WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

AI-Powered Resume Parser: An NLP-Based System for Automated Resume Information Extraction

[1] Ms Santhoshini, [2] Kushal Kumar V, [3] Logamithran C B, [4] Mohamed Aadil A, [5] Nagappan S P

[1 2 3 4 5] Sri shakthi Institute of Engineering and Technoogy, Coimbatore, 641062, India.

A B S T R A C T :

This project presents the design and development of an AI-powered resume parser aimed at automating the extraction and classification of key information from resumes. Leveraging machine learning techniques with Python, the system extracts fields such as name, contact information, education, skills, and work experience. It is deployed using Flask for web interface integration, with model serialization handled by Pickle and structured data storage managed by SQL databases. The parser addresses challenges related to diverse resume formats and entity extraction accuracy, applying optimization techniques to enhance model performance. Evaluation metrics such as precision, recall, and F1-score demonstrate the effectiveness and efficiency of the system in parsing resumes. The system enables seamless integration with applicant tracking systems, significantly reducing manual effort and improving recruitment workflows. Future improvements may include support for multiple languages and incorporation of deep learning models for enhanced accuracy.

Keywords : Artificial intelligence, Information extraction, Machine learning, Natural language processing, Recruitment automation, Resume parsing.

Introduction

The increasing use of artificial intelligence (AI) in recruitment has transformed traditional hiring workflows by enabling automation and enhancing efficiency. One significant advancement in this domain is the development of AI-powered resume parsers that automatically extract and classify key information from candidate resumes. Manual resume screening is often time-consuming and prone to human errors, whereas AI-based solutions offer faster processing and greater consistency. This paper presents the design, implementation, and evaluation of an AI-driven resume parser built using Python and Flask, with Pickle used for model serialization and an SQL database employed for structured data storage. The proposed system focuses on extracting essential resume fields such as name, contact information, education, skills, and work experience, handling diverse file formats including PDF and DOCX. By leveraging machine learning and natural language processing techniques, the system improves the speed and accuracy of candidate data extraction, facilitating seamless integration with applicant tracking systems.The remainder of this paper discusses the relevant background, describes the methodology, addresses the challenges encountered, presents evaluation results, and outlines future directions for enhancing the system's capabilities.

Related Work

Several resume parsing systems have been developed utilizing traditional rule-based methods and, more recently, AI-driven approaches. Early techniques primarily relied on keyword matching and regular expressions, which often struggled with diverse resume formats. Advances in NLP and ML have enabled the extraction of semantic meaning from resumes, improving entity recognition accuracy. Research has shown that machine learning models, including Random Forest, Support Vector Machines (SVMs), and deep learning-based Named Entity Recognition (NER), outperform manual and heuristic methods. Industry solutions like Sovren, HireAbility, and DaXtra have set benchmarks, though proprietary limitations restrict their customization. This project builds upon these developments by employing a custom-trained ML model designed for flexibility and easy integration with Applicant Tracking Systems (ATS).

Methodology

3.1. Data Collection

A diverse dataset of resumes in both PDF and DOCX formats was collected from various sources for the purpose of training and testing the system. These resumes varied in layout, structure, and content style to ensure generalization and robustness of the parser.

3.2. Preprocessing

Text was extracted from the resumes using libraries such as `pdfminer` for PDF files and `docx2txt` for DOCX files. The extracted text underwent preprocessing steps including tokenization, stopword removal, lemmatization, and noise reduction. This helped standardize the input and improve the accuracy of the subsequent extraction processes.

3.3. Feature Engineering

Key features were extracted using rule-based methods and pattern recognition via NLP. These features included personal details (e.g., name, email), educational background, professional experience, technical and soft skills, and certifications. Named Entity Recognition (NER) was used to identify and segment meaningful entities from the unstructured text.

3.4. Model Training and Serialization

A supervised machine learning model, specifically a Random Forest Classifier, was trained on labeled text segments to classify resume components accurately. Once trained and validated, the model was serialized using Python's Pickle library for efficient reuse during deployment.

