IntelliResume Classification and Job Recommendation System

Team Members:

Qin Jiayu	：	A0296744M
Zhu Yinge	：	A0295228W
Cheng Siyuan	：	A0287262X
Wang Xiang	：	A0298765A

1. System Introduction

This project aims to develop an AI-based system that aims to simplify the job search process and accurately classify suitable positions by deeply analyzing user resumes using machine learning techniques. The system can not only automatically extract personal information from resumes, such as name, email address, and contact information, but also identify the user's core skills and work experience to better match suitable positions.

In the front-end part, the system is developed using the React framework to provide an intuitive and smooth user experience, allowing users to easily upload resumes, view job recommendations, and get feedback. At the same time, the backend uses the Flask framework, which makes data processing and communication with classification models more efficient and flexible. Flask can easily integrate a variety of data processing tools, thereby ensuring the stability and efficiency of the system when processing large amounts of data.

In the process of matching user skills with job requirements, this system uses the KNN (K Nearest Neighbor) algorithm. The KNN algorithm, due to its superior performance in pattern recognition tasks, can quickly and accurately match users' skills and experience with the similarity between job descriptions. Specifically, the system extracts feature from the skills and experience in the user's resume and generates feature vectors, which are then compared with various positions in the job database using the KNN algorithm to identify the job roles that best match the user's background. This method ensures that job seekers can get job recommendations that match their abilities, increasing their job search success rate.

2. Business Case

The job classification system provides an innovative and effective solution to the key challenges faced by the job search and recruitment industry. Using artificial intelligence technology based on the KNN (K nearest neighbor) algorithm, the system can not only efficiently analyze the resume information of job seekers, but also automatically classify and recommend the most suitable positions based on the user's skills and experience, thereby realizing a personalized job search experience. This intelligent recommendation mechanism effectively alleviates the frustration encountered by job seekers in finding suitable positions, and also improves the efficiency of recruiters in screening suitable candidates.

In addition, the system has a clear revenue model, including multiple sources of revenue through advanced job recommendation services, data analysis reports, and recruitment advertising. With the growing market demand for personalized recruitment solutions—estimated to reach **$33 billion by 2027**, growing at a CAGR of **9.5%** from 2020 to 2027—this project has shown strong market potential, strong adaptability, and broad application prospects.

In the initial stage, the required development and marketing investment will lay a solid foundation for the subsequent expansion of the project. By optimizing resource allocation and improving user satisfaction, the scale of the system will gradually increase, thereby achieving improved resource utilization efficiency. Forecasts show that the project will bring considerable returns to investors and stand out in the highly competitive recruitment market.

In summary, this job classification system is not only a technology-driven solution, but also an investment opportunity with significant commercial potential. It can provide two-way value to job seekers and recruiters and promote the intelligent development of the entire industry.

3. Market Research

With the rapid development of artificial intelligence (AI) and machine learning technologies, the recruitment and job search industry are undergoing unprecedented changes. This change is not only a technological upgrade, but also a profound evolution of the industry model. At present, the continuous growth in the number of job seekers and the urgent need to improve the efficiency of the recruitment process have prompted the market to gradually transform to AI-driven tools. These tools can automate tedious tasks such as resume screening, job matching and candidate ranking, thereby significantly improving recruitment efficiency and improving the job search experience.

Industry Status Analysis

In the traditional recruitment model, job seekers often face the problem of information overload. Take LinkedIn and Indeed as examples. Although these platforms provide a large amount of job information, job seekers often feel confused and frustrated when looking for positions that match their qualifications and career goals. According to market research data, nearly 70% of job seekers said it is difficult to find a job that suits them among the massive number of positions. This situation not only reduces the enthusiasm of job seekers, but also makes recruitment agencies face huge challenges in screening suitable candidates.

In response to this market demand, our resume classification system focuses on solving the core problems faced by job seekers and recruitment agencies. The specific analysis is as follows:

Target user group

(1) Job seekers

Demographic characteristics: Job seekers are mainly professionals aged between 22 and 45 with diverse educational backgrounds, including holders of a bachelor's degree or above. They are not only concerned about career advancement, but also actively seek entry-level positions and opportunities for cross-industry transformation.

Behavioral characteristics: Job seekers often frequently use major recruitment platforms, especially LinkedIn, Glassdoor and Indeed. They tend to seek systems that can provide personalized recommendations so that they can find jobs that match their skills more efficiently.

Pain points: Job seekers often get lost in a large amount of job information and find it difficult to find jobs that truly match their qualifications, which directly affects their job search efficiency and satisfaction.

(2) Recruitment agencies and human resources departments

Demographic characteristics: Mainly concentrated in medium and large enterprises, these enterprises attach importance to the optimization and efficiency improvement of the recruitment process.

Behavioral characteristics: Recruitment agencies and HR departments are looking for tools that can reduce manual operations and improve candidate matching efficiency. They hope to use AI technology to improve the candidate screening process and job fit analysis.

Pain points: Faced with many resumes, recruitment agencies often find it difficult to quickly identify the best candidates. At this time, there is an urgent need for a tool that can automate screening and matching to improve recruitment efficiency and accuracy.

Competitiveness and Advantages

In such a market context, our resume classification system has demonstrated strong competitiveness and market value. First, the system uses advanced machine learning technology to achieve accurate resume parsing and intelligent matching, which greatly improves the user experience of job seekers and recruitment agencies. For example, by applying natural language processing (NLP) technology, we can extract key skills, work experience and other information from resumes and match them with the requirements of recruitment positions. This not only saves job seekers' time, but also improves the screening efficiency of recruiters.

Second, our system has personalized recommendation functions. Based on the user's historical behavior and preferences, the system can intelligently recommend positions that are highly matched with the user's qualifications. This personalized service significantly improves the success rate of job hunting, thereby enhancing users' trust and reliance on the system.

Finally, we plan to establish strategic partnerships with multiple recruitment platforms to further expand market share and enhance brand awareness. This strategy will provide strong support for the promotion of our system and create more value for users.

4. System Design

Module Design

1. Data Collection and Preprocessing Module

(1) Collection and Unification of Multi-Source Heterogeneous Data

The data collection and preprocessing module is the foundation of the resume analysis system. It is responsible for integrating resume data from multiple sources, including PDF, DOCX and TXT files, as well as information from online platforms such as LinkedIn. To ensure the consistency of data in different formats, the system adopts an ontology-based semantic unification model. The model defines the ontology structure of key information in resumes, such as education background, work experience and skills, to ensure that all data can be processed in a standardized manner. This process effectively avoids information loss due to inconsistent data formats. For example, the ontology model can identify and unify education information in different regions to maintain consistency in subsequent processing.

