

Original Article

Reliable Evaluation of Automated Resume Screening Through Leakage Control, Information Retrieval, and Bias Auditing

¹VIPIN BIHARI YADAV, ²APOORVA SINGH, ³NIRBHAY TIWARI

¹Department of Computer Science, Babasaheb Bhimrao Ambedkar University (A Central University), Satellite Center Amethi, Uttar Pradesh, India.

²Department of Information Technology, Babasaheb Bhimrao Ambedkar University, Lucknow, Uttar Pradesh, India.

³Department of Computer Science and Information Systems, Shri Ramswaroop Memorial University, Lucknow-Deva Road, Uttar Pradesh, India.

ABSTRACT: Resume screening software is widely used when employers receive far more applications than people can read, yet much of the published evidence for it is hard to trust. Models are often scored against labels derived from the very text they read, ranking quality is rarely measured, and a system is called fair without ever being tested. This paper evaluates resume screening along three lines: leakage control, information retrieval and bias auditing. We use the public Kaggle Resume Dataset (2,481 resumes in 24 occupational categories after cleaning) and compare five lexical classifiers and one Sentence-BERT classifier under stratified five-fold cross-validation, fitting the vectoriser on training folds only. The title line at the top of each resume turns out to leak the label. With it removed, the accuracy of the lexical models falls between 0.571 and 0.652 and macro-averaged F1 never exceeds 0.594. A stacked ensemble is not significantly better than plain logistic regression, and Sentence-BERT with logistic regression is lower still (0.595 accuracy, 0.560 macro-averaged F1). For retrieval we score 24 job-description queries against graded relevance. Latent semantic analysis gives the best NDCG@10 (0.746), ahead of BM25, TF-IDF cosine and Sentence-BERT, but none of these gaps survives correction for multiple comparisons. In a counterfactual audit, swapping male and female names and pronouns changes 0.23% of predictions on average. Because the dataset carries no demographic labels, this cannot establish fairness, and we spell out what a proper audit requires. Our figures sit far below the near-perfect scores often reported on rule-labelled data and offer a more honest baseline for later work.

KEYWORDS: Resume Screening, Leakage Control, Information Retrieval, Text Mining, TF-IDF, BM25, Latent Semantic Analysis, Sentence-BERT, NDCG, Bias Auditing, Counterfactual Fairness Audit, Reproducibility.

1. INTRODUCTION

Recruitment has become, to a large extent, an exercise in information overload. A single vacancy at a large employer can attract thousands of applications and reading them manually is slow, costly and inconsistent: two reviewers frequently disagree about the same candidate. Automated resume screening, in which software parses, classifies or ranks applicants before a person reads them, has consequently become a routine first stage of hiring, and vendors promote it as both faster and fairer than manual review [1].

Such claims carry real risk, and regulators have taken notice. The EU Artificial Intelligence Act classifies systems used for recruitment and selection as high-risk [2], and New York City's Local Law 144 of 2021 requires a bias audit before an automated employment decision tool may be used [3]. The concern is not hypothetical: Amazon abandoned an experimental recruiting engine after it was found to penalise resumes that mentioned women's activities [4], and reviews of hiring algorithms show how much harm an unchecked system can do [5], [6].

Responsible deployment therefore depends on trustworthy evidence about how well these systems actually work, yet the published evidence is uneven, and three weaknesses recur. The first is circular labelling. In some studies an eligibility label is produced by a fixed rule applied to the resume text, and a classifier is then trained to predict that label from the same text. A score near 100% in that setting shows only that the model has recovered the rule; it says nothing about whether the model can judge a candidate. This is a form of target leakage, a well-documented source of over-optimistic results in data-driven research [7], [8]. The second weakness is that systems meant to rank candidates are seldom measured with ranking metrics, so that accuracy on a labelling task stands in for ranking quality. The third is that claims about fairness and explainability are made without any group-level or counterfactual test.

This paper addresses all three weaknesses on a dataset that anyone can download, the public Kaggle Resume Dataset [9]. We treat leakage control, information retrieval and bias auditing as parts of a single evaluation, because a system can pass one of these checks and still fail another. The study makes five contributions. First, it provides a classification benchmark with leakage controls, in which the TF-IDF vectoriser is fitted on training folds only, duplicate resumes are removed, and the amount of label information carried by the title line of each resume is measured. Second, it provides a ranking benchmark for job-description queries with graded relevance, which compares BM25, TF-IDF cosine similarity, latent semantic analysis and Sentence-BERT on four standard metrics (NDCG@10, P@10, MAP and MRR), with bootstrap confidence intervals and paired significance tests. Third, it offers term-level explanations, namely an exact additive decomposition of cosine scores and the most informative terms of each class. Fourth, it reports a counterfactual name-and-pronoun audit, together with a plain statement of what such an audit can and cannot show. Fifth, it specifies a complete protocol (fixed seed, fold design, hyper-parameters and queries) that allows every reported number to be reproduced; the code is available from the corresponding author on request.

The scope of the study is deliberately limited. The resumes are private-sector examples, so the findings do not carry over to government hiring. The corpus contains no job descriptions and no protected attributes; ranking relevance is therefore derived from occupational categories, and fairness is probed only through counterfactual edits to the text. Sentence-BERT is used in both tasks as a frozen, general-purpose encoder (Sections 6.1 and 6.2), and fine-tuned encoders and cross-encoder re-rankers are left for future work.

The study is organised around four research questions. RQ1: how much of the accuracy reported for resume classification survives once label leakage is controlled? RQ2: do a stacked ensemble and a frozen sentence encoder outperform a plain linear model when the comparison is supported by a significance test? RQ3: which retrieval method (BM25, TF-IDF cosine, latent semantic analysis or Sentence-BERT) ranks resumes best for a job-description query under graded relevance? RQ4: what can a counterfactual name-and-pronoun audit show, and what can it not show, when no demographic labels are available?

The remainder of the paper is organised as follows. Section 2 reviews related work, Section 3 describes the data and the title-line leakage, Section 4 defines the methods and metrics, and Section 5 sets out the experimental protocol. Classification and ranking results are reported in Section 6, term-level explanations in Section 7 and the counterfactual audit in Section 8. Sections 9 and 10 discuss limitations and conclude.

