AI-Driven Resume Analysis and Enhancement Using Semantic Modeling and Large Language Feedback Loops

Achal Jagadeesh[1], Chinmayi Ravi Shankar[1], Sahithya Narayanaswamy Patel[1], Marco Polignano[2], Marco Levantesi[1,3], Giovanni Semeraro[2] and Ernesto William De Luca[1,3]

[1]*Otto-von-Guericke University, Universitätspl. 2, 39106 Magdeburg, Germany*
[2]*University Of Bari Aldo Moro, via E. Orabona 4, 70125, Bari, Italy*
[3]*Leibniz Institute for Educational Media | George Eckert Institute, Brunswick, Germany*

Abstract

Fairness is increasingly elusive in the current landscape of Artificial Intelligence and Large Language Models. These technologies can easily inject fake or inaccurate information into the data, often misrepresenting what truly exists. This problem is widely spread in many domain applications, including those dealing with user profiles. In particular, in the job market, this affects both recruiters and job seekers. Resumes are frequently optimized to fit the job call in rather than to reflect genuine qualifications, while automated screening tools may overlook authentic but non-standard profiles. This work proposes a resume analysis and enhancement system. It enables iterative improvement through the use of Large Language Models while preserving the original content. This leads to a consistent improvement in similarity and match quality with job applications. Fairness is achieved not by altering who the candidate is, but by ensuring their actual capabilities are accurately and contextually recognized, thus empowering both evaluators and applicants through authentic enhancement.

Keywords

Resume enhancement, ATS, NLP, semantic similarity, ethical AI, Sentence Transformers, fairness, GPT, LLaMA

1. Introduction

AI-driven resume screening systems, currently widely adopted in company recruitment scenarios, have redefined the process of candidate evaluation[1]. While these systems possess scalability and consistency, they often prioritize standardization over content [2, 3, 4]. As a result, applicants are implicitly encouraged to fit into rigid patterns using standard templates, inflated action verbs, and keyword-dense summaries that align with the parsing logic of Applicant Tracking Systems (ATS). This leads to a recruitment ecosystem where many resumes are optimized to pass automated filters rather than to authentically represent the candidate's qualifications, context, or potential. Such practices introduce a significant and often unacknowledged issue: *fairness*. In current automated systems, fairness is equated with the uniform application of algorithms[5]. However, uniformity is not the same as equity. Two candidates who pursue similar competencies may be treated differently based on how closely their resumes reflect the expected linguistic and structural patterns. Those from non-traditional backgrounds, interdisciplinary fields, or regions with different resume norms may be penalized due to the limitations of automated parsing logic rather than lack of ability. Moreover, candidates may feel compelled to deviate or artificially restructure their narratives just to be considered by the system [6].

This work presents a resume analysis and enhancement system designed around the principle of *"contextual fairness"* [6]. The system avoids modifying or artificially enhancing a candidate's narrative. Instead, it enhances what is already present suggesting section-wise improvements that improve clarity, alignment, and structure without distorting meaning. All suggestions are non-prescriptive and allow the candidate full control over the integration. To achieve this, the system employs two complementary AI components. A Sentence Transformer model (i.e., *"multi-qa-MiniLM-L6-cos-v1"*) [7] computes the semantic similarity between resume content and job descriptions. This enables the system to assess how well the candidate's wording aligns with the job description. Alongside this, an instruction-tuned LLaMA 3.2 model[8] generates fine-grained enhancement suggestions for individual resume sections such as Skills, Experience, and Summary. These suggestions are tailored to the job description's context but preserve the candidate's originality, offering ways to surface hidden strengths or clarify vague phrasing. The result is a system that recognizes intent and potential supporting candidates in expressing their capabilities authentically and enabling recruiters to evaluate resumes on substance rather than style. In a landscape increasingly shaped by automation, this approach represents a shift from optimization toward interpretation and from filtering toward understanding.

CLiC-it 2025: Eleventh Italian Conference on Computational Linguistics, September 24 — 26, 2025, Cagliari, Italy
achal@st.ovgu.de (A. Jagadeesh); chinmayi.ravi@st.ovgu.de (C. R. Shankar); sahithya.narayanaswamy@st.ovgu.de (S. N. Patel); marco.polignano@uniba.it (M. Polignano)

2. Related work

Traditional Applicant Tracking Systems (ATSs) rely on keyword-based filtering [9], which fails to capture the contextual nuances in resumes, leading to biased or inaccurate candidate evaluations. Recent approaches leverage transformer-based models to assess semantic similarity between resumes and job descriptions. Resume2Vec [9] introduced a framework using models like BERT, RoBERTa, and LLaMA [9] to generate embeddings and improve candidate–job alignment through cosine similarity. Their system outperformed conventional ATSs in both ranking accuracy and alignment with human judgment across multiple domains. Unlike keyword-centric methods, Resume2Vec emphasizes context and fairness by preserving the semantic richness of candidate data. This shift toward embedding-based analysis lays the foundation for more equitable and intelligent recruitment systems.

Lavi et al. (2021) [10] introduced conSultantBERT, a fine-tuned Siamese Sentence-BERT model tailored for resume-job matching, addressing challenges such as data heterogeneity, cross-linguality, and noisy resume formats. By leveraging cosine similarity between multilingual embeddings, their model significantly outperformed both TF-IDF and pre-trained BERT baselines in predicting resume-vacancy matches. Their findings affirm the importance of domain-specific fine-tuning to preserve semantic integrity in candidate profiles while improving matching accuracy. Like our system, conSultantBERT emphasizes contextual matching without resorting to superficial keyword overlap, highlighting the role of semantically grounded embeddings in achieving fair and scalable recruitment solutions. While conSultantBERT focuses on semantic matching between resumes and job descriptions using fine-tuned embeddings, our approach not only evaluates similarity but also provides customized resume enhancements using LLMs

Yadav et al. (2025) [11] developed a rule-based resume analysis system that integrates NLP and ATS scoring to enhance automated screening efficiency. Their system parses structured resume data and ranks candidates using metrics such as word count, skill match, and experience, delivering real-time feedback and improvement suggestions. While effective in increasing screening speed and ATS alignment, the model primarily focuses on formatting and keyword optimization. In contrast, **our work emphasizes semantic fairness by maintaining candidate authenticity**, going beyond surface-level optimizations to contextualize and enhance genuine qualifications[9, 12].

