to uncover cross-domain skill alignment and transferable competencies that traditional keyword-based systems might miss.

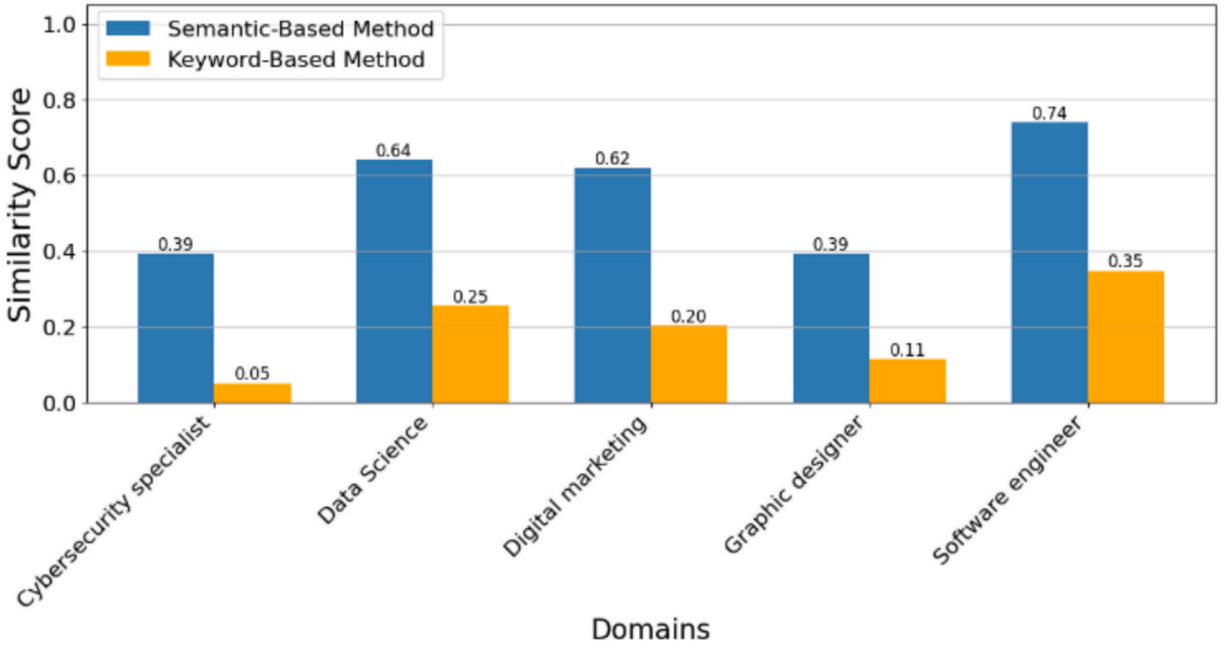

Fig. 4. A comparison of keyword-based and semantic-based methods shows the average similarity scores of the best-matching candidates for each domain using simulated data.

Table S3 illustrates a non-obvious, cross-domain match between an RFP Data Analyst role (Job ID 107) and a Civil Engineer profile (Resume ID 296). Despite differences in job title and domain, the semantic model identified alignment in project management, analytical reporting, and data documentation skills. For instance, the job emphasizes responsibilities such as "complete monthly/quarterly recurring DDQs and RFIs" and "run various analytical reports," which correlate with the candidate's experience in "QA/QC documents," "develop method statements," and "data flow for forecast"—highlighting transferable competencies that traditional keyword systems may overlook.

The semantic similarity scores serve not only as ranking metrics but also as indicators of conceptual fit. Although the absolute values may seem moderate, they are consistent with real-world linguistic variability and the diverse structure of resumes. Unlike controlled datasets, real resumes and job descriptions use inconsistent phrasing, unstructured layouts, and varied terminology across sectors. These factors naturally moderate score magnitudes.

Ultimately, the model demonstrates its real-world applicability by capturing semantically rich alignments between resumes and job descriptions, thereby improving the precision, fairness, and scope of recruitment processes.

