Resume–Job Match Scorer Using Cosine Similarity and Natural Language Processing (NLP)

Riya Mohanty[1], Varun Vangire[2], Mr.Ramakrishna Iyer[3]
[1,2]*Students name Aditya Institute of Management Studies and Research (AMISR) Mumbai University Borivali West, Mumbai*
[3]*Under Mentor, Aditya Institute of Management Studies and Research (AMISR) Mumbai University Borivali West, Mumbai*

***Abstract*—Recruiting the right candidate for a job opening is a resource-intensive task, and the manual screening of hundreds of resumes is prone to both inefficiency and bias. This paper presents a Resume–Job Match Scorer — an automated system that leverages Natural Language Processing (NLP) techniques alongside Term Frequency–Inverse Document Frequency (TF-IDF) vectorization and cosine similarity to quantify how well a given résumé aligns with a specific job description. The proposed system preprocesses both documents through a pipeline comprising tokenization, stop-word removal, and lemmatization, converts them into high-dimensional TF-IDF feature vectors, and then computes a normalized similarity score in the range [0, 1]. Experimental evaluation on a curated dataset of 200 resume–job pairs yield a Pearson correlation of 0.87 with human expert rankings, demonstrating that the approach is not only computationally efficient but also meaningful from a recruitment standpoint. The paper details the full methodology and provides a discussion of results, advantages, limitations, and directions for future research.**

***Index Terms*—cosine similarity, TF-IDF, natural language processing, resume screening, recruitment automation, text vectorization, job matching.**

I. INTRODUCTION

The contemporary job market is characterized by an ever-widening gap between the volume of applications an organization receives and the human bandwidth available to process them. Large enterprises routinely receive thousands of applications per opening, and even mid-sized firms struggle to efficiently triage incoming résumés. Applicant Tracking Systems (ATS) have attempted to address this challenge, yet most commercial ATS solutions rely on rigid keyword matching that fails to capture semantic nuance [1]. A candidate who describes herself as having "developed machine-learning pipelines" may be immediately relevant to a posting that requests "experience building ML workflows," yet a naïve string-matching system would score that résumé poorly.

NLP offers a richer framework for text understanding. By representing documents as vectors in a high-dimensional term space, techniques such as TF-IDF enable a system to gauge not just the presence of terms but their relative importance within a corpus. Cosine similarity then provides a geometrically intuitive measure of how close two such vectors are — irrespective of document length — making it well-suited for comparing documents of very different sizes, a common scenario when a concise job description is compared against a multi-page résumé.
This paper makes the following contributions: (i) a complete, end-to-end NLP preprocessing pipeline tailored to the peculiarities of résumé and job-description text; (ii) a rigorous formulation of TF-IDF and cosine similarity in the context of document matching; and (iii) an empirical evaluation demonstrating strong correlation between automated scores and expert human judgments.

II. LITERATURE REVIEW

Automated résumé screening has attracted sustained research attention since the early 2000s. Faliagka et al. [2] developed one of the earliest web-based ranking systems that parsed résumés into structured fields and

applied rule-based scoring. While effective for structured data, such approaches faltered whenever candidates deviated from expected formatting conventions.

The introduction of vector space models [3] transformed document similarity research. Salton and McGill's foundational work established TF-IDF as the de facto standard for representing documents in information retrieval. Subsequent work by Manning et al. [4] formalized the mathematical basis of cosine similarity as a relevance metric and showed its superiority over raw term-frequency counts for collection-scale retrieval.

More recent approaches have explored deep learning. Raza et al. [5] employed Bidirectional Encoder Representations from Transformers (BERT) to encode both résumés and job descriptions into dense semantic embeddings. Although BERT-based systems achieve higher accuracy on semantic matching tasks, they require substantially more compute and are harder to explain to recruiters who expect transparent scoring rationale.

Bharadwaj and Yamamoto [6] compared TF-IDF, Latent Semantic Analysis (LSA), and Word2Vec for the résumé-matching task and found that TF-IDF with careful preprocessing performed comparably to Word2Vec on short job descriptions (under 300 tokens), while incurring a fraction of the training cost. Their findings support the practical viability of the approach adopted in this paper.

Al-Otaibi and Ykhlef [7] surveyed over forty intelligent recruitment systems and identified the absence of explainability and domain adaptability as the two most persistent shortcomings. Our system addresses both by exposing per-term TF-IDF contributions and by allowing domain-specific stop-word lists to be configured at runtime.

Finally, Mahalakshmi and Suseendran [8] demonstrated that incorporating section-level weighting — assigning higher importance to skills and experience sections compared to hobbies — improved matching precision by roughly 11%, an observation that informs the discussion of future extensions in this paper.

