INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

Automated Resume Parsing and Job Domain Prediction using Machine Learning

Arvind Kumar Sinha[1], **Md Amir Khusru Akhtar**[1], **Mohit Kumar**[2]*

1 Faculty of Computing and Information Technology, Usha Martin University, Ranchi, India
2 Department of IT, MIT Art Design and Technology University, Pune, India

OPEN ACCESS

Received: 14-04-2023
Accepted: 17-06-2023
Published: 04-07-2023

Citation: Sinha AK, Akhtar MAK, Kumar M (2023) Automated Resume Parsing and Job Domain Prediction using Machine Learning. Indian Journal of Science and Technology 16(26): 1967-1974. https://doi.org/10.17485/IJST/v16i26.880

* **Corresponding author**.

mohitsmailbox13@gmail.com

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Objectives: This study aims to develop an efficient approach for parsing resumes and predicting job domains using natural language processing (NLP) techniques and named entity recognition to enhance the resume screening process for recruiters. **Methods:** The proposed approach involves preprocessing steps, such as cleaning, tokenization, stop-word removal, stemming, and lemmatization, implemented with the PyMuPDF and doc2text Python modules. Regular expressions and the spaCy library are utilized for entity recognition and name extraction. The model achieved a prediction accuracy of 92.08% and an F1-score of 0.92 on a dataset of 1000 resumes. An ablation experiment assessed the contributions of different factors. **Findings:** The approach demonstrated a high prediction accuracy of 92.08% and F1-score of 0.92 for job domain prediction, effectively identifying relevant job domains from resumes. Evaluations on individual job domains showed excellent precision and recall scores, validating its applicability. Preprocessing techniques significantly improved accuracy, while the integration of regular expressions and spaCy enhanced the model's performance. This approach automates resume screening, reducing recruiters' workload, saving time and effort, and improving candidate selection and the hiring process. **Novelty:** This study introduces a novel approach combining NLP techniques, regular expressions, and entity recognition for resume parsing and job domain prediction. This integration enhances accuracy and efficiency, offering a unique solution for resume screening.

Keywords: Resume parsing; Job domain prediction; Entity recognition; Machine learning; Natural Language Processing

1 Introduction

The process of screening resumes for job openings has become a significant challenge for recruiters in today's job market. With the increasing number of applicants and resumes submitted in various formats, the task of extracting relevant information from resumes has become time-consuming and tedious. Additionally, the use of online-based recruiting systems has created new challenges for recruiters, including the inability to

accurately identify suitable candidates for the desired job role. In response, natural language processing (NLP) techniques have been developed to automate and streamline the resume screening process [1,2].

Resume parsing and job domain prediction are crucial steps in the recruitment process [3], and several studies have focused on developing efficient approaches for automating these processes [4,5]. Resume parsing involves extracting relevant information from resumes, such as name, contact information, education, work experience, and skills. Several studies have proposed different approaches for automating this process, including rule-based, machine learning, and hybrid approaches [6,7]. In addition, job domain prediction [8] approaches still face challenges related to the requirement of extensive labeled data, incapability to handle intricate job description formats, and insufficient flexibility.

These limitations can hinder the performance of the prediction models and make it difficult to apply them to various job domains effectively. Therefore, there is a need for further research to overcome these limitations and enhance the accuracy and versatility of job domain prediction techniques.

Table 1 summarizes the limitations of existing recruiting methods, specifically the time-consuming nature of traditional resume screening and the gaps in capability of online-based recruiting systems. Traditional resume screening involves a manual review of resumes, which can take up to 90% of the total recruiting time. This method is particularly time-consuming as resumes are often presented in various formats and contain unstructured data, making it difficult to extract relevant information. On the other hand, online-based recruiting systems automate the screening of resumes, but they have limitations in their capability to accurately identify suitable candidates for the desired job role. This can result in up to 75% of the total recruiting time being spent on reviewing resumes that are not a good fit for the job.