Apart from pure same-domain matches, the system also generated non-trivial alignments across domains, demonstrating its capability to identify transferable skills. For example, a Civil Engineer's resume mapped to an RFP Data Analyst position based on standard skill/attribute terms, including "analytical reports ↔ QA/QC documents" and "data documentation ↔ forecast flow" (Table 3). For example, an Operations Manager connected to a Data Scientist role through transferable skills like "process optimization ↔ workflow efficiency." These examples demonstrate that the model can capture relevant skills, also beyond superficial keyword matches, supporting the identification of competent candidates from closely related domains.

4.3. Comparison to keyword-based results

1) Simulated Data: Fig. 4 shows the comparative performance of the keyword-based and semantic-based matching methods across five simulated job domains. As can be seen from the results, the semantic-based method certainly outperforms the keyword-based method in all of the explored domains. It is worth noting that the semantic model achieves an average similarity score of 0.74 for the Software Engineer domain, whereas the keyword-based method yields only 0.35. The results show a similar trend in Data Science and Digital Marketing, where both action keywords achieved semantic scores of 0.64 and 0.62, which are significantly greater than those of their keyword-based counterparts.

This difference is more pronounced in keyword-based domains, such as Cybersecurity Specialist (0.05) and Graphic Designer (0.11), which indicates that the keyword model struggles to detect conceptual parallels when the precise wording is absent. In contrast, the semantic model preserves accuracy by using contextual embeddings to connect comfort with various forms of search projects and experiences, as well as different phrases.

In summary, the simulated results confirm the increased accuracy and reliability of the semantic-based approach, showing its

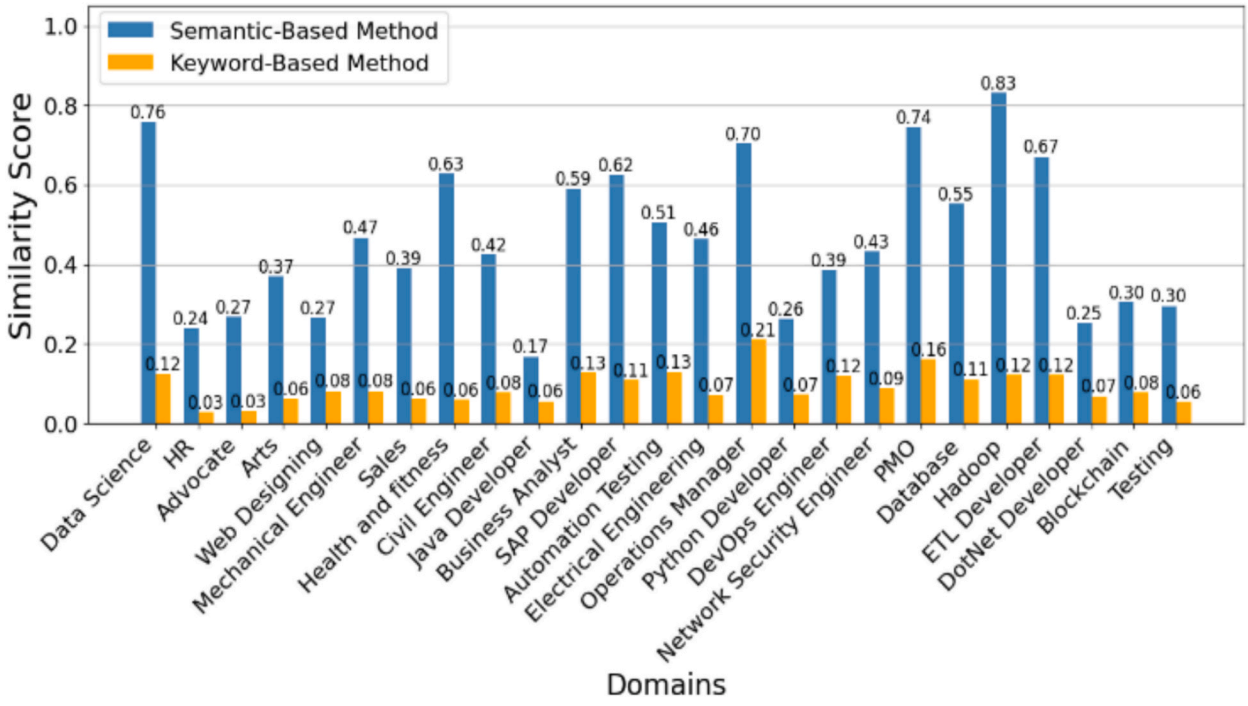

Fig. 5. Comparison of keyword-based *vs.* semantic-based methods showing the average similarity scores of best-matching candidates for each domain using real-world data.