Gan et al. (2024) [13] proposed a resume screening framework based on large language models (LLMs), utilizing agents such as LLaMA2 and GPT-3.5 to automate resume classification, scoring, and summarization. Their system is designed for high-throughput resume analysis, offering structured outputs that assist recruiters in candidate filtering. Similar to our work, their approach uses instruction-tuned LLMs for interpreting and processing resume content. However, the two systems diverge significantly in purpose and design philosophy. While Gan et al. focus on classification and summarization to streamline hiring pipelines, our system emphasizes "contextual fairness"—providing non-intrusive, section-wise suggestions that retain the candidate's narrative integrity. Instead of generating summaries or altering resume tone, our system enhances clarity and alignment using a hybrid model architecture: Sentence-Transformers multi-qa-MiniLM-L6-cos-v1[7] for semantic similarity scoring and LLaMA 3.2[8] for targeted feedback. However other LLMs (i.e. LLaMantino [14, 15]) or embedding strategies [16] cold be simply adopted by changing few lines of code.

3. Methodology

Our framework follows a pipeline with consecutive steps Figure 1. Such pipeline begins by taking in two primary inputs: ***the resume*** uploaded by the job seeker and the ***job description*** submitted by the recruiter.

Resume upload and processing. We support a resume uploading process for documents in Word (i.e. ".docx" extension) or PDF (i.e. ".pdf" extension) format. The system uses *python-docx* [1] for Word documents and *pdfplumber*[2] for PDFs. These libraries enable accurate extraction of plain text and preserves section structure as well as formatting semantics. Each parsed resume is stored in a document database (Firestore DB and Storage) alongside unique metadata including a *resume identifier*, *user email*, *timestamp*, and a designated *resume name* for future tracking and analysis purposes.

Job Description Submission and Structuring. Recruiters provide job descriptions through a structured template by inputting key fields such as job title, required experience, skills, responsibilities, and domain focus areas (e.g., questionnaireFocus)[3]. These structured fields are flattened into a consolidated textual representation, which makes them compatible with vector-based semantic models and term-frequency-based keyword extraction. To maintain consistency and modularity, the flattened job description is stored in parallel with its structured form within the same database, under a unique job identifier. This kind of dual representation allows the system to dynamically switch between structured access[17] (e.g.,

[1]https://python-docx.readthedocs.io/en/latest/
[2]https://pypi.org/project/pdfplumber/
[3]Currently, such aspects are not automatically extracted from the job position but we consider to do that as a future work.

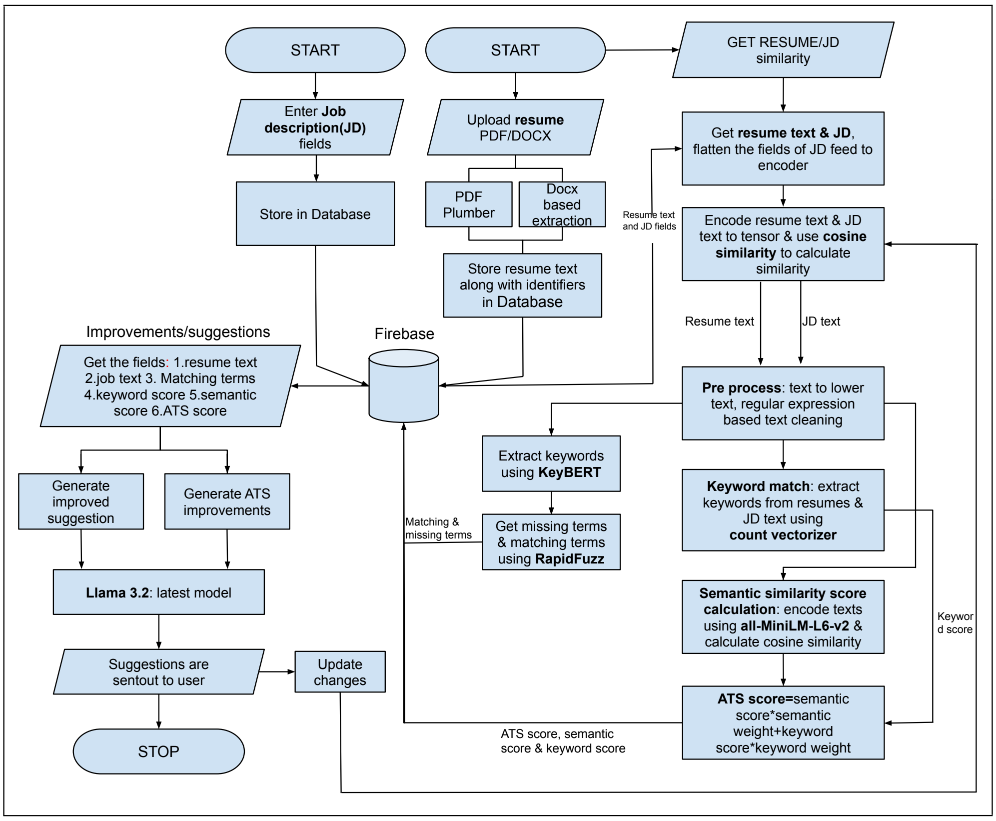

Figure 1: Flow diagram - Schematic flow diagram of the AI Resume Analyzer and Enhancer system. The process starts with uploading a resume (PDF/DOCX) and entering a job description. Extracted resume and job description texts are preprocessed and analyzed using two independent transformer models to compute semantic similarity, keyword relevance, and a final weighted ATS score. KeyBERT and RapidFuzz handle context-aware keyword extraction and matching. The pipeline also invokes LLaMA 3.2 to generate resume improvement suggestions and ATS feedback, which are stored and sent to users for review and updates.

for displaying details or generating questionnaires) and unstructured access (e.g., for semantic similarity and ATS scoring).

Data Cleansing. Both the resume and job description texts are normalized by converting to lowercase and applying regular expression-based cleaning into alphanumeric characters. This removes extraneous symbols, spacing irregularities, and control characters, ensuring input consistency before model encoding. This initial acquisition and preparation phase ensures that both resumes and job descriptions are available in clean, comparable formats for downstream tasks such as similarity computation, keyword relevance analysis, and improvement suggestion generation.

3.1. Similarity, Keyword Score and ATS Score Calculation

After resumes and job descriptions have been ingested and preprocessed, the system performs a multi-level alignment assessment through *semantic similarity* and *keyword relevance scores.* This step is central to producing a fair and interpretable *Applicant Tracking System (ATS) score* that reflects both explicit and contextual alignment between candidate profiles and job requirements.

