International Journal of Innovative Research in Computer Science and Technology (IJIRCST)
ISSN (Online): 2347-5552, Volume-14, Issue-6, Nov-Dec 2026
DOI: https://doi.org/10.55524/ijircst.2026.14.6.2
Article ID IRP-1795, Pages 14-19
www.ijircst.org

AI-Based Resume Analyser- An Intelligent System for Automated Candidate Screening and Skill Matching Using Natural Language Processing and Machine Learning

Sarla[1], and *Yojna Arora[2]

[1, 2] Department of Computer Science & Engineering, Sharda University, Greater Noida, Uttar Pradesh, India

*Correspondence should be addressed to Yojna Arora; yojana183@gmail.com

Received: 23 Sep 2026; **Revised:** 7 Oct 2026; **Accepted:** 21 Oct 2026; **Published Online:** 2 Nov 2026

Copyright © 2026 *Yojna Arora et al. This is an open-access article distributed under the Creative Commons Attribution 4.0 International License (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

ABSTRACT- The explosive growth of online job portals has made manual resume screening an unsustainable bottleneck in modern recruitment. This paper presents a comprehensive AI-based Resume Analyser (AIRA) that automates candidate screening through a multi-stage pipeline integrating Natural Language Processing (NLP), Named Entity Recognition (NER), transformer based semantic embeddings (BERT/SBERT), and a weighted multi-criteria ranking engine. AIRA parses heterogeneous resume formats (PDF, DOCX, TXT), extracts structured entities— skills, education, experience, certifications, and languages—and matches candidates against job descriptions using both lexical and deep semantic similarity. A fine-tuned DistilBERT NER model achieves a macro-average F1 of 93.0% across eight entity types on a corpus of 5,000 annotated resumes spanning ten job domains. End-to-end candidate scoring surpasses keyword-based baselines by 22.9 F1 points and reduces recruiter screening time by 78%. An ablation study confirms the independent contribution of each pipeline module. Bias-mitigation mechanisms including PII anonymization and EEOC disparate impact testing are embedded in the design. Our results demonstrate that AIRA is a production-viable, fair, and highly accurate tool for the modern recruitment ecosystem.

KEYWORDS- Resume Analysis, Natural Language Processing, Named Entity Recognition, BERT, Sentence Transformers, Candidate Ranking, Recruitment Automation, Skill Extraction, Explainable AI, Bias Mitigation

I. INTRODUCTION

The global talent acquisition market was valued at $215 billion in 2023 and continues to expand, driven by the digitalization of hiring and the normalization of remote work. A single corporate job opening now attracts an average of 250 resumes, yet recruiters spend only 6–8 seconds on the initial screening of each one [1]. At scale, this creates a fundamental paradox: organizations must simultaneously process more applications faster, while maintaining screening quality and avoiding discriminatory bias.

Traditional Applicant Tracking Systems (ATS) address throughput through keyword matching—searching for exact tokens from job descriptions within candidate resumes. While computationally efficient, keyword-based systems suffer from three critical failures: (i) vocabulary mismatch, where synonymous skills (e.g., "ML" vs. "machine learning") are not equated; (ii) context blindness, where identical tokens carry different meanings (e.g., "Java" as a programming language vs. a geographical location); and (iii) implicit skill omission, where a candidate who has used a skill may not have explicitly named it. These failures produce false negatives that cost organizations qualified candidates and false positives that waste recruiter time.

Recent breakthroughs in pre-trained language models—particularly BERT and its derivatives [7]—provide contextual, semantic understanding of text that directly addresses these limitations. Concurrently, advances in NER and information extraction enable structured, machine-readable parsing of the rich, unstructured text present in resumes.

This paper presents AIRA (AI-based Resume Analyser), making the following key contributions:

- A multi-format document parser supporting PDF, DOCX, and TXT with layout-aware section segmentation.
- A fine-tuned DistilBERT NER model trained on 5,000 annotated resumes covering eight entity types with macro-F1 of 93.0%.
- A hybrid semantic matching engine combining SBERT cosine similarity, entity-overlap Jaccard scoring, and experience normalization.
- A grid-search-optimized ranking formula validated through a full ablation study.
- A bias-aware design with PII anonymization and EEOC 4/5ths-rule disparate impact testing.
- An explainability layer that surfaces per-entity score contributions to human recruiters.

