Corpus and Baselines for Distinguishing Authentic, AI-Generated, and AI-Enhanced Resumes

Andrea Loizidou*, **Anshu Kiran Sharma***, **Adrian Esquivel**†
Mark A. Finlayson* **& Mustafa Ocal***
*Florida International University
Knight Foundation School of Computing and Information Sciences
†TECKpert
{aloizido, ashar076, markaf, mocal}@fiu.edu*, aesquivel@teckpert.com†

Abstract

Job applicants are increasingly turning to generative AI to create or enhance their resumes, leading to challenges in fairness, integrity, and efficiency of modern recruitment processes. We present the first curated corpus of resumes annotated as to whether they are authentic, AI-enhanced, or fully AI-generated. The corpus is balanced across the three classes, comprising 420 resumes spanning five job descriptions in the Information Technology (IT) sector, with the authentic resumes anonymized. We establish strong baselines for this task using traditional and neural supervised machine learning approaches, including Logistic Regression, SVM, Random Forest, XGBoost, BERT, and Longformer. For the featurized approaches, we pair sparse TF-IDF (word/character n-grams) with style features capturing length, punctuation, casing, contractions, lexical diversity (type-token ratio [TTR], number of hapax legomena), n-gram uniqueness, readability indices, and sentiment. Our analysis reveals systematic differences between the classes: AI-generated text features shorter, more uniform sentences, and fewer contractions; AI-enhanced text has the highest uniqueness and TTR; and authentic text has the widest variance across all features. XGBoost is the best performing method, achieving 95.29% accuracy and an F_1 of 0.953. We make the corpus available for other researchers to build upon our work. We also benchmark two leading off-the-shelf AI–text detectors on our 420-resume corpus. Despite strong reports in other domains, ORIGINALITY attains only 55.7% accuracy overall (71/140 authentic, 81/140 AI-generated, 82/140 AI-enhanced correct), and Writer attains 25.0%, with the largest failures on AI-enhanced resumes, highlighting domain shift and cautioning against uncalibrated deployment.

Keywords: Resume Classification, Gen-AI Detection, AI-Enhanced Text Detection

1. Introduction

Information technology (IT) positions are among the most in-demand jobs in today's labor market. Companies often rely heavily on resumes to quickly identify qualified candidates with the right technical skills. However, the hiring landscape has changed significantly with the rise of generative AI. Today, applicants can simply copy and paste a job description into a large language model (LLM) such as ChatGPT and obtain a "perfect" resume that mirrors the employer's requirements. Less problematically, applicants can feed their authentic resume into an LLM and ask the system to "enhance" the resume to better match a provided job description. These enhancements may run the gamut from merely tuning word choice or framing, all the way to completely fabricating select skills and experience.

Such activities create a major challenge for recruitment. Many automated applicant tracking systems (ATS) evaluate resumes by keyword matching against job descriptions. As a result, AI-generated or AI-enhance resumes—optimized to echo the posting—often achieve the highest match scores, unfairly outranking authentic candidates with genuine skills and experience. This is problematic for both sides of the hiring process. Companies waste time and resources evaluating candidates who may lack the advertised qualifications, while genuine applicants find themselves at a disadvantage because their authentic resumes do not score as highly.

Existing approaches to detecting AI-generated text have not proven effective for resumes. Tools designed for essays, articles, or longer documents typically rely on paragraph-level coherence or linguistic predictability. Resumes, however, are formatted as lists, bullet points, and short phrases, which are structures that obscure the stylistic signals traditional detectors depend on.

To enable research on addressing this challenge, we assemble and release the first curated and annotated corpus of resumes, which contains a balanced selection of authentic, AI-enhanced, and AI-generated resumes associated with five specific job descriptions in the Information Technology (IT) space. We also experiment with a number of baseline approaches that show very strong performance.

The authentic resumes in our corpus are real resumes submitted to our collaborator TECKpert, a talent acquisition firm specializing in connecting IT professionals with companies seeking technical expertise. TECKpert provided real job descriptions and authentic resumes predating the widespread adoption of generative AI (pre-2019), ensuring that the authentic dataset was free of AI

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 7331–7344
11–16 May 2026.

influence. From these materials, we constructed two additional classes: (1) AI-generated resumes, produced by prompting state-of-the-art LLMs with only the job descriptions, and (2) AI-enhanced resumes, created by rewriting selected sections of authentic resumes using LLMs while preserving original content. Further, we semi-automatically anonymized the real resumes to remove all personally identifiable information (PII).

For our featurized supervised machine learning baselines, we extracted both content-based features (e.g., TF-IDF word and character n-grams) and style-based linguistic features (e.g., readability scores, vocabulary richness, sentiment). The best-performing classifiers were Random Forest and XGBoost, with the latter achieving an accuracy of 95.29% and a macro F_1 of 0.953, and outperforming standard neural methods, demonstrating that, at least within this new corpus, AI-generated and AI-enhanced resumes can be reliably distinguished from authentic ones.

Off-the-shelf AI-text detectors are increasingly considered for compliance screening, yet it is unclear whether they generalize to resumes. To probe this, we evaluate two widely used systems on our three-way corpus. Performance varies sharply by class: the better detector correctly labels only 71/140 authentic, 81/140 AI-generated, and 82/140 AI-enhanced resumes (55.7% overall), while the other achieves 25.0% and largely fails on AI-generated and AI-enhanced documents. These results indicate substantial domain shift from essay/web text to professional resumes and motivate resume-specific calibration and human-in-the-loop review.

The paper is organized as follows. First, we review approaches to AI-generated text detection as well as prior work on automated resume processing (§2). Next, we explain our methodology in detail, including construction of the corpus, data preprocessing, feature engineering, exploratory data analysis, and model training (§3). Then we present our results and discuss their implications (§4). Finally, we provide a summary of the contributions (§5). We release the corpus to enable future work that builds upon these results[1].

2. Related Work

2.1. AI-Generated Text Detection

A growing ecosystem of commercial detectors aims to distinguish LLM-generated from human-authored text. Akram (2023) evaluated six widely used AI–text detectors on the AH&AITD corpus: GPTKit[2] (a meta-ensemble that aggregates several signals), GPTZero[3] (perplexity/burstiness cues tuned for student prose), Originality.ai[4] (commercial API scoring AI-likeness), Sapling[5] (writing assistant with an AI-content flagger), Writer[6] (governance suite with document-level judgments), and Zylalabs[7] (general AI-text detection API). Originality.ai stands out with a reported 97.0% accuracy and balanced F_1 for both AI and human classes. The remaining tools perform in a mid tier: Writer (69.05% accuracy), Zylalab (68.23%), Sapling (66.6%), and GPTZero (63.7%) show moderate, more balanced results. GPTKit trails with 55.29% accuracy and pronounced class imbalance (AI-class F_1 0.21 vs. human-class F_1 0.69), illustrating how aggregate signals don't necessarily translate to robust AI-text recall.