Semantic Similarity Score. To ensure robustness and fairness in semantic evaluation, the system leverages two independent transformer models from the `SentenceTransformers` library [4]:

[4] https://sbert.net/

- `multi-qa-MiniLM-L6-cos-v1`[5]
- `all-MiniLM-L6-v2`[6]

Each model independently encodes the cleaned resume text and job description text into tensor embeddings. Cosine similarity[18, 19] is then computed between these vectors to assess semantic alignment. If one model underperforms or introduces bias [20] in representation (e.g., due to phrasing variance), the other acts as a fallback, promoting score stability and fairness across domains and candidate profiles[10]. The all-MiniLM-L6-v2 model is used for ATS score calculation[21] due to its balanced ability to capture both semantic meaning and keyword-level relevance, making it ideal for evaluating overall resume compatibility. Meanwhile, multi-qa-MiniLM-L6-cos-v1 is reserved for pure semantic similarity scoring, as its QA-focused fine-tuning excels at understanding contextual alignment between resumes and job descriptions. This separation ensures accurate, fair, and domain-robust evaluations.

Keyword Relevance Score. Keyword-based scoring complements semantic alignment by focusing on lexical overlap. This scoring process follows the steps described below: (i) Initial Extraction. The job description is vectorized using `CountVectorizer` from `sklearn.feature_extraction.text`[7], allowing direct term frequency analysis. (ii) Resume Keyword Extraction.Resume keywords are extracted using `KeyBERT`[8], which identifies *top N* significant phrases based on contextual embedding similarity. Keyword extraction is essential and plays a vital role in ensuring fairness during evaluation. As shown in Figure 1 the extracted matching keywords are utilized by the LLM to generate context-aware suggestions, providing targeted improvements that align more closely with the job description. This step enhances both the relevance and fairness of the feedback provided to users.

Two different keyword extraction approaches are used to account for the inherent differences in data structure and consistency. Job descriptions are entered by users in a structured JSON format and are generally concise and standardized, making them ideal for keyword extraction using CountVectorizer, which captures raw term frequencies. In contrast, resumes are uploaded as binary files (PDF or DOCX) and converted to plain text, often in an unstructured and inconsistent manner - hence, KeyBERT is employed to extract context-aware key phrases using semantic embeddings, ensuring reliable keyword identification despite formatting noise or phrasing variability.

[5]https://huggingface.co/sentence-transformers/multi-qa-MiniLM-L6-cos-v1
[6]https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2
[7]https://scikit-learn.org/stable/modules/feature_extraction.html
[8]https://pypi.org/project/keybert/

Matching Score. The set intersection between extracted resume keywords and job description keywords is used to calculate a match ratio:

$$\text{match\_score} = \frac{|\text{matched\_keywords}|}{|\text{job\_keywords}|} \quad (1)$$

Fuzzy Matching. To account for synonyms and approximate matches, the system additionally uses `RapidFuzz`[9] to detect partial matches between keywords, further refining the keyword score. To account for synonyms, spelling variations, and approximate matches, the system incorporates RapidFuzz, a fast string matching library based on Levenshtein distance. RapidFuzz computes partial similarity ratios between extracted keywords from the job description and the resume, helping detect near-matches even when exact wording differs. This refinement step enhances the keyword score accuracy by capturing relevant but variably phrased skills or experiences.

Applicant Tracking System (ATS) Score. The final ATS score is computed as a weighted sum of semantic similarity and keyword relevance scores:

$$\text{ATS\_score} = (\text{sem\_score} \cdot w_1) + (\text{keyword\_score} \cdot w_2) \quad (2)$$

Where:

- w_1 = semantic weight (default: 0.5)
- w_2= keyword weight (default: 0.5)

This hybrid scoring formula balances surface-level term relevance with deep contextual alignment. By assigning separate weights, the system allows recruiters to prioritize either direct skill inclusion or holistic candidate-job compatibility.

The calculated ATS score serves as a crucial factor for both recruiters and job seekers by helping recruiters efficiently shortlist candidates based on relevance, while guiding job seekers in optimizing their resumes. Unlike traditional systems that rely solely on keyword matching, this score combines keyword relevance with semantic similarity, capturing not just the presence of required terms but also the contextual alignment between the resume and job description. This hybrid approach ensures greater fairness, adaptability across domains, and reduced bias, making it more insightful than conventional ATS scores that often overlook phrasing variations or implied competencies.

To prevent artificial score inflation and preserve candidate authenticity, the system avoids injecting new keywords or altering the resume's core content. Instead, it focuses on identifying and enhancing existing expressions—both semantically and lexically ensuring fairness to the job seeker while giving recruiters a transparent, accurate alignment signal.

[9]https://rapidfuzz.github.io/RapidFuzz/

3.2. Suggestion Generation and Section-Wise Improvements

To enhance specific resume sections by rephrasing, clarifying, or restructuring them using best practices in resume writing we design an enanchement step grounded on Large Language Models (LLMs) (i.e., `generate_improved_sections_with_llm`). It focuses on strengthening the candidate's input by:

- Reinforcing matched keywords in previous ATS Score Calculation step
- Improving clarity and formatting of the resume
- Highlighting quantifiable impacts and action-driven phrasing actions
- Focusing on improving sections like *Professional Summary, Experience, Skills, and Education*

A structured prompt is generated (see Table 1), including the resume text, job description, and the list of already matched keywords. The LLM is explicitly instructed not to introduce missing or hallucinated terms, ensuring that improvements remain factual and grounded in the candidate's original input. The model returns suggestions in strict JSON format, each linked to a specific resume section for traceable integration. We provide the LLM with the flattened job description and resume text, along with the matchingTerms, similarityScore, and atsScore, to give it fuller context for generating accurate, traceable suggestions. We pass the flattened job description and flattened resume text, matchingTerms, similarityScore, atsScore are also passed so that we are giving more context to the LLM model - LLaMA 3.2 latest.

Field	**Description**
Resume Text	Extracted, cleaned resume text uploaded by the user.
JobDescription	Flattened string of title, skills, experience, and role.
MatchedKeywords	Key terms found in both the resume and job description.
ExplicitInstructions	Directs model to avoid hallucination and ensure factual edits only.
OutputFormat	JSON array with fields: `sectionName`, `suggestion`.

Table 1
Elements of LLM prompt used for generating grounded, section-wise resume improvement suggestions.

The second service we designed, `generate_ats_score_and_improvements`, operates at a global resume level rather than focusing on specific sections. It analyzes the entire resume in the context of the job description and the list of matched keywords but deliberately avoids altering content or injecting new, unverified terms. Instead, it identifies opportunities for structural and stylistic enhancements that can improve ATS performance without compromising authenticity.