(2) Multi-stage processing of text extraction and format optimization

The text extraction step in the module adopts a two-layer reasoning architecture. The first layer uses rule reasoning technology to process resume text in standardized format to ensure accurate extraction of basic information; the second layer uses contextual semantic reasoning to identify potential semantic errors, especially when parsing non-standardized fields. The system can automatically identify and convert education backgrounds in different regions for subsequent analysis. For example, when processing non-standardized education information in certain regions, the system can map it to a unified education standard to maintain the validity of the data.

(3) Data cleaning and intelligent repair

During the **data cleaning** process, the system uses **NLTK** or **spaCy** libraries to remove stop words and unnecessary punctuation, ensuring the simplicity and consistency of the input data. By eliminating non-essential information in the resume text, such as common stop words (e.g., "a", "the", "and") and punctuation, the system effectively retains important information within resumes, enhancing the accuracy of subsequent feature extraction and data consistency.

2. Information Extraction and Resume Classification Module

(1) Feature Vector Construction and Accurate Classification

Information extraction and resume classification are the core functions of the system. In this module, the system first uses natural language processing technology to preprocess the resume text and generate feature vectors based on the TF-IDF (term frequency-inverse document frequency) method. After word segmentation, stop word removal and part-of-speech tagging, the system can extract key information and calculate the TF-IDF value of each word to generate a feature vector set. In addition, the system applies context-aware feature extraction to avoid information loss caused by semantic confusion. For example, when dealing with "data analyst" and "data scientist", the system can accurately distinguish the meanings of the two in different contexts.

(2) Training steps and feature extraction details

During the model training stage, the system divides the resume data into training and test sets, utilizing the encoded category labels for supervised learning. To improve the KNN model's performance, the system uses grid search to optimize hyperparameters, specifically fine-tuning the K value to ensure optimal accuracy and generalization. For feature extraction, the system exclusively uses TF-IDF to identify important terms and skills within the resume text, creating a feature representation that emphasizes keywords most relevant to each job category. This straightforward approach captures the essential information needed for classification without the use of word embeddings or dimensionality reduction. This TF-IDF-based representation is then fed into the KNN model, enabling it to categorize resumes accurately and efficiently. By relying solely on TF-IDF for feature extraction

(3) KNN classification and adaptive model update

After generating feature vectors, the system uses the KNN (K nearest neighbor) classification model to preliminarily classify the resume of job seekers. The KNN model quickly identifies the job category that best matches the resume content by calculating the similarity between the resume feature vector and the feature vector of each position. For example, for data science positions, the model prioritizes resumes that include Python and R skills. The classifier's adaptive learning ability enables it to continuously update model parameters based on historical resume data and classification feedback. The inference engine optimizes classification rules and model parameters by analyzing historical classification results and real-time feedback to improve classification accuracy.

3. Position matching and scoring module

(1) Similarity calculation based on multi-layer reasoning

The core of the position matching and scoring module is to calculate the matching degree between the job seeker's resume and different positions. To achieve this function, the system combines two indicators, cosine similarity and semantic similarity, to ensure that the resume matches the job requirements at the literal level and deep semantic association. For example, the system can identify the similarity between "back-end development engineer" and "software engineer", although they have differences in description and requirements.

(2) Dynamic feature weight adjustment

During the matching process, the reasoning engine can dynamically adjust the weights of different features. For technical positions, the system pays more attention to the job seeker's skills and project experience; for management positions, it emphasizes leadership and communication skills. The reasoning engine automatically updates the matching criteria based on historical recruitment data and changes in job requirements to generate specific matching scores. For example, the match degree of the job seeker's resume with the "back-end development" position is 97%, and the match degree with the "front-end development" position is 93%.

(3) Scoring and feedback mechanism

The scoring module combines historical job recruitment data and feedback, and dynamically adjusts the scoring criteria through the reasoning engine. The system analyzes the success rate of job seekers' previous resume submissions, the evaluation of recruiters, and changes in industry needs, and automatically optimizes the matching criteria between resumes and positions. For example, if the matching score of a position is significantly lower than the industry average, the inference engine will analyze the feature weight distribution behind the score and adjust the scoring strategy for the position to ensure the rationality of the score.

4. Recommendation and feedback module

(1) Intelligent job recommendation and adaptive strategy

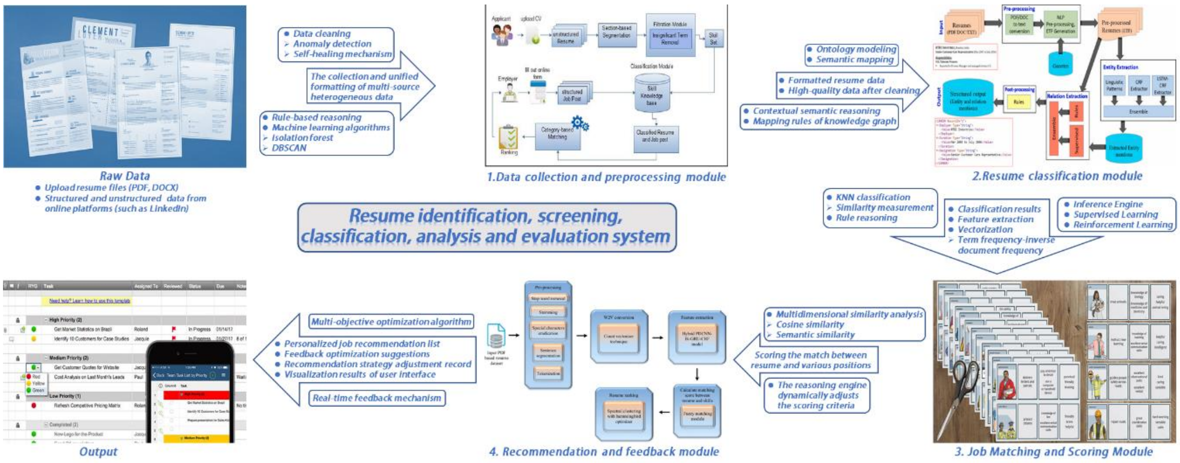
The recommendation and feedback module generates a personalized job recommendation list based on the results of the matching and scoring module. The recommendation engine combines historical data, job requirements and job seekers' intentions to design a multi-objective optimization algorithm to ensure that the recommended positions not only meet the job seekers' skill background, but also meet their career development intentions. For example, the system can recommend similar positions that meet the job seekers' career development direction based on the job seekers' historical submission records and feedback, and even provide more challenging or higher-paying related positions.

(2) Feedback-based recommendation optimization

Through the real-time feedback mechanism, the system continuously optimizes the recommendation strategy. When it is detected that a job recommendation is ignored or rejected by the job seeker, the reasoning engine analyzes the reasons and automatically adjusts the recommendation algorithm. For example, if the job seeker ignores low-matching positions many times, the system will reduce the frequency of recommending similar positions and give priority to highly matching positions. This adaptive recommendation system ensures that job seekers receive the best and accurate job recommendations.

