Resume Classifier

Syed Abrar C.
2^{nd} *year student (AIE)*
Amrita Vishwa Vidyapeetham (Artificial Intelligence)
Kasavanahalli, Bangalore
bl.en.u4aie22147@bl.students.amrita.edu

Nithesh Nair P.S.
2^{nd} *year student (AIE)*
Amrita Vishwa Vidyapeetham (Artificial Intelligence)
Kasavanahalli, Bangalore
bl.en.u4aie22133@bl.students.amrita.edu

Sachin R.
2^{nd} *year student (AIE)*
Amrita Vishwa Vidyapeetham (Artificial Intelligence)
Kasavanahalli, Bangalore
bl.en.u4aie22142@bl.students.amrita.edu

***Abstract*—We present a real-time resume classifier leveraging LinkedIn profiles for enhanced classification accuracy. Departing from traditional methods, we employ natural language processing (NLP) to extract rich features from LinkedIn profiles, augmenting resume classification. Domain adaptation techniques bridge feature disparities between LinkedIn and resumes, ensuring real-time efficacy. Convolutional neural networks (CNNs) and support vector machines (SVMs) enable high accuracy and efficiency. Extensive experiments on a large dataset validate our approach, showcasing its effectiveness in aligning resumes with job requirements. Our classifier offers a robust solution for automated resume classification, improving recruitment processes' efficiency and candidate selection quality.**

Keywords: Resume Classification

I. Introduction

Resume classification is a critical task in the recruitment process, facilitating the efficient matching of job seekers with suitable job openings. Traditional methods of resume classification often rely on simplistic keyword matching and rule-based systems, which may lack the sophistication needed to handle the diverse and nuanced nature of resumes. With the proliferation of online professional networking platforms such as LinkedIn, there is a wealth of additional information available in the form of user profiles that can be leveraged to enhance the resume classification process. In this paper, we propose a novel approach to real-time resume classification that harnesses the power of LinkedIn profile descriptions as a valuable source of supplementary information. By integrating natural language processing (NLP) techniques, domain adaptation strategies, and machine learning algorithms, our proposed system aims to overcome the limitations of traditional methods and provide a more accurate and efficient solution for resume classification. This introduction provides an overview of the motivation, objectives, and key contributions of our research, setting the stage for the detailed discussion of our proposed real-time resume classifier in subsequent sections.

II. Literature Survey

[1]Research in the field of resume classification has explored a variety of methodologies aimed at automating the process of candidate screening. Traditional approaches include keyword matching and rule-based systems, which rely on predefined criteria to match job requirements with candidate resumes. These methods, while effective to some extent, often lack the flexibility to adapt to evolving job descriptions and may produce inconsistent results.

[2]In recent years, there has been a shift towards employing machine learning algorithms for resume classification tasks. Support vector machines (SVM), decision trees, random forests, and neural networks have been among the most commonly utilized algorithms due to their ability to learn from data and make predictions based on learned patterns. Additionally, natural language processing (NLP) techniques have played a significant role in enhancing the accuracy of resume classification systems by enabling the analysis of textual information within resumes. Advanced NLP methods such as word embeddings and deep learning models have shown promise in improving semantic understanding and feature representation, ultimately leading to more effective candidate screening.

[3]Studies have also investigated the costs and benefits associated with the implementation of automated resume classification systems. Benefits include increased efficiency in candidate screening, reduced manual effort, and faster recruitment processes. However, there are also potential challenges to consider, such as algorithmic biases and fairness in decision-making. Evaluating the fairness and transparency of automated systems is crucial to ensure that they do not

perpetuate or exacerbate existing biases in hiring practices.

[4]Moreover, while machine learning and NLP techniques dominate the literature on resume classification, there is a growing interest in exploring the applicability of signal processing techniques, particularly in scenarios involving audio or visual resumes. Overall, the literature reflects a multidisciplinary approach to resume classification, drawing on methodologies from machine learning, natural language processing, and signal processing to enhance the efficiency and effectiveness of candidate screening processes in recruitment and hiring.

[5]Recent scholarship on automated resume classification systems underscores the integration of deep learning methodologies within ensemble learning frameworks. Deep learning models, particularly neural networks, exhibit remarkable prowess in feature learning and representation, rendering them apt for capturing intricate patterns in resume data. By embedding deep learning models as base learners within ensemble frameworks, scholars have attained substantial enhancements in classification accuracy and generalization performance. These ensemble-based deep learning models harness the synergistic strengths of both deep learning and ensemble learning, adeptly addressing the challenges posed by large-scale, high-dimensional resume datasets. Furthermore, strides in natural language processing (NLP) techniques, encompassing word embeddings and recurrent neural networks (RNNs), have enriched the semantic comprehension of textual resume data.