Across tools, key limitations recur. Several detectors exhibit high precision but low recall on AI text, which leads to many false negatives; conversely, false positives on genuine writing remain a practical concern in educational and publishing settings. Performance is sensitive to dataset composition and thresholding, and many systems are black-box APIs that limit reproducibility and granular error analysis. Prior work also indicates that paraphrasing can substantially erode detector performance, underscoring open challenges in domain transfer, style variation, and adversarial robustness (Akram, 2023).

Beyond commercial detectors, academic research has explored a wide spectrum of methodological families from watermarking and feature-based heuristics to neural classifiers, human-in-the-loop pipelines, and hybrids that aim to verify AI authorship under different deployment and threat models. Traditional featurized machine learning leverage stylistic and stylometric features, exploiting measurable differences in syntax, grammar, and linguistic cues between human and LLM text (Modupe et al., 2022; Schuster et al., 2020; Kumarage et al., 2023). Such features have long been studied in the field of stylometry and authorship attribution, where lexical diversity, punctuation usage, and structural patterns are used to characterize writing style and identify authorship signals (Stamatatos, 2009; Koppel et al., 2009; Potthast et al., 2016; Neal et al., 2017). Frequency-based and statistical approaches capture distributional irregularities such as token or word frequencies, perplexity,

[1] https://doi.org/10.34703/gzx1-9v95/HTRD9P

[2] https://gptkit.ai/

[3] https://gptzero.me/

[4] https://originality.ai/

[5] https://sapling.ai/ai-content-detector

[6] https://writer.com/ai-content-detector/

[7] https://zylalabs.com/

and burstiness indicative of machine generation (Fröhling and Zubiaga, 2021; Hans et al., 2024; Mitchell et al., 2023). Neural methods use pretraining, fine-tuned classifiers, or zero-shot prompting to learn representations that distinguish AI from human text across domains (Bakhtin et al., 2019; Solaiman et al., 2019; Crothers et al., 2022). Across benchmarks, reported performance typically falls in the F_1 = *0.76–0.92* range, varying with domain, text length, detector family, and adversarial pressure.

Some systems integrate human-aided pipelines, combining automated detectors with expert review for higher-confidence decisions in sensitive contexts (Hu et al., 2023; Clark et al., 2021), reporting detection AUROC up to *0.915* on GPT-4 generated text. Hybrid approaches fuse multiple signals, such as watermarking, linguistic features, and neural scores—to enhance robustness against obfuscation (Kushnareva et al., 2021; Liu et al., 2023b), with reported accuracies ranging from *85–97%* depending on the GPT model/version and domain.

Some approaches use data-driven watermarking to check whether a text was generated by a model that embeds a watermark signal (Gu et al., 2022; Lucas and Havens, 2023; Tang et al., 2023), while others rely on model-driven watermarking to inject identifiable patterns during generation for later verification (Kirchenbauer et al., 2023b,a; Liu et al., 2023a; Hou et al., 2023). Additional systems employ post-processing or neural watermarking to tag outputs after generation or via neural encodings, allowing provenance verification with minimal impact on quality (Por et al., 2012; Munyer et al., 2024; Abdelnabi and Fritz, 2021; Yoo et al., 2023).

Taken together, these strands trade off benefits and risks: watermarking can be effective and quality-preserving but requires provider adoption and can be obscured by later transformations of the text (e.g., paraphrasing); stylistic and frequency/metric approaches are flexible and transparent but can be perturbed by small edits and may not transfer across models or genres; neural methods capture nuanced cues and can be robust to small mutations but often depend on domain/language coverage and training data; human-aided workflows raise accuracy on edge cases but hinge on reviewer expertise and scalability; and hybrid designs improve resilience at the cost of added system complexity.

2.2. Resume Processing

Although both commercial and academic detectors are useful for spotting AI-generated text, they do not work well on resumes, which typically lack long, contiguous prose and instead preferentially use lists, bullet points, and short phrases. In our initial small experiments with existing AI-text generation systems, some clearly AI-generated resumes were labeled negative (missed), while polished human-written resumes triggered false positives, which underscores domain mismatch and calibration issues. This gap highlights the need for a dedicated, automatic AI-generated-resume detection tool tailored to resume structure (bullet points, section headers, skills lists), shorter contexts, and real-world Applicant Tracking Systems (ATS) workflows.

As far as we can tell, there is no prior scholarly work on AI-generated resume detection. Although there are emerging commercial resume "AI detectors" (e.g., Enhancv[8], AIApply[9], Jobscan[10]), we do not evaluate them here for three reasons: (i) they do not report standard evaluation metrics or public benchmarks, (ii) they provide no transparent architectural descriptions suitable for scientific comparison, and (iii) they do not distinguish AI-generated from AI-enhanced text as done here. Thus our focus in this section is in reviewing how researchers have approached automated processing of resumes more generally.

Automated resume screening has long been central to recruitment workflows, with ATSs widely used to filter candidates before human review. Traditional systems largely depended on keyword extraction and rule-based parsing to align resumes with job descriptions (Faliagka et al., 2012). While efficient, such approaches have been criticized for discarding qualified candidates whose resumes failed to mirror the precise terminology in postings (Brown and Campion, 1994).

Previous work on resume screening has explored a range of techniques, each with clear limitations. Early systems prioritized keyword matching, which was fast but unreliable, often missing candidates who used different terms to describe the same skills (Fuller and Raman, 2022). Word embedding methods such as word2vec improved synonym handling by mapping words into vector space, but they struggled to capture deeper semantic context, especially in technical roles where subtle distinctions matter (Pudasaini et al., 2021). Clustering methods grouped resumes with similar patterns but lacked precision for identifying specific qualifications (Bafna et al., 2019). Rule-based systems offered transparent decision-making but were brittle, breaking down as job requirements shifted. Later studies employed transformer-based models such as BERT, which better captured contextual meaning in resumes and job descriptions (Reddy and Parthiban, 2025). These models showed strong results but came at the cost of high data and computational demands, limiting practical deployment in many hiring contexts.

[8] https://enhancv.com/resources/resume-checker/

[9] https://aiapply.co/ai-resume-checker

[10] https://www.jobscan.co/

Research has also moved beyond retrieval into document classification. Luo et al. (2018) introduced ResumeNet, the first formal framework for Resume Quality Assessment (RQA) using deep learning. Their model encoded resume sections, applied attention mechanisms to highlight discriminative elements, and used pairwise and triplet learning strategies to handle limited labeled data. ResumeNet demonstrated that quality indicators such as section consistency, skill specificity, and organizational structure could be systematically modeled, achieving strong alignment with human judgments. Similarly, (Deshmukh and Raut, 2025) leveraged BERT to align resume vectors with job descriptions for improved candidate–job matching. These innovations advanced resume analysis, but they shared a common assumption: that resumes authentically reflect candidate authorship. As generative AI proliferates, this assumption is no longer safe.

Although these methods achieve 70–90% accuracy in specific contexts, significant challenges limit their broader organizational adoption, including difficulties with heterogeneous resume formats, dataset-driven bias, opacity in proprietary systems, imperfect soft-skill modeling, and concerns about generalizability; moreover, implementation is complicated by costs, workflow disruption, and training requirements.(Armstrong and Metaxa, 2025; Mujtaba and Mahapatra, 2024; Tewari et al., 2024).