(3) Interaction design and user experience optimization

The system's recommendation results are presented through a visual interface. Combined with the real-time feedback of the reasoning engine, users can make decisions based on the matching degree, job details, company evaluation and other information of the recommended positions. In addition, the feedback provided by the system is not limited to the job seeker experience, but also includes feedback data from recruiters (such as resume quality and matching accuracy) to further optimize the system and improve industry adaptability and user-friendliness.


Raw Data
Upload resume files (PDF, DOCX)
Structured and unstructured data from online platforms (such as LinkedIn)
Data cleaning
Anomaly detection
Self-healing mechanism
The collection and unified formatting of multi-source heterogeneous data
Rule-based reasoning
Machine learning algorithms
Isolation forest
DBSCAN
Applicant
upload CV
unstructured Resume
Section-based Segmentation
Filtration Module
Insignificant Term Removal
Skill Set
Employer
fill out online form
structured Job Post
Classification Module
Skill Knowledge base
Category-based Matching
Classified Resume and Job post
Ranking
1.Data collection and preprocessing module
Ontology modeling
Semantic mapping
Formatted resume data
High-quality data after cleaning
Contextual semantic reasoning
Mapping rules of knowledge graph
Input
Resumes (PDF DOC TXT)
Pre-processing
PDF/DOC to text conversion
NLP Pre-processing, ETF Generation
Pre-processed Resumes (ETF)
Gazetteer
Entity Extraction
Linguistic Patterns
CRF Extractor
LSTM-CRF Extractor
Ensemble
Extracted Entity mentions
Relation Extraction
Rules
Supervised
Ensemble
Post-processing
Rules
Output
Structured output (Entity and relation mentions)
2.Resume classification module
Resume identification, screening, classification, analysis and evaluation system
KNN classification
Similarity measurement
Rule reasoning
Classification results
Feature extraction
Vectorization
Term frequency-inverse document frequency
Inference Engine
Supervised Learning
Reinforcement Learning
Multi-objective optimization algorithm
Personalized job recommendation list
Feedback optimization suggestions
Recommendation strategy adjustment record
Visualization results of user interface
Real-time feedback mechanism
Input PDF based resume dataset
Pre-processing
Stop word removal
Stemming
Special characters eradication
Sentence segmentation
Tokenization
W2V conversion
Count vectorizer technique
Feature extraction
Hybrid PDCNN-Bi-GRU-CSF model
Calculate matching score between resume and skills
Fuzzy matching module
Resume ranking
Spectral clustering with hummingbird optimizer
Multidimensional similarity analysis
Cosine similarity
Semantic similarity
Scoring the match between resume and various positions
The reasoning engine dynamically adjusts the scoring criteria
Output
4. Recommendation and feedback module
3. Job Matching and Scoring Module

Resume identification, screening, classification, analysis and evaluation system module architecture diagram

Model training

1. Dataset expansion and cleaning

When building the dataset for the resume analysis system, we first collected resume data from multiple channels, including but not limited to social media (such as LinkedIn), recruitment websites, and company portals. By introducing key fields (such as name, gender, phone number, and email), the data structure is designed to be closer to the diversity of real resumes to improve the practicality and adaptability of the model. To ensure data quality, NLTK (Natural Language Toolkit) is used for multi-level data cleaning. The specific steps include removing stop words, punctuation marks, and extra spaces, and unifying the text format to increase the key information density of the text. The core of this process is to ensure the accuracy of feature extraction so that subsequent model training can be based on clear and standardized data.

2. Automatic extraction of key information

In the information extraction stage, the system uses regular expressions combined with the NLTK name library to automatically identify key information. By designing efficient regular expressions, standard format data such as emails and phone numbers can be quickly extracted. In addition, with the help of NLTK's named entity recognition capabilities, the system can automatically identify candidate names. This intelligent automated extraction technology not only improves the efficiency of data preprocessing, but also ensures that the subsequent classification and scoring modules can rely on accurate information foundation.

3. Feature Engineering and Keyword Extraction

Feature engineering is a key link to improve model performance. At this stage, the system performs numerical encoding (Label Encoding) on the job category labels to ensure that the model can understand the classification information. Feature extraction uses the TF-IDF method to identify important skills and technical keywords by analyzing the word frequency and inverse document frequency in the resume text, highlighting the ability and technical advantages of job seekers. In this way, the core content of the resume can be accurately extracted, thus providing strong support for subsequent classification and scoring.

4. Dataset Division and Model Training

After completing feature extraction, the dataset will be divided into training set and test set in a ratio of 8:2 to ensure the learning mode and generalization ability of the model. Next, the system uses a variety of machine learning models for training, including K-nearest neighbor algorithm (KNN), logistic regression, random forest, and support vector machine (SVM). Each model was subjected to detailed hyperparameter optimization and performance comparison during the training phase to select the best model for practical application.

5. Parameter optimization and cross-validation

To improve the robustness and prediction accuracy of the model, the k-fold cross-validation technique was used to tune the model parameters. For different models, different hyperparameters (such as K value of KNN, number of trees of random forest, kernel function type of SVM, etc.) were systematically tested to ensure the selection of the best parameter combination. This process is achieved by evaluating the performance of the model on different data subsets to ensure the stability and applicability of the final model, especially when facing complex and diverse resume data.

Examples in practical applications

1. Dataset cleaning: For example, when processing resumes, the system finds that a resume contains the field "email: example@domain.com". The regular expression quickly extracts the email information and considers it as an important feature in subsequent classification and scoring.

2. Feature extraction: When analyzing the skills of job seekers, if the resume mentions "familiar with Python, Java and machine learning", the system will use the TF-IDF algorithm to identify "Python" and "machine learning" as features with high importance. These features will be given higher weights in subsequent job matching and scoring.

3. Model training and optimization: When training the KNN model, assuming that the job seeker's resume scores low in a group of positions (such as 65% match), after cross-validation, it is found that when the K value is adjusted to 5, the prediction accuracy of the model is increased to 85%. This shows that the model's matching ability for specific occupations (such as data analysts) has been significantly improved.

5. System Development & Implementation

Front-end design

(1) Project Structure

The overall structure of the project is centered around the src folder, which is used to store the front-end source code. This folder contains two core components and related CSS rendering files:

ResumeUpload component: This component focuses on the upload function of user resumes, supports drag-and-drop upload and file type verification. In the specific implementation, the HTML5 File API is used to check the file format uploaded by the user and provide real-time user feedback to ensure that the user uploads a supported file type (such as PDF and Word). This component also uses React's state management to dynamically update the upload status through state variables to improve the user experience.