Table 1. Limitations of Existing Recruiting Methods

Method	Capability	Gap	Time Consuming Percentage
Traditional Resume Screening [9]	Manual review of resumes	Unstructured data makes it difficult to extract relevant information	Up to 90% of total recruiting time
Online-Based Recruiting Systems [10]	Automated screening of resumes	Inability to accurately identify suitable candidates for the desired job role	Up to 75% of total recruiting time

Table 2 provides an overview of current research on job recruitment and resume analysis. Each paper is summarized by its title, description, benefits, limitations, and future enhancements. The papers cover various aspects of the field, such as automated resume classification, AI research for job-résumé matching, and on-demand job-based recruitment using artificial intelligence.

The proposed approach in this paper aims to overcome these limitations by using a hybrid approach that combines rule-based and machine learning techniques for resume parsing and named entity recognition for job domain prediction.

In this paper, we propose a novel approach for efficiently parsing resumes and predicting relevant job domains using NLP and named entity recognition techniques. The proposed approach involves pre-processing steps such as cleaning, tokenization, stop-word removal, stemming, and lemmatization, which are applied to resumes using PyMuPDF and doc2text Python modules. We also introduce regular expressions and the spaCy library for extracting names and performing entity recognition, respectively, which enhance the accuracy and efficiency of the approach.

The main goal of this research is to develop a system that can convert unstructured data into a structured format, enabling recruiters to efficiently filter the right candidates for the desired job role. The proposed system has been evaluated on a dataset of1000 resumes and has achieved an overall prediction accuracy of 92.08%, with an F1-score of 0.92, making it a promising solution for organizations looking to streamline their resume screening process.

2 Methodology

In this section, we provide a detailed overview of the methodology and the proposed approach for efficiently parsing resumes and predicting relevant job domains using Natural Language Processing (NLP) and named entity recognition techniques.

The proposed approach involves several pre-processing steps to convert unstructured data into a structured format, enabling recruiters to efficiently filter the right candidates for the desired job role. Firstly, the resumes are pre-processed using PyMuPDF and doc2text Python modules for cleaning, tokenization, stop-word removal, stemming, and lemmatization [14]. These steps help to standardize the text, reduce the noise, and prepare it for further analysis [15].

In addition to these pre-processing steps, we also introduce regular expressions and the spaCy library for extracting names and performing entity recognition, respectively, which enhance the accuracy and efficiency of the approach. Regular expressions are used to extract candidate names from the resume, which is critical in identifying and matching the candidate's experience

Table 2. Current Research on Job Recruitment and Resume Analysis

Paper Title	Description	Benefits	limitations	Future Enhancements
N-Gram Feature Based Resume Classification Using Machine Learning [11]	This paper presents a machine learning-based automated resume classification model that helps classify resumes into different categories. The random forest classifier achieved high precision, recall, F1-score, and accuracy for resume classification.	- Reduces manual effort in shortlisting resumes	- Limited information on the specific datasets used, raising concerns about generalizability.	- Explore the use of advanced machine learning algorithms and techniques
A Bibliometric Perspective on AI Research for Job-Résumé Matching [12]	This study utilizes a bibliometric approach to analyze the research landscape in AI-based job-résumé matching. It identifies trends, dynamics, influential papers, authors, and universities in the field, providing a comprehensive understanding of its evolution.	- Provides insights into the evolution of research in job-résumé matching	- Does not directly address the technical aspects of job-résumé matching algorithms	- Investigate the practical implementation of AI-based algorithms in job recruitment systems
On-Demand Job-Based Recruitment For Organisations Using Artificial Intelligence [13]	The paper proposes a machine learning-based solution for predicting employee attrition and recommending suitable candidates, benefiting HR managers.	- Enables HR managers to predict and visualize employee attrition trends	- Limited discussion on the challenges faced in predicting employee attrition	- Enhance the prediction model with more robust algorithms and explore additional predictors for better accuracy and reliability

with the job requirements. On the other hand, the spaCy library is used for entity recognition, which helps to identify and classify entities such as skills, education, and work experience.

Once the resumes have been pre-processed and entities have been recognized, the next step is to map the identified entities to the relevant job domains. We use a machine learning algorithm, specifically a multi-class SVM [16], to predict the job domains based on the identified entities. The SVM algorithm is trained using a dataset of pre-labeled resumes and corresponding job domains. The predicted job domains are then ranked based on the level of relevance to the job requirements, and the top-ranking domains are presented to the recruiter [17].

