NLP-based Resume Classification and Ranking System

Krishna Vamsi Gujjula
MS in Computer Science
University Of Central Florida

Sai Sriram Kurapati
MS in Computer Science
University Of Central Florida

ABSTRACT

In today's highly competitive job market, companies receive a large volume of resumes for each job opening. This can make it challenging and time-consuming for hiring managers to manually review and screen every resume to identify the most qualified candidates. To address this issue, we propose an automated system that uses Natural Language Processing (NLP) techniques to parse and categorize resumes and match them to job descriptions. Our system can help reduce the time and resources required for manual resume review, while also improving the quality of hiring decisions. By leveraging NLP techniques, our system can accurately extract and analyze information from resumes, such as relevant skills, experience, education, and other qualifications. It can then compare this information with job descriptions and rank resumes based on their relevance and suitability for the job. In addition to improving the efficiency of the hiring process, our system can also help reduce bias and ensure a fair and objective evaluation of all resumes. It can eliminate human errors and inconsistencies that may occur in manual screening and ensure that every candidate is evaluated based on the same criteria. Overall, our NLP-based resume classification and ranking system can revolutionize the way companies hire employees. It can enable hiring managers to focus their time and resources on the most qualified candidates, resulting in better hiring decisions and a more productive workforce.

Keywords

Resume classification, resume ranking, natural language processing, NLP, job matching, job descriptions, automated system, supervised learning, data preprocessing, data cleaning.

1. INTRODUCTION

The recruitment process is a critical component of any organization's success, as it directly impacts on the company's performance and growth. An organization's hiring process starts with a job posting or opening, followed by the reception of resumes from job seekers. The resumes are typically reviewed by HR professionals and recruiters who are tasked with screening and identifying the most suitable candidates for the job. However, with the increase in the number of job seekers and the volume of resumes that companies receive, manually reviewing and identifying the best candidate can be a time-consuming and tedious task. Additionally, the manual screening process can be prone to errors, leading to the possibility of missing out on a qualified candidate.

To tackle this challenge, many organizations are turning to automated systems to assist in the recruitment process. One of the critical components of such systems is the ability to classify resumes into various categories based on their content. Resume classification involves analyzing resumes and extracting relevant information, such as work experience, education, and skills, and then categorizing the resumes into different groups based on the information extracted. These groups can be based on various factors, such as job titles, industry, or skill sets.

Another essential aspect of the recruitment process is resume ranking. Once resumes are classified into different categories, the next step is to rank them based on their relevance to the job opening. Resume ranking using NLP techniques involves assigning scores or weights to resumes based on their similarity to job requirements and their relevance to a job description. This process can help to identify the most suitable candidates for the job, based on their skills, experience, and other relevant factors.

To develop an automated system that uses NLP techniques for resume classification and ranking, we propose a system that will parse and categorize resumes based on their content, and then match them to job descriptions using NLP techniques. The proposed system will be designed to automate the recruitment process, saving time and resources while ensuring that the best candidates are identified for the job opening.

The proposed system will be built using various NLP techniques, such as tokenization, stemming, stop-word removal, and vectorization. These techniques will be used to preprocess and transform the raw text data into a format that can be used to train and evaluate machine

learning models. We will use various classification algorithms, such as Naive Bayes, Random Forest, and Gradient Boosting, to build and evaluate the models. These models will be trained using a dataset of resumes and job descriptions and then tested to evaluate their performance in classifying resumes and ranking them based on their relevance to job descriptions.

Overall, the proposed automated system that parses and categorizes resumes and matches them to job descriptions using NLP techniques can help organizations improve the quality of their hiring decisions, reduce costs, and streamline their recruitment process. The system will provide a faster and more accurate way to identify the most suitable candidates for a job opening, leading to a more efficient and effective recruitment process.

2. PROBLEM STATEMENT

The traditional approach to screening resumes and matching them to job descriptions involves manual review and comparison, which can be both time-consuming and prone to errors. Recruiters often spend hours reviewing resumes, searching for relevant keywords, and assessing candidates based on their qualifications and experience. This process can be particularly challenging for companies that receive a high volume of resumes for each job opening, making it difficult for recruiters to efficiently identify and evaluate the most promising candidates. Furthermore, the traditional resume screening process is often marred with biases that can impact the quality and diversity of the candidates selected.

