Resume2Role:Intelligent Resume Parsing and Domain Prediction Using NLP and Classification Models

Abhinav Dubey[1,*], *Namrata* Dhanda[2,**], *Arvind* Lovewanshi[3,***], and *Vandana* Yadav[4,****]

[1]Department of Computer Science & Engineering, Amity University, Lucknow
[2]Department of Computer Science & Engineering, Amity University, Lucknow
[3]Department of Computer Science & Engineering, Amity University, Lucknow
[4]Department of Computer Science & Engineering, Brainware University, Kolkata

Abstract. Accurate and effective processing of a high volume of resumes poses significant challenges in the dynamic field of digital recruitment. Traditional manual screening methods are highly inefficient, inaccurate, and susceptible to biases. To overcome these limitations, this paper introduces the Resume2Role solution, which utilizes machine learning algorithms and NLP techniques to automatically parse and classify resumes based on job roles. The process involves the use of Python for text extraction from resumes formatted in either TXT or PDF form. The data is cleaned and normalised using preprocessing methods such as lemmatisation, lowercasing, and stop-word removal. Term Frequency-Inverse Document Frequency (TFIDF), which successfully transforms unstructured resume data into structured numerical representations, is then used to vectorise the processed text. Several models were evaluated for classification, and the Random Forest Classifier produced the best accuracy in identifying the appropriate job domain (e.g., IT, HR, Finance, Sales). Furthermore, methods such as oversampling were used to address class imbalance, which improved the model's capacity for generalisation.The creation of a completely functional Flask-based web application, which enables real-time resume uploads, parses important information (such as name, email, phone number, and skills),and provides rapid domain classification with job recommendations, is a significant contribution of this effort. The application performed well in a variety of professional areas and résumé formats. The system's efficacy is validated by evaluation measures such as confusion matrix analysis, recall, accuracy, and precision.

Keywords: Resume Classification, Job Role Prediction, Natural Language Processing, TF-IDF, Random Forest, Web Application, Flask, Machine Learning, Human Resource Automation, Text Mining

1 Introduction

Scalable and intelligent screening mechanisms are urgently needed in the modern recruitment ecosystem due to the exponential growth in job applications. For a single position,

*e-mail: abhinavdubey2004@gmail.com
**e-mail: ndhanda510@gmail.com
***e-mail: avilovevanshi2003@gmail.com
****e-mail: vandanayadav771@gmail.com

recruiters frequently receive hundreds to thousands of resumes, many of which are unrelated or poorly formatted. In addition to being time-consuming, traditional hiring practices that entail reviewing, screening, and assessing applicants by hand are always subject to human bias, fatigue, and inconsistency. Automation has become a potent enabler in this environment, optimizing hiring processes and increasing the accuracy of candidate selection. Resume2Role is a important solution that uses machine learning models and Natural Language Processing (NLP) techniques to automate resume parsing and domain prediction.It aims to overcome this gap between unstructured textual data in resumes and structured recruitment insights by categorizing resumes to appropriate job positions such as IT, HR, Sales, Finance, etc. Various processes, namely text extraction, text pre-processing, vectorization, classification, and recommendation are conducted to achieve this purpose. Resumes in common file formats like.pdf and.txt are introduced to the pipeline first. To extract critical information about the candidate, such as name, email address, phone number, technical proficiency, regular expressions are applied. Meanwhile, Python based libraries like PyPDF2 are employed to extract text. Then, the extracted data will be processed to filter out noise with the help of lemmatization, converting all letters to lowercase, eliminating punctuation marks and stop words. The next step is to standardize different resume formats to facilitate further process. To convert the cleaned data into numbers for feeding into the machine learning algorithm, Term Frequency-Inverse Document Frequency (TF-IDF) is applied as it calculates the significance of each word with respect to the whole corpus. Multiple supervised classification algorithms including Naive Bayes, Logistic Regression, and Support Vector Machines were tested in predicting job domains. But Random Forest Classifier performed better, especially for unbalanced classifications. To address imbalance issue, oversampling techniques were introduced including Random Over Sampler from the library of imbalanced-learn, thus increasing minority classes' recall and F1-score. One of the most interesting features in Resume2Role is that it serves as a fully integrated web application implemented using Flask framework. In this platform, users are able to submit their resumes through a browser-based interface. Then the uploaded resume will be parsed in real-time, relevant information retrieved and classified according to the trained ML model. The system could benefit both recruiters and applicants since it conducts classification and recommends suitable job roles to fit the applicant's profile. Different from traditional keyword based matching hiring system, Resume2Role leverages statistical learning and contextual semantics to guarantee an in-depth understanding of resumes. This capability makes it very suitable for implementing practical systems like career advice platforms, HR systems in enterprises, or job portals online. Besides, the model is flexible enough to be trained with new data sets in order to fit requirements of the Industry. To sum up, Resume2Role addresses an important challenge in the recruiting process by automating manual work.