ClassificationResults component: This component is responsible for displaying classification results. Its design uses a responsive layout to ensure good display on different devices. By classifying resume information by position, users can easily browse and compare candidate information. To improve readability, the interface uses folding and expanding functions, and users can choose to view more detailed information according to their needs.

(2) Data interaction

The interaction between the front-end and the backend is implemented through Axios, using the RESTful API style. After the user uploads the resume, the system will initiate a POST request to /api/bulk_upload to send the resume data to the server. After the request is successful, the user will automatically jump to the ClassificationResults page to display the uploaded classification information. During this process, the front-end uses asynchronous requests to ensure that the page will not be blocked due to data loading, thereby improving the user experience.

(3) Style and user experience

In terms of front-end style, CSS modular management is used to decouple the style sheet from the component and enhance maintainability. To ensure the beauty and ease of use of the interface, modern UI elements and color schemes are used in the design to ensure the visual comfort of the user during operation. At the same time, the interface interactivity is improved by introducing animation effects. For example, when the file is successfully uploaded, the button will have a slight color change and prompt information will be displayed, further enhancing the user experience.

(4) State management

In the project, state management uses React's useState and useEffect hook functions to ensure that components can automatically re-render when data changes. In addition, the Context API is used to provide global state management, which simplifies data transfer between multiple components and enhances the clarity and structure of the code.

(5) Practical application and market value

The front-end design of this project is closely centered on resume classification, scoring and feedback functions, and is committed to providing users with accurate and efficient job search services. With the increasing application of artificial intelligence technology in the recruitment industry, the market demand for intelligent resume classification systems is also increasing. Through real-time feedback and personalized recommendations, the system can effectively solve the problem that job seekers have difficulty finding suitable positions in a large amount of recruitment information.

For example, after a user uploads a resume, the system can give job recommendations that match their skills and experience in a short period of time, greatly improving job search efficiency. In addition, for recruitment agencies, the system can automatically screen resumes, reduce the time cost of manual review, and thus improve recruitment efficiency. This two-way value creation demonstrates the project's potential for wide application in the actual market.

Backend design

The backend design uses the Flask framework and Python programming language to provide efficient and stable services and ensure smooth communication between the frontend and the machine learning module. The system's architecture focuses on modularity, making each function independent and collaborative, thereby improving the maintainability and scalability of the code.

(1) Multi-file upload and format processing

The backend application has powerful multi-file upload capabilities and supports PDF and DOCX formats. After receiving the POST request from the frontend, the system first verifies the uploaded file type to ensure that it meets the predetermined standards. Subsequently, the uploaded file is converted into a text list by using Python's pdfminer and python-docx libraries. This conversion process ensures the accuracy of the text and lays the foundation for subsequent processing.

(2) Personal information extraction

After the conversion is completed, the backend uses the NLTK library to extract personal information. This process includes but is not limited to extracting the employee's name, email, gender, and phone number. To ensure the accuracy of the extracted information, the system uses predefined regular expressions and entity recognition algorithms to identify key information in multiple formats. For example, when processing email addresses, the system uses specific regular expressions to accurately match standard email formats.

(3) Keyword extraction

In addition to personal information extraction, the backend also implements keyword extraction. This function uses TF-IDF vectorization technology to efficiently identify important keywords in the text. The specific steps are as follows:

Stop word processing: First, the system uses the stop word list of the NLTK library to filter out insignificant words.

TF-IDF calculation: Then, based on the text data and the preset keyword set, the system calculates the TF-IDF matrix and returns the keywords that match the resume content and their relevance scores. This process provides the necessary basic data support for resume classification.

(4) Job classification

After completing the information extraction, the system will classify the positions. The text content is first cleaned to remove all special characters and extra spaces to ensure the neatness of the input data. Subsequently, the backend loads the optimized KNN model from the pickle file. The model is fully trained to accurately analyze the user's skills and experience and recommend the most suitable positions for them. By introducing the KNN algorithm, the system can perform real-time matching based on historical data, improving the accuracy of classification.

(5) Result Feedback

Finally, all the extracted and classified information will be encapsulated in JSON format and returned to the front-end application for display. This process ensures the efficiency and reliability of data transmission, allowing users to quickly obtain job information and rating feedback related to their resumes. For example, after uploading a resume, users can immediately see the positions that best match their skills and their corresponding ratings, which undoubtedly improves the user experience.

6. Findings and Discussion

(1) System module design

System module design is the foundation of this project, and its goal is to ensure the efficiency of information extraction and classification. The survey results show that the current system module effectively implements the structured processing of resumes, extracting key fields including education background, work experience and skills through natural language processing technology. To further improve the accuracy of the system, it is recommended to introduce more advanced named entity recognition (NER) models in the module to capture more fine-grained information in resumes. For example, by training a customized NER model, diverse professional skills can be better identified and classified, thereby improving the system's adaptability to resumes in different fields.

(2) Model training

In terms of model training, our research shows that the current model is trained on multiple public datasets and achieves good classification performance through cross-validation. The survey results show that the model has an accuracy of more than 95% in identifying industry-related positions and skill matching. However, the model's performance is still insufficient when faced with a small number of samples or resumes in specific industries. Therefore, future improvements include using transfer learning technology to improve the adaptability to small sample datasets with the help of pre-trained models. In addition, data enhancement techniques can also be introduced to expand the diversity of training data and thus improve the generalization ability of the model.

(3) Front-end and back-end development

The survey results of front-end development show that the user interface design is intuitive, and user feedback shows that the overall user experience is good. However, there is still room for improvement in functionality and response speed. In terms of back-end development, the system's response time remains within an acceptable range in most cases, but the system may experience delays in high concurrency situations. To this end, future improvements can focus on optimizing database queries and load balancing strategies to improve the overall performance of the system.

(4) Market relevance

Market research results show that resume classification systems have strong application potential in the current job market. With the rapid development of artificial intelligence technology, companies are increasingly relying on automated resume screening tools in the recruitment process. According to the latest market data, about 70% of companies said they are considering or have implemented AI-based recruitment solutions. This provides an important market opportunity for our system. In addition, the existing resume classification systems on the market mostly focus on a specific field and lack cross-industry adaptability. By continuously optimizing the model and system design, our project will be able to stand out in multi-industry applications.

(5) Future improvement directions

To maintain the competitiveness of the system, future improvement directions should include the following aspects:

Algorithm optimization: Continue to study the latest deep learning algorithms and regularly update the model to ensure performance improvement.

User feedback loop: Establish a user feedback mechanism to collect user opinions in real time during use to continuously optimize user experience and system functions.

Dataset expansion: Expand the diversity of training data sets, introduce resume samples from more industries, and improve the universality and accuracy of the system.