Its outputs include:

- Parsing the resume text and job description.
- Evaluating aspects like formatting consistency (e.g., bullet points, section headers), action verb usage, and sentence clarity.
- Referencing the matched keywords to ensure better usage and placement, rather than adding unrelated terms.
- Returning the output in a strict JSON structure, which includes: (i) A list of factual, actionable suggestions; (ii) An estimated ATS score; (iii) Highlighted areas where improvements can be made to enhance readability and alignment.

This makes the output easily integrable into the system while keeping the suggestions grounded in the candidate's original input and safe from hallucinations.

Fairness and Transparency Considerations. Both services are governed by strict instruction constraints to:

- Prevent hallucination of unverified skills
- Avoid inflating match quality with artificial edits
- Respect candidate identity and experience as originally stated

By focusing solely on strengthening existing, verifiable content, this dual-LLM framework ensures that suggestions are ethical, transparent, and aligned with fair AI principles - providing job seekers with meaningful improvement pathways without compromising truthfulness.

4. Experimental Evaluation

To test the proposed approach, we decided to design and run two separate experiments to evaluate, how fair the process is and how effective it is.

4.1. Experiment 1: Fairness-Aware Resume Enhancement

This experiment evaluates whether resumes can be ethically enhanced to better align with job descriptions, without introducing fabricated content or misleading embellishments. The objective is to test whether a candidate's original experience and qualifications can be made more contextually relevant while preserving the integrity and authenticity of the resume. A representative set of 10 manually crafted synthetic resumes (refer Tables 3, 4 in appendix and for column name descriptions refer Table 5) were selected and evaluated against a curated synthetic job description for the role ReactJS Frontend Developer (API Integration & UI Frameworks) using four key metrics: *similarity score*, *ATS score*, *matching terms*, and *missing terms*. These metrics were computed against target job descriptions which were manually crafted by analyzing real listings for similar job roles. While individual scores may vary, the relative differences (score deltas) remain consistent across resumes. The resumes were then enhanced using our LLM-powered suggestion engine, which provides section-wise recommendations based solely on the candidate's original content and job relevance. Enhanced resumes were re-evaluated with the same approach previously used, for observing: (i) Increases in similarity and ATS scores; (ii) Growth in contextually valid matching terms; (iii) Retention of semantic integrity (i.e., no direct insertion of previously missing terms unless already implied).

Fairness Criteria. To ensure ethical enhancement, the system followed three key constraints:

- All newly introduced terms had to be contextually consistent with the original resume.
- Terms from the initial missing terms list were disallowed unless semantically implied or rephrased from existing content.
- No artificial keyword stuffing or hallucinated experiences were permitted.

The improvements were evaluated using changes in matching terms and missing terms metrics computed by comparing keyphrases from the job description with the resume text before and after enhancement (refer Table 2). These metrics served as our primary quantitative evidence, ensuring that enhancements improved alignment without introducing unrelated or fabricated content, as the suggestion engine operated strictly within the resume's original context.

Experimental results. Following enhancement using our system, all resumes demonstrated meaningful improvements while preserving fairness and integrity. New matching terms were successfully added in every case, and all additions were contextually aligned with the original resume content. Crucially, none of the original missing terms were directly reused, and no hallucinated or unrelated information was introduced (refer Table 3 and 4 in appendix). The outcomes of Experiment 1, which involved evaluating ten candidate resumes for the ReactJS Frontend Developer position, are summarized in Tables 3 and 4 in appendix. While both LLaMA 3.2 and GPT-4o raise the overall match counts, the `New_Terms_Added_by_LLaMA3.2` column grows only with fair, semantically grounded additions. In contrast, `New_Terms_Added_by_GPT-4o` reflects GPT-4o's blind injections of extra keywords—demonstrating how our approach upholds fairness by restricting edits to what the candidate's own language can support. The experiment confirms that our system provides significant improvements while maintaining fairness, i.e., enhancing the resume without misrepresenting the candidate's skills or experience.

4.2. Experiment 2: Effectiveness Comparison

This experiment compares the effectiveness of two resume enhancement strategies, both operating under strict non-hallucination constraints. The first method uses our domain-specific LLM-powered suggestion engine to improve resume-job alignment while preserving the candidate's original intent and language. The second method uses a general-purpose GPT-4o model instructed to rewrite resumes without adding any content not originally present. Each resume was evaluated in three forms: the original version, a system-enhanced version using our custom enhancement engine, and a GPT-enhanced (GPT-4o)[10] version rewritten by a large language model under strict non-hallucination instructions. All three versions were analyzed using the same backend evaluation pipeline (refer Figure 1) to compute similarity score, final ATS score, semantic similarity score, and keyword match score.

Experimental results. The system-enhanced resumes consistently outperformed the original versions in all key metrics. The summary of ATS and Similarity Scores (in %) Across Resume Enhancement Systems can be seen in Table 6. On average, similarity scores improved by 18.7% and ATS scores rose by 22.3% following enhancement. When comparing system-enhanced resumes to GPT-enhanced counterparts, our method achieved higher average similarity scores (43.52% vs. 34.13%) and comparable semantic similarity scores (46.77% vs. 47.21%), despite the GPT-enhanced versions showing a higher final ATS score (74.25%). However, a deeper inspection of the results reveals that the elevated ATS scores in GPT-enhanced resumes may be attributed to broader keyword coverage rather than meaningful contextual alignment. In Figure 1, once the suggestions from our system are updated, a parallel process generates and applies suggestions using Chat GPT-4o as well. Both updated versions, the one based on our system's suggestions and the one generated from GPT-4o's recommendations, are then re-evaluated. A comparison spreadsheet is generated containing the results of both evaluations, highlighting differences in ATS scores, similarity scores, and overall improvements. The ATS scores and similarity scores comparison across enhancement Systems can be seen

