

RESUME PARSER

PROJECT BASED LEARNING REPORT

Submitted in partial fulfilment of the requirements for the award of the degree of

Bachelor of Technology

In

COMPUTER SCIENCE AND ENGINEERING

By

BATCH-2B

MUDUNOORI SRI CHAITANYA VARMA	22331A05A5
NEKKALA GOWTHAM	22331A05B5
KOTTANA HEMANTH	22331A0583
NELLURI ENOSH RAJU	22331A05B7

Under the supervision of

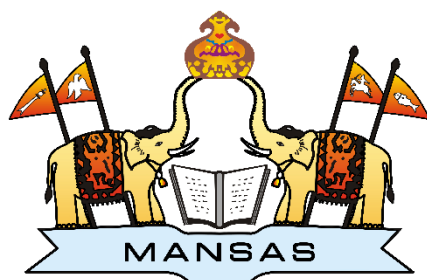
Mrs V. LAVANYA

Assistant professor



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
MVGR COLLEGE OF ENGINEERING (Autonomous)
VIZIANAGARAM-535005, AP (INDIA)
(Accredited by NBA, NAAC, and Permanently Affiliated to Jawaharlal Nehru
Technological University Kakinada)
OCTOBER, 2024

CERTIFICATE



This is to certify that the project report entitled “**RESUME PARSER**” is being submitted by **MUDUNOORI SRI CHAITANYA VARMA, NEKKALA GOWTHAM, KOTTANA HEMANTH, NELLURI ENOSH RAJU** bearing registered numbers **22331A05A5, 22331A05B5, 22331A0583, 22331A05B7** respectively, in partial fulfilment for the award of the degree of “**Bachelor of Technology**” in **Computer Science and Engineering** is a record of bonafide work done by them under my supervision during the academic year 2024-2025.

Dr. T. Pavan Kumar
Head of the Department,
Department of CSE,
MVGR College of Engineering,
Vizianagaram.

Mrs. V . Lavanya
Assistant Professor,
Department of CSE,
MVGR College of Engineering,
Vizianagaram.

ACKNOWLEDGEMENTS

We place on record and warmly acknowledge the continuous encouragement, invaluable supervision, timely suggestions and inspired guidance offered by our guide **Mrs. N. Lavanya**, Asst. Professor, Department of Computer Science and Engineering MVGR College of Engineering in bringing this report to a successful completion.

We consider it our privilege to express our deepest gratitude to **Dr T. Pavan Kumar**, Head of the Department for his valuable suggestions and constant motivation that greatly helped the project work to get successfully completed.

We sincerely thank all the members of the staff in the Department of Computer Science & Engineering for their sustained help in our pursuits. We thank all those who contributed directly or indirectly in successfully carrying out this work.

TABLE OF CONTENTS

S.NO	Contents	Page no
1	Abstract	5
2	Problem Statement	5
3	Introduction	6
4	Objective	6
5	Implementation & Output	7-10
7	References	11
8	Conclusion	11

ABSTRACT

In the era of data-driven decision-making, resume screening has become an essential process for recruiters to efficiently match candidates with job roles. Automated resume evaluation systems aim to streamline the hiring process by identifying relevant skills in candidates' resumes based on the job description. These systems leverage natural language processing (NLP) techniques to extract keywords from resumes and job descriptions, calculate similarity scores based on the overlap of relevant skills, and assess candidates against predefined criteria. Such automation not only saves time but also ensures a fairer and more consistent evaluation process. This system is designed to evaluate resumes using keyword matching and a similarity-based scoring system, with visual tools to assist HR professionals in identifying the best-fit candidates.

PROBLEM STATEMENT

Recruiters and HR professionals face the challenge of manually evaluating a large number of resumes for each job opening. This process is time-consuming, prone to human error, and often inconsistent. Additionally, matching the skills in candidates' resumes with those in job descriptions can be complex due to varying terminologies and formats. Without an automated system, there is a higher risk of overlooking qualified candidates, leading to inefficiencies in the hiring process.