2. RELATED WORK

Automated resume screening. Computational approaches to resume screening fall broadly into three groups: keyword and rule-based matching of resumes against stated requirements, supervised classification of resumes into occupational categories or suitability classes, and ranking of resumes against a job description. The first is transparent but brittle, because it depends on exact vocabulary; the second and third are statistical and are the focus of this paper. A difficulty common to all of them is that real hiring outcomes are private, so public studies must rely on proxy labels, such as the occupational category under which a resume was filed. Proxy labels make evaluation possible, but they also mean that a reported score measures agreement with the proxy and not the quality of a hiring decision, a limitation we return to in Section 9.

Text classification baselines. Linear models on sparse lexical features remain strong baselines for text categorisation. Joachims [10] argued that support vector machines suit text because document vectors are high-dimensional and sparse and most categorization problems are close to linearly separable, and Rennie et al. [11] proposed complementing naive Bayes to correct the poor assumptions of the multinomial model, particularly for skewed class distributions. Random forests [12] offer a non-linear alternative. All three families are included here, together with logistic regression, so that the comparison covers the models most often used for this task. Because resume categories are imbalanced (Table 1), we report macro-averaged F1 next to accuracy; Sokolova and Lapalme [13] analyse how such performance measures behave under class imbalance.

Lexical, latent and dense representations. TF-IDF weighting [14] and the BM25 ranking function [15], which grew out of the probabilistic retrieval work of Robertson and Walker [16], are decades old, yet they remain strong baselines for matching documents to a query; Manning et al. [17] give a standard textbook account. Latent semantic analysis (LSA) [18] goes a step further: it applies a truncated singular value decomposition to the term–document matrix, which lets it connect documents that share no exact words. Transformer encoders such as BERT [19] and Sentence-BERT [20] instead map a text to a dense vector that can be compared by cosine similarity, and the all-mpnet-base-v2 model used here is built on MPNet [21]. Dense encoders are not uniformly superior, however. The BEIR benchmark [22] reported that BM25 is a robust baseline that several neural retrievers did not consistently beat in zero-shot settings, which is why we evaluate a frozen dense encoder alongside the lexical rankers and not instead of them.

Ensembles and statistical comparison. Stacked generalisation [23] trains a second-level model on the out-of-fold predictions of several base learners, and ensembles are often reported to outperform single models. Such a claim is credible only when it is backed by a statistical comparison. When two classifiers are scored on the same items, McNemar's test [24] is the standard

paired comparison, and Dietterich [25] found it to have a low type I error among the tests he examined. For comparing methods across several data sets, Demšar [26] recommends the Wilcoxon signed-rank test [27], together with Holm's step-down procedure [28] to control the family-wise error rate when several comparisons are made. We follow this practice for per-query ranking scores and build confidence intervals by bootstrap resampling [29]. Smucker et al. [30] showed that the Wilcoxon test can disagree with permutation-based tests in retrieval evaluation, so we treat uncorrected p-values close to the 0.05 threshold as inconclusive. Ranking quality itself is reported with cumulated-gain metrics [31].

Leakage and reproducibility. Leakage, the use of information during training or preprocessing that would not be available at prediction time, is a well-documented cause of over-optimistic results. Kaufman et al. [7] formalise the concept and discuss how to detect and avoid it, and Kapoor and Narayanan [8] document leakage and the resulting reproducibility failures across machine-learning-based research in many fields, including leakage introduced by preprocessing steps applied before the train–test split. Two forms matter directly for resume screening: label leakage, in which an input field is a near copy of the target (the title line, Section 3.2), and preprocessing leakage, in which a vectoriser is fitted on the whole corpus before cross-validation (Section 5.1).

Bias in hiring algorithms. Bogen and Rieke [5] and Raghavan et al. [6] examine how hiring algorithms can absorb bias and how far vendors' claims deserve trust, and Sánchez-Monedero et al. [1] analyse the claims of automated-hiring vendors from technical and legal perspectives. Mehrabi et al. [32] survey where bias comes from and how fairness has been defined. Work on text models shows that bias can enter even without explicit demographic fields. De-Arteaga et al. [33] found gender-related gaps in the true-positive rates of occupation classifiers trained on online biographies, even after names and pronouns were removed, and Caliskan et al. [34] showed that word embeddings learned from text reproduce human-like biases, which is why dense encoders need an audit of their own. Our counterfactual test follows the audit-study tradition of Bertrand and Mullainathan [35], who sent out otherwise similar resumes under different names. Group-level criteria include demographic parity, the four-fifths rule [36] and equalised odds [37]; Barocas et al. [38] discuss where each of them breaks down.

Gap addressed here. In most of the work we reviewed, these strands are treated separately. Classification studies seldom report leakage checks or significance tests, retrieval studies seldom use resumes with graded relevance, and fairness studies seldom say what an audit can prove when demographic data are missing. We bring the three together in one reproducible protocol on a single public dataset.

3. DATA

3.1. RESUME CORPUS

We use the public Kaggle Resume Dataset [9], which its author describes as resume examples collected from the website livecareer.com. The archive holds 2,484 resumes as PDF files in 24 category folders, together with a table, Resume.csv, with the columns ID, Resume_str (plain text), Resume_html and Category. Only Resume_str and Category are used. After removing two exact duplicate texts and one resume with fewer than 20 words, 2,481 documents remain. Table 1 and Fig. 1 summarise the corpus. It contains no job descriptions, no hiring outcomes and no demographic fields.

TABLE 1 Corpus Summary after Cleaning

Property	Value
Resumes (raw → used)	2,484 → 2,481
Categories	24
Category size (min – max)	22 – 120 (BPO smallest; INFORMATION-TECHNOLOGY largest)
Majority-class rate (constant classifier)	0.048
Words per resume: mean (SD), median	812 (371), 757
Resumes with an identifiable title line	100%

Majority-class rate and all other figures are computed by our evaluation code.