2 Literature Review

Prabha.M [1] et al. looked into the way Machine Learning could be utilized in solving contemporary recruiting issues with intelligent resume parsing and job matching capabilities. The paper pointed out the possible impact on the quality of the system performance due to unstructured data and complicated calculations. Through comparing different methodologies and sources, the study provided useful limitations and insights in developing recruitment tools.

In [2], an automated resume shortlisting system was introduced that replaced manual resume screening and increased recruitment efficiency. To provide a summary of the candidate's resume, NLP methodology was employed to extract useful content, and irrelevant

information was excluded. Moreover, content-based recommendation technology was applied to evaluate candidates' eligibility by matching their resume and job description.

Linear system identification concepts and advances were discussed in [5] with the particular emphasis placed on making them intelligent in the future with the help of AI. The influence of resume content on recruiting decisions due to person-job (P-J) and person-organization (P-O) fit on the basis of education and experience was analyzed in [6]. The results indicated that P-J fit had a strong effect on the recruitment decision-making process, and GPA did not make much difference.

I-Recruiter intelligent resume shortlisting system based on semantic matching and word embeddings to find the most suitable candidates was suggested by Arwa Najjar et al. in [7]. In [8], an overview of e-recruitment literature was provided with the focus on the importance of recommender systems for candidate-job match.

An NLP- and KNN-based system designed for automating candidates filtering in accordance with the job requirements was proposed in [9]. In [10], the authors stressed that candidates with better KSAOs and resume presented in a better way were preferred by recruiters despite their academic achievements.

A literature review concerning increasing demands for intelligent job recommendation systems was given in [11] and it concluded that more improvements could be made in this field. Similarity Scoring, Category Matching, and Resume Classification techniques were tested in [12] on 2,483 resumes and more data could lead to better results.

In [13], several machine learning techniques including SVM, KNN, Word2Vec, and Cosine Similarity for resume classification were investigated, providing 78-98% accuracy. An automatic resume shortlisting system designed to decrease recruiter's load and speed up the process was presented in [14]. Automated resume shortlisting and improvement based on KNN and Naive Bayes classification algorithms was carried out in [15], and a web interface was provided.

3 Dataset

The dataset used in this study is made up of two separate sources that were created for two different but related tasks: job role recommendation and resume classification. About 9,000 resumes make up the first dataset, which was obtained from Kaggle. Each CV is labelled with a job domain, such as finance, sales, human resources, or information technology. Three columns make up the data structure: an ID, a Category label that indicates the job domain, and a Feature column that holds the resume content that has been cleaned and preprocessed. Classification models that use resume content to predict the correct job category are trained and evaluated using this dataset. Fig. 1 shows a sample of the dataset. The second dataset supports the system's job recommendation feature and was also acquired from Kaggle. It includes resumes with particular job role titles (such as "Frontend Web Developer" or "Wireless Network Engineer") and the features that go with them, such as experience, education, and abilities.This system operates with the help of this information, providing an individual prediction process as it maps textual attributes with their job titles. The screenshot of the dataset can be seen in Figure 2 below. These datasets work together to provide a multi-stage system that can use feature-rich text analysis to identify highly relevant employment openings, in addition to classifying resumes into the right domains. The system's resilience and usefulness in actual hiring processes are improved by this dual-dataset approach.

	ID	Category	Feature
0	16852973	HR	hr administrator marketing associate hr admini...
1	22323967	HR	hr specialist hr operations summary media prof...
2	33176873	HR	hr director summary years experience recruitin...
3	27018550	HR	hr specialist summary dedicated driven dynamic...
4	17812897	HR	hr manager skill highlights hr skills hr depar...