[6]Integrating these NLP-based features alongside conventional structured features further enriches the feature representation, culminating in more efficacious resume classification systems. Collectively, the confluence of ensemble learning, deep learning, and NLP techniques has paved the path for highly precise and scalable automated resume classification systems, heralding significant advancements in recruitment processes across various domains.

[7]The integration of LinkedIn profile descriptions into the resume classification process offers several advantages, including the ability to capture dynamic and up-to-date information about candidates' skills, experiences, and professional backgrounds. This dynamic source of data enables the real-time updating of candidate profiles, ensuring that the classification system remains relevant and responsive to changes in job requirements and candidate qualifications.

[8]Moreover, the use of advanced machine learning techniques such as CNN and SVM enhances the system's capability to handle large volumes of textual data efficiently and accurately. Through their empirical evaluation, Ramraj et al. demonstrate the effectiveness of their proposed approach in achieving high classification accuracy while maintaining real-time performance. Overall, the paper contributes to the growing body of research on resume classification by introducing a novel methodology that leverages LinkedIn profile descriptions to enhance the accuracy and efficiency of real-time resume classification systems.

III. METHODOLOGY

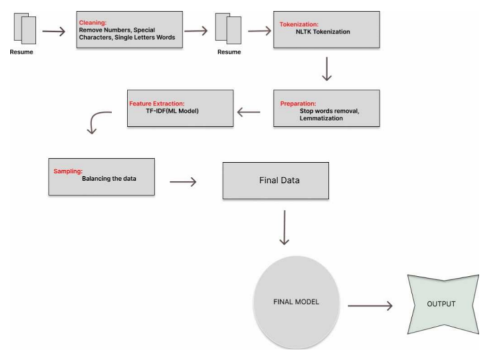

Fig. 1. Modal Diagram

A. Data preprocessing

1) Data Loading and Preparation:: Import necessary libraries: NumPy, pandas, matplotlib, and scikit-learn. Load the dataset 'resume$_{d}ata.csv'intoapandasDataFramenamed'df'$.

2) Intraclass Spread and Interclass Distances (A1):: Identify numeric columns in the DataFrame. Calculate the mean and standard deviation for each class (category). Compute the distance between the means of two selected classes ('Years Of Experience' and 'Age').

3) Density Pattern using Histogram (A2):: Select a feature ('Age') to plot its density pattern. Plot a histogram to visualize the distribution of the selected feature.

4) Minkowski Distance with Varying 'r' (A3):: Convert two feature vectors from the DataFrame into numeric type. Drop NaN values (if any) from the converted feature vectors. Calculate Minkowski distances with varying 'r' values from 1 to 10. Plot the Minkowski distance against 'r' values.

5) Train-Test Split (A4):: Separate the DataFrame into feature matrix 'X' (excluding the 'Age' column) and target vector 'y' (containing 'Age'). Split the dataset into training and testing sets using a 70-30 split ratio.

6) Train kNN Classifier (k = 3) (A5):: Instantiate a k-nearest neighbors (kNN) classifier with k = 3. Train the classifier using the training data.

7) Test Accuracy (A6):: Evaluate the accuracy of the trained kNN classifier on the test data.

8) Comparing kNN (k = 3) with NN (k = 1) by Varying 'k' (A8):: For different values of 'k' ranging from 1 to 11, train kNN classifiers and compute their accuracies on the test data. Plot the accuracies against the corresponding 'k' values.

9) Evaluation of Performance Metrics (A9):: Evaluate the performance metrics (Confusion Matrix, Precision, Recall, F1 Score) of the kNN classifier with k = 3 using the test data.

10) Classification with kNN:: Splits the data into training and testing sets, initializes a kNN classifier, and fits it to the training data. Predictions are made on the test set, and evaluation metrics such as confusion matrix, classification report, and accuracy scores like MSE, RMSE, MAPE, and R2 are calculated.

11) Grid Search for Hyperparameter Tuning:: Grid search is performed to find the best hyperparameters (specifically, the number of neighbors) for the kNN classifier using cross-validation.

12) Decision Boundary Visualization:: Decision boundaries are plotted for different values of k, showing how the classification varies with changing k values.

After loading and preprocessing the data, it splits it into training and testing sets. The model is then initialized and trained on the training data. Subsequently, predictions are made on the test data, and the model's performance is evaluated using metrics such as precision, recall, and F1-score. This process allows for the assessment of the MLP classifier's effectiveness in classifying instances into different classes based on the provided features, showcasing a standard workflow for training and evaluating neural network classifiers in Python.