ability to surface potentially relevant candidates through high-volume free-text classifications that strict keyword filters would otherwise miss. They also highlight the model's ability to leverage contextual meaning and account for the variability of language found in real-world resumes, which not only enhances the system's match quality but also supports more inclusive and flexible recruitment practices, particularly in a diverse and evolving employment context.

2) Real-World Data: Fig. 5 compares semantic versus keyword similarity scores across resume domains on a real dataset. The semantic-based method identifies more relevant candidate-job matches in all domains, with blue bars (semantic cores) consistently higher than the orange bars (keyword scores).

Domains such as Hadoop, Data Science, and PMP show the most significant performance gaps, with semantic scores reaching 0.83, 0.76, and 0.74, respectively, compared to much lower keyword scores of 0.12 and 0.16. It illustrates the semantic model's capability to infer contextual meaning, even when candidates use a different language from the job description.

Keyword-based models struggle to perform effectively when exact matches are scarce, particularly in domains such as blockchain, Testing, and Web Design, where string matching often leads to missed meaningful connections.

The Wilcoxon signed-rank test [27] also supports the above findings, which indicate that our semantic-based approach is significantly better than the traditional keyword-based approach, as evidenced by a p-value of 1.305×10^{-5} at a 5 % significance level. The Wilcoxon signed-rank test is more suitable and safer than the *t*-test, as it is a non-parametric test that does not require normality assumptions, especially when the nature of the problem does not guarantee that similarity scores follow a normal distribution. Moreover, the value of Cohen's D is 2.727, representing a huge effect size between the two groups.

4.4. Ranking evaluation

Although cosine similarity scores reveal how resumes and job descriptions align through pairwise comparison, recruitment systems primarily serve as ranking systems, presenting employers with a small subset of top candidates. To evaluate ranking quality more thoroughly, compare our TF-IDF–based approach against two baselines:

- Keyword-based method: A simple overlap model that represents the current practice in many recruitment platforms.
- LLM-based method: The all-MiniLM-L6-v2 model [28], a lightweight sentence transformer that generates 384-dimensional embeddings trained on over 1 billion training pairs. This baseline provides stronger contextual representations than keyword matching while maintaining reasonable efficiency.

Since all job postings in our dataset pertain to the Data Science domain, and ground-truth recruiter labels were unavailable, the evaluation focused on the most relevant resumes. Using a total of 82 resumes, spanning two domains: Data Science (40 resumes) and Hadoop (42 resumes). To measure Precision@10, count the number of retrieved resumes belonging to the target domain, such as Data Science, and divide by 10. To measure Recall@10, divide the exact count by the total number of relevant resumes (40 for Data Science and 42 for Hadoop).

Table 4 presents the results. Our TF-IDF–based method outperforms both baselines across domains. While the LLM-based model performs considerably better than the keyword-based baseline, as expected, our approach achieves the highest Precision@10 and Recall@10 in both Data Science and Hadoop. This advantage likely stems from the sensitivity of LLMs to jargon variability in recruitment contexts, whereas TF-IDF adapts more effectively to domain-specific terminology.

Overall, the semantic matching system proves far more effective in handling diverse and unstructured resume data, offering greater adaptability and precision for real-world recruitment tasks.

5. Discussion

5.1. Semantic matching vs. keyword-based approaches

Experimental results from both simulated and real-world datasets demonstrate a consistent advantage of semantic-based matching over traditional keyword-based methods for job–candidate alignment. Across all tested domains, the semantic similarity model achieved superior contextual understanding and greater flexibility.