To address these challenges, our proposed NLP-based resume classification and ranking system aims to streamline the recruitment process by automating resume screening and matching. Our system uses advanced NLP techniques to analyze and categorize resumes based on their content, including work experience, education, skills, and other relevant factors. By doing so, we can quickly identify the most qualified candidates for the job and provide an objective evaluation of their qualifications.

In summary, the proposed NLP-based resume classification and ranking system is designed to help organizations streamline their recruitment process, reduce bias, and improve the efficiency and effectiveness of their hiring decisions. By leveraging advanced NLP techniques, we can provide an objective evaluation of candidates based on their qualifications and experience, ultimately leading to a more diverse and qualified workforce.

3. RELATED WORK

The paper "A Machine Learning Approach for Automation of Resume Recommendation System" [1] proposes an innovative solution to the problem of resume ranking and classification using machine learning techniques. The authors acknowledge that the process of screening and ranking resumes manually can be a daunting task, especially for companies that receive a large number of resumes for each job opening. By developing an NLP-based system, the authors seek to reduce the amount of time and bias in the screening process by automating it. The proposed system leverages both content-based and collaborative filtering techniques, where resumes are represented as vectors using feature extraction techniques such as TF-IDF and LSA. The content-based approach uses the textual information in resumes to find similar resumes, while the collaborative filtering approach uses the similarity between employers to recommend resumes.

To evaluate the performance of the proposed system, the authors conducted experiments using different machine learning algorithms such as k-NN and SVM. They also compared the performance of the proposed system with baseline approaches using precision and recall metrics. The results showed that the proposed system outperformed baseline approaches in terms of precision and recall metrics, demonstrating its effectiveness in classifying and ranking resumes. This finding is encouraging, as it shows that the proposed system has the potential to significantly reduce the time and effort required for screening resumes and identifying the most suitable candidates for job openings.

The paper "Resume Screening using Machine Learning and NLP: A proposed system" [2] proposes a system for automating the process of resume screening using machine learning and NLP techniques. The proposed system utilizes various pre-processing techniques such as tokenization, stop word removal, stemming, and part-of-speech tagging to extract features from the resumes. The extracted features are then used to train machine learning models such as decision trees, support vector machines, and logistic regression to classify the resumes into different categories based on their relevance to the job descriptions.

The authors evaluated the proposed system on a dataset consisting of 500 resumes and achieved an accuracy of 85.2%. They also compared the proposed system with a traditional keyword-based approach and found that the machine learning and NLP-based approach outperformed the keyword-based approach in terms of

accuracy and efficiency. Overall, the paper presents a promising solution for automating the process of resume screening using machine learning and NLP techniques.

4. DATASET & PRE-PROCESSING

The dataset is acquired from Kaggle, which is a platform for data scientists and machine learning enthusiasts to find and share datasets. This dataset consists of 2208 resumes in .pdf or .docx format, which are commonly used file types for resumes.

The dataset also includes a .csv file which contains information about each resume. This includes a Unique ID to identify each resume, "Resume_str" which is the text content of the resume, "Resume_html" which is the HTML code for the resume (but is not relevant for our purposes), and Category which is the category or type of job the resume is intended for.

Since we only need the Resume_str and Category columns, we drop the Unique ID and Resume_html columns for the classification problem. This will make our data easier to work with and will eliminate any unnecessary information.

Next, we split the dataset into training and testing sets with a 80:20 ratio. This means that 80% of the data will be used for training our machine learning model, and the remaining 20% will be used for testing. The 'Stratify' parameter is used to ensure an equal distribution of categories in both the training and testing sets. This helps to prevent bias and ensure that our model is trained and tested on a representative sample of the data.

4.1. Data Pre-Processing

Preprocessing the resume data was a crucial step in our analysis as it allowed us to transform the raw text data into a standardized and more manageable format. We began by reading the resume data from a CSV file using the pandas library in Python. Once we had the data, we focused on the columns containing the resume text and dropped any unnecessary columns to simplify the dataset.

Next, we applied several pre-processing techniques to standardize the text data and remove any inconsistencies. This involved converting all characters to lowercase to avoid issues with case sensitivity, removing non-English characters, punctuation, and numbers to clean the text data. We then tokenized the words to separate them into individual units and removed stop words to eliminate commonly used words that do not add much value to the analysis. Lastly, we applied stemming to reduce words to their base form, making it easier to identify and analyze the main themes in the resume data. These pre-processing steps were essential in ensuring that the data was consistent, clean, and ready for analysis using machine learning algorithms.