3. Methodology

3.1. Data Collection & Generation

To obtain real world, authentic resumes, we collaborated with TECKpert, Inc., which provided authentic job descriptions and resumes from IT professionals that were submitted to the company before 2019. We focused on five representative roles spanning both technical and managerial responsibilities: .NET Application Developer, Database Analyst, Network Administrator, Project Manager, and Senior Web Application Programmer. For each job description, we started by collecting a total of 200 resumes per job description, for a total of 1000 raw authentic resumes.

Next, we generated AI-created resumes by prompting three state-of-the-art large language models with the job descriptions: OpenAI's GPT-5, which produces highly structured professional content (Brown et al., 2020); Google's Gemini 2.5 Flash, which integrates reasoning and natural text generation (Team et al., 2023); and Meta's LLaMA-3, an open-source alternative known for flexibility in formatting (Touvron et al., 2023). For each job description, we created 50 AI-generated resumes, carefully varying formatting conventions and linguistic styles to avoid uniform outputs. We provide the exact prompt templates (and decoding settings) used for generation in Appendix A.

To capture a middle ground between authentic and synthetic resumes, we produced 50 AI-enhanced versions for each job description. In this case, we provided authentic resumes as inputs to the same large language models along with the target job title, but without including the full job description. The models were instructed to refine these resumes into more polished and professional versions, improving style and clarity while retaining original content. This design was intended to simulate realistic editing scenarios in which applicants improve wording and presentation for a role without fully rewriting the document. We used a prompt-chaining procedure, and the full chained prompts are included in Appendix A. Because all AI-generated resumes were two pages, we constrained AI-enhanced outputs to two pages for comparability.

3.2. Data Balancing

Authentic submissions in our archive varied widely in length, with several much longer than two pages. AI-generated resumes, on the other hand, were always between one and two pages. To avoid a trivial length cue for detection we excluded authentic resumes exceeding two pages, leaving us with 140 authentic items. We then randomly sampled 140 AI-generated and 140 AI-enhanced resumes to balance the classes with the 140 authentic resumes, yielding 420 documents in total. The corpus spans five job descriptions, with 28 resumes per job description in each class, resulting in 140 resumes per class overall (i.e., 5 job descriptions × 28 per class × 3 classes).

Next, we anonymized personally identifiable information (PII). LLM-produced files often contained placeholder identities (e.g., "John/Jane Doe") or model usernames; to mitigate name-based bias, we removed or replaced such strings before modeling. Concretely, we used a hybrid spaCy+regex pass to detect and replace names, addresses, emails, phone numbers, and related identifiers with placeholders such as ‹NAME› and ‹ADDRESS›. Names were removed only when they appeared alone on a line to avoid corrupting content. Address handling combined regex (covering formats like "123 Main St, City, ST 12345," ZIP codes, city names, and state abbreviations) with named entity recognition (NER) over locations/facilities; repeated placeholders (e.g., consecutive ‹ADDRESS›) were merged. We initially stripped punctuation and extra spaces during tokenization, but restored capitalization and key punctuation after observing their discriminative value for downstream analyses (e.g., sentence length, n-gram features, domain strings like *C++* or *Ph.D.*).

3.3. Extracting Linguistic Features

To construct traditional featurized machine learning baselines, we engineered a mixed feature set that captures both content and style. Content is represented with sparse count vectors (TF-IDF on words and n-grams), while style is modeled with numeric linguistic attributes (length, casing and punctuation habits, lexical diversity, phrasing uniqueness, readability, and sentiment/tone).

Count vectors We computed TF-IDF on words (unigrams) to up-weight terms that are frequent within a resume but rare across the corpus, TF-IDF on word n-grams (bigrams/trigrams) to preserve short phrasal context (e.g., *machine learning*), and TF-IDF on character n-grams to capture subword style and formatting regularities (e.g., capitalization, punctuation, terms like *C++*). We implemented TF-IDF vectorization with `scikit-learn` (Pedregosa et al., 2011), an open-source ML library whose vectorizers are a reliable, interpretable baseline for converting raw text into model-ready features; we saved fitted vectorizers with joblib for reproducibility.

Basic Counts For each resume we derived total characters, words, and sentences, plus average sentence length, simple indicators of structural complexity that varied systematically by class.

Style-based Features We counted punctuation marks, Title-Case words, ALL-CAPS words, and contractions; these cues differentiate human from AI style, with contractions especially diagnostic for authentic resumes.

Lexical Diversity We measured Type-Token Ratio (TTR) and hapax legomena rate (share of words appearing once) to quantify vocabulary richness and repetition.

Uniqueness of Phrasing To gauge template-like versus varied language, we computed bigram and trigram uniqueness ratios, the proportion of n-grams that occur only once in the document.

Readability Scores We measured Flesch Reading Ease, Flesch-Kincaid Grade Level, SMOG, and Gunning Fog indices (Flesch, 1948; Kincaid et al., 1975) using `textstat`[11], a Python library that implements classic readibility formulas. These correlate with how constrained or edited the prose reads and helped separate classes.

Sentiment and Tone We extracted VADER compound scores to capture overall affect; flatter, more neutral tone was characteristic of fully AI-generated text. VADER is a rule-based sentiment analyzer tuned for short, informal English, to quantify tone with a single compound score (Hutto and Gilbert, 2014).

[11] https://pypi.org/project/textstat/

3.4. Exploratory Data Analysis

We performed an exploratory data analysis (EDA) to understand how writing style and structure differ across resume classes before training any models. This step helped identify which features were likely to be useful, with a focus on lexical indicators, readability scores, and phrasing patterns.

Sentence length and brevity As we can see in the Figure 1, AI-generated resumes were consistently shorter, with fewer characters and sentences; authentic resumes showed wider variance in sentence structure. This greater variance among authentic resume likely reflects differences in how individuals describe experience, where narrative explanations are combined concisely in bullets. In contrast, AI-generated and enhanced resumes appear to follow more uniform structual patterns across documents.

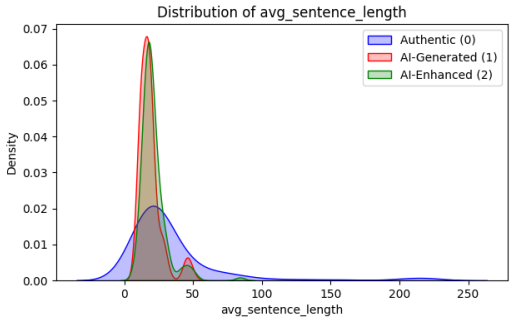

Figure 1: Average sentence length distribution.

Phrasing uniqueness Bigram uniqueness was highest in AI-enhanced resumes, likely reflecting a blend of human edits and AI suggestions, followed by authentic, with AI-generated the lowest (Figure 2). The relatively high uniqueness observed in AI-enhanced resumes suggests that localized rewriting may introduce lexical variation while preserving the underlying human-authored structure. While by comparison, fully AI-generated resumes occasionally show more repetitive phrasing patterns in the documents.