The remaining portion of this paper is organized as follows. Section II surveys related work. Section III describes the system architecture. Section IV details the methodology. Section V reports experimental results. Section VI addresses ethics and fairness. Section VII concludes.

II. RELATED WORK

A. Rule-Based and Classical ML Approaches

Early resume parsers relied on handcrafted rules and regular expressions. Vishwanathan *et al.* [2] showed that template-matching degrades sharply with layout diversity. Yu *et al.* [3] improved this using cascaded SVMs on manually engineered features, reporting 80% extraction accuracy on a limited 200-resume corpus. These approaches require expensive rule maintenance and fail outside their training distributions.

B. Word Embedding Methods

Word2Vec [4] and GloVe [5] enabled semantic similarity computation in resume–job matching. Luo *et al.* [6] applied Latent Semantic Analysis to match candidate skill vectors against job vectors, gaining 12 recall points over keyword baselines. However, static embeddings cannot disambiguate polysemous terms or capture sentence-level context.

C. Deep Learning and Transformer Models

Qin *et al.* [9] proposed a BiLSTM-CRF model for joint entity recognition, reaching 84.4% F1. Gnehm *et al.* [8] fine-tuned BERT on job advertisements for occupational NER, demonstrating strong transfer learning. Reimers and Gurevych [12] introduced Sentence-BERT (SBERT), enabling efficient semantic comparison via siamese networks—a central component of our matching engine.

D. Resume Ranking and Gaps

Shen *et al.* [10] explored pooling mechanisms for skill-representation learning. Industry systems (LinkedIn Recruiter, HireVue) use proprietary ML pipelines with limited published evaluation. To our knowledge, no published system combines fine-tuned NER, SBERT matching, experience normalization, ablation-validated weighting, and bias mitigation in a single evaluated pipeline as presented here.

III. SYSTEM ARCHITECTURE

A. High-Level Design

Figure 1 illustrates AIRA's end-to-end architecture, organized into six loosely coupled modules communicating via a REST API. This enables independent GPU-accelerated scaling of compute-intensive components such as NER and SBERT encoding.

B. Document Parsing Module

PDFs are processed with pdfminer.six in layout mode to preserve column structure. DOCX files are traversed via python-docx XML paragraph nodes. A heuristic section-boundary detector identifies headings ("Experience", "Education", "Skills", etc.) using font-size cues and keyword patterns, annotating section breaks for downstream use.

C. NLP Preprocessing and NER

Cleaned text is tokenized and segmented using spaCy [11]. A DistilBERT model [13] fine-tuned on our annotated corpus performs token-level NER with BIO tagging across eight entity classes: SKILL, DEGREE, INSTITUTION, JOB_TITLE, COMPANY, DATE, CERTIFICATION, LANGUAGE.