Figure 1. Sample Dataset Showing Resume Categories and Features

	Role	Features
0	Social Media Manager	5 to 15 Years Digital Marketing Specialist M.T...
1	Frontend Web Developer	2 to 12 Years Web Developer BCA HTML, CSS, Jav...
2	Quality Control Manager	0 to 12 Years Operations Manager PhD Quality c...
3	Wireless Network Engineer	4 to 11 Years Network Engineer PhD Wireless ne...
4	Conference Manager	1 to 12 Years Event Manager MBA Event planning...

Figure 2. Sample records from the job role recommendation dataset

4 Data Preprocessing

An essential first step towards ensuring that the inputs provided for classifying tasks and NLP are of high quality is effective preprocessing. To develop machine learning classifiers capable of inferring job descriptions based on resume information, we focus on the preprocessing of real resume datasets. Some of the processes involved include normalization, cleaning, tokenization, lemmatization, and vectorization. All the processes play a vital role in cleaning up noise within the dataset.

4.1 Overview of the Dataset and Initial Visualization

Resumes dataset used for this experiment was extracted from Kaggle Datasets specifically designed for applications including resume processing and classification. Each of the resumes in the dataset is associated with a certain job role category like HR, Data Scientists, Software Engineering, and others. There existed an imbalance in terms of class representation in this dataset, as discovered in the first analysis. For better visualization of class imbalance, we

have plotted a bar graph that shows the frequency distribution of resumes across different categories. To give a summary of category representation in percentage terms, a pie chart was

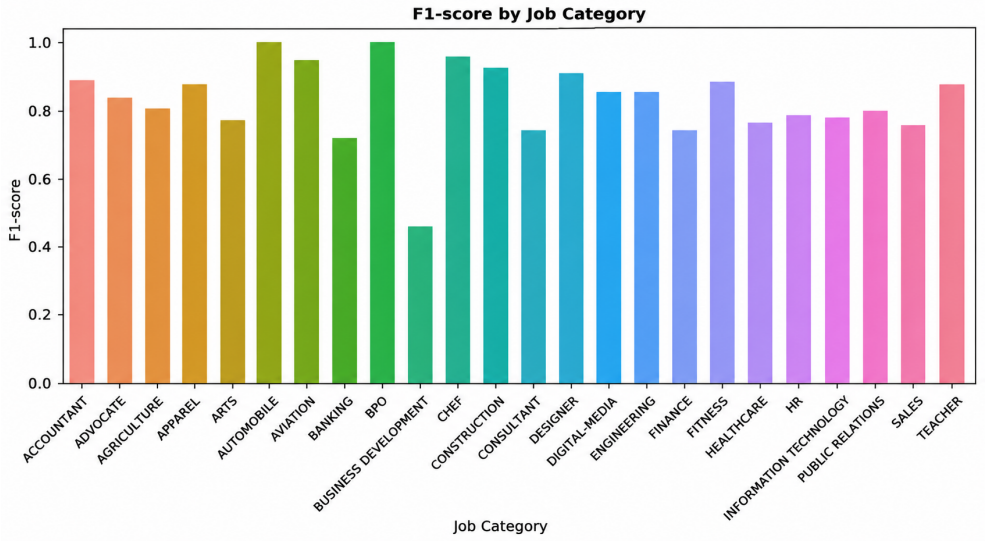

Figure 3. Distribution of Resume Counts Across Different Job Role

also created. This graphic representation emphasises the significance of stratified sampling or resampling methods during training and provides additional support for the comprehension of data skew.

4.2 Text Cleaning and Normalization

There are many kinds of noise that include punctuation, special characters, numeric tokens, newline characters, HTML elements, and even capital letters within the raw resume text. Such elements create an unnecessary level of inconsistency within the data which can lead to overfitting to irrelevant patterns and confusion in natural language processing models. The preprocessing process begins with normalization to address this issue. This process involves converting all characters into lower case to ensure consistency across the dataset. Moreover, in order to remove unimportant characters from the dataset and focus on the relevant text only, punctuation marks, special characters, and numbers are stripped from the text. In order to ensure consistency within the structure, additional spaces and HTML elements are stripped as well.