[10]https://openai.com/index/hello-gpt-4o/

Table 2
Comparison of matching terms across resume enhancement systems

Resume ID	Original_ Matching_ Terms	Original_ Missing_ Terms	LLaMA3.2_ Matching_ Terms	New_Terms_ Added_by_ LLaMA3.2	GPT-4o_ Matching_ Terms	New_Terms_ Added_by_ GPT-4o
resume_2_7_V7	frontend, react, developer, expertise, apis, ui, jest	reactjs, backend, skilled, freelance, axios, frameworks, typescript, redux, es6, components, component, agile, development	reactjs, frontend, react, developer, frameworks, expertise, apis, components, component, ui, jest, development	component, development, reactjs, components, frameworks	reactjs, frontend, react, skilled, developer, axios, frameworks, typescript, expertise, apis, redux, es6, components, component, ui, jest, agile, development	component, development, redux, reactjs, typescript, es6, axios, agile, components, frameworks, skilled
resume_2_6_V7	frontend, react, backend, skilled, developer, expertise, apis, ui, jest	reactjs, freelance, axios, frameworks, typescript, redux, es6, components, component, agile, development	reactjs, frontend, react, backend, skilled, developer, axios, expertise, apis, components, component, ui, jest, development	component, development, reactjs, axios, components	reactjs, frontend, react, backend, skilled, developer, axios, frameworks, typescript, expertise, apis, redux, es6, components, component, ui, jest, agile	component, development, redux, reactjs, typescript, es6, axios, agile, components, frameworks
resume_2_1_V7	frontend, react, developer, expertise, apis, ui, jest	reactjs, backend, skilled, freelance, axios, frameworks, typescript, redux, es6, components, component, agile, development	frontend, developer, frameworks, expertise, apis, redux, ui, jest, development	redux, development, frameworks	reactjs, frontend, react, backend, developer, axios, frameworks, typescript, expertise, apis, redux, es6, components, component, ui, jest, agile	component, redux, reactjs, es6, typescript, axios, backend, agile, components, frameworks

in Table 6. The system-enhanced resumes maintained a more focused and candidate-authentic tone while still improving discoverability (refer Table 3 and Table 4 in Appendix). In multiple cases, the system-enhanced versions outperformed GPT in similarity score by margins exceeding 16 percentage points, with the highest observed gain reaching 29.07% (refer Table 6).

Figure 2 shows that the augmented similarity scores (system_updated_similarityScore) markedly exceed both the baseline (original_similarityScore) and the GPT-4o derived scores (chatgpt4o_updated_similarityScore), indicating that our LLaMA 3.2–based methodology predicated on conservative, in situ enhancement of existing text yields the most substantial improvements in semantic alignment between resumes and job descriptions. Although GPT-4o's outputs show notable improvements over the unmodified baseline, they still fall short of the results achieved by our system. This supports the effectiveness of a fairness-oriented framework that prioritizes refining existing content rather than introducing extraneous terms.

In Figure 3 the bar chart compares ATS scores for ten resumes across three conditions: the original unmodified documents (blue bars), the LLaMA 3.2-based update methodology (red bars), and GPT-driven enhancements (green bars). LLaMA 3.2 updates yield the highest improvements boosting scores from approximately 18–25 at baseline to 35–55, whereas GPT enhancements produce moderate gains, raising baseline values to roughly 26–48. In every case, the LLaMA 3.2–adjusted resumes outper-

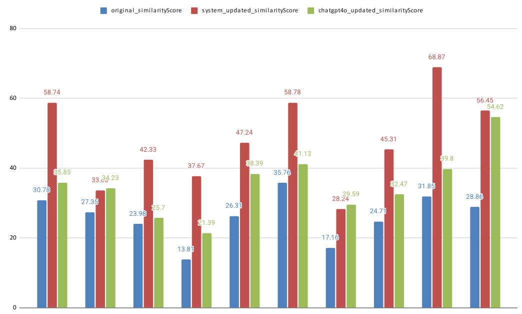

Figure 2: Similarity Score Comparison

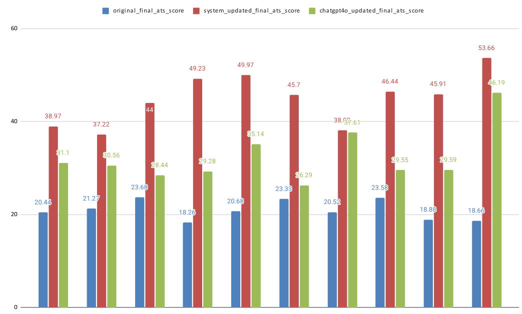

Figure 3: ATS Score Comparison

form both the original and GPT-enhanced versions, with the latter still delivering a substantial uplift relative to unmodified resumes.

Furthermore, the system's enhancements did not introduce any hallucinated content and preserved the resume's original structure and voice (refer Table 3 and Table 4 in Appendix). In contrast, GPT-enhanced rewrites, while constrained, occasionally drifted toward generalized language or tone inconsistencies. These observations reinforce the value of targeted, context-aware enhancement over generalized rewriting approaches.

5. Considerations and Limitations

The results from our experiments highlight the efficacy and robustness of the proposed AI-powered resume enhancement system, especially in terms of fairness, contextual integrity, and practical relevance for applicant tracking systems (ATS).

Fairness and Authenticity Preservation. Experiment 1 demonstrated that our system can meaningfully enhance resumes by adding relevant matching terms without compromising fairness or authenticity. The fact that none of the original missing terms were reused and no hallucinated or unrelated information was introduced is particularly encouraging. This shows that the system respects the candidate's true skills and experiences, avoiding unethical exaggeration or fabrication—a critical requirement in AI-assisted recruitment tools. The average improvements of 18.7% in semantic similarity and 22.3% in ATS scores indicate that the enhancements not only preserve but also amplify the relevance of candidate profiles to job descriptions, improving their discoverability without sacrificing honesty.

This balance between enhancement and fairness is a key differentiator compared to many automated systems that risk introducing biases or misrepresentations. The strict adherence to defined fairness criteria ensures the tool's suitability for real-world applications where ethical standards are paramount.

Comparative Effectiveness and Contextual Alignment. Experiment 2's comparative analysis between our system and GPT-based enhancements further reinforces the strengths of our approach. While GPT-enhanced resumes sometimes achieved higher ATS scores—likely due to broader keyword coverage—the system-enhanced resumes consistently showed superior or comparable semantic similarity scores, indicating a closer contextual match to the original resumes.

This distinction is important: higher ATS scores alone do not guarantee a better quality or more truthful resume. The tendency of GPT-based rewrites to introduce generalized language or tone inconsistencies could dilute the candidate's unique profile, potentially reducing perceived authenticity. In contrast, our system's targeted, context-aware enhancements retain the original voice and structure, offering improvements that are both meaningful and aligned with the candidate's actual background. The observed margin of improvement in similarity scores (up to 29.07 percentage points over GPT in some cases) suggests that our method excels at fine-grained semantic enhancement rather than broad-stroke rewriting. This focused approach is likely to yield better candidate-job matching outcomes in ATS environments that value precise and relevant keyword and phrase usage.

Additional limitations include the need for improved performance in domain-specific contexts, sensitivity to input formats, and the lack of multilingual support. Ethical concerns around bias, transparency, and resume over-optimization also warrant future exploration. Ensuring fairness, explainability, and data privacy in deployment environments will be crucial to responsible adoption [22].