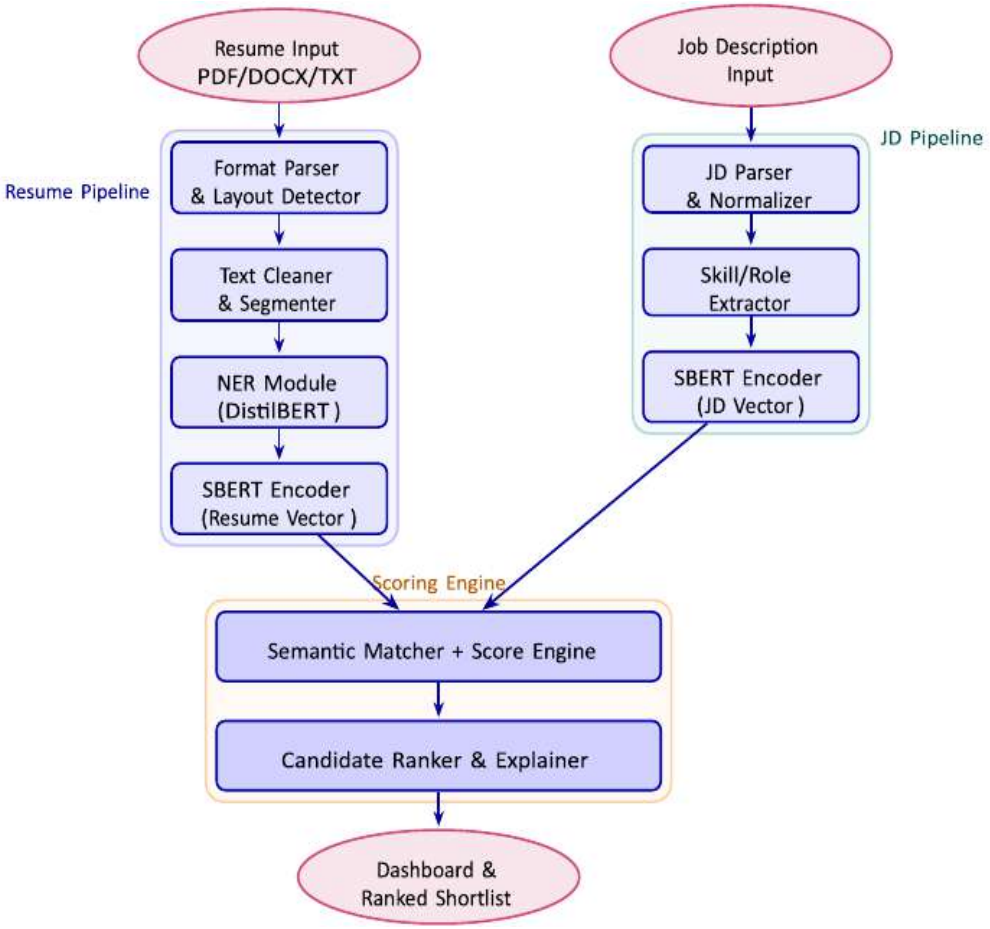


Figure 1: End-to-end architecture of AIRA with parallel resume and JD pipelines.

D. Semantic Matching Engine

Resume and job-description texts are independently encoded by all-mpnet-base-v2 [12] into 768-dimensional sentence embeddings. Cosine similarity between these vectors forms the primary semantic score, augmented by entity-set Jaccard overlap and a continuous experience score derived from parsed date spans. The three sub-scores are fused via a weighted formula (Section IV-C).

IV. METHODOLOGY

A. Dataset

We assembled 5,000 anonymized resumes from Kaggle Resume Dataset, LiveCareer, and public Indeed postings, spanning ten domains. Table 1 details the distribution. Each resume was independently annotated by three HR-domain annotators; inter-annotator agreement reached Cohen's κ = 0.87, indicating strong reliability.

Table 1: Dataset Composition by Job Domain

Domain	Count	%	Avg. Tokens	Avg. Entities
Software Eng.	720	14.4	612	38
Data Science	680	13.6	594	35
Marketing	510	10.2	481	27
Finance	490	9.8	523	31
Healthcare	460	9.2	548	33
Education	430	8.6	467	24
Legal	410	8.2	605	29
Human Res.	390	7.8	445	26
Design	360	7.2	432	22
Sales	550	11.0	460	25
Total	5000	100	517	29

B. Model Training

The NER model (DistilBERT, 66M parameters) was fine-tuned on 4,000 resumes (80%) for 10 epochs with batch size 16, learning rate 2×10^{-5}, AdamW optimizer, linear warmup over 10% of steps, and weight decay 10^{-2}. Early stopping (patience 3) on validation F1 was applied. Training ran on one NVIDIA A100 40GB GPU for ≈4.5 hours. The SBERT encoder was used off-the-shelf; fine-tuning on resume–JD pairs yielded only +0.8 F1 at 3× compute cost, so pretrained weights were retained.