Integrate other technologies: Explore the combination of image processing technology to deeply mine the unstructured information of resumes, such as automatically extracting information from images (such as certificates, awards, etc.).

Conclusion

Through in-depth investigation and analysis of system module design, model training, front-end and back-end development, and market demand, we have confirmed the effectiveness and market potential of the current system in the field of resume classification. Through continuous optimization and technological innovation, we are confident that we will establish a strong competitive advantage in the future job market.

Appendix1

PROJECT PROPOSAL

Date of proposal: 2024.10.10

Project Title: Intelligent resume recognition, screening, classification and analysis and evaluation system

Group ID (As Enrolled in Canvas Class Groups): Group 12

Group Members (name, Student ID):

Qin Jiayu A0296744M

Zhu Yinge A0295228W

Cheng Siyuan A0287262X

Wang Xiang A0298765A

Sponsor/Client: (Company Name, Address and Contact Name, Email, if any)

Background/Aims/Objectives:

In today's competitive job market, the match between job seekers and employers has become increasingly complex. As an important tool for job seekers to demonstrate their abilities, how to effectively classify, screen and analyze resumes has become an urgent problem to be solved. This project aims to develop a resume classification system based on artificial intelligence. Through efficient data processing and deep learning technology, it provides job seekers with intelligent resume analysis and feedback to help them better adapt to market needs and improve the screening efficiency of recruiters.

Goals include:

1. Improve the accuracy of resume classification and scoring, and automatically extract key features through deep learning models.

2. Provide personalized feedback and recommendations for job seekers to enhance their job search competitiveness.

3. Achieve real-time resume analysis to help recruiters quickly screen out candidates who meet job requirements.

Project Descriptions:

1. System Introduction:

This system integrates advanced natural language processing technology and machine learning algorithms to build a model of resume data sets, which can automatically analyze and evaluate resumes. The system aims to improve the automation of the recruitment process, thereby providing efficient and convenient services for job seekers and employers.

2. Business Case:

Most resume screening tools on the market still rely on manual intervention, which makes the screening process time-consuming and error prone. This system reduces the labor costs of recruiters through automated processing, while increasing the interview opportunities of job seekers. According to market research, it is expected that the system will be able to improve recruitment efficiency by more than 30% and improve the matching degree between candidates and positions.

3. Market Research:

According to market research results, about 70% of companies face the problem of low recruitment efficiency. Through the analysis of existing recruitment software, it is found that most tools lack flexible adaptive functions and are difficult to meet diverse recruitment needs. This project provides a more accurate resume analysis solution for this market pain point.

4. System Design:

The system design will be divided into the following four modules:

(1) Data collection and preprocessing module: collect job seekers' resumes and perform data cleaning to ensure data quality and consistency.

(2) Information extraction and resume classification module: Using natural language processing (NLP) toolkits (such as NLTK, spaCy) to parse resume content and extract key fields to ensure the accuracy of the extraction results.

(3) Job matching and scoring module: Match and score job seekers' resumes and job requirements and generate corresponding recommendation results.

(4) Recommendation and feedback module: Provide job seekers with personalized resume modification suggestions and career development recommendations based on system analysis results.

5. System development and implementation:

The system will adopt a front-end and back-end separation architecture. The front-end uses the React framework to provide a user interaction interface, and the backend uses Python and its related libraries (such as Flask) to implement business logic. The system will be deployed in the cloud to ensure data security and system stability. During the implementation process, it is planned to continuously optimize the system functions through multiple rounds of testing and user feedback to ensure that the system achieves the expected results.

Appendix2

Mapped System Functionalities against knowledge, techniques and skills of modular courses: MR, RS, CGS

1. Data collection and preprocessing module

(1) Multi-source data integration: This module uses a semantic ontology model for data integration. Through the knowledge graph method in the reasoning system (MR), it ensures the accurate integration of job seeker information in different file formats (such as PDF, DOCX, TXT) and online platforms (such as LinkedIn). This method makes the data structured, facilitates subsequent processing, and improves data consistency.

(2) Text extraction and cleaning: Using natural language processing (NLP) technology, the system applies regular expressions and named entity recognition (NER) methods in the extraction process to accurately identify and extract key information. This process uses the contextual semantic analysis capabilities of the reasoning system (RS) to ensure the integrity and accuracy of text information and eliminate potential noise.

(3) Anomaly detection and data repair: In the data cleaning stage, outlier detection is performed by using the isolation forest algorithm. This method is based on the ensemble learning idea of MR and can effectively identify and process inconsistent data. At the same time, the intelligent filling algorithm is used to repair missing information, which reflects the effectiveness of the cognitive system (CGS) in improving data quality management.

2. Information Extraction and Resume Classification Module

(1) Feature Vector Generation: The system generates feature vectors of job seekers' resumes by word segmentation, stop word removal, and part-of-speech tagging. In this process, the TF-IDF method is used to evaluate the importance of keywords to ensure that the extracted features can effectively represent the skills and experience of job seekers, demonstrating the application of MR in feature selection.

(2) Supervised Learning and Model Training: The system uses supervised learning algorithms such as support vector machine (SVM) and random forest for model training. During the training process, by using the k-fold cross-validation technique, the system can optimize model parameters and improve classification accuracy, reflecting the role of the reasoning system (RS) in model optimization.

(3) Adaptability of the classification algorithm: During the classification process, the system uses the KNN classification algorithm to match feature vectors based on the similarity between job descriptions. The adaptive learning ability of the KNN algorithm enables it to dynamically adjust the classification criteria based on historical data, reflecting the application of CGS in intelligent classification systems.

3. Job matching and scoring module

(1) Similarity calculation model: This module implements the matching calculation between the job applicant's resume and the job description, using two methods, cosine similarity and

Jaccard similarity, to ensure that the matching is evaluated from both the literal and deep semantics. This dual matching mechanism reflects the effectiveness of MR in handling complex matching tasks.

(2) Dynamic feature weight adjustment: During the matching process, the system reasoning engine dynamically adjusts the feature weight according to the job characteristics. For example, for technical positions, the system pays more attention to the job applicant's technical ability, while for management positions, it pays more attention to their leadership ability. This dynamic adjustment reflects the flexibility of RS in the feature weighting process.

(3) Comprehensive scoring mechanism: This module dynamically adjusts the scoring criteria based on historical recruitment data and user feedback, analyzes the success rate of job applicants' historical resume submissions and recruiters' evaluations, to optimize the matching score and improve the recommendation accuracy. This mechanism reflects the importance of CGS in feedback optimization.

4. Recommendation and feedback module

(1) Personalized recommendation strategy: Based on the matching and scoring results, the system generates a personalized job recommendation list. The recommendation engine combines historical data, job requirements, and job seekers' preferences, and uses a multi-objective optimization algorithm to ensure that the recommended positions are both in line with the job seekers' skill background and promote their career development. This strategy demonstrates the effective application of MR in personalized recommendations.