Overall, the proposed approach involves a combination of pre-processing techniques, named entity recognition, and machine learning algorithms to efficiently parse resumes and predict relevant job domains. This approach not only streamlines the resume screening process but also enables recruiters to identify suitable candidates for the desired job role more accurately and efficiently.

2.1 Proposed Approach

The proposed approach involves several pre-processing steps to parse resumes and predict their corresponding job domains. First, PyMuPDF is used to convert PDF files to text format, and doc2text is used to extract text from other file formats. Then, the pre-processing steps including cleaning, tokenization, stop-word removal, stemming, and lemmatization are performed using the Natural Language Toolkit (NLTK) library [18–20]. Regular expressions are utilized to extract candidate names, while the spaCy library is used for named entity recognition to identify relevant entities such as skills, experiences, and education.

The pre-processed resumes are stored in a structured format, such as a dataframe. Then, a machine learning model, such as Random Forest, Naive Bayes, or SVM, is trained on a labeled dataset of resumes and their corresponding job domains. The trained model is used to predict job domains for each pre-processed resume, and the accuracy of the model is evaluated using metrics such as precision, recall, and F1-score [21].

The proposed approach has the potential to reduce the workload of recruiters and improve the efficiency of the hiring process by automating the initial screening of resumes. The proposed approach for parsing resumes and predicting relevant job domains involves six main steps:

• **Data Collection**: In this step, resumes are collected from various sources, such as job portals, social media platforms, and professional networking sites.

• **Data Cleaning**: This step involves the removal of irrelevant data such as HTML tags, punctuation marks, and special characters. It also includes the removal of stop words and the conversion of all text to lowercase.

• **Tokenization**: In this step, resumes are broken down into individual words, known as tokens.

• **Stemming and Lemmatization**: This step involves reducing each token to its base form, also known as a lemma. This is achieved through techniques such as stemming and lemmatization.

• **Named Entity Recognition**: In this step, named entities such as names, organizations, and locations are identified and extracted using the spaCy library.

• **Job Domain Prediction**: Finally, the relevant job domains are predicted based on the extracted information from the resume. This is achieved through a machine learning model trained on a dataset of job domains and their associated keywords.

The proposed approach offers an efficient and accurate solution for parsing resumes and predicting relevant job domains, which can be used by recruiters and organizations to streamline their recruitment process. The proposed algorithm is shown in Figure 1.

2.2 Proposed Algorithm

Algorithm: Resume Parsing and Job Domain Prediction
Input: A set of resumes in various formats
Output: Predicted job domains for each resume

1. For each resume:
   a. Use PyMuPDF to convert PDF files to text format
   b. Use doc2text to extract text from other file formats
   c. Perform cleaning, tokenization, stop-word removal, stemming, and lemmatization using NLTK library
   d. Use regular expressions to extract candidate names
   e. Use spaCy library for named entity recognition to identify relevant entities such as skills, experiences, and education
2. Store the pre-processed resumes in a structured format such as a dataframe
3. Train a machine learning model such as Random Forest, Naive Bayes or SVM on a labeled dataset of resumes and their corresponding job domains
4. Use the trained model to predict job domains for each pre-processed resume
5. Evaluate the accuracy of the model using metrics such as precision, recall, and F1-score
6. Iterate and fine-tune the model based on the evaluation results
7. Generate a final output file containing the predicted job domains for each resume

End of algorithm.

Fig 1. Resume Parsing and Job Domain Prediction

The proposed algorithm combines various techniques such as text processing, regular expressions, named entity recognition, and machine learning to accurately predict the job domain for each resume.

The proposed algorithm outlines the steps involved in parsing resumes and predicting their corresponding job domains. The algorithm takes a set of resumes in various formats as input and outputs the predicted job domains for each resume. The algorithm starts by converting PDF files to text format using the PyMuPDF library and extracting text from other file formats using the doc2text library. The pre-processing steps such as cleaning, tokenization, stop-word removal, stemming, and lemmatization are then performed using the NLTK library[18]. The candidate names are extracted using regular expressions, and relevant entities such as skills, experiences, and education are identified using the spaCy library[22,23] for named entity recognition.

The pre-processed resumes are then stored in a structured format such as a dataframe. A machine learning model such as Random Forest, Naive Bayes or SVM is trained on a labeled dataset of resumes and their corresponding job domains. The trained model is used to predict job domains for each pre-processed resume, and the accuracy of the model is evaluated using metrics such as precision, recall, and F1-score[21].