Data visualization was another important aspect of our analysis. After pre-processing the data, we visualized it using bar charts and word clouds to gain insights into the data and understand the most common words and phrases used in each category. The bar charts helped us visualize the frequency of occurrence of each word in each category, while the word clouds provided a visual representation of the most used words in each category. This information helped us choose appropriate pre-processing techniques and understand which words and phrases to prioritize in our analysis. By visualizing the pre-processed data, we were able to gain a deeper understanding of the content and structure of resumes and make more informed decisions when analyzing the data.

5. TECHNIQUES

5.1. Resume Classification

Resume classification is a vital task in the recruitment process, as it helps recruiters and HR professionals to identify the most suitable candidates for specific job openings quickly. However, with the increasing volume of resumes received by organizations, the manual sorting of resumes becomes challenging, time-consuming, and error prone. Natural Language Processing (NLP) techniques can help automate this process by analyzing the content of resumes and assigning them to relevant categories.

To classify resumes based on their content, we used a combination of pre-processing techniques, data visualization, and classification models. After visualizing the preprocessed data, we split the data into training and test sets using the train_test_split function from the scikit-learn library. This step was crucial to ensure that our model could generalize well and perform well on new, unseen data.

We then vectorized the text data using the CountVectorizer function from scikit-learn, which converts the text into a matrix of word frequencies. This vectorized data was fed into various classification models such as Support Vector Classifier, Naive Bayes, Random Forest Classifier, Multi-Layer Perceptron, and Gradient Boosting Classifier. Each model was trained on the training data, and its

performance was evaluated using cross-validation techniques.

To improve the performance of each model, we used the GridSearchCV function from scikit-learn to tune the hyperparameters of each model. GridSearchCV performs a search over a range of hyperparameters and evaluates each combination using cross-validation, resulting in the best-performing model.

Finally, we evaluated the performance of the best-performing model on the test data using the classification report to measure its accuracy, precision, recall, and F1 score. The classification report provided a detailed summary of the model's performance for each category and helped us understand its strengths and weaknesses.

The process of resume classification involves several steps, including pre-processing the text data, visualizing the data to gain insights, selecting appropriate models, and evaluating their performance. This technique enables organizations to organize a large volume of resumes into distinct categories, making it easier to identify relevant candidates for specific job openings. NLP-based resume classification and ranking systems can significantly reduce the time and effort required for recruitment and help organizations make more informed hiring decisions.

5.2. Resume Ranking

In the second part, the objective is to determine the relevance of each resume to a specific job description. This involves analyzing the job description and comparing it to the content of each resume to determine the degree of match.

We used the same dataset as in the previous step. The dataset is in a .CSV file format and contains two fields: a unique identifier and a resume that is represented as a single string of text. We took the preprocessed data that was generated in the first part of the problem and added it as a third column to the .CSV file.

To accomplish this, we utilize the technique of TF-IDF (Term Frequency-Inverse Document Frequency) vectorization. This technique enables us to represent each resume and job description as a weighted vector, where the weight of each term corresponds to its importance in the document.

The TF-IDF technique is based on the concept of term frequency (TF) and inverse document frequency (IDF). The term frequency measures the frequency of each term in a given document, while the inverse document frequency measures how important a term is across all documents in the dataset. The combination of these two factors results in a weighted representation of the frequency of each term in a document.