(2) Feedback-based recommendation optimization: By monitoring job seekers' feedback on job recommendations in real time, the system can automatically adjust the recommendation strategy. For example, the system reduces the frequency of recommending positions that job seekers are not interested in and gives priority to highly matching positions. This mechanism demonstrates the value of CGS in dynamically adjusting the recommendation process.

(3) Visual interface and user interaction: The system displays the recommendation results through a visual interface, and users can make decisions based on information such as matching degree, job details, and company evaluation. In addition, the system also collects feedback data from recruiters to continuously improve the recommendation algorithm, enhance the industry adaptability and user-friendliness of the system, and reflect the application of RS in user interaction.

Appendix3

Installation and User Guide

Requirements:

System overview

This system is designed to simplify job searching for users by analyzing their resumes and classifying suitable job roles using machine learning. Also, the system can extract personal information like names and emails from resumes. The frontend is built using React for a smooth user experience, while the backend uses Flask to handle data processing and communication with the classification model. The KNN algorithm, known for its effectiveness in pattern recognition, is employed to match users' skills and experiences with potential job roles.

Frontend & Backend Applications

Our frontend is developed with the React framework. It uses APIs to communicate with backend which developed with Flask. The resume files will be sent from frontend to backend. And the results will return to the frontend application in JSON format.

Deployment

Our System is deployed in Windows 11. To run the system's backend, you will need to have a working Python installation with the necessary libraries installed:

- Flask
- Flask-Cors
- Pypdf
- Numpy
- Panda
- Nltk
- Scikit-learn
- Python-docx

To install the libraries above, key in the command "pip install <library's name>".

When the first time installing the backend application, you need to download relevant nltk resources:

- names
- words
- stopwords
- punkt_tab
- averaged_perceptron_tagger_eng
- maxent_ne_chunker_tab

To download the resources above, key in the command "python" to open python shell.

In python shell, key in the command “import nltk”.

Then key in the command “nltk.download(<resource’s name>)”

After all resources are downloaded, key in the command “exit()” to exit the Python shell.

To run the backend, simply open a terminal, enter:

- cd <path of the system>/Resume-Classification-System/SystemCode/backend
- python main.py

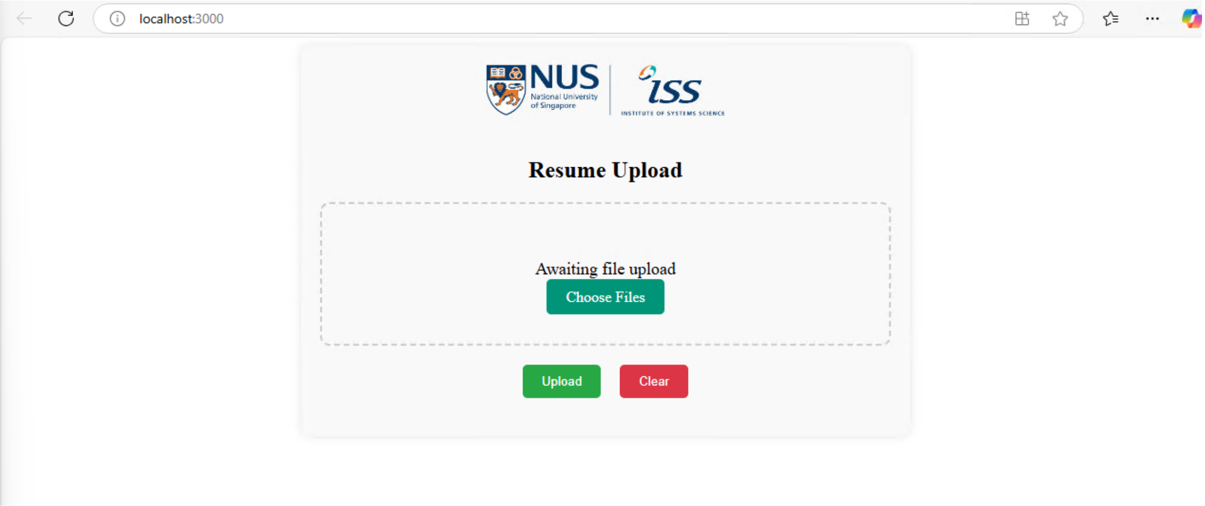
Next, you need to run the frontend application. It needs Node.js installation. After installing Node.js, open a new terminal, enter:

- cd <path of the system>/Resume-Classification-System/SystemCode/frontend
- npm install
- npm start