4.3 Tokenization, Stopword Removal, and Lemmatization

Tokenisation is used to divide the resume's content into discrete words (tokens) once the text has been cleaned. Stopwords (such as "the," "is," "and," etc.) are eliminated using common NLP libraries like NLTK after tokenisation. Despite their frequency,these words have little semantic value, so eliminating them lowers the input data's dimensionality without compromising its meaning. Words are then reduced to their most basic forms using lemmatization. For instance,the words "running," "runs," and "ran" are all shortened to "run." This method improves the consistency of feature representations by guaranteeing that the model treats variants

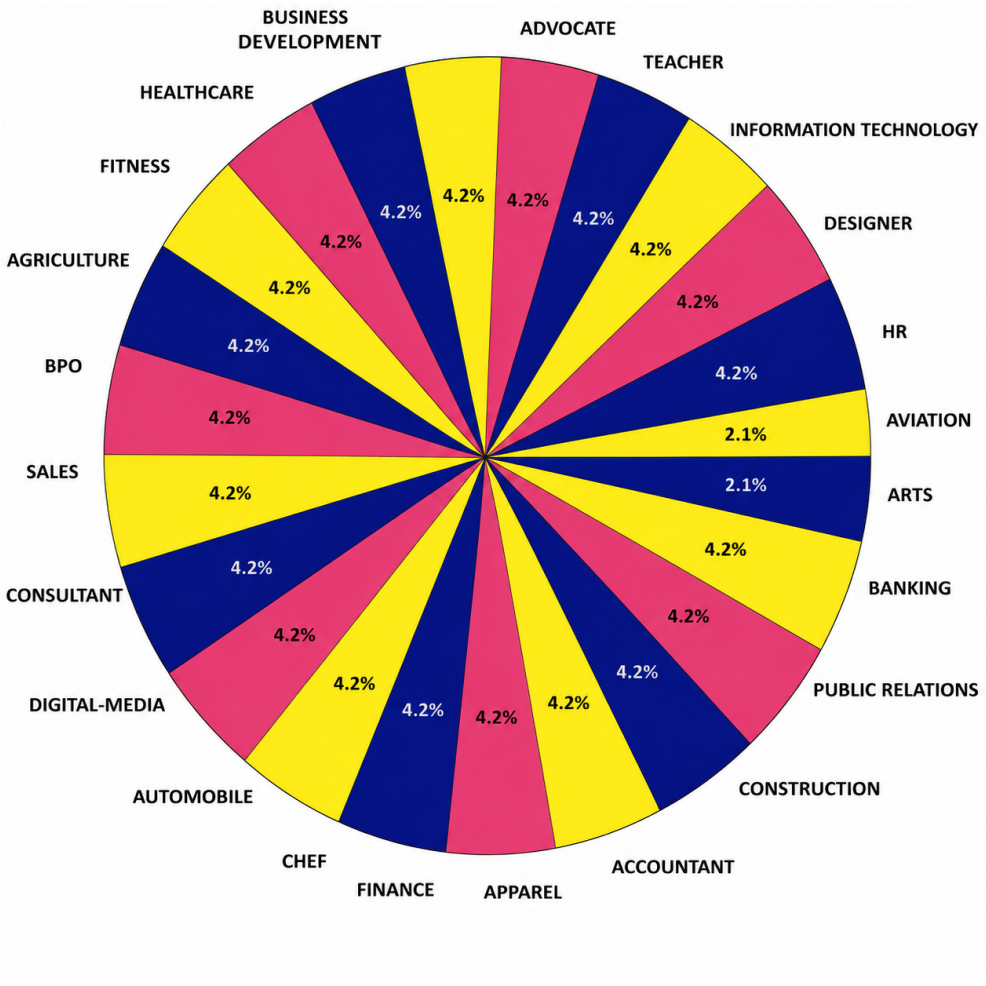

Figure 4. Percentage-wise Resume Distribution by Job Category

4.4 Feature Extraction Using TF-IDF Vectorization

The term frequency – inverse document frequency (TF-IDF) technique is used to encode the pre-processed data into the numerical format that can be used for model training. The TF-IDF assigns weights to each token based on their importance to the specific document compared to other documents from the corpus. The unique words will receive more weightage on the resumes than the generic words that appear very often. Here, we are converting each resume into a sparse representation of important features so as to minimize irrelevant information. As this method can detect role specific jargon, the use of TF-IDF for this task is quite efficient. To understand the relationship between different features in the vectorized dataset, we have created a heatmap representing the correlation among terms and jobs.