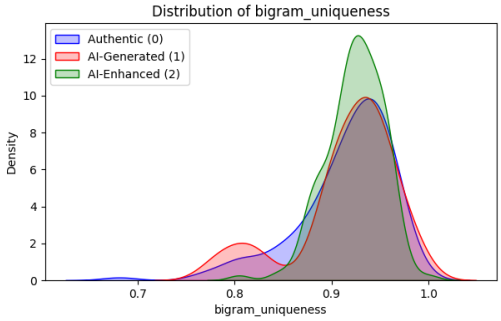

Figure 2: Bigram uniqueness across resume types.

Contractions Authentic resumes used the most contractions, AI-enhanced used some, and AI-generated used the least (Figure 3), aligning with a more natural tone in human writing. This difference likely reflects variation in stylistic formality across documents, with authentic resumes incorporating conversational phrasing, while AI-generated resumes tend to maintain more standardized professional language with AI-enhanced resumes between these extremes.

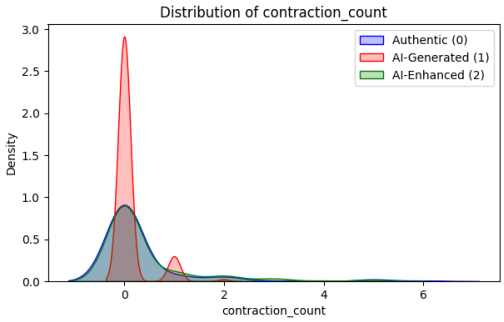

Figure 3: Contraction count distribution across resume types.

Lexical diversity Type-Token Ratio (TTR) was highest for AI-enhanced, then authentic, and lowest for AI-generated (Figure 4). This suggests that light human editing over AI assistance increases vocabulary richness relative to purely synthetic text. The higher TTR observed in AI-enhanced resumes may indicated paraphrasing applied to human-authored content, introducing additional lexical variation without substantially altering the document structure. In contrast, AI-generated resumes show more concentrated distributions, suggesting more consistent vocabulary usage.

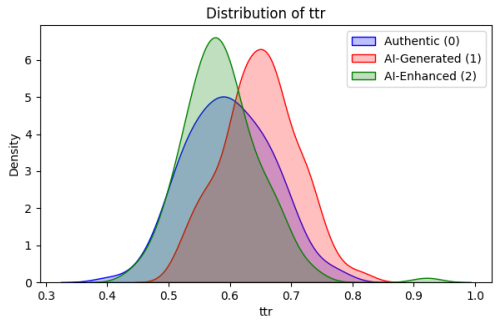

Figure 4: Type-Token Ratio (TTR) distribution.

Punctuation and casing signals AI-generated resumes tended to contain fewer punctuation marks and fewer Title-Case words (Figures 5 & 6). We also analyzed ALL-CAPS word counts (Figure 7). Authentic resumes exhibit greater dispersion in punctuation and capitalization counts, consistent with individualized formatting and section styling practices. In contrast, AI-generated resumes show more concentrated distributions across these features, suggesting more uniform formatting conventions, while AI-enhanced resumes again occupy an intermediate range between the two extremes.

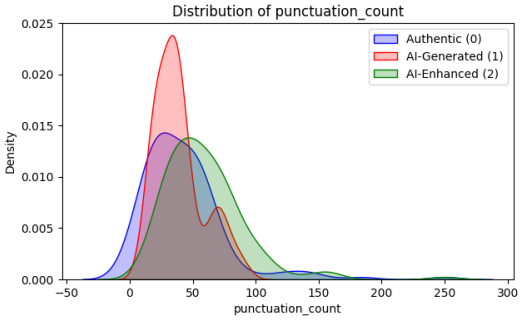

Figure 5: Punctuation count across resume types.

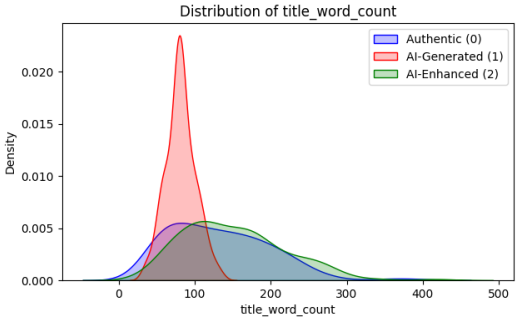

Figure 6: Title-case word count across resume types.

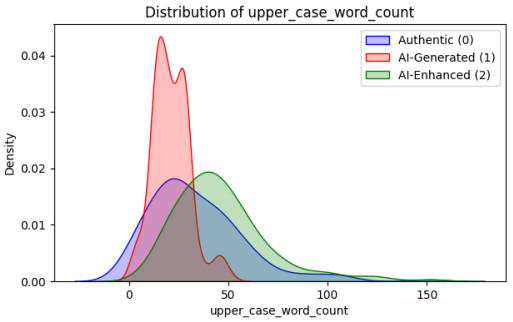

Figure 7: Upper case word count distribution.

Readability Readability indices separated classes: AI-generated resumes scored higher (i.e., were simpler/more uniform) on several measures; we report the Automated Readability Index (ARI) distribution in Figure 8 alongside other metrics. This suggests more consistent sentence construction in AI-produced text, while authentic resumes reflect a wider mix of concise statements and more complex descriptions across documents.

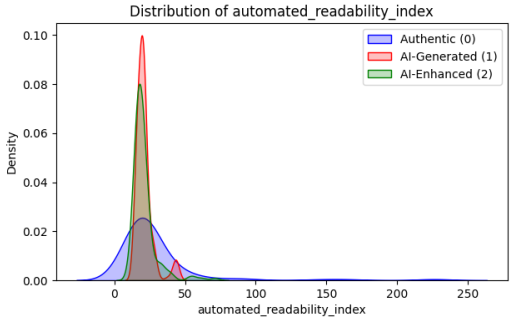

Figure 8: Automated Readability Index distribution.

Sentiment/tone Using VADER, the compound sentiment distribution (Figure 9) showed flatter, more neutral tone in AI-generated text, with authentic and AI-enhanced exhibiting broader spread. All three classes exhibit strongly positive sentiment values, reflecting the self-promotional nature of resumes. However, AI-enhanced resumes show a more concentrated distribution, indicating more uniform tonal expression, whereas authentic and AI-generated resumes display greater variability.

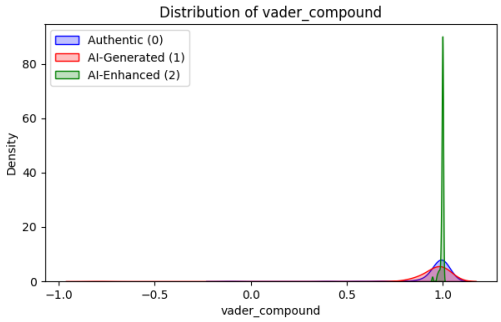

Figure 9: VADER compound sentiment distribution.

We have focused on the most diagnostic signals above, but our analysis also explored a broader set of indicators: additional readability metrics (SMOG, Gunning Fog, Dale–Chall, Linsear Write), length/volume cues (character and sentence counts), higher-order phrasing (trigram uniqueness alongside bigrams), and cross-feature relationships (correlation heatmap). These complementary views reinforced the class differences we report and helped guard against over-interpreting any single feature. The complete set of analyses are included in Appendix B.

