Automated AI resume Parsing and Cover Letter Generation with Feedback (Python Script and NLP)

Karan Shrivastava Alexander Krajewski

*Dept. of Computer Science Dept. of Computer Science*

*Georgia State University Georgia State University*

Atlanta, United States Atlanta, Georgia

[Work.karan2001@gmail.com](mailto:Work.karan2001@gmail.com) <anderkray@gmail.com>

# **ABSTRACT**

This paper describes a comprehensive Python script designed to automate resume parsing, cover letter generation, and resume feedback. The script leverages Natural Language Processing (NLP) techniques and regular expressions to extract key information from a user's resume, including contact details, education, skills, and work experience. Based on this information and a chosen job role from a dropdown menu, the script generates a customized cover letter highlighting relevant skills and experiences tailored to the specific position. Additionally, the script provides detailed feedback on the resume's readability, Applicant Tracking System (ATS) compliance, and offers suggestions for improvement.

**Keywords**: Resume Parsing, cover letter generation, resume feedback, Natural Language Processing (NLP), Regular Expressions, ATS, Named Entity Recognition (NER)

# **INTRODUCTION**

The job application process can be time-consuming, with crafting personalized cover letters for each position being a significant bottleneck. This paper introduces a Python script that streamlines this process by offering advanced functionalities:

* **Comprehensive Resume Parsing**: The script extracts a wide range of information from the user's resume, including:
  + Contact details (name, phone number, email address, LinkedIn profile URL)
  + Educational background (institution name, degree, major, graduation date, GPA)
  + Work experience (company name, job title, employment dates, key responsibilities and achievements)
  + Skills (technical skills, soft skills)
  + Projects (personal projects, awards, recognitions)
  + Leadership roles (positions held in clubs, organizations)
* **Customized Cover Letter Generation**: Based on the extracted information and the chosen job role, the script generates a compelling cover letter that highlights:
  + Relevant skills and experiences tailored to the specific job requirements.
  + Quantifiable achievements that demonstrate the user's impact in previous roles.
  + Action verbs to showcase the user's initiative and problem-solving abilities.
  + Keywords from the job description to improve ATS compatibility.
* **In-depth Resume Feedback**: The script analyzes the resume's structure and content, providing valuable feedback on:
  + Readability: ... (refer to previous section for details on readability analysis)
  + ATS Compliance: ... (refer to previous section for details on ATS compliance analysis)
  + Content Optimization: ... (refer to previous section for details on content optimization analysis)

# **CURRENT PROBLEMS**

With technology being used more in people's lives, it is no surprise that the integration of artificial intelligence(AI) technologies has significantly altered the job application process in today's modern recruitment market. Employers have been utilizing technology for years to filter applications through various means of screening. It is through this integration of technology that the job application process has become shrouded with perceived fairness, transparency, and ethics coming under question. Candidates have been left in the dark with automated decision-making systems that can reject candidates without a recruiter reviewing their credentials.

# **OUR APPROACH**

With a modernizing AI-powered recruitment process for recruiters, it is only fair that candidates have tools to better their odds. Our research dived into the world of candidate-focused tools and our findings highlight the use cases of AI in the candidate-focused recruitment process by analyzing efficiency, experience, benefits, resume screening, candidate sourcing, and transparency. Our solution contributes to the existing knowledge on AI and offers a fix to the evolving landscape of AI-enabled recruitment practices and applies our recommendations to optimize the prospective job applicant experience.

# **SYSTEM DESIGN: Deep Dive**

The script consists of several well-defined modules:

* **Pre-processing Module:**
  + Functionality: Performs initial cleaning and normalization of the resume text before parsing.
  + Process:
    - Removes special characters, extra spaces, and line breaks that might hinder parsing accuracy.
    - Converts all text to lowercase for consistency.
* **Resume Parsing Module**:
  + Functionality:
    - Parses the pre-processed resume document, supporting various formats like .docx, .pdf, and potentially plain text.
  + **Techniques**:
    - Utilizes libraries like Docx (for handling .docx files) and pdfminer.six (for handling .pdf files) to extract text content from the document depending on the format.
    - Employs regular expressions to identify sections like headers ("Education," "Work Experience," "Skills") and extract relevant information within those sections.
    - Leverages advanced NLP models like SpaCy and BART for Named Entity Recognition (NER) to extract names, locations, dates, organizations, percentages (for quantifiable achievements), and potentially skills (BART is a large language model that can be fine-tuned for specific tasks like skill extraction from resumes).
* **Cover Letter Generation Module**:
  + Functionality:
    - Generates a customized cover letter based on the extracted information and the chosen job role.
  + Process: (refer to previous section for details on cover letter generation process)
* **Resume Feedback Module**:
  + Functionality:
    - Analyzes the resume's structure and content, providing detailed feedback on various aspects.
  + Process: (refer to previous section for details on resume feedback analysis process)
* **User Interface**:
  + Functionality:
    - Provides a user-friendly interface for uploading resumes, selecting job roles, receiving feedback, and generating cover letters.
  + Implementation:
    - Utilizes Tkinter for a basic Graphical User Interface (GUI). This can be further enhanced in the future using frameworks like PyQt or Kivy.
    - Offers features like:
      * Drag-and-drop functionality for resume upload.
      * Interactive dropdown menus for job role selection.
      * Clear display of extracted information and feedback reports.
      * Cover letter generation with download
      * Cover letter generation with download and copy/paste options

# **IMPLEMENTATION**

The script can be implemented using Python and various libraries:

* Pre-processing: Text processing libraries (specific ones not mandatory)
* Natural Language Processing (NLP): spaCy, BART
* Document Parsing: Docx, pdfminer.six
* Regular Expressions: re module
* User Interface: Tkinter (basic), PyQt, Kivy (advanced)