While the system shows promising results, some areas merit further attention. Current performance is strongest on English-language resumes with consistent formatting;

improving support for varied layouts and multilingual inputs is a valuable direction. Our evaluation, centered on synthetic resumes for a specific domain (Frontend ReactJS), provides a solid foundation but would benefit from broader validation across job types and real-world data. Additionally, while basic bias detection is included, more comprehensive fairness auditing remains an important avenue for future development. As with all LLM-enhanced systems, results may vary slightly based on the quality of job description inputs. Addressing these aspects can help increase the system's robustness, fairness, and generalizability.

6. Conclusion

This project demonstrates that our AI-powered resume enhancement system effectively improves resume quality while upholding fairness and authenticity. By preserving resume integrity—without adding fabricated keywords or skills—the system consistently adds contextually relevant terms, resulting in substantial improvements in semantic similarity (18.7%) and ATS scores (22.3%). Compared to GPT-based rewrites, our approach achieves higher or comparable semantic alignment while maintaining the candidate's original voice and structure, avoiding generalized or inconsistent language. These findings highlight the advantage of targeted, context-aware enhancement methods that responsibly boost candidate discoverability and preserve authenticity. Consequently, our LLM-based enhancement system offers a practical, ethical, and superior solution for real-world recruitment pipelines. Future improvements could include support for multilingual resumes and enhanced robustness for unstructured or poorly formatted inputs.

7. Acknowledgments

This research is partially funded by PNRR project FAIR - Future AI Research (PE00000013), Spoke 6 - Symbiotic AI (CUP H97G22000210007) under the NRRP MUR program funded by the NextGenerationEU.

References

[1] A. Yerkebulan, M. Mansurova, A. Abdildayeva, O. Sharip, Research on the application of large language models (llm) for improving recruitment efficiency and accuracy, in: N. T. Nguyen, T. Matsuo, F. L. Gaol, Y. Manolopoulos, H. Fujita, T.-P. Hong, K. Wojtkiewicz (Eds.), Intelligent Information and Database Systems, Springer Nature Singapore, 2025, pp. 331–344. doi:10.1007/978-981-96-6008-7_24.

[2] S. Bharadwaj, R. Varun, P. Aditya, M. Nikhil, G. Babu, Resume screening using nlp and lstm, in: 2022 Fifth International Conference on Inventive Computation Technologies (ICICT), IEEE, 2022, pp. 238–241. URL: https://ieeexplore.ieee.org/document/9850889. doi:10.1109/ICICT54344.2022.9850889.

[3] G. Navarro, Fair and ethical resume screening: Enhancing ats with justscreen the resume-screeningapp, Journal of Information Technology, Cybersecurity, and Artificial Intelligence 2 (2025) 1–7. doi:10.70715/jitcai.2024.v2.i1.001.

[4] B. A. T. Dilshan, P. P. G. D. Asanka, Enhancing resume analysis: Leveraging natural language processing and machine learning for automated resume screening using ksa parameters, in: 2025 5th International Conference on Advanced Research in Computing (ICARC), 2025, pp. 1–6. doi:10.1109/ICARC64760.2025.10963312.

[5] A. K. Sinha, M. A. K. Akhtar, A. Kumar, Resume screening using natural language processing and machine learning: A systematic review, in: D. Swain, P. K. Pattnaik, T. Athawale (Eds.), Machine Learning and Information Processing, Springer Singapore, 2021, pp. 207–214. doi:10.1007/978-981-33-4859-2_21.

[6] S. Vaishampayan, S. Farzanehpour, C. Brown, Procedural justice and fairness in automated resume parsers for tech hiring: Insights from candidate perspectives, in: 2023 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), 2023, pp. 103–108. doi:10.1109/VL-HCC57772.2023.00019.

[7] D. Hamzic, M. Wurzenberger, F. Skopik, M. Landauer, A. Rauber, Evaluation and comparison of open-source llms using natural language generation quality metrics, in: 2024 IEEE International Conference on Big Data (BigData), 2024, pp. 5342–5351. doi:10.1109/BigData62323.2024.10825576.

[8] A. Grattafiori, et al., The llama 3 herd of models, 2024. URL: https://arxiv.org/abs/2407.21783. arXiv:2407.21783.

[9] R. V. K. Bevara, N. R. Mannuru, S. P. Karedla, B. Lund, T. Xiao, H. Pasem, S. C. Dronavalli, S. Rupeshkumar, Resume2vec: Transforming applicant tracking systems with intelligent resume embeddings for precise candidate matching, Electronics 14 (2025). URL: https://www.mdpi.com/2079-9292/14/4/794. doi:10.3390/electronics14040794.

[10] D. Lavi, V. Medentsiy, D. Graus, consultantbert: Fine-tuned siamese sentence-bert for matching jobs and job seekers, CoRR abs/2109.06501 (2021). URL: https://arxiv.org/abs/2109.06501. arXiv:2109.06501.