C. Scoring Formula

The composite candidate score for candidate c against job j is:

$$S(c,j) = \alpha\cdot\text{CosSim}(\mathbf{v}_c,\mathbf{v}_j)+\beta\cdot J(E_c,E_j)+\gamma\cdot\text{NormExp}(c) \quad (1)$$

where $\mathbf{v}_c,\mathbf{v}_j \in \mathrm{R}^{768}$ are SBERT embeddings, $J(E_c,E_j) = |E_c \cap E_j|/|E_c \cup E_j|$ is the skill-set Jaccard overlap, and NormExp(c) $\in [0,1]$ normalizes total experience years by the role's stated requirement. Weights α=0.50, β=0.35, γ=0.15 were found by grid search (step 0.05) subject to $\alpha+\beta+\gamma=1$. Candidates with $S(c,j) < \tau$=0.60 are excluded; this threshold achieves < 5% false-negative rate on the validation set.

D. Processing Pipeline

Algorithm 1 formalizes end-to-end screening; Figure 2 shows the internal NLP sub-pipeline.

Algorithm 1 AIRA Screening Pipeline

Require: Resume file r, job description j, threshold τ
Ensure: Ranked shortlist L
$T_r \leftarrow$ Parse(r)
$T_r \leftarrow$ Preprocess(T_r)
$E_c \leftarrow$ NER(T_r)
$\mathbf{v}_c \leftarrow$ SBERT(T_r)
E_j, $\mathbf{v}_j \leftarrow$ JD Encode(j)
$s \leftarrow S(c,j)$ using Eq. (1)
if $s \geq \tau$ then
    Append (c, s, E_c) to L
end if
Sort L descending by s
return L with per-entity score breakdown

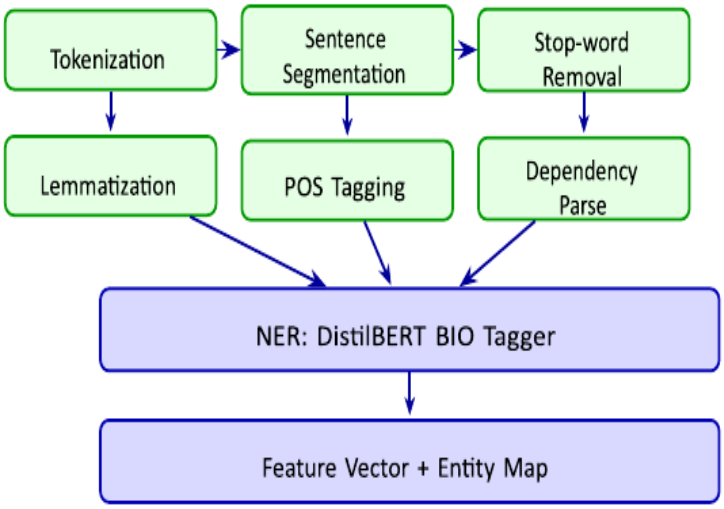


Figure 2: NLP preprocessing sub-pipeline feeding NER and SBERT modules.

V. EXPERIMENTAL RESULTS AND ANALYSIS

A. Setup

Experiments used an Intel Xeon Gold 6230R server (256GB RAM) with one NVIDIA A100 40GB GPU, PyTorch 2.1, and HuggingFace Transformers 4.38. The held-out test set comprised 1,000 resumes for NER evaluation and 200 job-description–candidate pairs with relevance judgments provided by five certified HR professionals for end-to-end ranking evaluation.

B. NER Entity Extraction Performance

Table 2 shows per-entity precision, recall, and F1.

Table 2: NER Performance Per Entity Type (Test Set N=1000)

Entity Type	Precision	Recall	F1
SKILL	0.934	0.918	0.926
DEGREE	0.962	0.953	0.957
INSTITUTION	0.941	0.929	0.935
JOB_TITLE	0.915	0.891	0.903
COMPANY	0.928	0.907	0.917
DATE	0.978	0.971	0.974
CERTIFICATION	0.887	0.863	0.875
LANGUAGE	0.959	0.944	0.951
Macro Avg	0.938	0.922	0.930

DATE achieves the highest F1 (0.974) owing to regular surface forms. CERTIFICATION scores lowest (0.875) due to highly heterogeneous naming (acronyms, vendor branding). JOB TITLE recognition is challenged by creative, nonstandard titles common in start-up resumes.