(Source: Self)

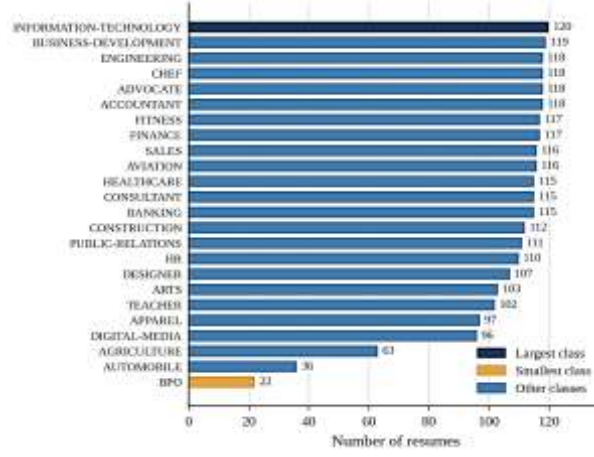


FIGURE 1 Number of resumes per category after cleaning. The largest class has 120 resumes and the smallest only 22

(Source: Self)

3.2. TITLE-LINE LEAKAGE

Every resume begins with a header line that gives a job title, for example “HR ADMINISTRATOR/MARKETING ASSOCIATE”. The category is the occupation under which the resume was filed, so this line is close to a copy of the label. To measure how much it helps, we evaluate every model twice: once on the full text and once with the title removed. The header is taken to be everything before the first run of five or more spaces. The title-removed setting is our main result.

3.3. PREPROCESSING

Text is lower-cased. URLs and e-mail addresses are deleted, and anything other than letters, digits, “+” and “#” becomes a space. We do not remove stop words or lemmatize. Sub linear TF-IDF already lowers the weight of very common words, and both steps would disturb the word order that sentence encoders rely on.

4. METHODS

Our evaluation has three parts that mirror the title of this paper. Leakage control decides how the data are split and which text a model may see (Sections 3.2 and 5.1). The information-retrieval part ranks resumes against job-description queries with graded relevance (Sections 4.2, 4.4 and 5.2). Bias auditing probes the classifier with counterfactual name-and-pronoun edits and states the limits of that probe (Sections 5.3 and 8). The subsections below define the representations, models and metrics that all three parts share.

4.1. PROBLEM FORMULATION

Let $D = \{d_1, \dots, d_N\}$ be the set of resumes and K the number of categories, with labels $i \in \{1, \dots, K\}$. Task A (classification) asks for a function $f: D \rightarrow \{1, \dots, K\}$ that predicts the category of a resume. Task B (ranking) takes a job-description query q and orders D by a scores (q, d) , judged against a graded relevancere $(q, d) \in \{0, 1, 2\}$. The category label comes from the data source. We do not compute it from the resume text with any rule.

4.2. LEXICAL, LATENT AND DENSE REPRESENTATIONS

Let N be the number of documents, $df(t)$ the number of documents that contain term t , and $tf(t, d)$ the count of t in d . We use the smoothed, sublinear TF-IDF weight over unigrams and bigrams, as implemented in scikit-learn [39], followed by Euclidean normalization:

$$u_{t,d} = (1 + \ln tf(t, d)) \left(\ln \frac{1+N}{1+df(t)} + 1 \right) \quad (1)$$

$$w_{t,d} = \frac{u_{t,d}}{\sqrt{\sum_{t'} u_{t',d}^2}} \quad (2)$$

Every document vector then has unit length. The score of a document for a query is their cosine similarity,

$$s(q, d) = \frac{q \cdot d}{\|q\| \|d\|} = \sum_{t=1}^V q_t d_t \quad (3)$$

Where V is the vocabulary size, the last form holds when both vectors have unit length. It also gives an exact additive explanation: term t contributes $q_t d_t$ to the score (Section 7).

BM25 [15] sums over the terms of the query:

$$\text{BM25}(q, d) = \sum_{t \in q} \text{IDF}(t) \cdot \frac{\text{tf}(t, d) (k_1 + 1)}{\text{tf}(t, d) + k_1 (1 - b + b \frac{|d|}{\text{avgdl}})} \quad (5)$$

$$\text{IDF}(t) = \ln \left(1 + \frac{N - n_t + 0.5}{n_t + 0.5} \right)$$

Here n_t is the document frequency of t , $\text{tf}(t, d)$ its count in d , $|d|$ the length of the document and avgdl the mean document length. We keep the common defaults $k_1 = 1.5$ and $b = 0.75$.

Latent semantic analysis [18] factorizes the TF-IDF matrix $X \in \mathbb{R}^{N \times V}$ with a truncated singular value decomposition. Documents and queries are projected into the resulting r -dimensional space, where they are again compared by cosine similarity. The projected, unit-length vector of a document is written $d^{(r)}$:

$$X \approx U_r \Sigma_r V_r^T, \quad d^{(r)} = \frac{d V_r}{\|d V_r\|} \quad (5)$$

We set $r = 200$.

For Sentence-BERT [20], we use the all-mpnet-base-v2 encoder, which returns 768-dimensional vectors. To get around the encoder's input limit, each resume is cut into chunks $z_1, \dots, z_m$ of 200 words, every chunk is encoded as a unit vector $e(z_j)$, and the document embedding is the re-normalized mean:

$$e(d) = \frac{\frac{1}{m} \sum_{j=1}^m e(z_j)}{\left\| \frac{1}{m} \sum_{j=1}^m e(z_j) \right\|} \quad (6)$$

4.3. CLASSIFIERS

We compare five lexical classifiers: multinomial logistic regression ($C = 10$), a linear support vector machine ($C = 0.5$), complement naive Bayes (smoothing parameter $\lambda = 0.3$), a random forest (200 trees on 5,000 unigram features) and a stacked ensemble. Stacking [23] fits a meta-learner g on the out-of-fold outputs of M base learners:

$$\hat{y} = g \left(f_1(x), \dots, f_M(x) \right) \quad (7)$$

Here the base learners are logistic regression, the linear SVM and complementary naive Bayes, and g is a logistic regression. The out-of-fold outputs come from 3-fold cross-validation inside each training fold. A sixth model feeds the frozen Sentence-BERT embeddings of the title-removed text into a logistic regression. Hyperparameters were fixed in advance and never tuned on test folds.

4.4. EVALUATION METRICS

Classification is scored by accuracy and macro-averaged F1. For class c with precision P_c and recall R_c ,

$$F_{1,c} = \frac{2P_c R_c}{P_c + R_c}, \quad \text{Macro-F1} = \frac{1}{K} \sum_{c=1}^K F_{1,c} \quad (8)$$

Macro-F1 is reported next to accuracy because the categories are imbalanced (Table 1). For ranking, let rel_i be the relevance grade of the document at rank i . The cumulated-gain metrics of Järvelin and Kekäläinen [31] discount the gain of each document by its rank. We use the common exponential-gain form:

$$\text{DCG}@k = \sum_{i=1}^k \frac{2^{\text{rel}_i - 1}}{\log_2(i+1)}, \quad \text{NDCG}@k = \frac{\text{DCG}@k}{\text{IDCG}@k} \quad (9)$$

Where $\text{IDCG}@k$ is the DCG of the ideal ordering, precision at k is the share of grade-2 documents among the top k . Average precision (AP) and reciprocal rank (RR) of a query are

$$\text{AP} = \frac{1}{n_2} \sum_{i=1}^N P @ i \cdot \mathbf{1}[\text{rel}_i = 2] \quad (10)$$

$$\text{RR} = \frac{1}{\text{rank}_{\text{first}}} \quad (11)$$

Where n_2 the number of grade-2 documents is, $\text{rank}_{\text{first}}$ is the rank of the first one, and $P@i$ is the precision at rank i . MAP and MRR are the means of AP and RR over queries. We use $k = 10$.