How can we develop an automated resume evaluation system that uses natural language processing (NLP) to accurately match the skills in resumes with the requirements in job descriptions, providing a similarity score and selection status for each candidate, while also offering visual insights into the resume selection process?

INTRODUCTION

In today's competitive job market, hiring the right candidate for a position has become a critical challenge for organisations. With hundreds or even thousands of resumes submitted for a single job opening, recruiters are under constant pressure to efficiently and accurately evaluate candidates. The traditional manual review process can be slow, subjective, and prone to errors, leading to delays in hiring and potentially overlooking qualified candidates. Furthermore, candidates often present their skills in diverse formats, making it difficult to identify and match relevant qualifications with the requirements of a specific role.

To address these challenges, automated resume evaluation systems powered by natural language processing (NLP) are gaining prominence. These systems can streamline the hiring process by extracting key information from resumes, comparing it against job descriptions, and providing recruiters with data-driven insights. By automating the keyword matching process, organisations can save time, reduce biases, and enhance the overall efficiency of resume screening.

This project focuses on developing an NLP-based resume evaluation system that extracts relevant skills from resumes, compares them with job keywords, and computes a similarity score to evaluate candidate suitability. Additionally, the system provides a clear visual representation of the hiring process, showing the selection status of candidates based on their similarity scores. This approach aims to help recruiters make faster, more informed hiring decisions while maintaining fairness and consistency throughout the process.

OBJECTIVES

The primary objective is to develop a system that extracts job-relevant skills from resumes using NLP and compares them against the required skills from the job description. The system generates a similarity score and provides an easy-to-understand status for each resume, indicating whether the candidate meets the criteria for further consideration. The system also offers visual insights into the hiring process by plotting the distribution of selected and non-selected candidates, aiding recruiters in making data-driven decisions.

IMPLEMENTATION

NECESSARY IMPORTS

```
import pandas as pd
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer
import matplotlib.pyplot as plt
import seaborn as sns
from tabulate import tabulate
```

Download NLTK resources

```
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')
```

Preprocess text: Tokenize, remove stopwords, and lemmatize

```
def preprocess_text(text):
    tokens = word_tokenize(text.lower())
    stop_words = set(stopwords.words('english'))
    return [WordNetLemmatizer().lemmatize(word) for word in tokens if
            word.isalnum() and word not in stop_words]
```

Extract relevant skills based on job keywords

```
def extract_skills(resume_text, job_keywords):
    return [word for word in preprocess_text(resume_text) if word in
            job_keywords]
```

Calculate similarity score based on keyword matches

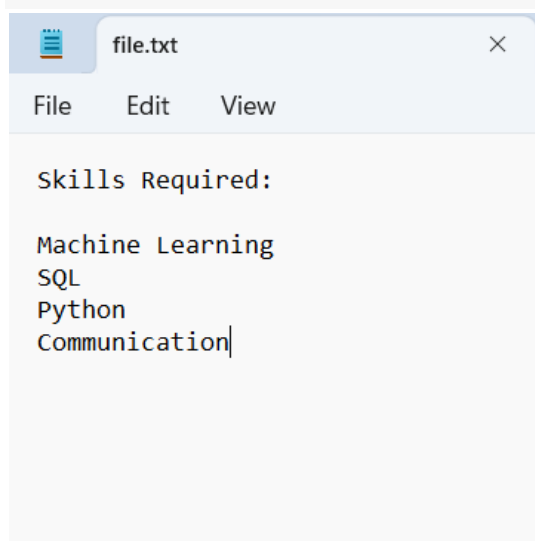
```
def calculate_similarity(found_skills, job_keywords):
    return len(set(found_skills).intersection(set(job_keywords))) /
            len(job_keywords)
```