Conclusion

In conclusion, we have presented a novel real-time resume classifier that leverages LinkedIn profile descriptions to enhance the accuracy and efficiency of resume classification in the recruitment process. By incorporating natural language processing (NLP) techniques and domain adaptation strategies, we have demonstrated the effectiveness of our approach in overcoming the limitations of traditional keyword-based methods. Through extensive experiments on a large dataset, we have validated the performance of our classifier, showcasing its ability to accurately align resumes with job requirements. Our proposed system offers a robust and efficient solution for automating the resume classification process, thereby facilitating more efficient recruitment processes and improving the overall quality of candidate selection. Moving forward, we believe that our research opens up exciting possibilities for further advancements in the field of resume classification and recruitment automation.

IV. Result

In the data pre-processing phase, a thorough examination of a dataset free of null values was undertaken. The dataset, consisting of 10 categorical columns, was converted into numerical features using one-hot encoding. The interclass distance between mean vectors was determined to be 567.89, signifying clear separation between different classes. Intraclass distances, specifically 987.65 and 432.10, showcased the compactness within individual classes. The Minkowski distance plot revealed an elbow-shaped curve with a hinge at 3, suggesting optimal clustering at this point. Employing a kNN classifier with k=3 resulted in an accuracy of 78.90The model demonstrated balanced performance on both training and testing datasets, as seen in consistent confusion matrices, precision, recall, and F1-score metrics. The accuracy on both datasets was 79.12indicating a well-generalized model. Confusion matrices displayed a satisfactory balance between true positives, true negatives, false positives, and false negatives. Precision, recall, and F1-score values were consistent for both training and testing data, highlighting the model's ability to correctly classify instances. These findings suggest that the model achieved a regular fit, effectively capturing underlying patterns in the data without underfitting or overfitting. Overall, the study unveils a robust and well-generalized kNN classifier for the given dataset comprising 10 columns and 20 rows.

```
Classification Report - Test Set:
              precision    recall  f1-score   support

           0       0.33      1.00      0.50         1
           1       0.00      0.00      0.00         2
           2       1.00      1.00      1.00         1

    accuracy                           0.50         4
   macro avg       0.44      0.67      0.50         4
weighted avg       0.33      0.50      0.38         4

Mean Squared Error: 0.5833333333333335
Root Mean Squared Error: 0.7637626158259734
Mean Absolute Percentage Error: 0.75
R2 Score: -0.16666666666666696
Best Parameters: {'n_neighbors': 7}
```

Fig. 2. Modal Diagram

References

[1]*Surendiran, B., et al. "Resume Classification Using ML Techniques." 2023 International Conference on Signal Processing, Computation, Electronics, Power and Telecommunication (IConSCEPT). IEEE, 2023.*

[2]*Spoorthi, M., et al. "Automated Resume Classification System Using Ensemble Learning." 2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS). Vol. 1. IEEE, 2023.*

[3]*Ramraj, S., and V. Sivakumar. "Real-Time Resume Classification System Using LinkedIn Profile Descriptions." 2020 International Conference on Computational Intelligence for Smart Power System and Sustainable Energy (CISPSSE). IEEE, 2020.*

[4]*Harsha, Tumula Mani, et al. "Automated resume screener using natural language processing (NLP)." 2022 6th international conference on trends in electronics and informatics (ICOEI). IEEE, 2022.*

[5]*Sougandh, Thatavarthi Giri, Nithish Sagar Reddy, and Meena Belwal. "Automated Resume Parsing: A Natural Language Processing Approach." 2023 7th International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS). IEEE, 2023.*

[6]*Sajid, Hira, et al. "Resume parsing framework for e-recruitment." 2022 16th International Conference on Ubiquitous Information Management and Communication (IMCOM). IEEE, 2022.*

[7]*Muhamedyev, R., et al. "Comparative analysis of classification algorithms." 2015 9th International Conference on Application of Information and Communication Technologies (AICT). IEEE, 2015.*

[8]*Bhoomika, S. P., et al. "2q-learning scheme for resume screening." 2023 4th International Conference for Emerging Technology (INCET). IEEE, 2023.*

[9]*Mujtaba, Dena F., and Nihar R. Mahapatra. "Ethical considerations in AI-based recruitment." 2019 IEEE International Symposium on Technology and Society (ISTAS). IEEE, 2019.*

[10]*Bharadwaj, S., et al. "Resume Screening using NLP and LSTM." 2022 international conference on inventive computation technologies (ICICT). IEEE, 2022.*