4.5. STATISTICAL INFERENCE

We compute percentile bootstrap confidence intervals with 2,000 resamples [29], over resamples for classification accuracy and over queries for ranking metrics. For two classifiers scored on the same cross-validated predictions, let n_{10} be the number of resumes that only the first classifier gets right and n_{01} the number that only the second gets right, with $n_d = n_{10} + n_{01}$. The exact two-sided McNemar test [24] then gives

$$p = \min \left\{ 1, 2 \sum_{i=0}^{\min(n_{10}, n_{01})} \binom{n_d}{i} \left(\frac{1}{2} \right)^{n_d} \right\} \quad (12)$$

For ranking, we compare the per-query NDCG@10 of two methods with the paired Wilcoxon signed-rank test [27]. When h comparisons are made within one metric, we apply Holm's step-down correction [28]. The p-values are sorted in ascending order, and the i -Th smallest is declared significant only if

$$p_{(i)} \leq \frac{\alpha}{h-i+1} \quad (13)$$

Moreover, all smaller p-values were also significant, with $\alpha = 0.05$. Uncorrected p-values close to 0.05 are treated as inconclusive.

5. EXPERIMENTAL PROTOCOL

5.1. CLASSIFICATION

We use stratified five-fold cross-validation with a fixed seed (42). In each fold the TF-IDF vocabulary and weights are fitted on the four training folds only and then applied to the held-out fold, so no information from held-out resumes reaches the vectoriser. This closes the leakage route found in pipelines that fit the vectoriser before splitting. The Sentence-BERT encoder is frozen and unsupervised, which means only its logistic-regression head is trained inside the folds. Results are reported as mean $\pm$ standard deviation over folds, for the full text and for the title-removed text (Sentence-BERT is run on the title-removed text only).

5.2. RANKING

The corpus has no job descriptions, so we wrote one short, realistic job-description query for each of the 24 categories. The queries name the skills, duties and tools typical of the role and are available together with the code from the corresponding author. Ranking is run over the title-removed text so that a query cannot simply match a title. Relevance is graded. A resume gets 2 when its category equals the category of the query, 1 when it belongs to a related family defined in advance (for example ACCOUNTANT, FINANCE and BANKING, or SALES, BUSINESS-DEVELOPMENT and CONSULTANT; the families were fixed in the code before any scoring), and 0 otherwise. Random ordering serves as a sanity check.

5.3. COUNTERFACTUAL AUDIT

A classifier is fitted on 80% of the resumes (stratified) and audited on the other 20%, which is 497 resumes. For each of eight matched pairs of conventionally male- and female-coded full names, every held-out resume is submitted twice. Once it carries the male name and the pronoun ‘‘he’’, and once the female name and ‘‘she’’, both injected into the text. Let $f(\cdot)$ be the predicted label, $\pi(\cdot)$ the largest predicted probability, and $\oplus$ the injection of a name-and-pronoun variant, with a^{male} and a^{female} the two variants. The flip rate and the confidence gap are

$$\text{FR} = \frac{1}{n} \sum_{i=1}^n \mathbf{1} [f(x_i \oplus a^{\text{male}}) \neq f(x_i \oplus a^{\text{female}})] \quad (14)$$

$$\Delta\pi = \frac{1}{n} \sum_{i=1}^n |\pi(x_i \oplus a^{\text{male}}) - \pi(x_i \oplus a^{\text{female}})| \quad (15)$$

We also report the mean absolute difference in cosine score (Eq. 3) between the two variants for four representative queries. A scorer that ignores names and pronouns should give $\text{FR} \approx 0$ and negligible score gaps.

Where demographic labels exist, for instance in an employer's own applicant records, a deployable audit should add group metrics: the demographic parity difference, the disparate-impact ratio (the four-fifths guideline asks for a ratio of at least 0.8 [36]) and equalized odds, which asks for equal true-positive and false-positive rates across groups [37]. None of these can be computed on the present dataset, so we describe them only in words.

6. RESULTS

6.1. CLASSIFICATION

Table 2 and Fig. 2 give the cross-validated results. Without the title line, the five lexical classifiers reach accuracy of 0.571–0.652 and macro-F1 of 0.499–0.594, against 0.048 for a constant classifier. On the full text the best macro-F1 is 0.630 (linear SVM). Removing the title lowers macro-F1 by 0.01–0.04 depending on the model, most for the random forest and the linear SVM. The title therefore carries some label information, but most of the signal stays in the body of the resume. The bootstrap 95% confidence interval for the accuracy of logistic regression on the title-removed text is 0.626–0.662. Logistic regression serves as the reference model below because it has the highest mean macro-F1 on that text.

TABLE 2 Five-Fold Cross-Validated Classification Results (Mean $\pm$ SD over Folds)

Model	Acc. (full text)	Macro-F1 (full text)	Acc. (title removed)	Macro-F1 (title removed)
Logistic regression	0.671 $\pm$ 0.015	0.618 $\pm$ 0.016	0.644 $\pm$ 0.017	0.594 $\pm$ 0.017
Linear SVM	0.684 $\pm$ 0.016	0.630 $\pm$ 0.016	0.650 $\pm$ 0.017	0.592 $\pm$ 0.016
Complement naive Bayes	0.584 $\pm$ 0.015	0.510 $\pm$ 0.013	0.571 $\pm$ 0.014	0.499 $\pm$ 0.013
Random forest	0.690 $\pm$ 0.023	0.626 $\pm$ 0.018	0.651 $\pm$ 0.017	0.586 $\pm$ 0.018
Stacking (LR+SVM+NB $\rightarrow$ LR)	0.686 $\pm$ 0.011	0.625 $\pm$ 0.007	0.652 $\pm$ 0.012	0.593 $\pm$ 0.007
Sentence-BERT + LR	—	—	0.595 $\pm$ 0.024	0.560 $\pm$ 0.023