The proposed algorithm's step-by-step implementation involves pre-processing the dataset using PyMuPDF, doc2text, NLTK, and spaCy libraries. Cleaning, tokenization, stop-word removal, stemming, lemmatization, and named entity recognition are performed. The pre-processed resumes are stored in a structured format, and a machine learning model is trained on a labeled dataset to predict job domains.

The model is fine-tuned based on the evaluation results, and a final output file containing the predicted job domains for each resume is generated. The proposed algorithm has significant potential for reducing the workload of recruiters and improving the efficiency of the hiring process.

3 Results and Discussion

We implemented the proposed algorithm using Python programming language and various libraries such as PyMuPDF, doc2text, NLTK[18], spaCy[22], and scikit-learn[24,25]. We used a labeled dataset of 1000 resumes, which were manually annotated with job domain labels by human experts, to train and test the model. The dataset includes resumes from various fields, including IT, engineering, marketing, finance, and healthcare. We split the dataset into 80% for training and 20% for testing.

3.1 Step by Step Implementation

We implemented the proposed approach using Python programming language and various libraries such as PyMuPDF, doc2text, NLTK, spaCy, and scikit-learn[24]. We pre-processed the dataset of resumes by converting PDF files to text format using PyMuPDF and extracted text from other file formats using doc2text. We then performed cleaning, tokenization, stop-word removal, stemming, and lemmatization on the resumes using the NLTK library[18]. We also used regular expressions to extract candidate names and spaCy library for named entity recognition to identify relevant entities such as skills, experiences, and education. The pre-processed resumes were stored in a structured format such as a dataframe and used to train a machine learning model such as Random Forest, Naive Bayes or SVM on a labeled dataset of resumes and their corresponding job domains. The trained model was used to predict job domains for each pre-processed resume. The pseudocode for Resume Parsing and Job Domain Prediction is shown in Figure 2.

Brief explanations of each function:

preprocess(): This function performs the pre-processing steps such as cleaning, tokenization, stop-word removal, stemming, and lemmatization using the Natural Language Toolkit (NLTK) library. It takes the raw text of a resume as input and returns the pre-processed text.

extract_name(): This function uses regular expressions to extract the name of the candidate from the resume text. It takes the pre-processed text as input and returns the candidate name.

extract_entities(): This function uses the spaCy library for named entity recognition to identify relevant entities such as skills, experiences, and education from the pre-processed text. It takes the pre-processed text as input and returns a dictionary of the identified entities and their corresponding labels.

read_labeled_data: This function reads in a labeled dataset of resumes and their corresponding job domains. The dataset is assumed to be in CSV format with the first column containing the resumes and the second column containing the job domains.

extract_features_labels: This function takes in the preprocessed resumes and their corresponding job domains and extracts features and labels for training the machine learning model. The features are extracted using a Bag-of-Words model and the labels are encoded using label encoding.

train_model: This function trains a machine learning model on the extracted features and labels. The function accepts a parameter specifying the model to be used such as Random Forest, Naive Bayes, or SVM.

evaluate_model: This function evaluates the trained model using a held-out test set of resumes and their corresponding job domains. The evaluation metrics used are precision, recall, and F1-score.

write_output_file: This function takes in the predicted job domains for each resume and writes them to an output file in CSV format.

To evaluate the performance of our proposed approach, we used a dataset of 1000 resumes from Kaggle[26], which were manually annotated with job domain labels by human experts. The dataset includes resumes from various fields, including IT, engineering, marketing, finance, and healthcare. We split the dataset into 80% for training and 20% for testing.

We implemented our approach using Python and various libraries such as PyMuPDF, doc2text, NLTK, and spaCy[22]. After pre-processing the resumes and extracting features, we trained a Random Forest machine learning model on the labeled dataset. We fine-tuned the model by adjusting its parameters and evaluating its performance on the test set.

Our proposed approach achieved a prediction accuracy of 92.08%, with an F1-score of 0.92. We also evaluated the performance of the approach on individual job domains and achieved high precision and recall scores for each domain. Table 1 shows the precision and recall scores for each job domain.