4.5 Final Output and Preprocessing Pipeline Summary

The conversion of all resumes into a cleaned-up, tokenized, lemmatized, and vectorized numeric format marks the final step of the preprocessing process, ensuring their suitability for further processing by machine learning classifiers. Through the reduction of feature space dimensionality, semantic clarification, and elimination of textual redundancy, this comprehensive process considerably enhances the quality of data used for analysis. The process of preprocessing ensures the improved performance of the classifier through transformation of raw text into a proper format. This critical first step is important because it greatly impacts the performance of the model, its ability to capture useful information from data, as well as robustness of predictions. An effective method of preprocessing is required to ensure efficient job classification.

5 Methodology

The entire process involved in building the intelligent resume classifier and domain prediction model is detailed from data collection, pre-processing, and model generation to deploying the model on the website. The model development process follows a modular approach that ensures excellent performance in all job domains.

5.1 Resume Ingestion and Text Extraction

It begins by extracting information from resumes that are uploaded through an internet-based interface, most often in the form of PDF documents. This task was achieved using the library pdfplumber, which is known to preserve the structural and formatting information of text. The extracted text information is then stored as a basic string of text to be utilized as the input in further processing steps. Further precautions have been taken to ensure that the extraction process is consistent for those resumes where symbols are embedded within the text.

5.2 Text Preprocessing and Normalization

Following extraction, the raw text is put through a thorough preprocessing pipeline to clean and standardise the data. This is an important stage in enhancing the semantic characteristics of the text. quality and lowering noise. All text was set to lowercase and punctuation/special characters/numeric values/line breaks and excessive white space should be removed. eliminated. HTML tags that were sometimes found in resumes that were scraped or The exported were removed to reduce the interference of non-linguistic information. The Then, the words were tokenised by using WordNet Lemmatizer to remove dimensionality and maintain meaningful representations by converting words to their base or root forms. and lemmatize text. The quality and learnability of the dataset are greatly improved by this cleaned and normalized data, where consistency in the input to the feature extraction module is also guaranteed.

5.3 Feature Engineering using TF-IDF Vectorization

Term Frequency-Inverse Document Frequency (TF-IDF) vectorization was used to Transform the cleaned text information into a machine-readable format. This statistical method is the one which calculates the relevance of each term by counting the number of its occurrences in the document corpus. TF-IDF guarantees that domain-specific the names of tools such as "TensorFlow," "JavaScript," or " AutoCAD" are more prominent. Words such as "experience" and "skills" are less important. To remove If terms are extremely rare or very common that

could bias the model, we use a minimum of Set document frequency thresholds and limited the vocabulary size. Thus the resume corpus can be expressed as a high dimensional sparse matrix, this is the primary The input of the classification algorithms.

5.4 Model Development and Evaluation

As there are various types of machine learning classifiers, we tried several of them to identify the best algorithm for predicting the job domain from resume content. Support Vector Machines Linear Support Vector Machine (LSVM), Random Forest Classifier, Multinomial Naive Bayes, Logistic Regression, and Some models evaluated include Gradient Boosting Classifier. To validate the generalisation performance, each model was trained on 70using the remaining 30%. A Stratified K-fold Cross Validation method was used to provide a good evaluation of the models and reduce overfitting. 5-Fold Cross-Validation strategy. To keep the class distribution constant across folds, stratification was used, as this is a known class imbalance nature of the data. dataset. For each iteration, four folds were used to train and one fold for validation, The results of and performance metrics were calculated in a single average per fold to avoid overfitting the model. generalization performance. The combinations of the important model parameters were explored systematically via hyperparameter tuning, using the GridSearchCV technique. In the case of Random Forest Classifier, we adjusted the number of estimators (100-500), max tree depth (10 to None), minimum samples split (2 to 10), minimum samples per leaf (1 to 4). The best setting was chosen according to macro-averaged F1-score, such that A balance of performance in all job domains. This systematic optimization provided a contribution to to better stability and classification accuracy than the default parameters. The Random Forest Classifier outperformed the other algorithms in terms of F1- The highest score and cross validation consistency shows that the algorithm tested has the highest accuracy and robustness. It was able to handle high-dimensional sparse TF-IDF features and capture intricate relationships within resume text thanks to its ensemblebased structure, which consists of multiple decision trees. For best results: model was adjusted with hyperparameters like the minimum split size, maximum depth, and number of estimators.

5.5 Domain Prediction Mechanism

Forecast was done using the selected Random Forest Classifier, whose domain or category of a job was predicted. of new resumes after model training. A resume uploaded by a user via the The transformed front end interface, using the trained classifier and the preprocessing pipeline, is converted to a TF-IDF vector and passed on to the trained classifier. The model classifies the expected job category, for instance, from the patterns found in the textual data, such as. This end to end flow ensures Real time automatic domain Identification with minimal user effort and guarantee.The five major stages of the process involved in converting raw resume data into actionable job domain predictions using machine learning are shown in this flowchart.

