Resume Tune

Shubham Pal
IIITD,
CSAI, 2021564

Garv Makkar
IIITD,
CSAI, 2021530

Rishav Raj
IIITD,
CSAI, 2021556

Parth Kaushal
IIITD,
CSAI, 2021548

Samyak Jain
IIITD,
CSAI, 2021560

Sachin Sharma
IIITD
CSAI, 2021559

November 2024

1 Abstract

In today's competitive job market, many job seekers face the challenge of aligning their resumes with job descriptions and ensuring they pass through Applicant Tracking Systems (ATS). This project aims to create an application that helps users improve their resumes using advanced Large Language Models (LLMs) and Natural Language Processing (NLP) techniques. Our application leverages the power of LLMs, specifically trained on relevant data, to analyze resumes and job descriptions, identifying any gaps in skills or qualifications. Acting upon this analysis, our system automatically acts upon the suggested improvements and generates the suitable improved Resume for the user. The effectiveness of the system will be measured through relevance, job-fit scores, and improved readability.

2 Introduction

In today's competitive job market, many job seekers face the challenge of aligning their resumes with job descriptions and ensuring they pass through Applicant Tracking Systems (ATS). Even qualified candidates often miss out on opportunities because their resumes don't effectively showcase the right skills and experience. In recent years, the use of Large Language Models (LLMs) and Natural Language Processing (NLP) has gained significant attention for Natural Language Understanding and Generation tasks. While LLMs are widely used for resume-job description matching, their potential for generating in-depth, domain-rich personalized resume enhancements remains relatively unexplored.

This literature survey explores relevant research on existing and state-of-the-art Machine Learning, Deep Learning, and LLM-based approaches and their applications in analyzing and enhancing resumes. By reviewing prior work on resume parsing, job matching algorithms, and Applicant Tracking System (ATS) requirements, this survey will provide a foundation for designing an intelligent system capable of delivering personalized ATS-compliant resume suggestions.

The insights gained will guide the project's technical development and model training.

3 Literature Survey

Recent advancements in resume optimization have been driven by the use of Large Language Models (LLMs) and Natural Language Processing (NLP). This section highlights key works in this domain:

CareerBoost: Revolutionizing the Job Search with Resume Enhancement and Tailored Recommendations

Asoke Nath et al. propose a system that enhances the job search process using advanced NLP and machine learning techniques, including Random Forest Classifier and K-means Clustering. The system features a *Resume Enhancer* that evaluates resumes and job descriptions to recommend skills, and a *Job Recommendation* tool that matches job listings to users' qualifications. Cosine Similarity is employed to assess skill matches, while job opportunities are grouped for easier exploration. Future enhancements include user feedback integration, personalized learning paths, and automatic resume updates for continuous improvement.

Resume Building Application Based on LLM

R. J. Sunico et al. (2023) introduce a resume-building application aimed at helping students, particularly from underprivileged backgrounds, create effective resumes using an LLM. The application converts natural language input into professional resume bullet points, assesses content for improvements, and ensures resumes follow the STAR (Situation, Task, Action, Result) format. It comprises three key modules: *Resume Generation*, *Resume Assessment*, and *User I/O*, utilizing GPT-3.5 Turbo to enhance resume quality. Initial tests demonstrate its ability to simplify resume creation for users with limited career support.

ResumeFlow: An LLM-Facilitated Pipeline for Personalized Resume Generation and Refinement

Zinjad et al. (2024) present a system that automates resume creation by extracting information from user-provided resumes and job descriptions. The approach excels in generating tailored resumes quickly and with minimal user input. However, it may lack depth in evaluating the final product's effectiveness compared to human-crafted resumes, raising concerns about quality assurance.

Comparative Analysis of ML and LLM in Resume Parsing: A Paradigm Shift in Talent Acquisition

This paper discusses the transition from traditional machine learning techniques to LLMs in resume parsing. Esranur Kaygin (2023) highlights the advantages of LLMs, such as improved accuracy and contextual understanding. However, challenges related to integrating these models into existing Applicant Tracking Systems (ATS) are not sufficiently addressed, potentially limiting practical application.

Generating Synthetic Resume Data with Large Language Models for Enhanced Job Description Classification

Skondras, Zervas, and Tzimas (2023) explore the use of LLMs to create synthetic resume data for improving job description classification. By combining real-world resume data from Indeed with synthetic data generated by ChatGPT, the authors produce a comprehensive dataset for training classification models. Models trained on this augmented dataset, particularly using the BERT architecture, achieve high accuracy and precision, outperforming traditional training methods. However, reliance on synthetic data raises concerns about generalizability, as the quality and realism of generated resumes may vary.