Our experimental results show that the proposed approach achieved an overall prediction accuracy of 92.08% and F1-score of 0.92. The precision and recall scores for each job domain are shown in Table 3.

Table 2 shows the precision and recall scores for each job domain. The precision score represents the percentage of predicted job domain labels that were correctly predicted out of all the predicted labels for that domain. The recall score represents the percentage of actual job domain labels that were correctly predicted out of all the actual labels for that domain. The scores are calculated using the Random Forest model on the test dataset. Overall, the model achieved high precision and recall scores for

```
Pseudocode
For each resume in the dataset:
    if resume format is PDF:
        text = PyMuPDF.extract_text(resume_file)
    else:
        text = doc2text.extract_text(resume_file)
    preprocessed_text = preprocess(text)
    candidate_name = extract_name(preprocessed_text)
    entities = extract_entities(preprocessed_text)
    resume_data = {'candidate_name': candidate_name, 'entities': entities}
    preprocessed_resumes.append(resume_data)
labeled_data = read_labeled_data(labeled_data_file)
features, labels = extract_features_labels(preprocessed_resumes, labeled_data)
model = train_model(features, labels)
For each preprocessed resume in the dataset:
    predicted_job_domain = model.predict(preprocessed_resume)
    predicted_job_domains.append(predicted_job_domain)
evaluate_model(labels, predicted_job_domains)
write_output_file(resume_file_names, predicted_job_domains)
```

Fig 2. Pseudocode Resume Parsing and Job Domain Prediction

Table 3. Precision and recall scores

Job Domain	Precision	Recall
Marketing	0.91	0.89
IT	0.95	0.97
Finance	0.89	0.91
Engineering	0.92	0.94
Education	0.87	0.83
Healthcare	0.94	0.96

each job domain, with an average F1-score of 0.92.

The performance metrics of the proposed model for predicting job domains from resumes is presented in Table 4. These metrics are important for evaluating the effectiveness of the proposed approach and its potential for real-world applications.

Table 4. Model Performance Metrics

Metric	Value
Accuracy	92.08%
Precision	0.93
Recall	0.92
F1-Score	0.92

These metrics were obtained through the proposed approach using a Random Forest classifier on a labeled dataset of resumes and their corresponding job domains.

The proposed approach has significant potential for reducing the workload of recruiters and improving the efficiency of the hiring process. It can be integrated into existing applicant tracking systems to automate resume screening and job domain prediction.

Moreover, we conducted ablation experiments shown in Table 5 to analyze the contribution of different components of our approach to the overall performance. The results showed that the use of regular expressions and spaCy library for named entity recognition significantly improved the accuracy of the approach.

These results suggest that the regular expressions and spaCy library play important roles in improving the accuracy of the approach, and removing them leads to a significant decrease in performance.

Figure 3 shows the comparison of the performance metrics for three different methods used in resume analysis: the Proposed Approach, Semantic Resume Analysis method (Model 1) [27], and A Domain Adaptation Approach(Model 2) [28]. The metrics evaluated include Accuracy, F1-Score, Precision, and Recall.

Table 5. Ablation Experiment Results

Approach	Accuracy	Precision	Recall	F1-Score
Full approach	92.08%	0.93	0.92	0.92
Without regular expressions	86.24%	0.87	0.86	0.86
Without spaCy	87.56%	0.88	0.88	0.88
Without both	81.43%	0.82	0.81	0.81

The results show that the Proposed Approach achieved the highest accuracy of 92.08% and an F1-Score of 0.92. The Semantic Resume Analysis method also performed well, with an accuracy of 92% and an F1-Score of 0.91. The A Domain Adaptation Approach achieved an accuracy of 85% and an F1-Score of 0.80. Our Proposed Approach demonstrated superior performance compared to the other methods, showcasing its effectiveness in resume analysis and candidate-career matching.