3.5. Model Training

For the classical machine learning baselines, we represent each resume using a combination of sparse TF-IDF features and dense linguistic features. We compute TF-IDF vectors over word unigrams using a vectorizer fit on the training split and then applied unchanged to the validation data. In parallel, we extract numeric linguistic attributes (e.g. length statistics, punctuation and case counts, lexical diversity measures, readability indices, and sentiment scores) from the same documents. We then concatenate the sparse TF-IDF representation with the dense linguistic feature vector to a unified feature space prior to model training. The same combined representation is used across all traditional classifiers.

We tested four supervised classifiers: Logistic Regression (LR), Support Vector Machines (SVM), Random Forest (RF), and XGBoost. We used them because they cover complementary inductive biases for our mixed feature space (sparse TF-IDF + dense linguistic features). LR provides a strong, transparent linear baseline for high-dimensional text and is effective at flagging AI-style phrasing; SVMs are likewise well-suited to separating sparse, high-dimensional data and help with subtle boundaries such as AI-enhanced vs. authentic. In contrast, RF and XGBoost are tree ensembles that model non-linear interactions between content and style features; RF captures deeper stylistic patterns (e.g., punctuation/sentence structure) from mixed inputs, and XGBoost sharpens this with gradient boosting to detect small phrasing/structure shifts typical of lightly edited text.

All models were trained and compared in a unified pipeline with MLflow logging[12] (hyperparameters and metrics such as accuracy, precision, recall, macro F_1), ensuring reproducibility and fair selection of the best system for inference.

Consistent with these design reasons, the tree ensembles (RF/XGBoost) ultimately performed best on our combined feature set, whereas the linear baselines trailed, providing evidence that modeling content/style interactions is beneficial in this domain.

To complement these traditional models, we also fine-tuned transformer-based architectures to explore capturing deeper contextual dependencies. First, we trained a standard BERT-base classifier in the raw text, using the 512-token context window to establish a baseline for transformer representations. However, since most resumes exceeded this limit, we implemented a chunked BERT variant that segmented each document into overlapping 512-token windows with a stride of 128 and aggregated across chunks to produce document-level predictions. Finally, we trained Longformer, which extends BERT's architecture to accommodate up to 4096 tokens natively, allowing full-document encoding without truncation.

[12] https://mlflow.org/

3.6. Baselines: External AI–text detectors

Prior work reports strong performance from commercial detectors; for example, Akram (2023) finds that ORIGINALITY and WRITER achieve the highest accuracies among tested tools (97% and 69.05%, respectively). Motivated by this, we include these two systems as external baselines and ask whether they generalize to resume text and, in particular, to our three-way labeling scheme (authentic, AI-enhanced, AI-generated).

We applied both detectors to all 420 resumes in our corpus, obtaining each system's AI-likelihood score per document. Following standard evaluation practice, we computed overall accuracy as well as class-conditional accuracy for each of the three types. We reported accuracy by class on our corpus.

4. Results & Discussion

We evaluated the models LR, SVM, RF, XGBoost, BERT, BERT with chunk, and Longformer on the 3-way classification task (authentic, AI-generated, AI-enhanced). We used an 80-20 train-test set split of the data. Tree ensembles clearly outperformed linear baselines: RF reached 0.941 accuracy, and XGBoost reached 0.953 accuracy on the test set, with XGBoost attaining the highest macro F_1 = 0.953. LR and SVM trailed with macro F_1 of 0.751 and 0.791, respectively. To visualize error patterns, we report the confusion matrix of the best model (XGBoost) in Figure 10

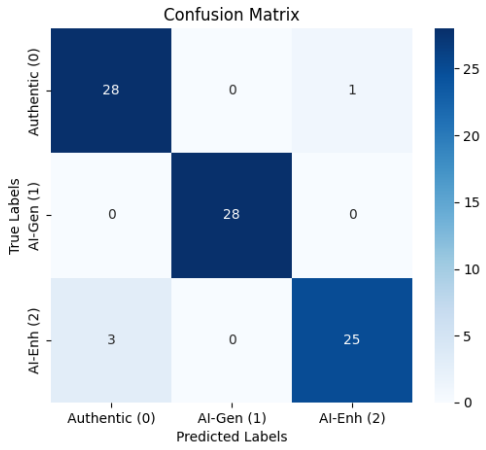

Figure 10: Confusion Matrix of Best Model

At the class level, AI-generated (class 1) was easiest, achieving perfect precision/recall/F_1; authentic (class 0) and AI-enhanced (class 2) also scored highly (F_1 = 0.95 each). Most errors came from confusion between authentic and AI-enhanced, consistent with our EDA showing that AI-enhanced writing blends human structure with AI-style phrasing.

The tree-based family handled our mixed feature space (sparse TF-IDF + dense linguistic features) better than linear separators. In particular, XGBoost's gradient-boosted trees flexibly captured content-style interactions (e.g., the joint pattern of n-gram usage with contraction/ punctuation habits and readability levels), which are crucial to separating lightly edited AI-enhanced text from authentic submissions. In particular, XGBoost seemed to most effectively explore the complementary cues present in the sentence length, contraction count, bigram/trigram uniqueness, vocabulary richness (TTR), and readability indices.

Model	Accuracy	Macro F_1
Logistic Regression	0.753	0.751
SVM	0.788	0.791
Random Forest	0.941	0.942
XGBoost	**0.953**	**0.953**
BERT	0.812	0.808
BERT with stride	0.906	0.907
Longformer	0.938	0.939

Table 1: Performance across models

The results confirm that modeling nonlinear couplings between content and style is decisive for this task, especially for the subtle authentic vs. AI-enhanced boundary explaining why XGBoost achieved the top macro F_1.

Among the transformer-based architectures, the results follow a consistent trend reflecting the impact of context length. The standard BERT model with a 512-token limit performed competitively (macro F_1=0.808) but was constrained by truncation, losing information in longer resumes. Extending BERT with the sliding-window chunking strategy improved performance substantially (macro F_1=0.907) by preserving global context through aggregated representations across segments. The Longformer, with the longest context window, further narrowed the gap with tree ensembles (macro F_1=0.939) confirming that full-document context strengthens discrimination between authentic and AI-influenced writing in resumes. The improved performance of chunked BERT and Longformer suggests that differentiating cues for AI-assisted resume writing extends beyond local lexical patterns to document-level structure and style continuity.

4.1. Off-the-shelf AI–text detectors on resume data

We evaluate two commercial detectors—ORIGINALITY and WRITER—class by class on our corpus.

Authentic (140). ORIGINALITY correctly labeled 71/140 as authentic, marked 58/140 as "partially AI" (some as high as 40% AI), and flagged 11/140 as fully AI, even though all authentic resumes predate ChatGPT. By contrast, WRITER correctly labeled 105/140 as authentic and marked the remaining 35/140 as "partially AI" with low scores ($<$10% AI each); none were flagged as fully AI. (Per-class accuracies: ORIGINALITY 50.7%; WRITER 75.0%.)