As shown in Fig. 4, results from simulated data highlight that the semantic model significantly outperformed keyword techniques, particularly in Data Science and Software Engineering. Keyword-based methods underperformed in areas such as Cybersecurity and Graphic Design, where reliance on exact lexical matches caused missed connections. The semantic approach, by contrast, captured

Table 4
Comparative ranking performance of semantic (TF-IDF), keyword-based, and LLM (SBERT) models.

	Domain			
	Data Science		Hadoop	
Method	Precision@10	Recall@10	Precision@10	Precision@10
Keyword-Based	0.078	0.020	0.004	0.001
LMM-Based (SBERT)	0.262	0.065	0.188	0.045
TF-IDF-Based (our method)	**0.400**	**0.100**	**0.598**	**0.142**

context and intent, recognizing relevant qualifications despite variations in terminology.

5.2. Robustness in real-world contexts

Validation with real-world recruitment data (Fig. 5) further confirmed the model's advantage over keyword-based baselines. Domains such as Hadoop, PMP, and Data Science yielded similarity scores above 0.70, compared to keyword-based scores of less than 0.16. The model's ability to detect transferable skills and interpret diverse professional language supports technical, interdisciplinary, and non-traditional career paths. It enhances recruiter efficiency, broadens inclusivity, and facilitates the identification of underrepresented talent pools. These findings provide the foundation for system-level considerations in real-world deployments, discussed next.

As shown in Table 4, the TF–IDF–based model consistently outperforms both the keyword baseline and the LLM (SBERT) baseline in terms of Precision@10 and Recall@10 across the Data Science and Hadoop domains. While the LLM baseline improves over keyword matching, TF-IDF achieves the highest top-10 precision and recall, indicating better retrieval of relevant resumes in the shortlist. This pattern suggests that TF-IDF adapts more effectively to recruitment-specific jargon and domain terminology, thereby reducing the likelihood that recruiters overlook qualified candidates.

While some of the real-world similarity scores fall within the range of 0.66–0.72, others exceed 0.80–0.85, indicating substantial contextual similarity. Considering scores around 0.65–0.70 are actionable thresholds, as they consistently distinguish between relevant and irrelevant matches across domains. In contrast, higher scores above 0.80 reflect strong alignment. Expecting recruiters to interpret these values as relative ranking signals rather than strict cutoffs, using them to prioritize candidates for review while applying domain expertise to finalize decisions. As illustrated in the Results, candidate scoring in the 0.65–0.70 range often constituted the best-fit matches available, confirming their practical utility despite appearing modest in absolute terms.

5.3. System-level design and scalability

Beyond performance, the system provides a deployable, end-to-end recruitment platform that integrates data ingestion, semantic parsing, real-time similarity scoring, and automated ranking within a unified architecture. Its modular design enables domain portability and rapid adaptation to changing organizational needs.

The system supports large-scale deployments, but high-dimensional text representations can increase computational costs. Optimizations include dimensionality reduction, approximate nearest neighbor search, and batch processing to enhance scalability and efficiency. These measures will help maintain responsiveness while supporting real-time recruitment scenarios with large candidate pools.

5.4. Ethical considerations and bias mitigation

Supporting ethical deployment through fairness-aware features, including anonymized processing, exclusion of protected attributes from similarity computations, and one-to-one matching constraints. Anonymization is a multi-layered process, replacing personal identifiers with synthetic placeholders and generalizing free-text fields to prevent re-identification.

The one-to-one matching constraint supports fairness by preventing candidates from being matched to multiple roles. It is also important to note that TF-IDF can encode implicit bias if terms disproportionately associated with particular groups are upweighted (e.g., "leadership" or "senior" in predominantly male areas). To address this, generic or stereotyped terms are normalized and down-weighted, with curated keyword dictionaries and frequency analysis ensuring that such terms do not dominate the similarity score. Future work will extend the system by systematically auditing TF-IDF term distributions to identify and alleviate implicit demographic or occupational biases.

However, in domains where transferable skills are in high demand, this constraint may limit the system's ability to allocate talent optimally. Future refinements may include flexible or probabilistic matching strategies to strike a balance between fairness and utility.