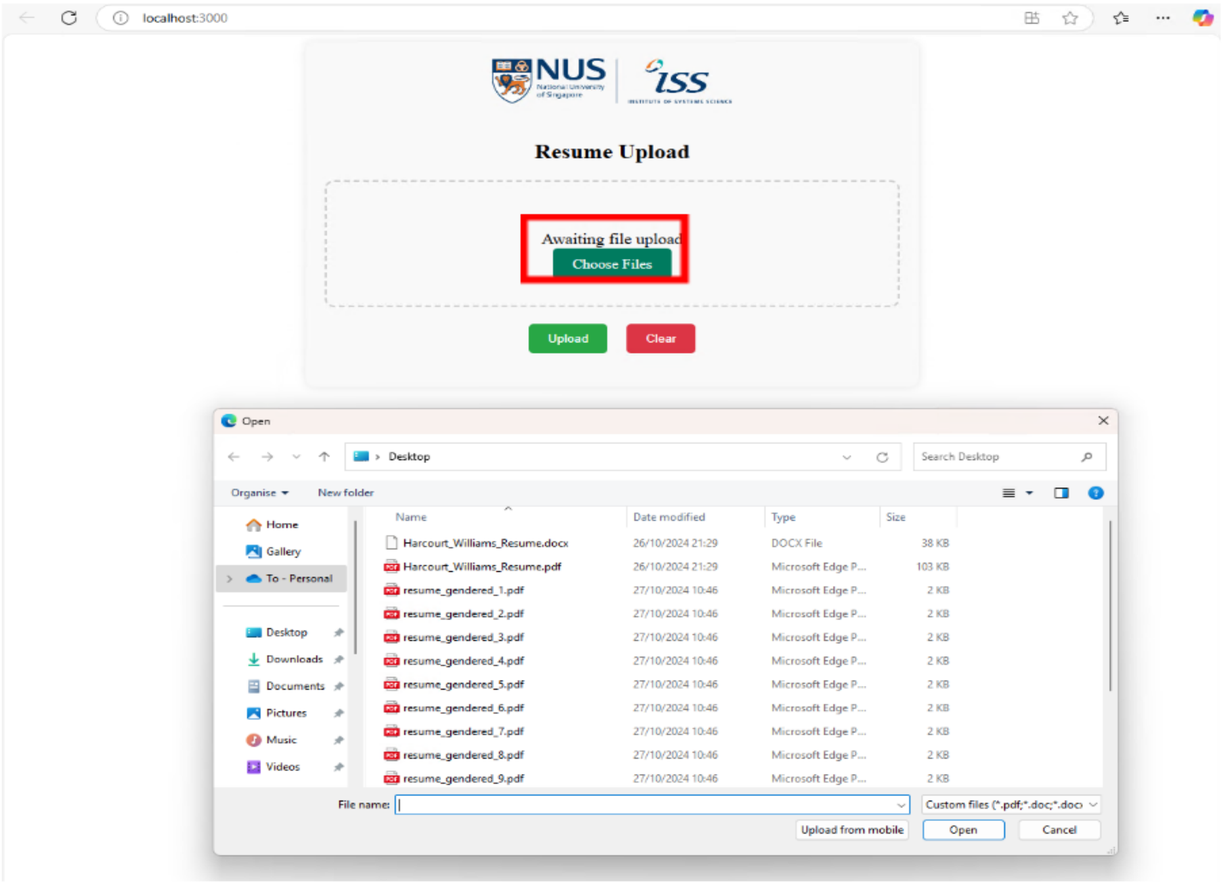
Start

Open your browser and go to http://127.0.0.1:3000/.

This is the home page of the system. It provides functionality of resume uploading. You can simply upload your resume here. Our system supports .pdf and .docx.



You can select one or more files to upload your resume after clicking “Choose Files” Button.

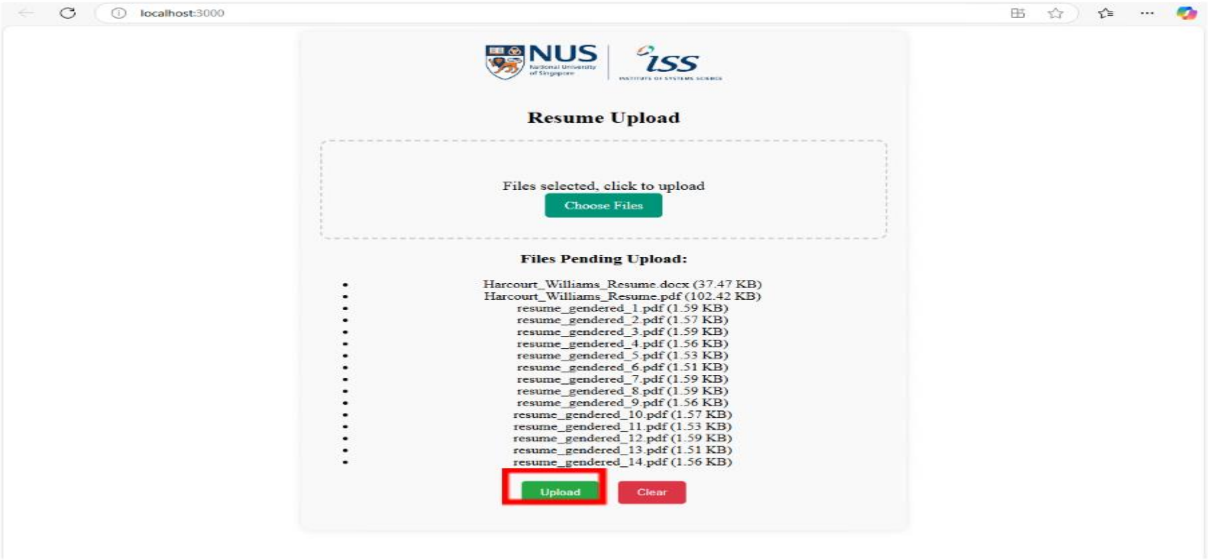


Name	Date modified	Type	Size
Harcourt_Williams_Resume.docx	26/10/2024 21:29	DOCX File	38 KB
Harcourt_Williams_Resume.pdf	26/10/2024 21:29	Microsoft Edge P...	103 KB
resume_gendered_1.pdf	27/10/2024 10:46	Microsoft Edge P...	2 KB
resume_gendered_2.pdf	27/10/2024 10:46	Microsoft Edge P...	2 KB
resume_gendered_3.pdf	27/10/2024 10:46	Microsoft Edge P...	2 KB
resume_gendered_4.pdf	27/10/2024 10:46	Microsoft Edge P...	2 KB
resume_gendered_5.pdf	27/10/2024 10:46	Microsoft Edge P...	2 KB
resume_gendered_6.pdf	27/10/2024 10:46	Microsoft Edge P...	2 KB
resume_gendered_7.pdf	27/10/2024 10:46	Microsoft Edge P...	2 KB
resume_gendered_8.pdf	27/10/2024 10:46	Microsoft Edge P...	2 KB
resume_gendered_9.pdf	27/10/2024 10:46	Microsoft Edge P...	2 KB

File name:

Custom files (*.pdf;*.doc;*.doc

Then you can click "Upload" button to upload chosen resumes. Also, you can click "Clear"button to remove and re-select files.

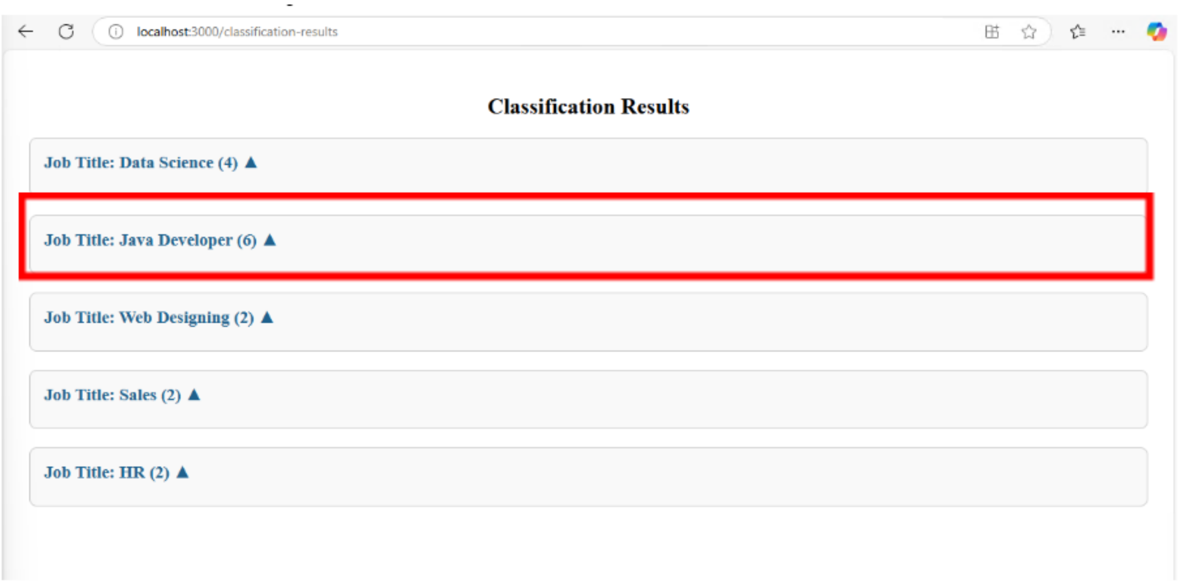


Files selected, click to upload

Files Pending Upload:

- Harcourt_Williams_Resume.docx (37.47 KB)
- Harcourt_Williams_Resume.pdf (102.42 KB)
- resume_gendered_1.pdf (1.59 KB)
- resume_gendered_2.pdf (1.57 KB)
- resume_gendered_3.pdf (1.59 KB)
- resume_gendered_4.pdf (1.56 KB)
- resume_gendered_5.pdf (1.53 KB)
- resume_gendered_6.pdf (1.51 KB)
- resume_gendered_7.pdf (1.59 KB)
- resume_gendered_8.pdf (1.59 KB)
- resume_gendered_9.pdf (1.56 KB)
- resume_gendered_10.pdf (1.57 KB)
- resume_gendered_11.pdf (1.53 KB)
- resume_gendered_12.pdf (1.59 KB)
- resume_gendered_13.pdf (1.51 KB)
- resume_gendered_14.pdf (1.56 KB)

When classification is completed, you will be redirected to the result page. In this page, the classification results are group by the most suitable job titles based on the content of each resume. You can click job title cards to see details.



Classification Results

Job Title: Data Science (4) ▲

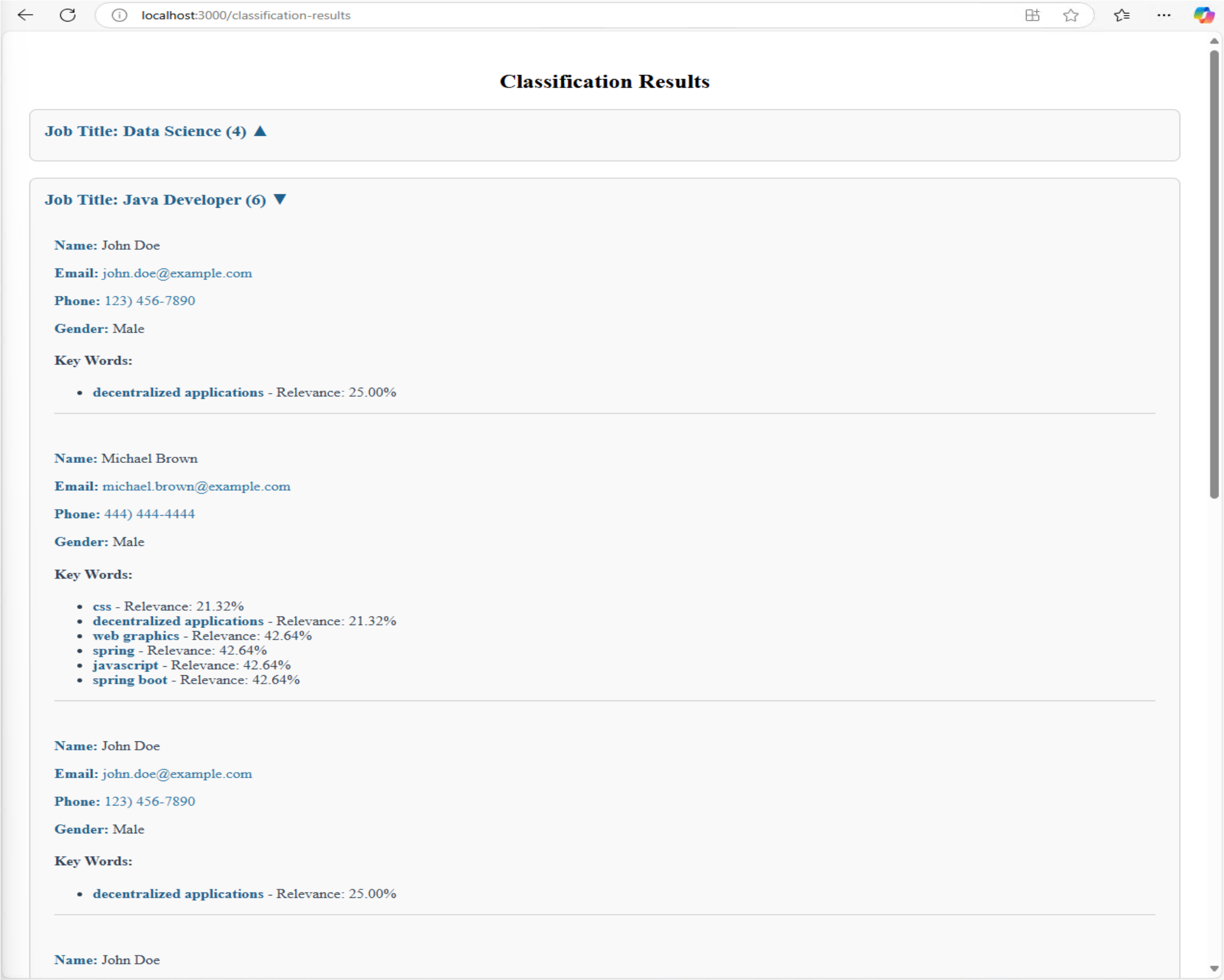
Job Title: Java Developer (6) ▲

Job Title: Web Designing (2) ▲

Job Title: Sales (2) ▲

Job Title: HR (2) ▲

In the expanded job title list, you can view the information extracted from resume files. The information includes name, email, phone number, gender and keywords of each resume.



Classification Results

Job Title: Data Science (4) ▲

Job Title: Java Developer (6) ▼

Name: John Doe

Email: john.doe@example.com

Phone: 123) 456-7890

Gender: Male

Key Words:

- **decentralized applications** - Relevance: 25.00%

Name: Michael Brown

Email: michael.brown@example.com

Phone: 444) 444-4444

Gender: Male

Key Words:

- **css** - Relevance: 21.32%
- **decentralized applications** - Relevance: 21.32%
- **web graphics** - Relevance: 42.64%
- **spring** - Relevance: 42.64%
- **javascript** - Relevance: 42.64%
- **spring boot** - Relevance: 42.64%

Name: John Doe

Email: john.doe@example.com

Phone: 123) 456-7890

Gender: Male

Key Words:

- **decentralized applications** - Relevance: 25.00%

Name: John Doe