Sentence-BERT + LR uses frozen all-mpnet-base-v2 embedding's of the title-removed text only, so the full-text columns are empty.
(Source: Self)

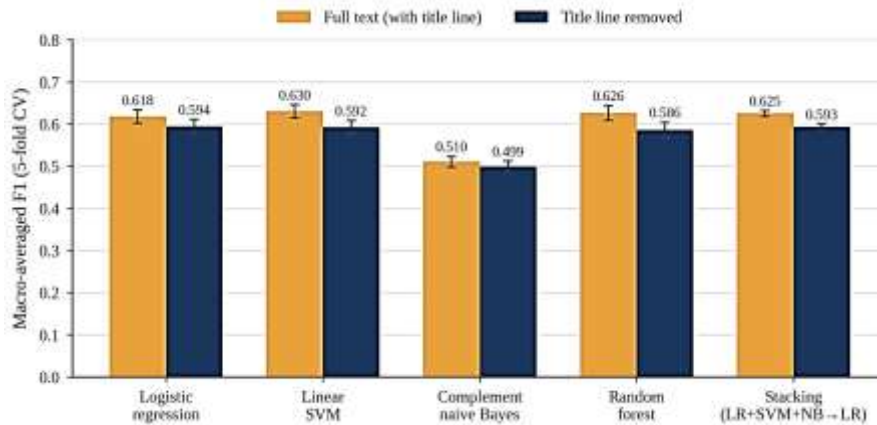


FIGURE 2 Macro-F1 of the Five Lexical Classifiers with and without the Title Line (Error Bars: SD over Five Folds)

(Source: Self)

Sentence-BERT with logistic regression reaches 0.595 ± 0.024 accuracy and 0.560 ± 0.023 macro-F1. That is about 0.05 and 0.03 below logistic regression on TF-IDF, though still above complementary Naive Bayes. We did not store the fold-level predictions of this model, so no paired test was run for it, and we describe the gap without calling it significant.

Paired exact McNemar tests against logistic regression (Table 3) show that complement naive Bayes is significantly worse ($p < 0.001$). The linear SVM ($p = 0.261$), the random forest ($p = 0.444$) and the stacked ensemble ($p = 0.132$) cannot be told apart from it. In other words, the ensemble brings no demonstrable gain over a single linear model here, which runs against claims of ensemble superiority made without a significance test.

TABLE 3 Exact McNemar Test Versus Logistic Regression (Title Removed)

Compared model	Only LR correct	Only other model correct	p-value
Linear SVM	60	74	0.261
Complement Naive Bayes	261	80	< 0.001
Random forest	184	200	0.444
Stacking (LR+SVM+NB $\rightarrow$ LR)	62	81	0.132

For complement naive Bayes the exact p-value is about 1.6×10^{-23} .

(Source: Self)

Performance differs sharply across categories (Fig. 3). Recall runs from 0.045 for BPO ($n = 22$; one resume out of 22 recovered) to 0.883 for INFORMATION-TECHNOLOGY ($n = 120$), a spread of 0.838. The smallest classes (BPO, AUTOMOBILE, and AGRICULTURE) and the broad, overlapping ones (CONSULTANT, ARTS) do worst. The most frequent confusions (Table 4) are FINANCE $\rightarrow$ ACCOUNTANT, ARTS $\rightarrow$ TEACHER, CONSULTANT $\rightarrow$ INFORMATION-

TECHNOLOGY, ADVOCATE → HEALTHCARE and APPAREL → SALES. Most of these are genuine overlaps between occupations, which shows that the category label is only a noisy stand-in for suitability. Accuracy by resume-length quartile falls from 0.663 for the shortest quartile to 0.630 for the longest, so length has only a modest effect.

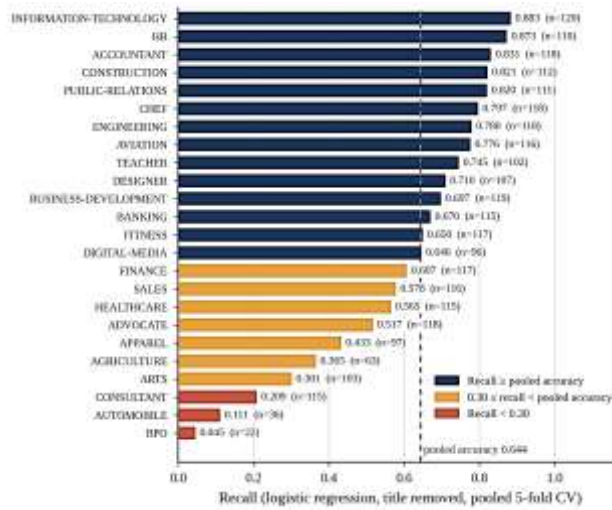


FIGURE 3 Per-Category Recall of Logistic Regression on Title-Removed Text (Pooled Predictions from Five-Fold CV). Bars are Coloured by Recall Relative to the Pooled Accuracy (Dashed Line)

(Source: Self)

TABLE 4 Most Frequent Confusions of Logistic Regression (Title Removed, Pooled over Five Folds)

True category	Predicted as	Resumes	Share of true class
Finance	Accountant	26	22.2%
Arts	Teacher	22	21.4%
Consultant	Information-Technology	20	17.4%
Advocate	Healthcare	18	15.3%
Apparel	Sales	17	17.5%
Accountant	Finance	11	9.3%

(Source: Self)

6.2. RANKING JOB-DESCRIPTION QUERIES

Table 5 and Fig. 4 report ranking quality over the 24 queries. LSA obtains the highest mean on all four metrics (NDCG@10 of 0.746), followed by BM25 (0.705), TF-IDF cosine (0.687) and Sentence-BERT (0.629). Random ordering scores 0.065, so every method carries real signal. The confidence intervals, however, overlap widely. In paired Wilcoxon tests on per-query NDCG@10, LSA against BM25 gives $p = 0.086$, and LSA against TF-IDF gives $p = 0.030$. With Holm's correction over the three comparisons (BM25, TF-IDF, random), the second of these rises to about 0.060 and no longer passes the 0.05 threshold. The comparison with random ordering is clearly significant ($p < 0.001$). Sentence-BERT was not part of this test family, so its lower scores should be read as descriptive. With only 24 queries the methods cannot be separated reliably, and the ordering is suggestive at best.