A comprehensive empirical evaluation of fairness remains a priority for future work. Planned assessments include subgroup representation tracking, fairness auditing, and evaluation of non-standard candidate profiles, such as those of career changers and freelancers. Additionally, social impact metrics—such as diversity among matched candidates and recruiter satisfaction—will provide further validation of fairness in practice. Future Work will also extend the system to perform fairness audits with synthetic demographic attributes, such as gender and race-neutral names, used to test metrics such as demographic parity and equal opportunity, so that we can empirically evaluate if the system's anonymization, one-to-one matching, and exclusion of protected attributes are effectively combating bias in job-candidate matching.

5.5. Multilingual and cross-regional applicability

Currently, the system processes only English-language resumes and job descriptions, which limits its applicability in the global recruitment market. Future Work will extend the framework to support multilingual recruitment through several strategic methods. One approach is to incorporate translation APIs (e.g., Google Translate, DeepL) as a preprocessing step, translate non-English resumes into English before calculating similarity. A more advanced strategy is to adopt cross-lingual embeddings, such as LASER or multilingual BERT, to perform semantic alignments across languages without translation. Furthermore, adaptive terminology mapping is a crucial step for capturing regional variations in job titles, skills, and certifications, ensuring consistent representation of local terms in

the similarity model. Combined, these strategies provide a clear roadmap for expanding inclusivity and improving the system's practical utility across multilingual and cross-regional recruitment contexts.

5.6. Limitations and threats to validity

Several limitations influence the system's performance and applicability. The system's accuracy depends on the completeness and clarity of resumes and job descriptions; ambiguity, insufficient detail, or inconsistent formatting—common in real-world submissions—may cause the semantic model to misinterpret intent or overlook relevant qualifications. Fragmented or sparse documents particularly reduce TF-IDF's ability to capture domain-relevant terminology, which can lower similarity scores even when candidates may otherwise be a good fit.

Additionally, the model's results are inherently dependent on the chosen semantic representation—TF-IDF in this study—which offers transparency and interpretability but may underperform large language model (LLM)-based sentence encoders in contextual understanding, albeit with trade-offs in bias sensitivity, explainability, and computational efficiency. Scalability also presents a constraint, as cosine similarity calculations scale linearly with the number of job-candidate pairs. For huge pools, this approach may introduce latency or resource bottlenecks. Approximate nearest-neighbor (ANN) search methods, such as FAISS, Annoy, or HNSW, provide promising alternatives for future optimization.

Another limitation arises in cold-start scenarios when extending to new domains with unfamiliar terminology. Without sufficient domain-specific vocabulary, the system may initially underperform. Bootstrapping strategies, such as seeding keyword dictionaries with publicly available ontologies, incorporating recruiter-provided domain terms, or using unsupervised clustering to detect emerging jargon, can help address this challenge in future iterations.

To address these limitations, future work will explore the integration of debiased or fairness-aware embeddings, adaptive thresholds for ambiguous or sparse inputs, computational optimizations such as dimensionality reduction and ANN search, as well as enhanced capabilities for multilingual and domain coverage. These improvements aim to strengthen the system's fairness, scalability, and robustness in varied and complex real-world recruitment contexts.

Additionally, residual re-identification risks may arise when unique attribute combinations occur within small applicant pools or niche domains. Even without direct identifiers, unusual patterns—such as rare skill sets, education histories, or job sequences—can potentially reveal individual identities. To mitigate this, the system applies dimensionality reduction techniques, including vector truncation and principal component analysis (PCA), which eliminate fine-grained identifying details while preserving the semantic structure necessary for matching. Future iterations will incorporate formal privacy-preserving mechanisms, such as differential privacy, to further reduce re-identification risks and strengthen compliance with data protection regulations. For the operational privacy workflow, see Appendix A.

