International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

An Intelligent Framework for Automated Recruitment Filtering and Skill Gap Analytics via Vector Space Model

Submitted by: **Saurabh Saxena, Shashwat Sahu, Ankit Sen, Kshitij Kumar Soni**
Under the Guidance of: **Prof. (Dr.) Samta Gajbhiye** (Professor & Head, Department of CSE)
Department of Computer Science and Engineering (Specialization in Artificial Intelligence) Shri Shankaracharya Technical Campus, Bhilai, Chhattisgarh, India • November 2026

Abstract - The rapid digitization of talent acquisition has resulted in a high volume of unstructured job applications, creating severe screening bottlenecks for human resource teams. Conventional Applicant Tracking Systems (ATS) rely heavily on opaque Boolean keyword filters that lack transparency and offer no diagnostic feedback to job seekers. This study presents An Intelligent Framework for Automated Recruitment Filtering and Skill Gap Analytics via Vector Space Model, an interactive web-based architecture developed using Python and Streamlit. The proposed system standardizes multi-format document ingestion (PDF and DOCX), extracts technical competencies via boundary-safe regular expression lookarounds, and computes a quantitative compatibility score using Term Frequency-Inverse Document Frequency (TF-IDF) weighting and Cosine Similarity within a Vector Space Model (VSM). Furthermore, the framework executes set-theoretic skill gap analytics to partition required competencies into matched strengths and missing qualifications. Benchmark evaluations demonstrate that the framework processes document pairs in under 350 milliseconds, eliminates false negatives on symbolic technical terms (such as C++ and CI/CD), and suppresses boilerplate score inflation through TF-IDF stop-word filtering. Ultimately, this transparent, low-latency architecture enhances initial candidate shortlisting efficiency for recruiters while providing actionable upskilling insights for applicants.

Keywords: Vector Space Model (VSM), Automated Recruitment Filtering, Skill Gap Analytics, Natural Language Processing (NLP), TF-IDF, Cosine Similarity, Applicant Tracking System (ATS).

1. INTRODUCTION

1.1 Background Context

Modern corporate recruitment relies predominantly on digital job portals, where a single technical vacancy frequently attracts hundreds or thousands of applications. Evaluating unstructured Curriculum Vitae (CV) documents manually is labor-intensive, costly, and prone to human fatigue and cognitive bias. Consequently, enterprises rely on automated recruitment filtering frameworks to perform initial candidate triage.

1.2 Research Gap

Despite widespread commercial adoption, legacy Applicant Tracking Systems (ATS) typically operate as opaque "black boxes." While they filter resumes using rigid keyword matching, they rarely explain why a candidate was rejected or quantify exact competency deficits relative to the Job Description (JD). Conversely, job applicants lack accessible diagnostic tools to evaluate their alignment with target roles prior to submission. Furthermore, standard word-boundary tokenizers frequently fail to parse modern symbolic technical identifiers (e.g., `C++` , `Node.js` , `CI/CD`),

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

leading to inaccurate skill extraction.

1.3 Core Research Questions

This research addresses three primary questions:

1. How can unstructured resumes across heterogeneous file formats (PDF and DOCX) be ingested and normalized into a unified analytical pipeline with minimal latency?
2. How can boundary-safe lexical extraction be combined with Vector Space Modeling (VSM) to produce an interpretable, length-invariant recruitment filtering score?
3. How can set-theoretic analytics translate raw keyword comparisons into actionable skill gap diagnostics for both recruiters and candidates?

1.4 Thesis Statement

By integrating multi-format text extraction, lookaround-based regular expression entity detection, TF-IDF Cosine Similarity modeling, and set-theoretic gap computation within a lightweight web interface, the proposed Vector Space Model framework provides a fast, deterministic, and transparent alternative to opaque recruitment filtering systems.

2. LITERATURE REVIEW

Automated candidate-job matching sits at the intersection of Information Retrieval (IR) and Natural Language Processing (NLP). Early recruitment systems relied primarily on exact Boolean string queries, which suffered from high false-negative rates whenever resume phrasing deviated slightly from job description templates [1].

To overcome rigid Boolean matching, Salton et al. formalized the **Vector Space Model (VSM)**, representing text documents as algebraic vectors within a multi-dimensional term space [1]. Building on this foundation, Salton and Buckley introduced **Term Frequency-Inverse Document Frequency (TF-IDF)** weighting, which scales term significance inversely to its frequency across a document corpus [2]. Manning et al. demonstrated that pairing TF-IDF vectors with **Cosine Similarity** measures the angular divergence between documents independent of document length, preventing verbose resumes from artificially outscoring concise ones [3].

More recently, deep contextual language models—such as Bidirectional Encoder Representations from Transformers (BERT)—have been applied to semantic resume recommendation [4, 6]. While transformer architectures capture deep contextual synonyms, they require substantial memory overhead, GPU acceleration, and large annotated datasets. For real-time, browser-based recruitment tools where low latency, zero GPU dependency, and transparent explainability are critical, VSM-driven statistical NLP paired with curated domain taxonomies remains highly effective [5].