TABLE 5 Ranking Quality over 24 Job-Description Queries (Mean, with 95% Bootstrap CI over Queries Below)

Method	NDCG@10	P@10	MAP	MRR
BM25	0.705 [0.599, 0.803]	0.654 [0.529, 0.767]	0.449 [0.364, 0.531]	0.816 [0.691, 0.924]
TF-IDF cosine	0.687 [0.575, 0.789]	0.638 [0.517, 0.750]	0.454 [0.368, 0.536]	0.843 [0.717, 0.962]
LSA cosine ($r = 200$)	0.746 [0.632, 0.847]	0.692 [0.563, 0.808]	0.482 [0.387, 0.572]	0.853 [0.734, 0.958]
Sentence-BERT (all-mpnet-base-v2)	0.629 [0.515, 0.738]	0.583 [0.467, 0.696]	0.387 [0.306, 0.468]	0.738 [0.590, 0.873]
Random	0.065 [0.038, 0.098]	0.046 [0.025, 0.071]	0.043 [0.039, 0.047]	0.149 [0.083, 0.246]

Relevance: 2 = same category, 1 = related family, 0 = other (Section 5.2). Best mean per column in bold.

(Source: Self)

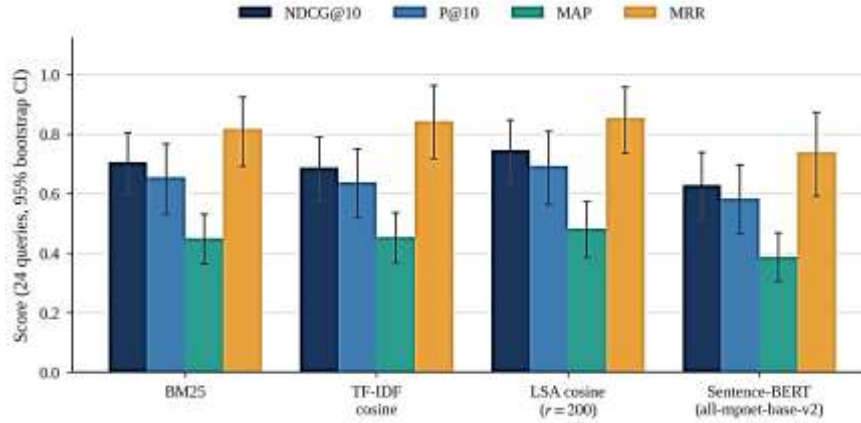


FIGURE 4 Ranking Metrics with 95% Bootstrap Confidence Intervals over 24 Queries (the Random Baseline is Omitted; Its NDCG@10 is 0.065)

(Source: Self)

The per-query scores explain part of this picture. LSA reaches a perfect NDCG@10 on nine of the 24 queries, against five for BM25 and three for TF-IDF. It scores higher than BM25 on 13 queries and lower on seven (16 and five against TF-IDF). A few queries are hard for every method: ADVOCATE stays between 0.08 and 0.14, and SALES between 0.23 and 0.37. BPO is best served by BM25 (0.262) and worst by LSA (0.022). A method can therefore win on average and still lose clearly on individual occupations. The very low ADVOCATE scores match the confusion with HEALTHCARE seen in Table 4, because the same vocabulary of patients, families and care appears in both categories.

Sentence-BERT finishing last calls for some explanation. One possible reason is that a single mean-pooled vector blurs a long resume that covers many topics, while the exact skill terms in a query are matched directly by the lexical methods. Another is that a general-purpose encoder has not been tuned to separate neighbouring occupations. We have not tested either explanation. Note also that MAP is far lower than P@10 for every method, because MAP rewards finding all of the roughly 100 same-category resumes and not only the first ten. Since relevance comes from category labels, a high score means that a method retrieves resumes of the right occupation. It does not mean that the top-ranked person is the best candidate for a vacancy.

7. EXPLAINABILITY

Two kinds of explanation come almost for free. First, for the linear classifier, the largest positive coefficients of each class list its most telling terms (Table 6). Most make sense (chef: kitchen, culinary, menu; teacher: classroom, lesson, students). Some, however, expose quirks of the data, such as “navy” for AVIATION and “child” and “patient” for ADVOCATE. These reflect who happened to post resumes rather than the occupation itself.

TABLE 6 Highest-Weight Terms of Selected Classes (Logistic Regression, Title Removed)

Category	Top Terms (descending coefficient)
Information-Technology	information technology, technology, it, technology specialist, hardware, network, information, security
Healthcare	healthcare, medical, care, patient, clinical, patients, health, physicians
Chef	chef, kitchen, food, culinary, cooking, menu, restaurant, cook
Teacher	teacher, classroom, teaching, lesson, learning, students, children, teachers
Aviation	aviation, aircraft, flight, navy, safety, maintenance, repair, of aviation
Advocate	advocate, health, child, care, advocacy, patient, advocate company, families

(Source: Self)

Second, because the cosine score is a sum over terms (Eq. 3), every ranking can be broken down exactly. If $\gamma_t = q_t d_t$ is the contribution of term t , the share of the score explained by a set T of terms is

$$\gamma_t = q_t d_t, \quad S_T = \frac{\sum_{t \in T} \gamma_t}{s(q, d)} \quad (16)$$

For the INFORMATION-TECHNOLOGY query, the top-ranked resume (category INFORMATION-TECHNOLOGY) has a score of 0.105. Its eight largest term contributions are “network administration” (0.0147), “systems support” (0.0102), “java” (0.0102), “network” (0.0085), “software and” (0.0082), “sql” (0.0079), “help desk” (0.0060) and “and network” (0.0058), which together account for 68% of the score. A reviewer can thus see which skills and phrases drove a given rank. The

explanation is faithful for the lexical scorer, but it does not carry over to dense embeddings, which would need added attribution methods.

8. COUNTERFACTUAL NAME-AND-PRONOUN AUDIT

Swapping a male name and pronoun for a female one changes on average 0.23% of predictions (at most 0.60% for any single name pair) and moves the cosine score by about 1.5×10^{-5} (Table 7). The effect is tiny, which is what one expects from a bag-of-words model, because personal names are rare tokens that carry little weight.