AI-generated (140). ORIGINALITY identified 81/140 as fully AI, labeled 22/140 as partially AI (65–90% AI), and misclassified 17/140 as authentic. WRITER failed on this class, assigning every document a $>$70% human score ($<$30% AI) and thus never labeling them AI. Likely contributors include the resume domain's short, bullet-style sentences, heavy noun-phrase fragments, and templated sectioning, which reduce the reliability of perplexity and burstiness-based cues and blur stylometric signals tuned on essay/web domains.

AI-enhanced (140). ORIGINALITY labeled 82/140 as partially AI (the intended outcome, detecting edited spans while leaving human content), 45/140 as fully authentic, and 13/140 as fully AI. WRITER again largely failed, rating each resume $>$90% human and explicitly marking 68/140 as fully human. This behavior is consistent with conservative calibration and domain shift: light, style-only edits from our prompt-chaining procedure dilute detector-visible cues, while resume-specific conventions (concise bullet fragments, jargon lists) further mask stylistic differences relative to detectors' training distributions.

Overall. Aggregating across classes, ORIGINALITY achieves 234/420 correct (55.7% accuracy), whereas WRITER achieves 105/420 (25.0% accuracy). Despite prior reports of strong performance in other genres, these in-domain results indicate that off-the-shelf detectors, without resume-specific calibration, are unreliable for AI-resume detection, especially on AI-enhanced documents.

5. Contributions

In this paper, our contributions are four-fold. **First,** we have presented the first corpus of resumes, annotated for whether they are authentic, AI-assisted, or AI-generated. We release this corpus for use by other researchers; the corpus and prompts we developed will allow updating of the corpus with new AI-enhanced and AI-generated resumes in the future, as generative AI technology advances. **Second,** we developed a set of baselines for detecting AI-generated and AI-enhanced resumes, and benchmarked them on our data. **Third,** we performed a comprehensive exploratory data analysis of the differences between the classes, showing clear differences in class characteristics that leads to very high performance for traditional featurized machine learning methods. **Fourth,** we provide the first in-domain evaluation of two off-the-shelf AI–text detectors on a three-way resume corpus, revealing large class-dependent performance gaps and highlighting domain shift and the need for resume-specific calibration.

6. Limitations & Future Work

Despite the encouraging results, several limitations remain. First, the dataset is restricted to a single professional domain of Information Technology/Software Engineering and a controlled set of job descriptions, which may limit generalization to resumes from other fields with different conventions and stylistic norms. Second, while the balancing strategy helps reduce bias that could arise due to document length and formatting, the overall dataset size remains modest compared to large-scale text classification benchmarks. Third, our modeling primarily relies on lexical and stylistic features extracted from normalized texts; structural characteristics of resumes such as sections and formatting conventions were not explicitly modeled and may provide additional signals. Finally, the study reflects a snapshot of AI-assisted writing practices at a particular point in time, whereas LLMs continue to evolve rapidly, potentially introducing shifts in writing style that may affect detection performance.

Future work will focus on expanding the corpus to include additional domains, job roles, and larger number of resumes across all three classes, enabling more comprehensive evaluation of robustness. We also plan to explore richer linguistic representations, including document structure signals that can capture resume organization beyond surface text. In addition, data collection over time may provide insight into how AI assisted resume writing evolves and how detection models can adapt to emerging LLM writing capabilities. Ongoing collaboration with recruitment practitioners may also enable future collection of additional annotated data under appropriate privacy safeguards.

Acknowledgments

This work was funded by a research grant to Dr. Finlayson & Dr. Ocal from TECKpert LLC. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of TECKpert LLC.

7. Bibliographical References

Sahar Abdelnabi and Mario Fritz. 2021. Adversarial watermarking transformer: Towards tracing text provenance with data hiding. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 121–140. IEEE.

Arslan Akram. 2023. An empirical study of ai generated text detection tools. *arXiv preprint arXiv:2310.01423*.

Lena Armstrong and Danaé Metaxa. 2025. Navigating automated hiring: Perceptions, strategy use, and outcomes among young job seekers. *Proceedings of the ACM on Human-Computer Interaction*, 9(2):1–26.

Prafulla Bafna, Shailaja Shirwaikar, and Dhanya Pramod. 2019. Task recommender system using semantic clustering to identify the right personnel. *VINE Journal of Information and Knowledge Management Systems*, 49(2):181–199.

Anton Bakhtin, Sam Gross, Myle Ott, Yuntian Deng, Marc'Aurelio Ranzato, and Arthur Szlam. 2019. Real or fake? learning to discriminate machine from human generated text. *arXiv preprint arXiv:1906.03351*.

Barbara K Brown and Michael A Campion. 1994. Biodata phenomenology: Recruiters' perceptions and use of biographical information in resume screening. *Journal of Applied Psychology*, 79(6):897.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Elizabeth Clark, Tal August, Sofia Serrano, Nikita Haduong, Suchin Gururangan, and Noah A Smith. 2021. All that's' human'is not gold: Evaluating human evaluation of generated text. *arXiv preprint arXiv:2107.00061*.

Evan Crothers, Nathalie Japkowicz, Herna Viktor, and Paula Branco. 2022. Adversarial robustness of neural-statistical features in detection of generative transformers. In *2022 international joint conference on neural networks (IJCNN)*, pages 1–8. IEEE.

Asmita Deshmukh and Anjali Raut. 2025. Applying bert-based nlp for automated resume screening and candidate ranking. *Annals of Data Science*, 12(2):591–603.

Evanthia Faliagka, Athanasios Tsakalidis, and Giannis Tzimas. 2012. An integrated e-recruitment system for automated personality mining and applicant ranking. *Internet research*, 22(5):551–568.

Rudolph Flesch. 1948. A new readability yardstick. *Journal of applied psychology*, 32(3):221.

Leon Fröhling and Arkaitz Zubiaga. 2021. Feature-based detection of automated language models: tackling gpt-2, gpt-3 and grover. *PeerJ Computer Science*, 7:e443.

Joseph B Fuller and Manjari Raman. 2022. Building from the bottom up. *Harvard Business School, January*.

Chenxi Gu, Chengsong Huang, Xiaoqing Zheng, Kai-Wei Chang, and Cho-Jui Hsieh. 2022. Watermarking pre-trained language models with backdooring. *arXiv preprint arXiv:2210.07543*.

Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2024. Spotting llms with binoculars: Zero-shot detection of machine-generated text. *arXiv preprint arXiv:2401.12070*.

Abe Bohan Hou, Jingyu Zhang, Tianxing He, Yichen Wang, Yung-Sung Chuang, Hongwei Wang, Lingfeng Shen, Benjamin Van Durme, Daniel Khashabi, and Yulia Tsvetkov. 2023. Semstamp: A semantic watermark with paraphrastic robustness for text generation. *arXiv preprint arXiv:2310.03991*.

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. 2023. Radar: Robust ai-text detection via adversarial learning. *Advances in neural information processing systems*, 36:15077–15095.

Clayton Hutto and Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the international AAAI conference on web and social media*, volume 8, pages 216–225.