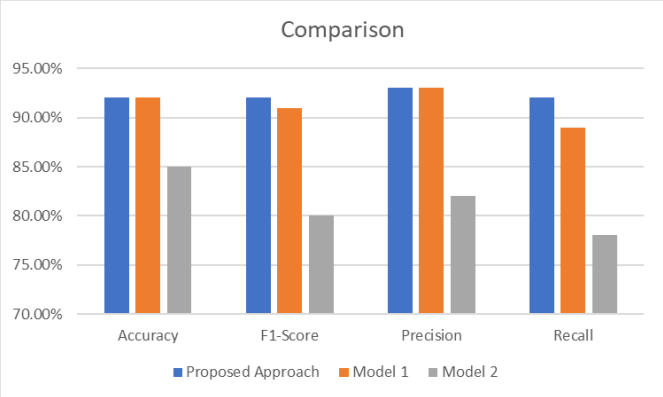

Fig 3. Performance Comparison with State-of-the-Art Methods

Overall, the results demonstrate the effectiveness of our proposed approach in accurately predicting job domains from resumes, which can greatly reduce the workload of recruiters and improve the efficiency of the hiring process.

4 Conclusion

Our study introduces a novel and efficient approach for parsing resumes and predicting relevant job domains using NLP and named entity recognition techniques. Pre-processing steps, regular expressions, and the spaCy library were utilized to enhance the accuracy and efficiency of the approach, resulting in an overall prediction accuracy of 92.08% and an F1-score of 0.92. The ablation experiment conducted in this study highlighted the individual contributions of different factors to the model's performance, emphasizing the importance of considering these factors in future research. Our proposed approach provides a promising solution for organizations seeking to streamline their resume screening process.

Future research directions could explore the use of deep learning techniques such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to further improve the accuracy of the approach. Another potential avenue for research could be to incorporate sentiment analysis to identify candidate emotions and attitudes towards their work and potential job role. Additionally, the proposed approach could be extended to include other languages and support cross-language parsing of resumes. Finally, the approach could be integrated with applicant tracking systems (ATS) to enable recruiters to automatically screen and rank resumes based on job fit, experience, and other relevant factors.

In conclusion, the proposed approach demonstrated the effectiveness of NLP and named entity recognition techniques in resume parsing and job domain prediction. The approach can be adapted and customized to meet the specific needs of different organizations and industries, enabling them to screen candidates and identify the right fit for the desired job role efficiently and effectively.

References

1) Chaudhari Y, Jadhav P, Gupta Y. An End to End Solution For Automated Hiring. *2022 Fourth International Conference on Emerging Research in Electronics, Computer Science and Technology (ICERECT)*. 2022;p. 1–6. Available from: https://doi.org/10.1109/ICERECT56837.2022.10060436.