TABLE 7 Counterfactual Audit of the TF-IDF Logistic-Regression Classifier

Quantity	Value
Held-out resumes audited	497
Matched male/female name pairs	8
Prediction flip rate FR (mean over pairs)	0.23%
Prediction flip rate FR (largest pair)	0.60%
Mean $\Delta\pi$ in top-class probability	0.0019
Mean absolute change in cosine score (4 queries)	1.51×10^{-5}

(Source: Self)

Limits of interpretation. This result must not be read as evidence that the system is fair. (i) The model was trained to predict occupational categories, not past hiring decisions, so it holds no learned hiring bias that could be exposed. (ii) Only two signals, first name and pronoun, and one binary attribute, gender, were tested; ethnicity, age, disability, nationality and proxies such as institution names, employment gaps, addresses and extracurricular activities were not. (iii) Eight name pairs are a small sample. (iv) The test covers the TF-IDF classifier and lexical scoring only. Dense embeddings can carry demographic information in very different ways and need their own audit. A valid audit of a deployed system needs the employer's demographic and outcome data, the group metrics of Section 5.3, and repeated testing after every change of model or data.

9. DISCUSSION

Realistic performance. With leakage controlled and labels that we did not generate ourselves, logistic regression on a 24-way resume task reaches about 0.644 accuracy and 0.594 macro-F1. Accuracy near 100% should make a reader suspicious. It usually means that the labels are a function of the input text, that the vectoriser was fitted before the split, or that test resumes also appear in training. The design used here rules out each of these.

Simple methods hold their own. TF-IDF with a linear model is statistically level with a random forest and a stacked ensemble, and the lexical rankers cannot be told apart on 24 queries. Frozen Sentence-BERT beat none of them on either task. For screening problems like this one, fast and inspectable methods are a sensible default, and extra complexity should be justified by a significance test and not only by a higher point estimate. This finding concerns frozen, general-purpose embeddings; it does not mean that fine-tuned neural models would behave the same way.

The labels set the ceiling. Many errors are understandable overlaps, such as finance and accounting or consulting and IT. A category is a coarse, partly self-assigned stand-in for suitability. What a screening tool really has to predict, whether someone would do the job well or be shortlisted by a fair process, is not observed in any public dataset.

Why the three checks belong together. Leakage control tells us how much of a score is real, retrieval metrics tell us whether a system orders candidates sensibly instead of merely labelling them, and bias auditing tells us whether protected signals move the output. A system can pass one of these checks and still fail another, so reporting only one gives an incomplete picture of reliability.

9.1. A RESPONSIBLE SCREENING WORKFLOW

Drawing on these results, we suggest the following workflow for organisations that use text-mining tools in recruitment. We have not evaluated it.

- Eligibility by explicit rules. Mandatory criteria stated in the vacancy (a required degree, a license, minimum experience) are checked by transparent, documented rules that a candidate can contest, not by a learned classifier.
- Ranking as decision support. A lexical or semantic ranker orders the eligible applicants for human review, with the term-level explanation of Section 7 attached to each score. It never rejects an applicant by itself.
- Human oversight and records. A trained reviewer makes every shortlisting and rejection decision. Inputs, scores and decisions are logged so that outcomes can be audited and contested.
- Regular bias audits. Counterfactual tests (Section 5.3) and group metrics on the organization's own data are run before deployment and after every change, in line with the audit and high-risk requirements noted in Section 1.

9.2. THREATS TO VALIDITY

- Internal. Hyper-parameters were fixed in advance rather than tuned so that some scores may be slightly pessimistic. The 24 ranking queries were written by the authors, and the family map behind relevance grade 1 is a judgment call. Fold-level predictions of the Sentence-BERT classifier were not kept, so it has no paired test.
- Construct. Category labels approximate occupational fit, not hiring suitability. The corpus is private-sector and English-language, so no claim is made for government or non-English recruitment.
- External. All results come from one dataset gathered from a single website. Its resumes are examples that follow a few templates and may not represent real applicant populations.
- Statistical. With 24 queries, the ranking comparisons have low power, and the audit uses only eight name pairs.

10. CONCLUSION

We evaluated text-mining methods for resume classification and job-description ranking on the public Kaggle Resume Dataset, using leakage controls, independent labels, ranking metrics and significance tests. Once the title line is removed, the lexical classifiers reach 0.571 to 0.652 accuracy. Stacking gives no significant gain over logistic regression, and a frozen Sentence-BERT classifier scores below the lexical models. In ranking, latent semantic analysis has the best average but cannot be separated reliably from BM25 or TF-IDF, and Sentence-BERT comes last. The counterfactual name test found almost no effect, but without demographic data it says nothing conclusive about fairness. The main lesson is methodological: a reported score means little unless leakage is controlled, ranking is measured as ranking, and a bias claim states what the audit could actually detect. Next steps are fine-tuned encoders and cross-encoder rankers under the same protocol, real job descriptions with expert relevance judgments, audits of dense models, and evaluation on the demographic and outcome data of a cooperating employer.

DATA AND CODE AVAILABILITY

The dataset used in this study is publicly available on Kaggle [9]. The code that regenerates every number, table and figure (seed 42) is available from the corresponding author upon reasonable request.

CONFLICTS OF INTEREST

The authors declare that there is no conflict of interest concerning the publishing of this paper.