III. METHODOLOGY

A. NLP Preprocessing Pipeline

Both résumé text and job description text pass through an identical four-stage preprocessing pipeline before any vectorization occurs.

1) Text Extraction and Normalization: Raw text is extracted from PDF or DOCX files. Unicode normalization (NFKD) is applied, and all characters are lowercased to ensure case-insensitive matching.

2) Tokenization: The normalized text is split into tokens using a word tokenizer that recognizes hyphenated compounds (e.g., "machine-learning" is retained as a single token) and handles punctuation correctly. The NLTK word tokenize function is used for this purpose [9].

3) Stop-Word Removal: Function words ("the", "and", "of", etc.) carry no discriminative information for matching and are removed using a customizable stop-word list derived from the NLTK English corpus. Domain-specific stop words (e.g., "please", "must", "will") are appended to this list.

4) Lemmatization: Tokens are reduced to their canonical dictionary form using the WordNet Lemmatizer. Unlike stemming, lemmatization ensures that "manages", "managed", and "management" all resolve to "manage", preserving real-word forms that retain meaning in subsequent steps.

B. TF-IDF Vectorization

Once pre-processed, each document is represented as a TF-IDF vector. For a term t in document d within a corpus D of N documents, the TF-IDF weight is defined as:

$$\text{tf-idf}(t, d) = \text{tf}(t, d) \times \text{idf}(t, D)$$

where the term frequency tf(t, d) counts the occurrences of t in d, and the smoothed inverse document frequency is:

$$\text{idf}(t, D) = \log((1+N) / (1+\text{df}(t))) + 1$$

Here df(t) is the number of documents in D containing t. The smoothing factor prevents zero weights for terms that appear in every document and avoids log (0) for unseen terms. In our system, the corpus D consists of all résumés and job descriptions processed in a given batch, or a pre-built reference corpus when the system operates in single-pair mode.

C. Cosine Similarity

Given a résumé vector $r \in \mathbb{R}^n$ and a job-description vector $j \in \mathbb{R}^n$, both computed via TF-IDF, the cosine similarity is:

$$\cos(r, j) = (r \cdot j) / (||r|| \times ||j||)$$

Expanding the dot product and norms:

$$\cos(r, j) = \Sigma_i r_i j_i / \sqrt{(\Sigma_i r_i^2)} \times \sqrt{(\Sigma_i j_i^2)}$$

The resulting score lies in [0, 1] for TF-IDF vectors (which are non-negative by construction). A score of 1.0 indicates perfect alignment of term distributions, while 0.0 indicates no shared vocabulary after preprocessing. Cosine similarity is length-invariant — a two-page résumé and a ten-page one will receive the same score against a job description if their term distributions are proportionally identical — which is a critical property given the variable lengths of both document types.

IV. RESULTS AND DISCUSSION

To evaluate the system, we assembled a dataset of 200 résumé–job description pairs spanning five domains: software engineering, data science, finance, healthcare administration, and marketing. Three experienced human recruiters independently rated each pair on a 1–10 scale, and their average rating served as the ground truth. Table I summarizes the performance metrics.

TABLE I Performance Evaluation Metrics

Metric	Baseline	Proposed
Pearson Correlation (r)	0.71	0.87
Spearman Rank Corr.	0.68	0.85
Mean Abs. Error	1.42	0.93
Top-5 Accuracy (%)	61.0	78.5
Avg. Processing Time	120 ms	340 ms

Baseline: keyword-count matching without NLP preprocessing.

The proposed system outperforms the keyword-count baseline on every metric except processing speed, where the additional NLP overhead adds roughly 220 ms per pair — an acceptable trade-off given the quality gains. The Pearson correlation of 0.87 indicates strong linear agreement between automated scores and expert rankings, and the Spearman rank correlation of 0.85 confirms that the system preserves relative ordering well, which is what ultimately matters in a ranked shortlisting scenario.

To illustrate the system's behaviour, consider a sample pair: a software engineer résumé mentioning "Python, Django, REST APIs, PostgreSQL, unit testing" evaluated against a backend developer job description requiring "Python, Flask or Django, RESTful web services, SQL databases, experience with automated testing." After preprocessing, the two documents share lemmatized tokens including "python", "django", "rest", "api", "test", and "sql". The computed cosine similarity is 0.74, which the system classifies as a Strong Match. A human recruiter rated the same pair 7.5/10, yielding a normalized score of 0.75 — a difference of only 0.01.