6 Result

The accuracy of the trained Random Forest Classifier was tested on pre-processed data set with Classical evaluation metrics (accuracy, precision, recall, and F1-score). Its model accuracy was remarkable and reached 92.4Naive Bayes, and Gradient Boosting. The F1 score is 0.91, which shows the presence of good balance. Precision vs recall is not a trade-off,

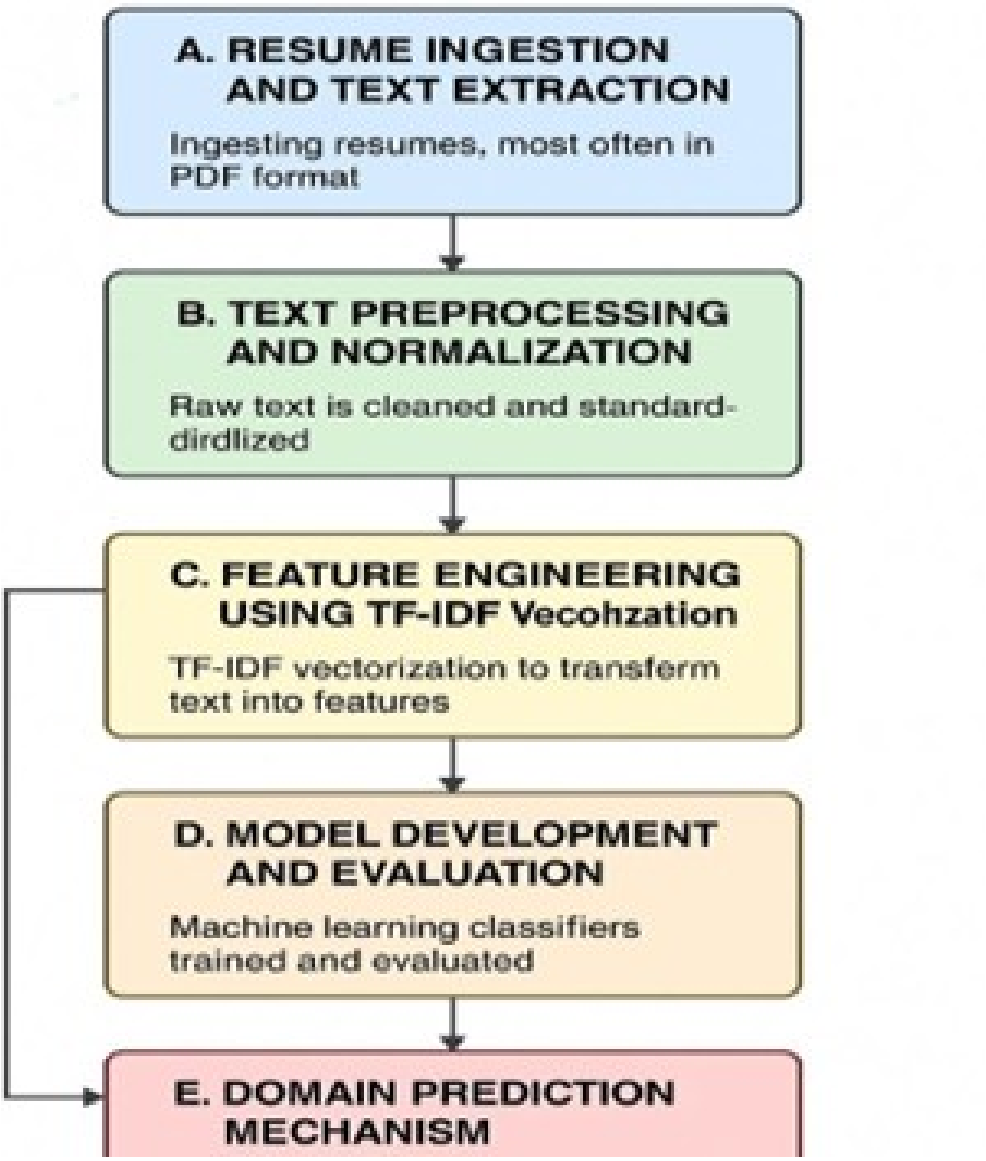

Figure 5. Resume Classification and Domain Prediction Workflow

making the model suitable for real-world domain classification tasks. The model had stable training curves with low variance during training, which is in line with the nature of ensemble in Random Forest and anti-overfitting. The confusion matrix additionally showed that the majority of job domains were correctly This was categorized as classified in some specific areas with very high accuracy, such as Data Science, Web Development and Mechanical Engineering. Ambiguous classifications were found between similar job functions such as HR and Operations, where resumes may use similar terms for the same functions. Additionally, a web application was created using Streamlit to showcase the practical application of