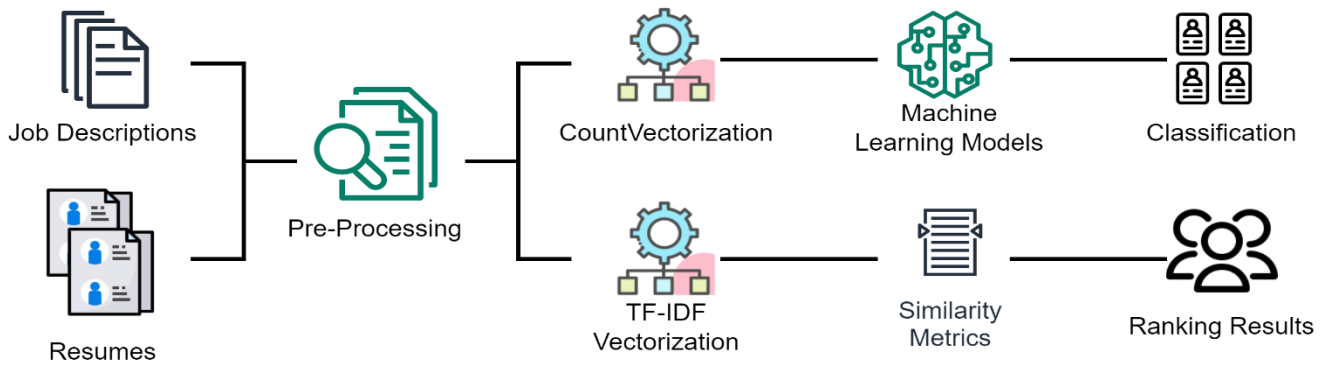
We use the max_df and min_df parameters to set the upper and lower bounds on the document frequency of terms that are included in the TF-IDF matrix. This helps to filter out the terms that are too common or too rare and only keep the terms that are most relevant.

Once we have the TF-IDF vectors for the job descriptions and resumes, we use four different text similarity metrics to determine the similarity between the job description and each resume. These metrics are Jaccard similarity, Sørensen–Dice coefficient, cosine similarity, and overlap similarity.

Jaccard similarity and Sørensen–Dice coefficient are both based on the overlap between the two sets of terms in the documents. The Jaccard similarity is defined as the size of the intersection of the two sets of terms divided by the size of the union of the two sets, while the Sørensen–Dice coefficient is defined as twice the size of the intersection divided by the sum of the sizes of the two sets. Both of these measures range from 0 to 1, with 1 indicating a perfect match between the two sets of terms.

Cosine similarity, on the other hand, measures the cosine of the angle between the two vectors that represent the documents in a multi-dimensional space. In this space, each dimension represents a different term, and the value of each dimension represents the frequency or importance of that term in the document[3].

Finally, overlap similarity measures the ratio of the number of common terms to the total number of terms in both documents. This measure also ranges from 0



to 1, with 1 indicating a perfect match.
By using multiple metrics, we can ensure a more comprehensive and accurate assessment of the similarity between the strings. We then calculate a score for each resume based on the similarity metrics, with higher scores indicating a greater relevance to the job description.
The ranking of the resumes based on their score allows us to identify the most qualified candidates for a specific job opening, making the recruitment process more efficient and effective. By automating the process of resume classification and ranking, we can save time and resources while ensuring that the best candidates are identified for each job opening.

6. EVALUATION

For the resume classification task, the table summarizes the evaluation results for five different models on a resume classification task. Gradient Boosting achieved the highest F1 score, precision, recall, and accuracy among the models tested. MLP also performed well with an F1 score of 69.2% and precision of 71.2%. Naive Bayes had the lowest performance with an F1 score of 59.9% and accuracy of 60.8%. Overall, Gradient Boosting is recommended for this task due to its high performance in all evaluation metrics.

Model	**F1**	**Precision**	**Recall**	**Accuracy**
Naive Bayes	**59.9%**	**63%**	**60.8%**	**60.8%**
Random Forest	**64%**	**68.8%**	**65%**	**65.8%**
SVC	**68.7%**	**70.4%**	**72%**	**67.6%**
MLP	**69.2%**	**71.2%**	**68%**	**68.4%**
Gradient Boosting	**71.9%**	**75.5%**	**72%**	**74.4%**

Overall, improving the accuracy of the model requires careful consideration of the specific characteristics of the dataset and appropriate preprocessing techniques to handle them.
Another way to improve accuracy is to address class imbalance. In the given dataset, some categories may have many more examples than others, making it more difficult for the model to accurately classify the under-represented classes.

To evaluate the second part of our project, which involves ranking resumes based on their relevance to a job description, we wanted to ensure that our evaluation was fair and unbiased. To achieve this, we decided to add a new column to our job description table, which represents the category of each job description. This allowed us to select 20 job descriptions, one from each category, to use in our evaluation.
After generating the top five resumes for each job description, we calculated the accuracy of our algorithm for each category. The accuracies that we obtained for each category are presented in the table below. These results provide insight into the effectiveness of our algorithm and can be used to further refine and improve the ranking system in the future.