5.7. TF-IDF vs. BERT/SBERT in comparative aspect

Although the methodology section presents justification for applying IF-IDF and cosine similarity, it is essential to compare this approach within a broader context. Transformer representations derived from systems like BERT and SBERT perform more complex contextual modelling, but come with significant downsides. First, they are computationally expensive: the inference time requires GPU acceleration, and the latency and energy consumption generated are much higher than those of IF-IDF, making them unsuitable for massive online recruiting due to a lack of scalability [29]. Second, customizing LLMs for recruitment domains is costly and limited by the scarcity of labeled ground-truth data, which hinders cross-domain (industry-specific) generalization [30]. Third, fairness remains an issue: empirical studies demonstrate that BERT representations encode occupational and gender biases, raising concerns about fairness and ethics in high-stakes applications such as recruiting [19]. In contrast, TF-IDF offers clear term weights, rapid adaptation to domain-specific terms, and reduced bias amplification. These trade-offs suggest that, although embeddings based on LLMs may enhance contextual sensitivity, using TF-IDF embeddings is a more pragmatic and responsible approach for scalable and ethical recruiting systems.

6. Conclusion

This study presents and evaluates an end-to-end semantic similarity–based job-matching system, utilizing both simulated and real recruitment datasets, and benchmarks its performance against traditional keyword-based approaches. Across multiple domains, the semantic model consistently outperformed a keyword-based baseline, demonstrating superior contextual understanding, recognition of transferable skills, and the ability to capture relevant qualifications beyond exact keyword matches. These improvements enhance job–candidate alignment, recruiter efficiency, and inclusiveness in identifying diverse talent.

While the results confirm the advantages of semantic similarity methods, the system's performance remains sensitive to data quality, domain coverage, and scalability constraints in large-scale, real-time deployments. Additionally, developers may encode societal biases into semantic representations even when they exclude protected attributes, underscoring the importance of fairness-aware design.

Future work will focus on extending multilingual and cross-regional applicability, integrating debiased or fairness-aware embeddings, and optimizing computational performance through techniques such as dimensionality reduction and approximate nearest neighbor search. Further exploration of hybrid approaches—combining semantic similarity with domain-specific ontologies or machine learning classifiers—may yield additional gains in precision and adaptability. By addressing these areas, the system can evolve

into a scalable, fair, and globally deployable solution for intelligent recruitment.

CRediT authorship contribution statement

Mohammed-Hassan Ajjam: Software. **Hamed S. Al-Raweshidy:** Software.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper: The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Hamed S Al-Raweshidy reports financial support and article publishing charges were provided by Brunel University London. Hamed S Al-Raweshidy reports a relationship with Brunel University London that includes: employment. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A:. Privacy, Security, and Regulatory compliance (GDPR/CCPA)

The platform complies with GDPR/CCPA regulations through a privacy-by-design process that spans the entire end-to-end recruitment workflow. The following subsections detail the operational measures that extend Section F. Privacy and Security Analysis.

A.1 Data Minimization and Anonymization

At the level of the direct identifiers, information such as names, email addresses, phone numbers, addresses, URLs, and IDs is either deleted or substituted with a placeholder (pseudonymization).

- Dates and locations are generalized (e.g., month–year, region). See Table 4 S1-S3 for examples of generalized fields.
- Feature extraction excludes sensitive attributes (e.g., gender, race) and their proxies.
- Dimensionality reduction (e.g., vector truncation) reduces the risk of re-identification.

A.2 Pseudocode: Anonymization and Feature Preparation

The preposing pipeline ensures compliance before similarity computations. Steps include anonymization, pseudonymization, generalization, and suppression of protected features, all of which occur before semantic analysis.

```
for document in dataset:
    # Step 1: Remove direct identifiers
    remove identifiers (email, phone, URL, address)

    # Step 2: Pseudonymize named entities
    detect entities (names, organizations, locations)
    replace each entity with placeholder <PERSON>, <ORG>, <LOC>

    # Step 3: Generalize quasi-identifiers
    generalize dates to month-year
    generalize locations to region

    # Step 4: Tokenize and filter protected attributes
    tokenize text
    remove protected terms (gender, race, proxies, etc.)

    # Step 5: Feature preparation
    apply TF-IDF vectorization with configured parameters
    apply feature reweighting with domain-specific weights

    # Step 6: Store processed vector for downstream analysis
    store processed vector
```

Fig. 6. A pseudocode example of the anonymization and feature preparation pipeline for GDPR/CCPA compliance. The system suppresses direct identifiers, pseudonymizes entities such as names and organizations, generalizes quasi-identifiers like dates and locations, and removes protected terms before applying TF-IDF vectorization and feature selection.