REFERENCES

- [1] J. Sánchez-Monedero, L. Dencik, and L. Edwards, "What does it mean to 'solve' the problem of discrimination in hiring? Social, technical and legal perspectives from the UK on automated hiring systems," *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, Barcelona, Spain, 2020, pp. 458–468, Jan 2020, doi: 10.1145/3351095.3372849
- [2] European Parliament and Council of the European Union, "Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act)," *Official Journal of the European Union*, L series, Jul. 12, 2024. [Online]. Available: <http://data.europa.eu/eli/reg/2024/1689/oj>
- [3] New York City Department of Consumer and Worker Protection, "Automated Employment Decision Tools (AEDT): Local Law 144 of 2021." [Online]. Available: <https://www.nyc.gov/site/dca/about/automated-employment-decision-tools.page>
- [4] J. Dastin, "Amazon scraps secret AI recruiting tool that showed bias against women," *Reuters*, 2018. [Online]. Available: <https://www.reuters.com/article/us-amazon-com-jobs-automation-insight/amazon-scraps-secret-ai-recruiting-tool-that-showed-bias-against-women-idUSKCN1MK08G/>
- [5] M. Bogen and A. Rieke, "Help wanted: An examination of hiring algorithms, equity, and bias," *Upturn*, Washington, DC, USA, Dec. 2018. [Online]. Available: <https://www.upturn.org/work/help-wanted/>
- [6] M. Raghavan, S. Barocas, J. Kleinberg, and K. Levy, "Mitigating bias in algorithmic hiring: Evaluating claims and practices," *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, Barcelona, Spain, vol. 1, no. 1, pp. 469–481, 2020, doi: 10.1145/3351095.3372828
- [7] S. Kaufman, S. Rosset, C. Perlich, and O. Stitelman, "Leakage in data mining: Formulation, detection, and avoidance," *ACM Transactions on Knowledge Discovery from Data*, vol. 6, no. 4, pp. 1–21, 2012, doi: 10.1145/2382577.2382579
- [8] S. Kapoor, and A. Narayanan, "Leakage and the reproducibility crisis in machine-learning-based science," *Patterns*, vol. 4, no. 9, pp. 100804–100804, Aug. 2023, doi: 10.1016/j.patter.2023.100804.
- [9] S. Bhawal, "Resume Dataset," Kaggle. [Online]. Available: <https://www.kaggle.com/datasets/snehaanbhawal/resume-dataset>
- [10] T. Joachims, "Text Categorization With Support Vector Machines: Learning With Many Relevant Features," *Machine Learning: ECML-98*, vol. 1398, pp. 137–142, 1998, doi: 10.1007/bfb0026683
- [11] J. D. M. Rennie, L. Shih, J. Teevan, and D. R. Karger, "Tackling the poor assumptions of naive Bayes text classifiers," *ICML'03: Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, Washington, DC, USA, pp. 616–623, 2003.
- [12] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324
- [13] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing and Management*, vol. 45, no. 4, pp. 427–437, 2009, doi: 10.1016/j.ipm.2009.03.002
- [14] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing and Management*, vol. 24, no. 5, pp. 513–523, 1988, doi: 10.1016/0306-4573(88)90021-0

- [15] S. Robertson and H. Zaragoza, "The probabilistic relevance framework: BM25 and beyond," *Found. Trends Inf. Retr.*, vol. 3, no. 4, pp. 333–389, 2009, doi: 10.1561/15000000019
- [16] S. E. Robertson, and S. Walker, "Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval," 17th Annual International ACM SIGIR Conference Research and Development in Information Retrieval, Dublin, Ireland, pp. 232–241, 1994.
- [17] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*, Cambridge, U.K.: Cambridge Univ. Press, 2008.
- [18] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990, doi: 10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASII>3.0.CO;2-9
- [19] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, MN, USA, pp. 4171–4186, 2019, doi: 10.18653/v1/N19-1423.
- [20] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China, pp. 3982–3992, 2019, doi: 10.18653/v1/D19-1410
- [21] K. Song, X. Tan, T. Qin, J. Lu, and T.-Y. Liu, "MPNet: Masked and permuted pre-training for language understanding," *Advances in Neural Information Processing Systems*, vol. 33, 2020, doi: 10.48550/arXiv.2004.09297
- [22] N. Thakur, N. Reimers, A. Rüklé, A. Srivastava, and I. Gurevych, "BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models," *Proceedings of 35th Conference Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, arxiv, Nov 2021, doi: 10.48550/arXiv.2104.08663
- [23] D. H. Wolpert, "Stacked generalization," *Neural Network*, vol. 5, no. 2, pp. 241–259, 1992, doi: 10.1016/S0893-6080(05)80023-1
- [24] Q. McNemar, "Note on the sampling error of the difference between correlated proportions or percentages," *Psychometrika*, vol. 12, no. 2, pp. 153–157, 1947, doi: 10.1007/BF02295996
- [25] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural Computers*, vol. 10, no. 7, pp. 1895–1923, 1998, doi: 10.1162/089976698300017197
- [26] Janez Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research (JMLR)*, vol. 7, no. 1, pp. 1–30, 2006.
- [27] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945, doi: 10.2307/3001968
- [28] Sture Holm, "A simple sequentially rejective multiple test procedure," *Scandinavian Journal of Statistics*, vol. 6, no. 2, pp. 65–70, 1979. [Online]. Available: <https://www.jstor.org/stable/4615733>
- [29] B. Efron, "Bootstrap methods: Another look at the jackknife," *Breakthroughs in Statistics*, Springer Series in Statistics. Springer, NY vol. 7, no. 1, pp. 1–26, 1979, doi: 10.1214/aos/1176344552
- [30] M. D. Smucker, J. Allan, and B. Carterette, "A comparison of statistical significance tests for information retrieval evaluation," *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, Lisbon, Portugal, pp. 623–632, Nov 2007, doi: 10.1145/1321440.1321528
- [31] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of IR techniques," *ACM Transactions on Information Systems*, vol. 20, no. 4, pp. 422–446, 2002, doi: 10.1145/582415.582418
- [32] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *ACM Computing Surveys*, vol. 54, no. 6, pp. 1–35, July 2021, doi: 10.1145/3457607
- [33] M. De-Arteaga et al., "Bias in bios: A case study of semantic representation bias in a high-stakes setting," *FAT* '19: Proceedings of the Conference on Fairness, Accountability, and Transparency*, Atlanta, GA, USA, pp. 120–128, 2019, doi: 10.1145/3287560.3287572
- [34] A. Caliskan, J. J. Bryson, and A. Narayanan, "Semantics derived automatically from language corpora contain human-like biases," *Science*, vol. 356, no. 6334, pp. 183–186, Apr 2017, doi: 10.1126/science.aal4230
- [35] M. Bertrand and S. Mullainathan, "Are Emily and Greg more employable than Lakisha and Jamal? A field experiment on labor market discrimination," *American Economic Review*, vol. 94, no. 4, pp. 991–1013, 2004, doi: 10.1257/0002828042002561.
- [36] M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian, "Certifying and removing disparate impact," *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Sydney, Australia, pp. 259–268, 2015, doi: 10.1145/2783258.2783311.
- [37] M. Hardt, E. Price, and N. Srebro, "Equality of opportunity in supervised learning," *Advances in Neural Information Processing Systems*, vol. 29, 2016, doi: 10.48550/arXiv.1610.02413
- [38] S. Barocas, M. Hardt, and A. Narayanan, *Fairness and Machine Learning: Limitations and Opportunities*, Cambridge, MA, USA: MIT Press, 2023. [Online]. Available: <https://fairmlbook.org>
- [39] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011. [Online]. Available: <https://jmlr.org/papers/v12/pedregosa11a.html>