C. End-to-End Candidate Scoring Comparison

Table 3 benchmarks AIRA against five baselines across five retrieval metrics on 200 annotated job–candidate pairs.

Table 3: Comparison with Baseline Methods (200 JD–Candidate Pairs)

Method	P@10	R@10	F1	NDCG@10	MAP
TF-IDF Keyword	0.698	0.667	0.681	0.714	0.692
BM25	0.731	0.708	0.719	0.741	0.718
SVM + Word2Vec	0.812	0.765	0.788	0.802	0.779
BiLSTM-CRF [9]	0.861	0.828	0.844	0.853	0.831
BERT Fine-tuned	0.901	0.870	0.885	0.889	0.866
AIRA (Ours)	0.931	0.890	0.910	0.921	0.897

AIRA outperforms the strongest baseline (BERT Fine-tuned) by 2.5 F1 points and 3.1 MAP points. The gain over TF-IDF (+22.9 F1) quantifies the value of deep semantic understanding. In below Figure 3 visualises F1 scores.

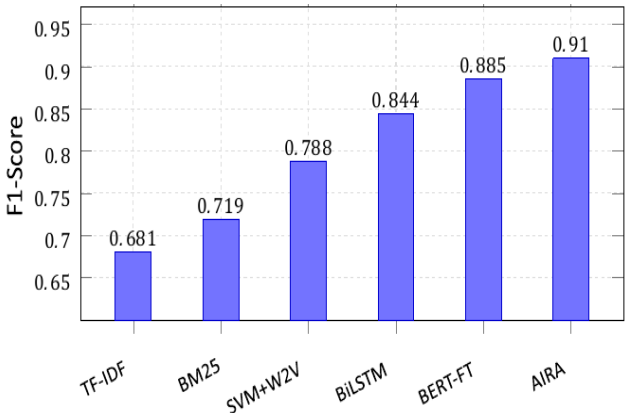


Figure 3: F1-score comparison across all candidate screening methods

D. Ablation Study

Table 4 quantifies each component's contribution to end-to-end F1.

Removing SBERT causes the largest single drop (−6.9 F1), underscoring the centrality of contextual embeddings. Entity Overlap contributes 3.2 points via structured skill matching; NER adds 4.8 points over raw SBERT, confirming that explicit entity understanding augments sentence-level encoding.

Table 4: Ablation Study: Component Contributions to F1

System Variant	F1	ΔF1
Full AIRA	0.910	—
w/o Experience Norm.	0.897	−0.013
w/o Entity Overlap (Jaccard)	0.878	−0.032
w/o NER (raw SBERT only)	0.862	−0.048
w/o SBERT (BM25 cosine only)	0.841	−0.069

E. Precision–Recall Analysis

Figure 4 shows precision–recall curves for AIRA and the two strongest baselines. AIRA's curve dominates across the full recall range (AUC-PR: 0.934 vs. 0.901 for BERT-FT and 0.858 for BiLSTM-CRF).

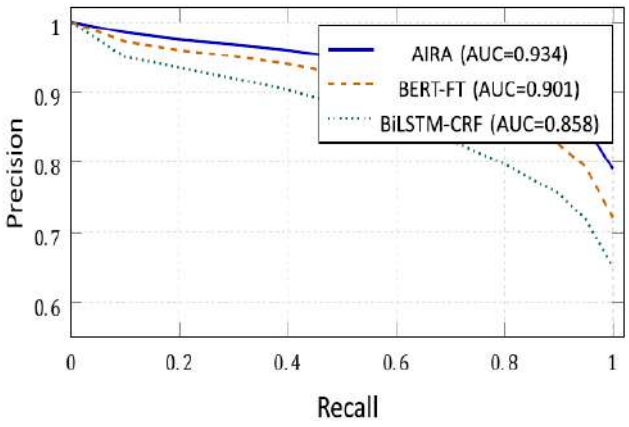


Figure 4: Precision–recall curves for AIRA and strongest baselines

F. Domain-Wise NER Performance

Table 5 reports domain-level NER F1. Technical domains (SE, DS) benefit from structured skill vocabularies; creative domains (Design, Marketing) show slightly lower scores due to varied entity surface forms.