Category	**Accuracy**
ACCOUNTANT	80%
ADVOCATE	100%
AGRICULTURE	100%
APPAREL	100%
AVIATION	80%
BANKING	80%
BUSINESS-DEVELOPMENT	100%
CHEF	100%
CONSTRUCTION	100%
DESIGNER	60%
DIGITAL-MEDIA	100%
ENGINEERING	80%
FINANCE	80%
FITNESS	100%
HEALTHCARE	100%
HR	100%
INFORMATION-TECHNOLOGY	80%
PUBLIC-RELATIONS	100%
SALES	100%
TEACHER	100%
TOTAL	**92.5%**

The statement indicates that the accuracy of resume ranking was evaluated based on the top 5 resumes for each category, and the accuracy obtained was 92.5%. This suggests that increasing the number of resumes beyond the top 5 for each category may result in a decrease in accuracy. This evaluation helps in understanding the performance of the resume ranking system and may guide future improvements.

7. LIMITATIONS

However, this project also had some limitations. One of the main limitations of this project is the lack of available labeled data for the resume ranking task. This made it difficult to evaluate the effectiveness of the ranking algorithm since we could not compare the

predicted rankings with the actual rankings of the candidates.

Another limitation of the project is the reliance on keyword matching to extract skills and experiences from resumes. This approach may not capture the full context and nuances of the candidate's abilities and may result in false positives or false negatives. Furthermore, the project was limited by the size and quality of the data set used for training and testing the machine learning models. In future work, we aim to expand the data set and improve its quality to further enhance the accuracy of our models.

8. DISCUSSION & CONCLUSION

In this conclusion, we developed an NLP-based system for resume classification and ranking using machine learning techniques. The goal of this project was to create a tool that can help recruiters and hiring managers quickly and accurately identify the most qualified candidates for a job opening.

To address this problem, we conducted a thorough review of related work in the field of NLP and machine learning for resume analysis. We found that there is a significant body of research on this topic, with many studies focused on developing models that can accurately classify resumes into different categories and rank them based on their relevance to a job posting.

After reviewing the literature, we began the data preprocessing stage, which involved cleaning and structuring the raw resume data. We used techniques such as tokenization, stop word removal, stemming, and lemmatization to prepare the data for machine learning analysis.

We then experimented with different machine learning techniques, including Naïve Bayes, support vector Classifier, Random Forest classifier, MLP Classifier and Gradient Boosting Classifier. We found that the random forest classifier outperformed the other models, achieving an accuracy of 74% with Gradient Boosting Classifier in resume classification.

In addition to resume classification, we developed a resume ranking system that assigns a score to each resume based on its relevance to a job posting. We used techniques such as cosine similarity, Jaccard similarity, Sørensen–Dice coefficient, overlap similarity and TF-IDF weighting to compare the content of the resume to the job posting and assign a ranking score.

Our key findings from this project were that NLP-based techniques can be highly effective for automating the process of resume screening and ranking. By using machine learning models and advanced NLP techniques, we were able to achieve good accuracy in resume classification and relevance ranking.

In future work, we can focus on expanding the dataset to include resumes and job descriptions from different domains can help to improve the generalizability of the model. This can be achieved by incorporating data from various industries, including healthcare, finance, and technology, to name a few. Another avenue for future work is to explore the use of deep learning techniques such as neural networks to improve the accuracy of the model. While the current approach of using cosine similarity works well, other algorithms such as PageRank or neural network-based models can be investigated for the ranking task to potentially improve performance.

Furthermore, given the unavailability of labeled data, the evaluation of the resume ranking model was difficult. Collecting more labeled data can enable more comprehensive evaluation and potentially improve the model's performance. This can be achieved by manually labeling resumes based on their relevance to a specific job or by using active learning techniques to collect relevant labels from domain experts.

9. REFERENCES

[1] Roy, Pradeep & Chowdhary, Sarabjeet & Bhatia, Rocky. (2020). “A Machine Learning approach for automation of Resume Recommendation system”. Procedia Computer Science. 167. 2318-2327. 10.1016/j.procs.2020.03.284.

[2] 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, 253–258. https://doi.org/10.32628/CSEIT228240

[3] Daryani, Chirag & Chhabra, Gurneet & Patel, Harsh & Chhabra, Indrajeet & Patel, Ruchi. (2020). "An automated resume screening system using natural language processing and similarity".99-103.