AI-Based Automatic Resume Analysis

The article *AI-Based Automatic Resume Analysis* (2022) provides insights into how LLM-based AI applications assist HR departments during recruitment. It describes how these tools extract detailed information from candidates' resumes, analyze work experiences and projects, and rate expertise in various skills and technological tools. This article serves as a valuable reference for the first phase of our application, particularly in the extraction of user information from their profiles and resumes. subsection* These studies collectively contribute to the understanding of how LLMs can enhance resume optimization. They also reveal gaps in practical application and user-centric design, which the proposed project aims to address.

4 Methodology

Data Preprocessing and Tagging

- **Resume Input and Tokenization:** Resume PDFs have been converted to JSON format. Each extracted JSON file provides personal details and information about education, skills, work experience, projects, and any other possible information under the `extras` section. Converting resume PDFs to JSON files provides a more structured and consistent schema of the resume in comparison to the `.txt` format. The Llama 3.1 70B model has been used for this extraction task. Due to the high number of parameters, this model shows better Named Entity Recognition (NER) capabilities, better structure understanding, good support for customization, and high accuracy in parsing complex information.

- **Token Cleaning and Preprocessing:** Standard cleaning techniques were applied to remove stop words, punctuation, and other irrelevant tokens.

- **Section Identification and Tagging:** Different sections of the resume were identified and labeled with appropriate tags (e.g., `<Work Experience>`, `<Achievements>`, `<Skills>`, etc.). This structured representation facilitates downstream analysis and tuning.

Skill Extraction and Cleaning

- **Relevant Skill Extraction:** Skills mentioned in the resume were extracted using techniques like keyword matching.

- **Skill Cleaning and Normalization:** Extracted skills were cleaned to ensure consistency and accuracy. This involved tasks such as removing stop words and normalizing variations.

Information Extraction from Job Descriptions (JD)

Similarly to resumes, job descriptions have also been converted to JSON format using the Llama 3.1 70B model. As part of the experimentation, we have extracted three sections from the job descriptions:

- **Role:** Cosine similarity was used between the job description and an exhaustive list of all roles to determine the appropriate role.
- **Required Skills:** The Llama model was used to extract the required skills.
- **Eligibility Criterion:** Eligibility requirements were extracted using the same model.

Zero-Shot Prompting and Skill Tuning

- **Model Selection:** Multiple language models, including Llama3_8b, Gemma2_9b, and Gemini, were employed for zero-shot prompting.
- **Prompt Engineering:** Prompts were carefully crafted to guide the models in refining the extracted skills based on the provided job descriptions.
- **Skill Tuning:** The models generated improved versions of the extracted skills by leveraging their understanding of the job requirements and the context provided in the prompts.

Resume Enhancement Using RAG

The goal of our project is to enhance resumes by comparing the skills required for specific job roles with the skills present in a candidate's existing resume. To achieve this, we have implemented a Retrieval-Augmented Generation (RAG) system that extracts relevant skills from job descriptions and enhances resumes accordingly.

Setting up the Environment

- **Google Generative AI Gemini:** We used the Gemini-1.5-pro-latest model, which allows us to create content by responding to prompts, making it well-suited for enhancing resumes.
- **LangChain Library:** Streamlined document processing.
- **FAISS:** Used Facebook AI Similarity Search (FAISS) to store and retrieve relevant data efficiently.

Processing Knowledge Base for RAG

We created an extensive knowledge base consisting of an exhaustive list of important skills in each identified domain, represented as PDFs.

- **Text Splitting:** Text was split into manageable chunks using the `CharacterTextSplitter` with chunks of 1,000 characters and a 200-character overlap to preserve context.

Creating the Vector Store

- **Embeddings:** We used the `HuggingFace sentence-transformers/all-MiniLM-L6-v2` model to convert text chunks into embeddings. These embeddings represent the semantic meaning of the text.
- **FAISS Storage:** The embeddings were stored using FAISS for quick retrieval when matching skills between job descriptions and resumes.

Building the RAG System

The RAG system retrieves relevant chunks of skills from the documents and uses the Gemini model to generate enhanced resumes.

- **Retrieval Mechanism:** The system retrieves the top 3 most relevant pieces of text from the vector store, ensuring high relevance in the content passed to the model.