A.3 Data Retention and Deletion Policies

- Retention: For active requisitions, the system stores data for 12 months and allows archival storage for up to 24 months.
- Deletion: If a user cancels a job or submits a request, the system deletes all vectors and raw text associated with that job. The system purges backups every 30 to 60 days.

- Data-subject rights: Requests for access, rectification, erasure, portability, and objection are processed within GDPR (1 month) and CCPA (45 days) timeframes.

A.4 Security and Audit Controls

- Encryption in motion (TLS 1.2 +) and at rest (AES-256).
- Role-based Access Control (RBAC) with Multi-Factor Authentication (MFA).
- Immutable audit logs for all access and changes.
- Schedule integration of differential privacy for future releases to further reduce residual re-identification risks.

A.5 Ethical and Transparency Practices

To maintain user trust and fairness in recruitment:

- Transparency: Candidates are informed about how their data is collected, processed, and protected, as well as who has access to it.
- Fairness audits: Reviewers periodically audit the matching algorithm to detect and mitigate bias.
- Ethical safeguards: The system restricts data use to recruitment purposes only and prevents the exploitation of candidates' personal information for unrelated objectives.

Appendix B. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.ins.2025.122728.

Data availability

I have shared the data links in the references

References

[1] R. Lobo, P.E. Daga, H. Alani, M. Fernandez, Semantic Web technologies and bias in artificial intelligence: A systematic literature review, Semantic Web 14 (4) (2023) 745–770.
[2] J. Aram Khasro, et al., Machine Learning for Recruitment: Analyzing Job-Matching Algorithms, Mach. Learn. 27 (2025) 1.
[3] A.C. Oihab, et al., Intelligent recruitment: How to identify, select, and retain talents from around the world using artificial intelligence, Technol. Forecast. Soc. Chang. 169 (2021) 120822.
[4] P. Shimpi, B. Balinge, T. Golait, S. Parthasarathi, C.J. Arunima, Y. Mali, Job Crafter - The One-Stop Placement Portal, in: 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), 2024, pp. 1–8, https://doi.org/10.1109/ICCCNT61001.2024.10725010.
[5] KN, Pavan Kumar, and ML. Gavrilova, Latent personality traits assessment from social network activity using contextual language embedding, IEEE Trans. Comput. Social Syst. 9 (2) (2021) 638–649.
[6] N. Gangoda, K.P. Yasantha, C. Sewwandi, N. Induvara, S. Thelijjagoda, N. Giguruwa, Resume Ranker: AI-Based Skill Analysis and Skill Matching System, in: 2024 Sixth International Conference on Intelligent Computing in Data Sciences (ICDS), 2024, pp. 1–8, https://doi.org/10.1109/ICDS62089.2024.10756304.
[7] Z. Jing, J. Wang, M. Sigdel, B. Zhang, P. Hoang, M. Liu, and M. Korayem. "Embedding-based recommender system for job to candidate matching on scale." arXiv preprint arXiv:2107.00221, 2021.
[8] V. Swanand, H. Leary, Y. Berhanu Alebachew, L. Hickman, B.A. Stevenor, W. Beck, C. Brown, Human and LLM-Based Resume Matching: An Observational Study, in: In Findings of the Association for Computational Linguistics: NAACL, 2025, 2025,, pp. 4808–4823.
[9] M.I. Kurniawan, Candidate experiences in AI-driven recruitment: A phenomenological study on algorithmic bias and fairness perceptions, Journal of Management & Economics Review 2 (5) (2025) 52–63.
[10] W. Zongda, H. Zhu, G. Li, Z. Cui, H. Huang, J. Li, E. Chen, G. Xu, An efficient Wikipedia semantic matching approach to text document classification, Inf. Sci. 393 (2017) 15–28.
[11] M.C. Urdaneta-Ponte, A. Méndez-Zorrilla, I. Oleagordia-Ruiz, Lifelong learning courses recommendation system to improve professional skills using ontology and machine learning, Appl. Sci. 11 (9) (2021) 3839.
[12] C. Janneth, J. López, N. Piedra, O. Martínez, E. Tovar, Usage of social and semantic web technologies to design a searching architecture for software requirement artefacts, IET Softw. 4 (6) (2010) 407–417.
[13] F. Elaine, R.S. Thomas, M.K. Higgins, C.J. Williams, I. Choi, L.A. McCauley, Finding the right candidate: Developing hiring guidelines for screening applicants for clinical research coordinator positions, J. Clin. Transl. Sci. 6 (1) (2022) e20.
[14] K.A. Ayşegül, M. Meeuwisse, M. Gorgievski, G. Smeets, Uncovering important 21st-century skills for sustainable career development of social sciences graduates: A systematic review, Educ. Res. Rev. 39 (2023) 100528.
[15] A. Irfan, N. Mughal, Z.H. Khand, J. Ahmed, G. Mujtaba, Resume classification system using natural language processing and machine learning techniques, Mehran University Research Journal Of Engineering & Technology 41 (1) (2022) 65–79.
[16] N.H. Dilusha, et al., AI Bot to Increase the Accuracy and Efficiency of Hiring Process of Business Organizations, in: 2024 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES), 2024, pp. 1–6, https://doi.org/10.1109/ICSES63760.2024.10910737.
[17] J. Wang, Y. Yang, S. Wang, C. Chen, D. Wang, Q. Wang, Context-Aware Personalized Crowdtesting Task Recommendation, IEEE Trans. Softw. Eng. 48 (8) (2022) 3131–3144, https://doi.org/10.1109/TSE.2021.3081171.
[18] A. Sabina, P. Buono, R. Lanzilotti, Recruitment chatbot acceptance in a company: a mixed-method study on human-centered technology acceptance model, Pers. Ubiquit. Comput. 28 (6) (2024) 961–984.
[19] Kurita. K, N. Vyas, A. Pareek, A.W. Black, and Y. Tsvetkov. "Measuring bias in contextualized word representations." arXiv preprint arXiv:1906.07337, 2019.
[20] L. Zhou, N. C., & Mehta, R, Fairness in AI-driven recruitment: A systematic review of challenges and solutions, Journal of Artificial Intelligence Ethics and Society 3 (1) (2024) 22–41.
[21] Wang, Y. "Artificial intelligence in recruitment: A qualitative analysis of ethical risks and mitigation strategies". Preprint, 2025, available at SSRN: https://ssrn.com/abstract=4759978.