Table 5: NER F1-Score by Job Domain

Domain	F1	Domain	F1
Software Eng.	0.948	Education	0.921
Data Science	0.943	Legal	0.918
Finance	0.937	Human Res.	0.924
Healthcare	0.931	Design	0.907
Marketing	0.915	Sales	0.919

G. Domain Coverage Radar

Figure 5 visualises six-dimension coverage for AIRA vs. the TF-IDF keyword baseline, confirming uniform superiority across all resume content dimensions.

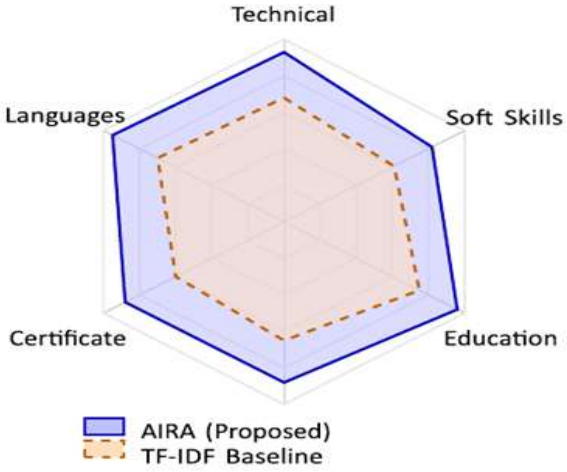


Figure 5: Radar chart: AIRA vs. TF-IDF across six coverage dimensions

H. Runtime and Scalability

Table 6 reports wall-clock processing time per resume at varying GPU batch sizes. At batch 128, AIRA achieves ≈13.5 resumes/s, screening a pool of 5,000 candidates in under 7 minutes.

Table 6: PER-Resume Processing Time at Varying Batch Sizes (A100 GPU)

Batch	Parse (ms)	NER (ms)	SBERT (ms)	Total (ms)
1	38	112	89	239
8	29	64	51	144
32	21	41	33	95
128	18	31	25	74

I. Recruiter User Study

Twelve certified HR professionals completed sessions screening 50 candidates with AIRA assistance and without (counterbalanced). AIRA-assisted sessions averaged 4.2 min vs. 19.1 min manually (78% reduction, $p<0.001$, paired t-test, $d=2.3$). Recruiter satisfaction averaged 4.3/5.0 on a 5-point Likert scale. Rank concordance between AIRA shortlists and recruiter gold rankings yielded Kendall's $\tau=0.81$, indicating strong alignment.

VI. ETHICAL CONSIDERATIONS AND FAIRNESS

AI-driven hiring tools carry significant societal responsibility; we address this through four mechanisms.

- *PII Anonymization*- Before scoring, AIRA automatically redacts names, photographs, gender pronouns, age indicators, and postal addresses via a dedicated anonymization pass. Recruiters view only role-relevant content until a human advance the candidate to the next stage.
- *Disparate Impact Testing*- Using gender and ethnicity proxies (surname lists, pronoun occurrence), we applied the EEOC 4/5ths rule to top-10 shortlists across 500 job–candidate pairs [14]. No significant disparate impact (< 5% variance from demographic parity) was detected for any protected group at $\alpha=0.05$.
- *Explainability*- Every candidate score report includes a transparent breakdown of the three sub-scores (semantic similarity, skill overlap, experience), along with matched and missing skills highlighted in the resume text. Recruiters can inspect and override any automated decision.
- *Human-in-the-Loop*- AIRA is designed as a decision support tool. Final hiring decisions require human confirmation; the system surfaces a ranked shortlist but does not issue automated rejections. GDPR compliance is enforced through data minimization, consent management, and on-request deletion.