Evaluate resume based on similarity score

```
def evaluate_resume(similarity_score, threshold=0.4):  
    return "Selected" if similarity_score >= threshold else "Not  
Selected"
```

Function to read resume from a file

```
def read_resume_from_file(file_path):  
    with open(file_path, 'r') as file:  
        return file.read()
```



Function to extract keywords from the job description

```
def extract_keywords_from_job_description(file_path):  
    resume_text = read_resume_from_file(file_path)  
    keywords = preprocess_text(resume_text)  
    return list(set(keywords))
```

Sample data

```
job_keywords = extract_keywords_from_job_description('file.txt')  
  
resumes = {'resume_id': [1, 2, 3],  
           'resume_text': [  
               "Experienced in Python and SQL, with a strong background  
in data analysis and machine learning.",
```



```

        "Skilled in communication and teamwork, proficient in SQL
and leadership.",
        "Knowledgeable in C++, data structures, and data analysis,
with some experience in Python."
    ]}

df = pd.DataFrame(resumes)

```

Process each resume

```

df['skills_found'] =df['resume_text'].apply(lambda x: extract_skills(x,
job_keywords))
Df['similarity_score']=df['skills_found'].apply(lambda
x:calculate_similarity(x, job_keywords))
Df['status']=df['similarity_score'].apply(lambda x: evaluate_resume(x))

```

Print resume status in a clean tabular format

```

headers = ["Resume ID", "Skills Found", "Similarity Score", "Status"]
table_data = df[['resume_id', 'skills_found', 'similarity_score',
'status']].values.tolist()
print(tabulate(table_data, headers, tablefmt="fancy_grid"))

```

Resume ID	Skills Found	Similarity Score	Status
1	['python', 'sql', 'machine', 'learning']	0.571429	Selected
2	['communication', 'sql']	0.285714	Not Selected
3	['python']	0.142857	Not Selected

Bar chart for resume selection status

```
def plot_bar_chart(df):  
    plt.figure(figsize=(10, 6))  
    sns.set_style("whitegrid")  
    palette = {"Selected": "#5cb85c", "Not Selected": "#d9534f"}  
  
    bar_chart = sns.countplot(x='status', data=df, palette=palette,  
hue='status')  
    bar_chart.set_title("Resume Selection Status", fontsize=18,  
fontWeight='bold')  
    bar_chart.set_xlabel("Selection Status", fontsize=14,  
fontWeight='bold')  
    bar_chart.set_ylabel("Number of Resumes", fontsize=14,  
fontWeight='bold')  
    for p in bar_chart.patches:  
        bar_chart.annotate(f'{p.get_height()}',  
                           (p.get_x() + p.get_width() / 2., p.get_height()),  
                           ha='center', va='baseline', fontsize=12, fontWeight='bold')  
    plt.tight_layout()  
    plt.show()
```

Display the bar chart

```
plot_bar_chart(df)
```



REFERENCES

- ❖ **NLTK Documentation**
<https://www.nltk.org/>
- ❖ **Seaborn Documentation**
<https://seaborn.pydata.org/>
- ❖ **Matplotlib Documentation**
<https://matplotlib.org/stable/users/index.html>
- ❖ **Removing stop word using NLTK**
<https://youtu.be/LLl3bQXhhzI?si=UPpxv8hBxSoYShFi>
- ❖ **Lemmatization using NLTK**
<https://youtu.be/L8jWbCxvrOU?si=J60Bv1DXozz3fOY5>

CONCLUSION

This project successfully demonstrates how natural language processing (NLP) can be used to automate resume screening by extracting relevant skills from resumes and matching them with job requirements. By implementing keyword-based similarity scoring and visualising the results, the system enhances the efficiency of the hiring process, reducing manual effort and improving consistency. The use of tools like NLTK, Pandas, and Seaborn allows for a robust solution that not only screens resumes effectively but also provides insights through data visualisation, aiding recruiters in making informed, data-driven hiring decisions.