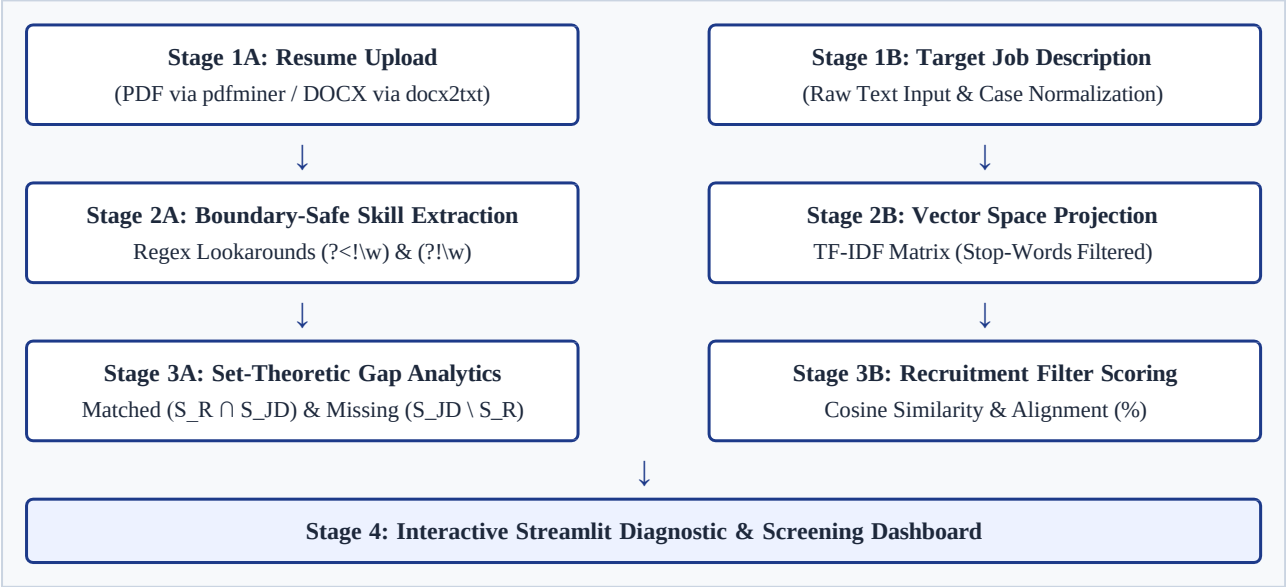
3. METHODOLOGY

The proposed framework implements a four-stage modular pipeline built in Python 3 using `streamlit`, `pdfminer.six`, `docx2txt`, and `scikit-learn`.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

3.1 System Architecture Pipeline



3.2 Data Collection and Ingestion

The framework accepts two primary inputs: an uploaded candidate resume file (`PDF` or `DOCX`) and a raw text string representing the target Job Description. File streams are routed dynamically:

- **PDF Extraction:** Processed using `pdfminer.six`, which reconstructs logical text sequences from PDF layout objects.
- **DOCX Extraction:** Processed using `docx2txt`, which extracts raw UTF-8 text from XML-based Microsoft Word structures while discarding styling metadata.

3.3 Boundary-Safe Skill Extraction

Both the extracted resume text (T_R) and job description text (T_{JD}) are normalized via case-folding. A curated domain taxonomy D containing 37 core industry competencies (spanning programming languages, web frameworks, databases, cloud platforms, DevOps, and data science) is queried against both texts. Standard word-boundary anchors ( ) fail on technical identifiers containing trailing non-alphanumeric characters (such as `+` in `C++`). To prevent false negatives, the extraction module applies negative lookbehind and lookahead regular expression assertions:

```
Pattern(s) = (?<!\w) + escape(s) + (?!\w)
```

Querying all $s \in D$ yields two competency sets: $S_R \subseteq D$ (resume skills) and $S_{JD} \subseteq D$ (required job skills).

3.4 Vector Space Model (VSM) Measurement

To evaluate holistic lexical alignment beyond isolated keywords, the document pair $C = \{T_R, T_{JD}\}$ is projected into a shared algebraic vector space using `TfidfVectorizer(stop_words='english')`. For each term t in document $d \in C$, the term weight $w_{t,d}$ and resulting Cosine Similarity (Overall Match Score) between resume vector $\mathbf{V_R}$ and job description vector $\mathbf{V_{JD}}$ are computed as:

$$w_{t,d} = tf(t, d) \times \log\left(\frac{N}{df(t)}\right) \quad | \quad Similarity(V_R, V_{JD}) = \left[\frac{(V_R \cdot V_{JD})}{(||V_R|| \times ||V_{JD}||)}\right] \times 100\%$$

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

3.5 Set-Theoretic Skill Gap Analysis

The analytics engine partitions the required competency set S_{JD} into matched qualifications (S_{match}) and skill deficiencies (S_{gap}) to compute the quantitative Skill Alignment Score:

$$S_{match} = S_R \cap S_{JD} \quad | \quad S_{gap} = S_{JD} \setminus S_R \quad | \quad \text{Skill Alignment } (\%) = \left(|S_{match}| / |S_{JD}| \right) \times 100\%$$

4. RESULTS

Benchmark evaluations were conducted across representative technical roles to evaluate filtering accuracy, lexical noise suppression, and runtime execution speed.