Resume Enhancement with RAG

The system takes both resume and job description JSON files as input and performs the following:

- Queries the RAG system for industry-specific knowledge about the role.
- Enhances resume sections (skills, experience, projects, extras) using retrieved knowledge.
- Incorporates job requirements and industry best practices.
- Saves the enhanced resume with references to sources used from the knowledge base.

DPO

We sampled a set of 30 resumes for the role of Software Development Engineer (SDE) and labeled them on a scale of 0 to 5. After labeling, we fine-tuned the model using the **Direct Preference Optimization (DPO)** technique. However, due to memory constraints, we were unable to fine-tune the model or run the code at this stage. Despite these limitations, we decided to adopt this method.

5 Evaluation Metrics and Results Analysis

We designed various evaluation metrics and criteria to critically analyze the different aspects by which a resume can be evaluated and improved.

5.1 Evaluation Criteria

We've set the following metrics to evaluate the performances of different approaches for the resume enhancement.

Criteria A: Skill Matching

- Parses and normalizes skills from both the resume and job description.
- Uses fuzzy matching to account for variations in skill names.
- Calculates a skill match score based on required vs. present skills.

Criteria B: Grammar

- Evaluates writing quality using *LanguageTool*.

Criteria C: Resume Alignment to Job Description

- **Keyword Analysis:** Extracts and compares relevant keywords.
- **Text Similarity:** Uses TF-IDF for content comparison.
- **Semantic Matching:** Employs BERT embeddings.

The system combines these factors to generate a final matching score, weighing:

- Semantic Understanding: 40%
- Keyword Matching: 30%
- Content Similarity: 30%

This provides a comprehensive assessment of how well a resume matches a job description.

Final Criteria The overall score is calculated as the average of the above three criteria.

5.2 Pipeline

- Connects all the above components in order and saves the results.
- Selects the best model and method and saves it.

5.3 Results Analysis and Visualization System

This system implements a comprehensive analysis framework to evaluate and visualize the performance of different resume enhancement approaches.

5.3.1 Score Analysis

- Processes three types of evaluation scores (A, B, C) for each version:
  - **R:** Original resume.
  - **R1:** Basic enhanced resume.
  - **R2:** RAG-enhanced resume.
- Calculates average scores across all test pairs.

5.3.2 Visualization

- Creates comparative line plots for each evaluation metric.
- Shows performance trends across different resume pairs.
- Generates separate visualizations for:
  - Skills matching (A).
  - Grammar quality (B).
  - Overall matching (C).
  - Combined total score.

5.3.3 Best Model Selection

- Identifies the best-performing combination of:
  - **LLM model:** 70B vs. 90B.
  - **Resume version:** R vs. R1 vs. R2.
- Stores results for final model selection.

5.4 Results for Existing Work

The test set currently consists of four pairs.

Results for LLaMA-3.1-70B-Versatile:

- **Average Evaluation Score for R:** 0.4353264725093886
- **Average Evaluation Score for R1:** 0.6270193125967944
- **Average Evaluation Score for R2:** 0.6350886409211411

Results for LLaMA-3.1-90B-text-preview:

- **Average Evaluation Score for R:** 0.4371544996145654
- **Average Evaluation Score for R1:** 0.5774336770940104
- **Average Evaluation Score for R2:** 0.5776433689482731

6 Conclusion

This research delves into the application of Large Language Models (LLMs) and Natural Language Processing (NLP) techniques to enhance resumes for better job application outcomes. The methodology involved a multi-step process, including data preprocessing, skill extraction, job description analysis, and resume enhancement using Retrieval-Augmented Generation (RAG).

Findings

LLM Power: The 70B LLaMA model demonstrated superior performance in various tasks, including resume parsing, skill extraction, and job description analysis. This highlights the potential of large language models in understanding and generating human-quality text.

RAG Effectiveness: The implementation of RAG proved to be a valuable approach for enhancing resumes. By retrieving relevant information from a knowledge base, the system effectively improved the quality and relevance of the generated content. Evaluation and Improvement: A rigorous evaluation process was established to assess the performance of different models and techniques. The use of metrics such as skill matching, grammar, and overall alignment to job descriptions provided a comprehensive evaluation framework.

Future Directions:

Model Optimization: Further fine-tuning and optimization of the LLM models can lead to even better performance and more accurate resume enhancements.