[11] Resume analysis using nlp and ats algorithm, International Journal of Latest Technology in Engineering Management and Applied Science 14 (2025) 761–767. URL: https://www.ijltemas.in/submission/index.php/online/article/view/1937. doi:10.51583/IJLTEMAS.2025.140400090.
[12] C. Daryani, G. Chhabra, H. Patel, I. Chhabra, R. Patel, An automated resume screening system using natural language processing and similarity, 2020, pp. 99–103. doi:10.26480/etit.02.2020.99.103.
[13] C. Gan, Q. Zhang, T. Mori, Application of llm agents in recruitment: A novel framework for resume screening, 2024. URL: https://arxiv.org/abs/2401.08315. arXiv:2401.08315.
[14] P. Basile, E. Musacchio, M. Polignano, L. Siciliani, G. Fiameni, G. Semeraro, Llamantino: Llama 2 models for effective text generation in italian language, CoRR abs/2312.09993 (2023). URL: https://doi.org/10.48550/arXiv.2312.09993. doi:10.48550/ARXIV.2312.09993. arXiv:2312.09993.
[15] M. Polignano, P. Basile, G. Semeraro, Advanced natural-based interaction for the italian language: Llamantino-3-anita, CoRR abs/2405.07101 (2024). URL: https://doi.org/10.48550/arXiv.2405.07101. doi:10.48550/ARXIV.2405.07101. arXiv:2405.07101.
[16] M. Polignano, P. Basile, M. de Gemmis, G. Semeraro, A comparison of word-embeddings in emotion detection from text using bilstm, CNN and self-attention, in: G. A. Papadopoulos, G. Samaras, S. Weibelzahl, D. Jannach, O. C. Santos (Eds.), Adjunct Publication of the 27th Conference on User Modeling, Adaptation and Personalization, UMAP 2019, Larnaca, Cyprus, June 09-12, 2019, ACM, 2019, pp. 63–68. URL: https://doi.org/10.1145/3314183.3324983. doi:10.1145/3314183.3324983.
[17] M. Mochol, H. Wache, L. Nixon, Improving the accuracy of job search with semantic techniques, 2007, pp. 301–313. doi:10.1007/978-3-540-72035-5_23.
[18] I. Singh, A. Garg, Resume ranking with tf-idf, cosine similarity and named entity recognition, in: 2024 First International Conference on Data, Computation and Communication (ICDCC), 2024, pp. 224–229. doi:10.1109/ICDCC62744.2024.10961659.
[19] C. Daryani, G. Chhabra, H. Patel, I. Chhabra, R. Patel, An automated resume screening system using natural language processing and similarity, 2020, pp. 99–103. doi:10.26480/etit.02.2020.99.103.
[20] S. D'Amicantonio, M. K. Kulangara, H. D. Mehta, S. Pal, M. Levantesi, M. Polignano, E. Purificato, E. W. D. Luca, A comprehensive strategy to bias and mitigation in human resource decision systems, in: M. Polignano, C. Musto, R. Pellungrini, E. Purificato, G. Semeraro, M. Setzu (Eds.), Proceedings of the 5th Italian Workshop on Explainable Artificial Intelligence, co-located with the 23rd International Conference of the Italian Association for Artificial Intelligence, Bolzano, Italy, November 26-27, 2024, volume 3839 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2024, pp. 11–27. URL: https://ceur-ws.org/Vol-3839/paper1.pdf.
[21] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, CoRR abs/1908.10084 (2019). URL: http://arxiv.org/abs/1908.10084. arXiv:1908.10084.
[22] M. Polignano, C. Musto, R. Pellungrini, E. Purificato, G. Semeraro, M. Setzu, Xai.it 2024: An overview on the future of AI in the era of large language models, in: M. Polignano, C. Musto, R. Pellungrini, E. Purificato, G. Semeraro, M. Setzu (Eds.), Proceedings of the 5th Italian Workshop on Explainable Artificial Intelligence, co-located with the 23rd International Conference of the Italian Association for Artificial Intelligence, Bolzano, Italy, November 26-27, 2024, volume 3839 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2024, pp. 1–10. URL: https://ceur-ws.org/Vol-3839/paper0.pdf.

Appendix

Table 3
Comparison of matching terms across resume enhancement systems (Part 1)

Resume ID	Original_ Matching_ Terms	Original_ Missing_ Terms	LLaMA3.2_ Matching_ Terms	New_Terms_ Added_by_ LLaMA3.2	GPT-4o_ Matching_ Terms	New_Terms_ Added_by_ GPT-4o
resume_2_7_V7	frontend, react, developer, expertise, apis, ui, jest	reactjs, backend, skilled, freelance, axios, frameworks, typescript, redux, es6, components, component, agile, development	reactjs, frontend, react, developer, frameworks, expertise, apis, components, component, ui, jest, development	component, development, reactjs, components, frameworks	reactjs, frontend, react, skilled, developer, axios, frameworks, typescript, expertise, apis, redux, es6, components, component, ui, jest, agile, development	component, development, redux, reactjs, typescript, es6, axios, agile, components, frameworks, skilled
resume_2_6_V7	frontend, react, backend, skilled, developer, expertise, apis, ui, jest	reactjs, freelance, axios, frameworks, typescript, redux, es6, components, component, agile, development	reactjs, frontend, react, backend, skilled, developer, axios, expertise, apis, components, component, ui, jest, development	component, development, reactjs, axios, components	reactjs, frontend, react, backend, skilled, developer, axios, frameworks, typescript, expertise, apis, redux, es6, components, component, ui, jest, agile	component, development, redux, reactjs, typescript, es6, axios, agile, components, frameworks
resume_2_1_V7	frontend, react, developer, expertise, apis, ui, jest	reactjs, backend, skilled, freelance, axios, frameworks, typescript, redux, es6, components, component, agile, development	frontend, developer, frameworks, expertise, apis, redux, ui, jest, development	redux, development, frameworks	reactjs, frontend, react, backend, developer, axios, frameworks, typescript, expertise, apis, redux, es6, components, component, ui, jest, agile	component, redux, reactjs, es6, typescript, axios, backend, agile, components, frameworks
resume_2_3_V7	frontend, react, developer, expertise, ui	reactjs, backend, skilled, freelance, axios, frameworks, typescript, apis, redux, es6, components, component, jest, agile, development	frontend, react, developer, expertise, ui	–	reactjs, frontend, react, developer, axios, frameworks, typescript, expertise, apis, redux, es6, components, component, ui, jest, agile, development	component, apis, development, redux, reactjs, typescript, es6, axios, agile, components, frameworks, jest
resume_2_2_V7	frontend, react, developer, expertise, ui, jest	reactjs, backend, skilled, freelance, axios, frameworks, typescript, apis, redux, es6, components, component, agile, development	frontend, react, developer, expertise, apis, components, component, ui, jest	component, apis, components	reactjs, frontend, react, backend, skilled, developer, axios, frameworks, typescript, expertise, apis, redux, es6, components, component, ui, jest, agile, development	component, apis, development, redux, reactjs, typescript, es6, axios, frameworks, backend, agile, components, skilled
resume_2_5_V7	frontend, react, backend, developer, expertise, apis, agile, development	reactjs, skilled, freelance, axios, frameworks, typescript, redux, es6, components, component, ui, jest	reactjs, frontend, react, backend, developer, frameworks, expertise, apis, ui, jest, agile, development	reactjs, ui, jest, frameworks	reactjs, frontend, react, skilled, developer, axios, frameworks, typescript, expertise, apis, redux, es6, components, component, ui, jest, agile, development	component, redux, reactjs, es6, typescript, axios, ui, components, frameworks, jest, skilled
resume_2_9_V7	frontend, developer, expertise, apis, ui, jest	reactjs, react, backend, skilled, freelance, axios, frameworks, typescript, redux, es6, components, component, agile, development	reactjs, frontend, react, developer, expertise, apis, components, component, ui, jest, development	component, development, reactjs, react, components	reactjs, frontend, react, backend, skilled, developer, axios, frameworks, typescript, expertise, apis, redux, es6, components, component, ui, jest, development	component, development, redux, reactjs, typescript, es6, axios, backend, react, components, frameworks, skilled
resume_2_8_V7	react, backend, developer, expertise, apis, ui, jest	reactjs, frontend, skilled, freelance, axios, frameworks, typescript, redux, es6, components, component, agile, development	react, backend, developer, expertise, components, component, ui, jest, development	component, components, development	reactjs, frontend, react, backend, developer, axios, frameworks, typescript, expertise, apis, redux, es6, components, component, ui, jest, agile, development	frontend, component, development, redux, reactjs, typescript, es6, axios, agile, components, frameworks