J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, Institute for Simulation and Training.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023a. A watermark for large language models. In *International Conference on Machine Learning*, pages 17061–17084. PMLR.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Manli Shu, Khalid Saifullah, Kezhi Kong, Kasun Fernando, Aniruddha Saha, Micah Goldblum, and Tom Goldstein. 2023b. On the reliability of watermarks for large language models. *arXiv preprint arXiv:2306.04634*.

Moshe Koppel, Jonathan Schler, and Shlomo Argamon. 2009. Computational methods in authorship attribution. *J. Am. Soc. Inf. Sci. Technol.*, 60(1):9–26.

Tharindu Kumarage, Joshua Garland, Amrita Bhattacharjee, K Trapeznikov, S Ruston, and Huan Liu. 2023. Stylometric detection of ai-generated text in twitter timelines. arxiv 2023. *arXiv preprint arXiv:2303.03697*.

Laida Kushnareva, Daniil Cherniavskii, Vladislav Mikhailov, Ekaterina Artemova, Serguei Barannikov, Alexander Bernstein, Irina Piontkovskaya, Dmitri Piontkovski, and Evgeny Burnaev. 2021. Artificial text detection via examining the topology of attention maps. *arXiv preprint arXiv:2109.04825*.

Aiwei Liu, Leyi Pan, Xuming Hu, Shu'ang Li, Lijie Wen, Irwin King, and Philip S Yu. 2023a. An unforgeable publicly verifiable watermark for large language models. *arXiv preprint arXiv:2307.16230*.

Xiaoming Liu, Zhaohan Zhang, Yichen Wang, Hang Pu, Yu Lan, and Chao Shen. 2023b. Coco: Coherence-enhanced machine-generated text detection under low resource with contrastive learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16167–16188.

Evan Lucas and Timothy Havens. 2023. Gpts don't keep secrets: Searching for backdoor watermark triggers in autoregressive language models. In *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*, pages 242–248.

Yong Luo, Huaizheng Zhang, Yongjie Wang, Yonggang Wen, and Xinwen Zhang. 2018. Resumenet: A learning-based framework for automatic resume quality assessment. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 307–316. IEEE.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International conference on machine learning*, pages 24950–24962. PMLR.

Abiodun Modupe, Turgay Celik, Vukosi Marivate, and Oludayo O Olugbara. 2022. Post-authorship attribution using regularized deep neural network. *Applied Sciences*, 12(15):7518.

Dena F Mujtaba and Nihar R Mahapatra. 2024. Fairness in ai-driven recruitment: Challenges, metrics, methods, and future directions. *arXiv preprint arXiv:2405.19699*.

Travis Munyer, Abdullah All Tanvir, Arjon Das, and Xin Zhong. 2024. Deeptextmark: a deep learning-driven text watermarking approach for identifying large language model generated text. *Ieee Access*, 12:40508–40520.

Tempestt Neal, Kalaivani Sundararajan, Aneez Fatima, Yiming Yan, Yingfei Xiang, and Damon Woodard. 2017. Surveying stylometry techniques and applications. *ACM Comput. Surv.*, 50(6).

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

Lip Yee Por, KokSheik Wong, and Kok Onn Chee. 2012. Unispach: A text-based data hiding method using unicode space characters. *Journal of Systems and Software*, 85(5):1075–1082.

Martin Potthast, Sarah Braun, Tolga Buz, Fabian Duffhauss, Florian Friedrich, Jörg Marvin Gülzow, Jakob Köhler, Winfried Lötzsch, Fabian Müller, Maike Elisa Müller, Robert Paßmann, Bernhard Reinke, Lucas Rettenmeier, Thomas Rometsch, Timo Sommer, Michael Träger, Sebastian Wilhelm, Benno Stein, Efstathios Stamatatos, and Matthias Hagen. 2016. Who wrote the web? revisiting influential author identification research applicable to information retrieval. In *Advances in Information Retrieval*, pages 393–407, Cham. Springer International Publishing.

Shushanta Pudasaini, Subarna Shakya, Sagar Lamichhane, Sajjan Adhikari, Aakash Tamang, and Sujan Adhikari. 2021. Scoring of resume and job description using word2vec and matching them using gale–shapley algorithm. In *Expert Clouds and Applications: Proceedings of ICOECA 2021*, pages 705–713. Springer.

B Vamsi Nath Reddy and S Parthiban. 2025. Recognizing the required and skill-based resume to the companies with bert compared with random forest. In *AIP Conference Proceedings*, volume 3270, page 020155. AIP Publishing LLC.

Tal Schuster, Roei Schuster, Darsh J Shah, and Regina Barzilay. 2020. The limitations of stylometry for detecting machine-generated fake news. *Computational Linguistics*, 46(2):499–510.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, et al. 2019. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*.

Efstathios Stamatatos. 2009. A survey of modern authorship attribution methods. *J. Assoc. Inf. Sci. Technol.*, 60:538–556.

Ruixiang Tang, Qizhang Feng, Ninghao Liu, Fan Yang, and Xia Hu. 2023. Did you train on my dataset? towards public dataset protection with cleanlabel backdoor watermarking. *ACM SIGKDD Explorations Newsletter*, 25(1):43–53.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Shikha Tewari, Dr Amit Joshi, and Deeksha Tewari. 2024. Ai powered hr: impact, benefits and challenges. *BENEFITS AND CHALLENGES (October 17, 2024)*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

KiYoon Yoo, Wonhyuk Ahn, Jiho Jang, and Nojun Kwak. 2023. Robust multi-bit natural language watermarking through invariant features. *arXiv preprint arXiv:2305.01904*.

A. Prompts for the Experiments

In order to generate AI-generated resumes based on the job descriptions, we used the following prompt:

I am applying for the following job: [job_title]
[copy-pasted job_description]
Generate the perfect resume for the above job description

In order to enhance resumes using LLMs based on the job descriptions, we used the following prompt:

Prompt 1: *Read the attached resume and wait for my prompt: [attached resume]*

Prompt 2: *I am applying for the following job: [job_title]. Please enhance and optimize this resume to make it more compelling and relevant to [job_title]. Keep the improved version professional and concise. Return only the revised resume (no commentary), two pages, single-column, professional layout.*

B. Supplementary Exploratory Analyses: Readability, Length, and Phrase Structure

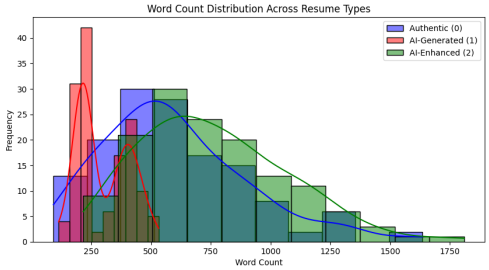

Figure 11: Word count distribution across resume types.

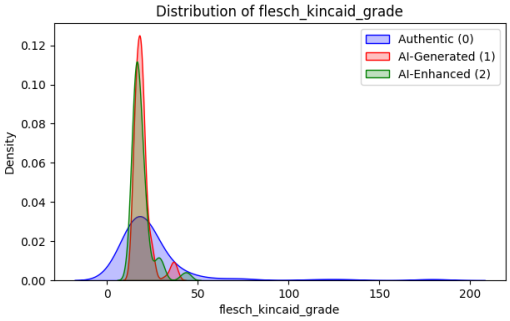

Figure 12: Flesch-Kincaid grade distribution.