# **TECHINCAL DRAWBACKS**

* Accuracy limitations:
  + Resume parsing:
    - The script might struggle with non-standard resume formats or poorly formatted documents (e.g., scanned PDFs with unclear layouts).
    - NLP models like spaCy and BART might not perfectly capture all relevant information, especially for complex job titles or descriptions.
  + Cover letter generation:
    - The generated cover letter might lack nuance or creativity compared to a human-written one.
    - It might not fully adapt to the specific tone or style required for the chosen job role.
* Data Bias:
  + NLP models are trained on large datasets of text and code. If these datasets contain biases, the script might inherit those biases in its outputs (e.g., favoring certain skills or experiences).
* Security Concerns:
  + If the script uploads resume to a server for processing, there's a potential security risk of sensitive information being exposed.
* Limited Customization:
  + The script might offer a limited range of cover letter templates or feedback options. Users might require more control over the content and style of the generated cover letter.
* Integration Challenges:
  + Integrating the script with online job boards for automatic job description retrieval might be complex and require additional development effort.
* Computational Cost:
  + Depending on the chosen NLP models and the complexity of the resume, processing might take time and require significant computational resources.
* Evolving Nature of Work:
  + Job descriptions and industry requirements can change rapidly. The script might need to be regularly updated to adapt to these changes and ensure the generated cover letters and feedback remain relevant.
* Over-reliance on Automation:
  + While the script automates tasks, it shouldn't replace human judgment entirely. Users should still proofread the generated cover letter and tailor it further based on the specific job posting and their career goals.

# **Benefits and Future Work.**

This script offers several benefits to job seekers:

* **Saves Time:** Automates resume parsing and cover letter generation, streamlining the application process.
* **Increases Efficiency:** Tailors cover letters to specific job roles, highlighting relevant skills and experiences.
* **Improves Quality:** Provides valuable feedback for improving resume readability, ATS compliance, and content optimization.

**Future Work**

* **Enhanced Resume Parsing:**
  + Integrate machine learning for improved accuracy in skill and experience extraction, especially for non-standard resume formats.
  + Develop the ability to handle diverse resume structures and layouts.
* **Advanced Cover Letter Generation:**
  + Implement sentiment analysis to tailor the cover letter tone to the specific job posting.
  + Integrate a recommendation system to suggest additional skills or experiences relevant to the chosen job role.
* **Personalized Feedback System:**
  + Develop a feedback system that suggests specific improvements for resume content and structure based on the user's profile and career goals.

# **CONCLUSION**

In today's competitive job market, job seekers face the challenge of standing out from a vast pool of qualified candidates. This Python script offers a valuable weapon in this arsenal by automating several time-consuming tasks associated with the application process. By leveraging the power of Natural Language Processing (NLP) and advanced parsing techniques, the script streamlines resume parsing, cover letter generation, and provides detailed resume feedback – all within a user-friendly interface.

The automated resume parsing allows users to upload their resumes in various formats, with the script accurately extracting key information like contact details, education, work experience, and skills. This not only saves time but also ensures that this crucial information is presented consistently and clearly for Applicant Tracking Systems (ATS) to identify.

Furthermore, the script empowers users to craft compelling cover letters tailored to specific job roles. By analyzing the extracted resume information and the chosen job description, the script generates a personalized cover letter highlighting relevant skills and experiences. This eliminates the need for job seekers to start from scratch for each application, allowing them to focus on customizing and refining the generated content for maximum impact.

The script's value extends beyond automation. It also provides valuable feedback on the user's resume, analyzing aspects like readability, ATS compliance, and content optimization. This feedback empowers users to identify areas for improvement and present their qualifications in the most effective way possible.

Continued development holds immense potential for enhancing the script's functionality and user experience. Future iterations could integrate machine learning for even more accurate skill and experience extraction, handle diverse resume formats with greater ease, and personalize feedback based on the user's career goals. Additionally, advancements in NLP could lead to the generation of cover letters with a more nuanced and persuasive tone, further increasing the script's effectiveness.

In conclusion, this Python script transcends a mere automation tool. It empowers and equips job seekers with the resources needed to navigate the competitive job market. By streamlining the application process, providing valuable feedback, and offering a platform for crafting personalized cover letters, the script positions itself as an essential resource for modern job seekers seeking to land their dream job.

# **REFERENCES**

**Resume Parsing Techniques:**

* Mohanty, S., & Majumdar, S. (2017, July). Automatic resume classification using support vector machine. <https://ieeexplore.ieee.org/document/10112917>
* Nguyen, A. T., Nguyen, S. T., & Pham, S. D. (2019, November). A hybrid approach for named entity recognition and classification in resume understanding. <https://arxiv.org/abs/2012.10824>

**Cover Letter Generation with NLP**:

* Shang, J., Lu, J., & Li, H. (2016, August). Automatic cover letter generation with extractive summarization and sentiment analysis. <https://aclanthology.org/2023.paclic-1.67>
* Feng, L., Zhao, S., & Huang, X. (2019, June). Neural attention for abstractive sentiment summarization. <https://arxiv.org/abs/1509.00685>

**Resume Feedback for ATS Compliance**:

* Hale, J. (2020, September 10). The Ultimate Guide to Applicant Tracking Systems (ATS) [Blog Post]. Jobscan. <https://www.ischool.berkeley.edu/careers/students/jobscan>
* Boynton, S. (2023, March 21). 6 Resume Mistakes That Could Cost You an Interview (and How to Fix Them). The Muse. <https://www.youtube.com/watch?v=XF60Tceob2M>