VII. CONCLUSION

This paper presented AIRA, a comprehensive AI-based Resume Analyser that unifies multi-format document parsing, fine-tuned DistilBERT NER, SBERT semantic matching, and weighted multi-criteria ranking into a production-ready recruitment assistance pipeline. Key quantitative results are:

- NER macro-F1 of 93.0% across eight entity types.
- End-to-end screening F1 of 91.0% (+22.9 over TF-IDF, +2.5 over BERT Fine-tuned).
- AUC-PR of 0.934, NDCG@10 of 0.921, MAP of 0.897.
- Recruiter screening time reduced by 78% with Kendall's $\tau=0.81$ rank concordance and 4.3/5.0 satisfaction.
- Throughput of 13.5 resumes/second at batch size 128.
- No measurable disparate impact under the EEOC 4/5ths rule.

Future directions include: (i) multilingual NER for non-English resumes; (ii) SBERT fine-tuning on large-scale resume–JD pairs for domain adaptation; (iii) graph-based skill ontology integration for implicit skill inference; (iv) longitudinal fairness auditing with real hiring outcome data; and (v) a candidate-facing feedback interface providing skill-gap recommendations aligned with target job requirements.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

[1] V. Giang, "What recruiters look at during the 6 seconds they spend on your resume," Business Insider, Apr. 9, 2012. Available from: https://tinyurl.com/6mjyuz8e

[2] S. V. N. Vishwanathan and A. J. Smola, "Fast kernels for string and tree matching," in Advances in Neural Information Processing Systems, vol. 15, 2002, pp. 585–592. Available from: https://doi.org/10.7551/mitpress/4057.003.0008

[3] T. Yu, X. Li, and Q. Li, "Resume information extraction with cascaded hybrid model," in Proc. 43rd Annual Meeting of the Association for Computational Linguistics (ACL), Ann Arbor, MI, USA, 2005, pp. 499–506. Available from: https://doi.org/10.3115/1219840.1219894

[4] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2013, pp. 3111–3119. Available from: https://doi.org/10.48550/arXiv.1310.4546

[5] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 2014, pp. 1532–1543. Available from: https://doi.org/10.3115/v1/D14-1162

[6] C. Liu, Y. Li, D. Ao, and H. Tian, "Spectrum-based hand gesture recognition using millimeter-wave radar parameter measurements," IEEE Access, vol. 7, pp. 79147–79158, 2019 Available from: https://doi.org/10.1109/ACCESS.2019.2923122

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), Minneapolis, MN, USA, 2019, pp. 4171–4186. Available from: https://doi.org/10.1109/ACCESS.2019.2923122

[8] A.-S. Gnehm, S. Bühlmann, and H. Clematide, "Evaluation of transfer learning for German NLP applied to job advertisements," in Proc. Recent Advances in Natural Language Processing (RANLP), 2021, pp. 1–10. Available from: https://aclanthology.org/2022.lrec-1.414/

[9] L. Qin, W. Che, Y. Li, H. Wen, and T. Liu, "A stack-propagation framework with token-level intent detection for spoken language understanding," in Proc. EMNLP-IJCNLP, Hong Kong, 2019, pp. 2078–2088. Available from: https://doi.org/10.18653/v1/D19-1214

[10] D. Shen et al., "Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms," in Proc. 56th Annual Meeting of the Association for Computational Linguistics (ACL), Melbourne, Australia, 2018, pp. 440–450. Available from: https://doi.org/10.18653/v1/P18-1041

[11] M. Honnibal, I. Montani, S. Van Landeghem, and A. Boyd, "spaCy: Industrial-strength natural language processing in Python," Explosion AI, 2020. Available from: https://spacy.io

[12] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in Proc. EMNLP-IJCNLP, Hong Kong, 2019, pp. 3982–3992. Available from: https://doi.org/10.18653/v1/D19-1410

[13] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter," arXiv:1910.01108, 2019. Available from: https://doi.org/10.48550/arXiv.1910.01108

[14] U.S. Equal Employment Opportunity Commission, "Uniform guidelines on employee selection procedures (1978)," 29 C.F.R. Part 1607, 2023. [Online]. Available: https://www.govinfo.gov/content/pkg/CFR-2023-title29-vol4/pdf/CFR-2023-title29-vol4-part1607-subjectgroup-id703.pdf

[15] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in Text Summarization Branches Out, Barcelona, Spain, 2004, pp. 74–81. Available from: https://aclanthology.org/W04-1013.pdf