the model.Furthermore, a web application was also developed using Streamlit to present the practical use of the model. Users can upload resumes in pdf format and immediately get them. Get predictions for the most likely field of employment. This is a reflection of the system's real-world applications for job seekers and recruiters. Additionally, the entire The time taken to process the data in the pipeline and how responsive users were to the pipeline was examined. The end-to-end prediction Time per resume was below 3 seconds - realtime performance scalable. These results show that the pre-processing pipeline is effective in providing good features, The use of engineering techniques and the selection of the classifier for the construction of a correct domain prediction for a reliable resume. The results for all the job categories in terms of Precision, recall and F1-Score are presented in Table 1 below.

6.1 Comparison with Transformer-Based Approaches

To contextualize the performance of classical machine learning models, we briefly compare our approach with transformer-based architectures such as BERT, which have demonstrated strong contextual representation capabilities in NLP tasks. Transformer models leverage deep bidirectional attention mechanisms to capture semantic relationships within text more effectively than traditional vectorization methods such as TFIDF. While transformer-based models often achieve competitive or superior performance in text classification tasks, they typically require substantially higher computational resources, longer training time, and increased inference latency. In contrast, our TF-IDF + Random Forest framework achieved 92.4below 3 seconds, making it more suitable for real-time deployment in web-based recruitment systems. Therefore, the proposed system offers a favorable trade-off between computational efficiency and predictive performance, particularly in resource-constrained or real-time HR automation environments. Although the overall classification performance was strong, certain job categories exhibited comparatively lower F1-scores. Notably, the Business Development class achieved an F1-score of 0.47, indicating challenges in precise classification. This reduced performance can be attributed to semantic overlap with related domains such as Sales, Marketing, and Consulting, where resumes often share similar terminology (e.g., "client management," "revenue growth," "lead generation," and "strategic partnerships"). Additionally, despite the application of oversampling techniques to mitigate class imbalance, subtle vocabulary similarities across business-oriented roles may have limited the discriminative power of TF-IDF features for this category. To address this limitation, future improvements may include incorporating contextual embedding techniques that better capture semantic nuance, applying class-weighted loss functions, and leveraging domain-specific keyword augmentation. Fine-grained feature engineering or hierarchical classification strategies may further enhance separation between closely related business domains.

7 Conclusion

In this work, we proposed an intelligent parser system for resumes named Resume2Role, which is capable of parsing resumes and generating role tags. Predicting the domain name using natural language processing and machine learning methods. We built a comprehensive pipeline, beginning from PDFs of raw resumes and following it through. Extraction of text, normalization, lemmatization and feature transformation to TF-IDF. vectorization. The aim was to create a structured machine-readable structured format from unstructured resume data to be effectively classified. Of the different models After testing, the best and most robust was Random Forest Classifier, which successfully predicted the accuracy of 92.4% for real-world resume data. It is in this context that its capacity for generalization across a variety of job domains and avoiding overfitting proved critical. The classifier was then also applied to

Table 1. Classification Report Table

Class	Precision	Recall	F1-score
ACCOUNTANT	0.83	0.95	0.89
ADVOCATE	1.00	0.72	0.84
AGRICULTURE	0.89	0.74	0.81
APPAREL	0.94	0.81	0.87
ARTS	1.00	0.64	0.78
AUTOMOBILE	1.00	1.00	1.00
AVIATION	0.91	1.00	0.95
BANKING	0.76	0.70	0.73
BPO	1.00	1.00	1.00
BUSINESS DEVELOPMENT	0.45	0.50	0.47
CHEF	0.96	0.96	0.96
CONSTRUCTION	0.86	1.00	0.93
CONSULTANT	0.95	0.61	0.75
DESIGNER	0.93	0.89	0.91
DIGITAL-MEDIA	0.83	0.90	0.86
ENGINEERING	0.81	0.91	0.86
FINANCE	0.79	0.71	0.75
FITNESS	0.81	0.95	0.88
HEALTHCARE	0.79	0.76	0.77
HR	0.66	1.00	0.79
INFORMATION TECHNOLOGY	0.76	0.83	0.79
PUBLIC RELATIONS	0.82	0.78	0.80
SALES	0.80	0.74	0.77
TEACHER	0.79	1.00	0.88