Table 4
Comparison of matching terms across resume enhancement systems (Part 2)

Resume ID	Original_ Matching_ Terms	Original_ Missing_ Terms	LLaMA3.2_ Matching_ Terms	New_Terms_ Added_by_ LLaMA3.2	GPT-4o_ Matching_ Terms	New_Terms_ Added_by_ GPT-4o
resume_2_4_V7	react, developer, expertise, apis, ui, jest	reactjs, frontend, backend, skilled, freelance, axios, frameworks, typescript, redux, es6, components, component, agile, development	reactjs, react, developer, expertise, apis, components, component, ui	reactjs, component, components	reactjs, frontend, react, skilled, developer, axios, frameworks, typescript, expertise, apis, redux, es6, components, component, ui, jest, agile, ux, github, flexbox, integrations, scss, design, applications, javascript, api, responsive, interfaces, bootstrap, fetch, building, experience, devtools, integration, integrating, css, scalable, layouts, collaborate, applying, ensuring, functional, chrome, managing, router, performance, hands, authentication, vanilla, gitlab, deliver, environment, university, tailwind, computer, styling, form, grid, token, accessibility, git, science, user, handling, high, testing, practices, teams, rest, state, modern, years, based, best, library, contract, hooks, performant, quality, implementation, time, hook, query, like, context	scss, accessibility, performant, design, devtools, integrations, git, university, frontend, component, applying, environment, styling, scalable, years, quality, hooks, teams, ensuring, ux, science, integrating, redux, typescript, high, layouts, authentication, frameworks, like, bootstrap, testing, javascript, computer, css, responsive, collaborate, query, fetch, applications, components, deliver, library, functional, flexbox, chrome, handling, github, axios, interfaces, form, time, best, vanilla, integration, grid, reactjs, rest, experience, agile, hands, skilled, api, practices, tailwind, modern, token, user, hook, building, router, performance, implementation, contract, state, context, gitlab, es6, based, managing
resume_2_10_V7	react, developer, expertise, apis, redux, ui, jest, scss, api, experience, candidate, managing, university, computer, form, grid, science, user, high, rest, state, years, best, contract, hooks, time, hook	reactjs, frontend, backend, skilled, freelance, axios, frameworks, typescript, es6, components, component, agile, development, ux, github, flexbox, integrations, web, design, applications, javascript, responsive, bachelor, interfaces, formik, bootstrap, fetch, building, devtools, integration, integrating, css, scalable, layouts, initiative, responsibilities, collaborate, degree, applying, ensuring, functional, chrome, router, looking, performance, hands, authentication, vanilla, gitlab, deliver, environment, tailwind, styling, token, startup, accessibility, git, working, handling, testing, practices, teams, modern, based, library, negotiable, performant, quality, implementation, validation, query, include, like, ideal, equivalent, context	reactjs, frontend, react, developer, expertise, apis, redux, components, component, ui, jest, development, ux, integrations, applications, api, building, experience, integration, integrating, scalable, managing, deliver, university, computer, form, grid, science, user, high, rest, state, years, best, contract, quality, time, like	frontend, component, development, reactjs, integrations, scalable, building, quality, applications, integrating, components, like, deliver, ux, integration	reactjs, frontend, react, backend, developer, axios, frameworks, typescript, expertise, apis, redux, es6, components, component, ui, jest, agile, development, ux, github, flexbox, integrations, scss, design, applications, javascript, api, responsive, interfaces, bootstrap, fetch, building, experience, devtools, integration, integrating, css, scalable, layouts, collaborate, ensuring, functional, chrome, managing, performance, hands, authentication, vanilla, gitlab, deliver, environment, university, tailwind, computer, styling, form, grid, token, accessibility, git, science, user, handling, high, testing, practices, teams, rest, state, modern, years, based, best, library, hooks, performant, quality, time, hook, validation, query, like, context	functional, accessibility, performant, flexbox, design, chrome, devtools, integrations, handling, github, axios, interfaces, git, vanilla, integration, frontend, component, environment, styling, reactjs, scalable, quality, agile, teams, hands, ensuring, practices, ux, tailwind, validation, integrating, modern, token, typescript, building, layouts, backend, authentication, performance, frameworks, like, bootstrap, testing, javascript, context, gitlab, responsive, css, development, collaborate, query, es6, fetch, applications, based, components, deliver, library

Table 5
Column Names description for Tables 3 and 4

Column Name	Description
Original_Matching_Terms	The set of job-description keywords that already appeared in the candidate's resume before any edits.
Original_Missing_Terms	Keywords required by the job but absent from the unmodified resume.
LLaMA3.2_Matching_Terms	After our LLaMA 3.2 "in-place" enhancement, this column lists all keywords in the resume that now match the job description—combining the original matches with those preserved by conservative rewriting.
New_Terms_Added_by_LLaMA3.2	Of the matches in the previous column, these are the new terms introduced by LLaMA 3.2. Crucially, each is semantically equivalent to language already used by the candidate.
GPT-4o_Matching_Terms	The total set of matched keywords after GPT-4o editing—again including both originally present terms and those retained or reordered by GPT.
New_Terms_Added_by_GPT-4o	The new keywords injected by GPT-4o. Unlike our method, these often include terms that were not semantically aligned with the candidate's original phrasing.

Table 6
ATS and Similarity Scores, in %, Across Resume Enhancement Systems

Resume Name	Original Similarity Score	Original ATS Score	Proposed System Updated Similarity	Proposed System Updated ATS Score	GPT-4o Updated Similarity	GPT-4o Updated ATS Score
resume_2_7	30.78	20.44	58.74	38.97	35.85	31.10
resume_2_6	27.35	21.27	33.63	37.22	34.23	30.56
resume_2_1	23.98	23.68	42.33	44.00	25.70	28.44
resume_2_3	13.81	18.26	37.67	49.23	21.39	29.28
resume_2_2	26.31	20.68	47.24	49.97	38.39	35.14
resume_2_5	35.76	23.33	58.78	45.70	41.13	26.29
resume_2_9	17.16	20.52	28.24	38.09	29.59	37.61
resume_2_8	24.71	23.58	45.31	46.44	32.47	29.55
resume_2_4	31.85	18.88	68.87	45.91	39.80	29.59
resume_2_10	28.86	18.66	56.45	53.66	54.62	46.19