[22] Glassdoor, Access date 01/09/2024, "Data Science Job Posting on Glassdoor," Source [Online]. Available: https://www.kaggle.com/datasets/rashikrahmanpritom/data-science-job-posting-on-glassdoor.
[23] Kaggle, Access date 01/09/2024, "Resume Dataset," Source [Online]. Available: https://www.kaggle.com/datasets/gauravduttakiit/resume-dataset.
[24] B.A. Ur Rehman, T. Mahmood, T. Saba, S.A.O. Bahaj, F.S. Alamri, M.W. Iqbal, A.R. Khan, An optimized role-based access control using trust mechanism in e-health cloud environment, IEEE Access 11 (2023) 138813–138826.
[25] Y. Mariam, "Ensuring Compliance with GDPR, CCPA, and Other Data Protection Regulations, Challenges and Best Practices," (2023).
[26] V. Mohith, and D. Y. Kumar. "Enhancing Recruitment Efficiency: A Proposal for an Automated Resume Screening and Job Suggestion System on the'Dreams Job'Online Platform." In Proceedings of the International Conference on Computational Innovations and Emerging Trends (ICCIET 2024), vol. 112, p. 303, 2024.
[27] F. Wilcoxon, Individual comparisons by ranking methods, Biometrics 1 (1945) 80–83.
[28] W. Wang, F. Wei, L.i. Dong, H. Bao, N. Yang, M. Zhou, Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers, Adv. Neural Inf. Proces. Syst. 33 (2020) 5776–5788.
[29] J. Devlin, M.W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers), 2019, pp. 4171–4186.
[30] Reimers. N, and I. Gurevych. "Sentence-bert: Sentence embeddings using siamese bert-networks." arXiv preprint arXiv:1908.10084, 2019.