3.5. Web Deployment

The system was deployed using Flask, enabling a simple and interactive web interface where users could upload resumes. The backend processed the file and returned a structured JSON output, displaying the extracted fields in an organized manner.

3.6. Database Management

All parsed and structured data was stored in an SQL database. This allowed for efficient querying, analytics, and integration with other systems such as Applicant Tracking Systems (ATS). The database design ensured scalability and easy retrieval of candidate information.

Results

1.1. Performance

The performance of the resume parser was evaluated using standard metrics:

- Precision: 89.2%
- Recall: 87.5%
- F1-Score: 88.3%
- Parsing Speed: Average of 1.32 seconds per resume.

The system demonstrated high accuracy in entity extraction across various resume formats. Testing involved 200 resumes, with feedback from HR professionals validating the parser's practical usability. The parser handled diverse layouts and mixed formatting with minimal errors.

Table 1 - Group Statistics

Metric	N	Mean	Std. Deviation	Std. Error Mean
Precision	**50**	**0.892**	**0.045**	**0.0064**
Recall	**50**	**0.875**	**0.051**	**0.0072**
F1-Score	**50**	**0.883**	**0.048**	**0.0068**
Parsing Speed (s)	**50**	**1.32**	**0.15**	**0.021**

Table 2 - Independent Samples Test

Variances	t	df	Siq. (2-tailed)	Mean Difference	Std. Error Difference

Equal variances assumed	**-1.564**	**98**	**0.121**	**-0.032 sec**	**0.020**
Equal variances not assumed	**-1.589**	**96.4**	**0.115**	**-0.032 sec**	**0.019**

Discussion

The proposed AI-powered resume parser successfully automated the extraction of key information from resumes with high efficiency and accuracy. Handling multiple document formats, particularly inconsistent DOCX files, posed a significant challenge that was mitigated through enhanced preprocessing techniques. Although the model performed well, minor inaccuracies in complex layouts suggest potential benefits from exploring deep learning models such as Bidirectional LSTM or transformers like BERT. Integration with real-time applicant tracking systems (ATS) was smooth due to the system's modular architecture, but scaling to high-volume parsing scenarios would require optimization of database and server resources.

Conclusion

This study demonstrated the development and deployment of an AI-powered resume parsing system that improves recruitment workflows by automating the extraction and classification of critical candidate information. Leveraging machine learning and natural language processing, the system achieved high accuracy while maintaining low parsing times. Future enhancements may include support for multiple languages, incorporation of deep learning-based entity recognition, and integration with larger HRMS platforms for seamless end-to-end recruitment automation.

Acknowledgements

I would like to express my sincere gratitude to my mentor, the department staff, and the Head of Department (HoD) for their invaluable guidance, support, and encouragement throughout the course of this project. Their expertise and constant assistance have been instrumental in the successful completion of this work.

REFERENCES :

1. Jurafsky, D., & Martin, J. H. (2021). Speech and Language Processing (3rd ed.). Draft.
2. HireAbility Resume Parser Documentation.
3. Sovren Resume/CV Parser Whitepaper.
4. Chiticariu, L., Li, Y., & Reiss, F. R. (2013). "Rule-Based Information Extraction is Dead! Long Live Rule-Based Information Extraction Systems!" Proceedings of EMNLP.
5. Bird, S., Klein, E., & Loper, E. (2009). Natural Language Processing with Python. O'Reilly Media
6. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL. [https://arxiv.org/abs/1810.04805]
   (Introduces BERT, which can be fine-tuned for entity extraction from resumes)
7. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). Neural Architectures for Named Entity Recognition. NAACL.
   (LSTM-CRF-based sequence tagging methods for structured resume data extraction)
8. LinkedIn Talent Insights & Job Posting Schema.
   (Real-world schema and fields that align with industry expectations in parsed resumes)
9. Ratinov, L., & Roth, D. (2009). Design Challenges and Misconceptions in Named Entity Recognition. CoNLL.
   (Important NER baseline techniques and evaluation insights).