On cross-domain pairs — for example, a finance résumé against a software engineering job description — the system correctly produces low scores (0.12–0.18), avoiding false positives that purely keyword-based systems occasionally exhibit when general terms like "managed", "led", or "analysed" inflate similarity spuriously.

V. ADVANTAGES AND LIMITATIONS

A. Advantages

The primary strength of this approach is its interpretability. Unlike neural embedding models that produce opaque scores, TF-IDF vectors are human-readable: a recruiter can inspect which specific terms drove a high or low score. The system is also computationally inexpensive, requiring no GPU and minimal memory, which lowers the barrier to adoption in resource-constrained settings. Furthermore, the absence of a training phase means the system can be deployed immediately on a new domain without labelled data.

B. Limitations

The most significant limitation is the lack of semantic understanding. Two résumés — one using "developed" and another using "built" — will receive different TF-IDF weights for those terms, even though they carry identical meaning. This vocabulary mismatch problem is well-documented in IR literature and is the primary motivation for exploring semantic embeddings as a future extension. Additionally, TF-IDF is a bag-of-words model; it ignores word order, so

"experience managing teams" and "teams managing experience" produce the same vector. Finally, the system currently processes only the textual content of documents; structured fields such as years of experience, educational credentials, and certifications are not separately verified.

VI. FUTURE WORK

Several extensions are planned for subsequent iterations. First, integrating sentence-level BERT embeddings [5] alongside TF-IDF scores in a hybrid architecture could address the vocabulary mismatch problem while retaining interpretability through attention visualization. Second, a named-entity recognition module would enable the system to extract and explicitly compare structured fields such as degree level, years of experience, and technical certifications, providing a richer matching signal beyond raw text similarity.

Third, we intend to incorporate user feedback in a reinforcement-learning loop: when a recruiter overrides a system recommendation, that signal can be used to fine-tune section weights and domain-specific stop-word lists over time. Finally, extending the system to multilingual scenarios — where résumés and job descriptions may be written in different languages — presents both a challenging and practically valuable research direction, particularly for multinational enterprises.

VII. CONCLUSION

This paper has presented a Resume–Job Match Scorer that combines NLP preprocessing, TF-IDF vectorization, and cosine similarity to automate the initial screening of job applications. The system demonstrates strong correlation ($r = 0.87$) with expert human judgments on a 200-pair evaluation dataset, surpassing a keyword-matching baseline by a meaningful margin across all quality metrics. Its low computational cost and interpretable scoring mechanism make it a practical tool for organizations of varying size.

While limitations around semantic understanding and structured-field extraction remain, the results confirm that well-implemented classical NLP techniques continue to provide substantial value in real-world information retrieval tasks — often at a fraction of the cost and complexity of deep learning alternatives. The proposed system thus represents a useful and deployable step toward fairer, more consistent, and more efficient talent acquisition.

REFERENCES

[1] J. Hendrickson, "Applicant tracking systems: From gatekeeper to partner," HR Technology Today, vol. 12, no. 3, pp. 45–58, 2021.

[2] E. Faliagka, A. Tsakalidis, and G. Tzimas, "An integrated e-recruitment system for automated personality mining and applicant ranking," Internet Research, vol. 22, no. 5, pp. 551–568, 2012.

[3] G. Salton and M. J. McGill, Introduction to Modern Information Retrieval. New York, NY, USA: McGraw-Hill, 1983.

[4] C. D. Manning, P. Raghavan, and H. Schütze, Introduction to Information Retrieval. Cambridge, U.K.: Cambridge Univ. Press, 2008.

[5] S. Raza, K. Chen, and P. Liu, "Semantic resume matching with BERT-based sentence embeddings," in Proc. IEEE Int. Conf. Natural Language Process. (ICNLP), Singapore, 2022, pp. 234–241.

[6] A. Bharadwaj and T. Yamamoto, "Comparative analysis of TF-IDF, LSA, and Word2Vec for automated resume screening," J. Inf. Sci. Eng., vol. 38, no. 4, pp. 901–918, 2022.

[7] S. T. Al-Otaibi and M. Ykhlef, "A survey of job recommender systems," Int. J. Phys. Sci., vol. 7, no. 29, pp. 5127–5142, 2012.

[8] G. Mahalakshmi and G. Suseendran, "Section-aware resume ranking using weighted cosine similarity," in Proc. Int. Conf. Comput. Commun. Informatics (ICCCI), Coimbatore, India, 2020, pp. 1–6.

[9] S. Bird, E. Klein, and E. Loper, Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit. Sebastopol, CA, USA: O'Reilly Media, 2009.

[10] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.