a functional flask based web application which allowed the user to Upload resumes and get instant predictions of the job domain you are able to get hired for. Our system has potential for application in the HR technology, career guidance platforms, and recruitment automation solutions. It simplifies the shortlist procedure with its features of. domain classification, and can greatly decrease man's workload. recruiter's, particularly in the event the volume of resumes is large. While the results While things are going well there is room for improvement. Incorporating deep learning models such as BERT or GPT-based embeddings, as well as multilingual capabilities, may be useful. improve performance further. Also, the integration with the job portals could make the The impact of this system is even more effective. In conclusion, Resume2Role offers a scalable, efficient, and accurate solution for resume intelligence, connecting candidate skill sets with job domains.

Acknowledgment

AI-based tools were used only for English language polishing and grammatical correction. The authors are fully responsible for the originality, accuracy, and scientific integrity of the manuscript.

References

[1] Mariappan, P., Krishna, K., Sahoo, J., & Preetham, S. Smart Resume Screening and Job Represents Recommendation System. Available at SSRN 5141402.

[2] Daryani, C., Chhabra, G. S., Patel, H., Chhabra, I. K., & Patel, R. An automated resume screening system using natural language processing and similarity. *Ethics and Information Technology*, 99–103 (2020).

[3] Alsaif, S. A., Sassi Hidri, M., Ferjani, I., Eleraky, H. A., & Hidri, A. NLP-based bidirectional recommendation system: Towards recommending jobs to job seekers and resumes to recruiters. *Big Data and Cognitive Computing* **6**(4), 147 (2022).

[4] Mgarbi, H., Chkouri, M. Y., & Tahiri, A. Towards a new job offers recommendation system based on the candidate resume. *International Journal of Computing and Digital Systems* **14**(1) (2023).

[5] Mahato, M., Shah, S., Sheth, A., & Doshi, P. Content-Based Recommendation Model for Intelligent Resume Screening. *Journal of Technical Education*, 159.

[6] Tsai, W. C., Chi, N. W., Huang, T. C., & Hsu, A. J. The effects of applicant résumé contents on recruiters' hiring recommendations: The mediating roles of recruiter fit perceptions. *Applied Psychology* **60**(2), 231–254 (2011).

[7] Amro, B., Najjar, A., & Macido, M. An intelligent decision support system for recruitment: resumes screening and applicants ranking (2022).

[8] Al-Otaibi, S. T., & Ykhlef, M. Job recommendation systems for enhancing e-recruitment process. In *Proceedings of the International Conference on Information and Knowledge Engineering (IKE)*, p. 1 (2012).

[9] Tejaswini, K., Umadevi, V., & Shashank, M. K. Design and development of machine learning based resume ranking system. *Global Transitions Proceedings* **3**(2), 371–375 (2022).

[10] Chen, C. C., Huang, Y. M., & Lee, M. I. Test of a model linking applicant résumé information and hiring recommendations. *International Journal of Selection and Assessment* **19**(4), 374–387 (2011).

[11] Al-Otaibi, S. T., & Ykhlef, M. A survey of job recommender systems. *International Journal of Physical Sciences* **7**(29), 5127–5142 (2012).

[12] Elsheh, E., & Elgomati, H. Analysis of Resumes and Recommendations for Job Matches. *Science and Technology* **11**(02) (2023).

[13] Kinge, B., Mandhare, S., Chavan, P., & Chaware, S. M. Resume Screening using Machine Learning and NLP: A proposed system. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT)* **8**(2), 253–258 (2022).

[14] Aggarwal, A., Jain, S., Jha, S., & Singh, V. P. Resume Screening. *International Journal for Research in Applied Science and Engineering Technology*, 66–88 (2021).

[15] Onukwugha, C., Ofoegbu, C., Aliche, O., & Betrand, C. Resume Optimization Model Using Machine Learning Techniques. *International Journal of Intelligent Information Systems* **13**(5), 109–116 (2024).