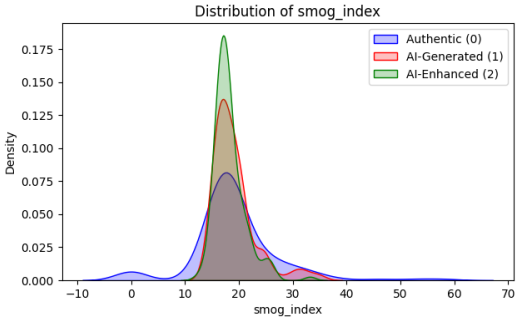

Figure 13: SMOG index distribution.

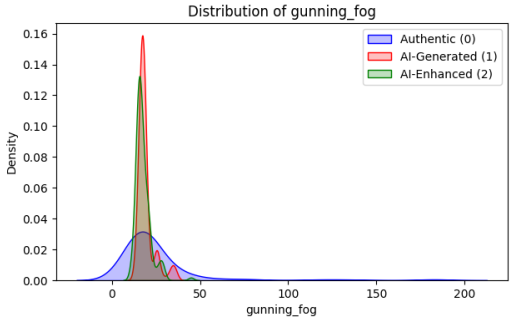

Figure 14: Gunning Fog index distribution.

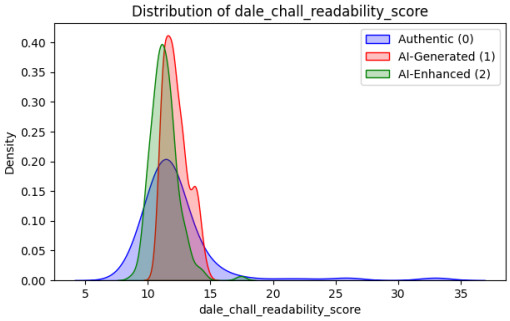

Figure 18: Dale-Chall readability score distribution.

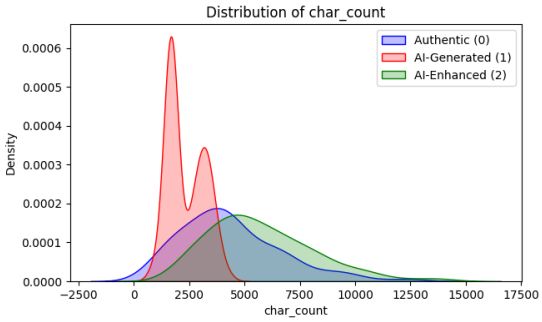

Figure 15: Character count distribution.

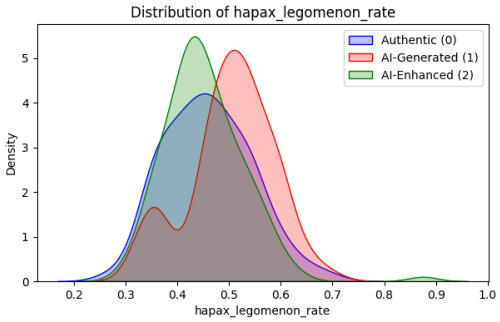

Figure 19: Hapax Legomenon Rate distribution.

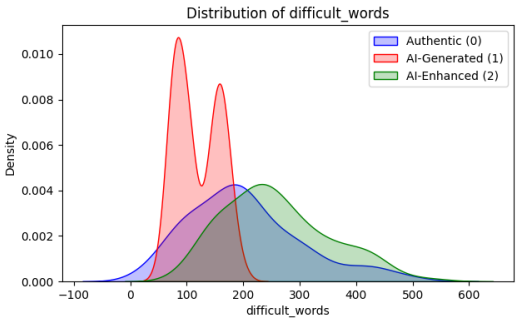

Figure 16: Difficult words distribution.

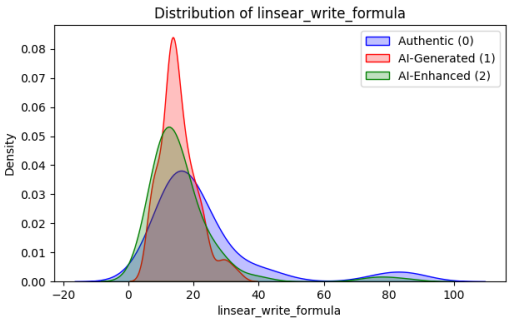

Figure 20: Linsear Write Formula distribution.

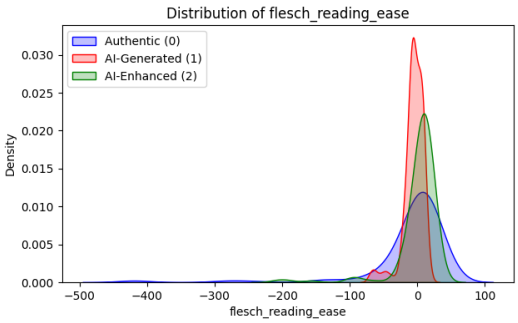

Figure 17: Flesch Reading Ease distribution.

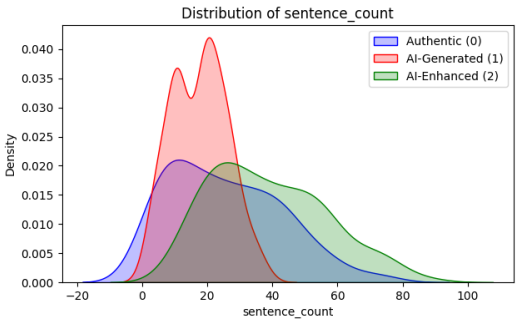

Figure 21: Sentence count distribution.

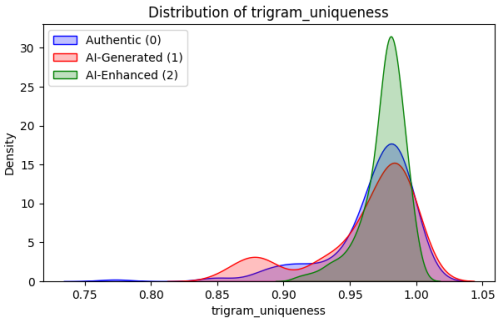

Figure 22: Trigram uniqueness distribution.

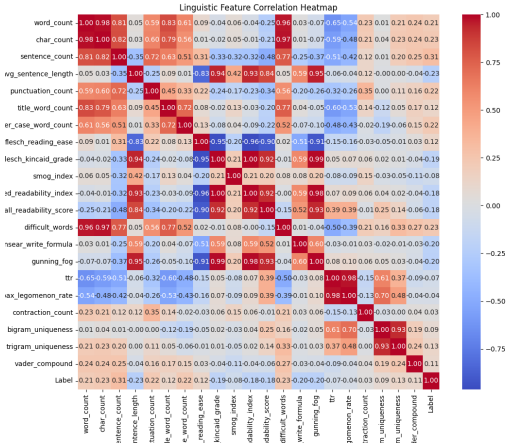

Figure 23: Feature correlation heatmap.