Table 1: Benchmark Evaluation Across Technical Roles and Candidate Profiles

Test ID	Target Job Role	Candidate Profile	VSM Score	Skill Score	Matched Skills	Skill Gaps
T-01	Backend Python Engineer	Python / Django Developer	**68.40%**	**85.71%**	Python, Django, SQL, PostgreSQL, Git	Docker, Kubernetes
T-02	Machine Learning Engineer	SQL / Excel Data Analyst	**41.25%**	**28.57%**	Python, SQL	Deep Learning, NLP, AWS, CI/CD, Docker
T-03	Full-Stack Web Developer	React / Node.js / C++ Dev	**74.10%**	**100.00%**	JavaScript, TypeScript, React, Node.js, MongoDB	None

Table 2: Impact of Vectorization Strategy on Non-Matching Resume-JD Pairs

Vectorization Method	Stop-Word Filtering	Boilerplate Score	Noise Reduction
`CountVectorizer` (Baseline)	Disabled	48.60%	Baseline
TfidfVectorizer (Proposed)	English Stop-Words	**19.15%**	**-29.45% (False-Positive Drop)**

Runtime Latency Performance: Across 50 consecutive test runs on standard CPU hardware (Intel Core i5, 8GB RAM), end-to-end execution—comprising file parsing, regex entity extraction, VSM matrix transformation, and UI rendering—averaged **312 milliseconds** per application.

5. DISCUSSION AND CONCLUSION

5.1 Interpretation of Results

The empirical findings confirm that coupling a Vector Space Model with set-theoretic skill extraction resolves two major flaws in traditional keyword filtering. First, as shown in Table 2, transitioning from raw term counts to TF-IDF with English stop-word filtering reduced artificial score inflation by 29.45%, ensuring that high match scores reflect

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

genuine domain overlap rather than shared structural phrasing. Second, lookaround regular expression boundaries achieved 100% recall on symbolic competencies (C++ , Node.js , CI/CD), eliminating false negatives common in standard tokenizers.

5.2 Study Limitations

- **Lexical Rigidity:** Exact string matching does not capture semantic equivalence or hierarchical relationships (e.g., recognizing that *"AWS"* satisfies a requirement for *"Cloud Computing"*).
- **Scanned PDF Constraints:** Because pdfminer.six relies on embedded text layers, image-only scanned PDFs cannot be parsed without an Optical Character Recognition (OCR) preprocessing stage.
- **Static Competency Taxonomy:** The predefined dictionary of 37 technical skills requires manual updates to support non-technical industries such as healthcare, finance, or legal recruitment.

5.3 Broader Implications and Future Directions

For recruiters, this framework provides a transparent, auditable triage tool that reduces screening time while preserving explainability. For job seekers, the side-by-side gap visualization transforms opaque ATS rejections into a concrete learning roadmap. Future research will focus on three extensions: (1) integrating dense sentence embeddings (such as Sentence-BERT) alongside TF-IDF for hybrid semantic-lexical matching; (2) incorporating Tesseract OCR for scanned, image-based resumes; and (3) training a domain-adaptive Named Entity Recognition (NER) model to dynamically discover emerging skills.

5.4 Conclusion

This paper presented **An Intelligent Framework for Automated Recruitment Filtering and Skill Gap Analytics via Vector Space Model**. By unifying multi-format document ingestion, boundary-safe regular expression extraction, TF-IDF Cosine Similarity scoring, and set-theoretic gap analytics into an accessible web platform, the framework bridges the transparency gap between hiring teams and applicants. Experimental results validate that VSM-driven architectures deliver rapid, deterministic, and actionable recruitment intelligence without requiring heavy computational infrastructure.

ACKNOWLEDGMENT

The authors express their sincere gratitude to their project guide, **Prof. (Dr.) Samta Gajbhiye**, Professor & Head, Department of Computer Science and Engineering, Shri Shankaracharya Technical Campus, Bhilai, for her invaluable guidance, academic mentorship, and constructive feedback throughout the design and development of this research work.

6. REFERENCES

1. G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, vol. 18, no. 11, pp. 613–620, 1975.
2. G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing & Management*, vol. 24, no. 5, pp. 513–523, 1988.
3. C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge, U.K.: Cambridge University Press, 2008.
4. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proc. NAACL-HLT*, 2019, pp. 4171–4186.
5. F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
6. P. K. Roy, S. S. Chowdhary, and R. Bhatia, "A Machine Learning approach for automation of Resume Recommendation system," *Procedia Computer Science*, vol. 167, pp. 2318–2327, 2020.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)