Expanded Knowledge Base: Enriching the knowledge base with a wider range of industry-specific information can improve the quality of the generated content. User Interface Development: Creating a user-friendly interface to facilitate interaction with the system and allow for customization of the enhancement process.

Ethical Considerations: Addressing potential biases and ensuring fairness in the application of AI-powered tools.

In conclusion, this research demonstrates the promising potential of LLMs and NLP techniques in automating and improving the resume enhancement process. By leveraging these advancements, job seekers can significantly enhance their chances of success in today's competitive job market.

References

[1] Yu, K., Guan, G., & Zhou, M. (2005). Resume Information Extraction with Cascaded Hybrid Model. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)* (pp. 499–506). Ann Arbor, Michigan: Association for Computational Linguistics.

[2] Chen, J., Gao, L., & Tang, Z. (2016). Information extraction from resume documents in pdf format. *Electronic Imaging*, 2016(17), 1–8.

[3] Chen, J., Zhang, C., & Niu, Z. (2018). A two-step resume information extraction algorithm. *Mathematical Problems in Engineering*, 2018.

[4] Zu, S., & Wang, X. (2019). Resume information extraction with a novel text block segmentation algorithm. *International Journal of Natural Language Computing*, 8, 29–48.

[5] Linet, N. (2020). "ATS BREAKER"- A System for Comparing Candidate Resume and Company Requirements. *International Journal of Engineering Research*, V9. https://doi.org/10.17577/IJERTV9IS060403.

[6] Gu, Z., Meng, C., Wang, K., Lan, J., Wang, W., Gu, M., & Zhang, L. (2022). XYLayoutLM: Towards Layout-Aware Multimodal Networks For Visually-Rich Document Understanding. *arXiv preprint arXiv:2203.06947*.

[7] Chou, Y.-C., & Yu, H.-Y. (2020). Based on the application of AI technology in resume analysis and job recommendation. In *2020 IEEE International Conference on Computational Electromagnetics (ICCEM)* (pp. 291-296). Singapore. https://doi.org/10.1109/ICCEM47450.2020.9219491.

[8] Surendiran, B., Paturu, T., Chirumamilla, H. V., & Reddy, M. N. R. (2023). Resume Classification Using ML Techniques. In *2023 International Conference on Signal Processing, Computation, Electronics, Power and Telecommunication (IConSCEPT)* (pp. 1-5). Karaikal, India. https://doi.org/10.1109/IConSCEPT57958.2023.10169907.

[9] Nath, N. A., Saha, N. S., Sarkar, N. S. D., & Bose, N. A. (2024). CareerBoost: Revolutionizing the Job Search with Resume Enhancement and Tailored Recommendations. *International Journal of Scientific Research in Computer Science Engineering and Information Technology*, 10(3), 281–290. https://doi.org/10.32628/cseit24103106.

[10] ResumeFlow: an LLM-facilitated pipeline for personalized resume generation and refinement. (n.d.). https://arxiv.org/html/2402.06221v2.

[11] Grine, A. E. (2024, March 21). Resume scanner: Leverage the power of LLM to improve your resume. *Medium*. https://medium.com/thedeephub/resume-scanner-leverage-the-power-of-llm-to-improve-your-resume-401a0cb49cd7.

[12] Sunico, R. J., Pachchigar, S., Kumar, V., Shah, I., Wang, J., & Song, I. (2023). Resume Building Application based on LLM (Large Language Model). In *2023 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)* (pp. 486-492). Greater Noida, India. https://doi.org/10.1109/ICCCIS60361.2023.10425602.

[13] Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., Quentin, D. L., Gesmundo, A., Attariyan, M., & Gelly, S. (2019). Parameter-Efficient Transfer Learning for NLP. *arXiv preprint arXiv:1902.00751*.

[14] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2021). LORA: Low-Rank adaptation of Large Language Models. *arXiv preprint arXiv:2106.09685*.

[15] Li, X. L., & Liang, P. (2021). Prefix-Tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

[16] Lester, B., Al-Rfou, R., & Constant, N. (2021). The power of scale for Parameter-Efficient Prompt Tuning. *arXiv preprint arXiv:2104.08691*.

[17] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., ... Amodei, D. (2020). Language Models are Few-Shot Learners. *arXiv preprint arXiv:2005.14165*.

[18] Stiennon, N., Ouyang, L., Wu, J., Ziegler, D. M., Lowe, R., Voss, C., Radford, A., Amodei, D., & Christiano, P. (2020). Learning to summarize from human feedback. *arXiv preprint arXiv:2009.01325*.