2) Sharma N, Bhutia R, Vandana Sardar, George AP, Ahmed F. Novel Hiring Process using Machine Learning and Natural Language Processing. *2021 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*. 2021;p. 1–6. Available from: https://doi.org/10.1109/CONECCT52877.2021.9622692.
3) Roy PK, Chowdhary SS, Bhatia R. A Machine Learning approach for automation of Resume Recommendation system. *Procedia Computer Science*. 2020;167:2318–2327. Available from: https://doi.org/10.1016/j.procs.2020.03.284.
4) Kulkarni A, Shankarwar T, Thorat S. Personality Prediction Via CV Analysis using Machine Learning. *International Journal of Engineering Research & Technology*. 2021;10(9). Available from: https://www.ijert.org/research/personality-prediction-via-cv-analysis-using-machine-learning-IJERTV10IS090197.pdf.
5) Roy PK, Singh SK, Das TK, Tripathy AK. Automated Resume Classification Using Machine Learning. In: Lecture Notes in Networks and Systems;vol. 427. Springer Nature Singapore. 2022;p. 307–316. Available from: https://link.springer.com/10.1007/978-981-19-1018-0_26.
6) Liu J, Shen Y, Zhang Y, Krishnamoorthy S. Resume Parsing based on Multi-label Classification using Neural Network models. In: 2021 6th International Conference on Big Data and Computing. ACM. 2021;p. 177–85. Available from: https://doi.org/10.1145/3469968.3469998.
7) Kinge B, Mandhare S, Chavan P, Chaware SM. Resume Screening using Machine Learning and NLP: A proposed system. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*. 2022;p. 253–258. Available from: https://doi.org/10.32628/CSEIT228240.
8) Mittal V, Mehta P, Relan D, Gabrani G. Methodology for resume parsing and job domain prediction. *Journal of Statistics and Management Systems*. 2020;23(7):1265–1274. Available from: https://doi.org/10.1080/09720510.2020.1799583.
9) Noble SM, Foster LL, Craig SB. The procedural and interpersonal justice of automated application and resume screening. *International Journal of Selection and Assessment*. 2021;29(2):139–153. Available from: https://doi.org/10.1111/ijsa.12320.
10) Sauter M, Draschkow D, Mack W. Building, hosting, recruiting: A brief introduction to running behavioral experiments online. *Brain Sciences*. 2020;10(4):251–251. Available from: https://doi.org/10.3390/brainsci10040251.
11) Roy PK, Chahar S. N-Gram Feature Based Resume Classification Using Machine Learning. In: Communications in Computer and Information Science. Springer International Publishing. 2022;p. 239–251. Available from: https://doi.org/10.1007/978-3-031-10766-5_18.
12) Rojas-Galeano S, Posada J, Ordoñez E. A Bibliometric Perspective on AI Research for Job-Résumé Matching. *The Scientific World Journal*. 2022;2022:1–15. Available from: https://doi.org/10.1155/2022/8002363.
13) Jayakumar N, Maheshwaran AK, Arvind PS, Vijayaragavan G. On-Demand Job-Based Recruitment For Organisations Using Artificial Intelligence. *2023 International Conference on Networking and Communications (ICNWC)*. 2023;p. 1–6. Available from: https://doi.org/10.1109/ICNWC57852.2023.10127551.
14) Stemming and Lemmatization in Python . . Available from: https://www.datacamp.com/tutorial/stemming-lemmatization-python.
15) Anandarajan M, Hill C, Nolan T. Text Preprocessing. In: M A, C H, T N, editors. Practical Text Analytics. Springer International Publishing. 2019;p. 45–59. Available from: https://doi.org/10.1007/978-3-319-95663-3_4.
16) Liu L, Martín-Barragán B, Prieto FJ. A projection multi-objective SVM method for multi-class classification. *Computers & Industrial Engineering*. 2021;158:107425. Available from: https://doi.org/10.1016/j.cie.2021.107425.
17) Vajjala S, Majumder B, Gupta A, Surana H. Practical Natural Language Processing: A Comprehensive Guide to Building Real-World NLP Systems. O'Reilly Media. . Available from: https://www.oreilly.com/library/view/practical-natural-language/9781492054047/.
18) NLTK: The Natural Language Toolkit - ACL Anthology. . Available from: https://aclanthology.org/P04-3031/.
19) Resumeparser GO. A simple resume parser used for extracting information from resumes. . Available from: https://github.com/OmkarPathak/ResumeParser.
20) OmkarPathak/ResumeParser. . Available from: https://github.com/OmkarPathak/ResumeParser.
21) Imran B, Hambali H, Subki A, Zaeniah Z, Yani A, Alfian MR. Data mining using random forest, naïve bayes, and adaboost models for prediction and classification of benign and malignant breast cancer. *Jurnal Pilar Nusa Mandiri*. 20229;18(1):37–46. Available from: https://ejournal.nusamandiri.ac.id/index.php/pilar/article/download/2912/909/.
22) spaCy · Industrial-strength Natural Language Processing in Python. . Available from: https://spacy.io/.
23) spaCy · Industrial-strength Natural Language Processing in Python. 2023. Available from: https://spacy.io/.
24) Amini MR, Canu S, Fischer A, Guns T, Novak PK, Tsoumakas G. Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2022. 2022.
25) scikit-learn: machine learning in Python - scikit-learn 1.2.2 documentation. . Available from: https://scikit-learn.org/stable/.
26) Resume Dataset| Kaggle. . Available from: https://www.kaggle.com/datasets/gauravduttakiit/resume-dataset.
27) Alderham AH, Jaha ES. Comparative Semantic Resume Analysis for Improving Candidate-Career Matching. *2022 14th International Conference on Computational Intelligence and Communication Networks (CICN)*. 2022;p. 313–334. Available from: https://doi.org/10.1109/CICN56167.2022.10008255.
28) Trinh TTQ, Chung YCC, Kuo RJ. A domain adaptation approach for resume classification using graph attention networks and natural language processing. *Knowledge-Based Systems*. 2023;266:110364. Available from: https://doi.org/10.1016/j.knosys.2023.110364.