Resume Screening System Using Artificial Intelligence

Student Name: ___________________
Department: **CSE (AI & ML)**
Semester: **II Year**
College Name: ___________________

Abstract

This project presents an AI-based Resume Screening System that automates the process of shortlisting resumes by comparing them with a given job description using Natural Language Processing techniques and machine learning algorithms.

Problem Statement

Manual resume screening is time-consuming and prone to bias. Recruiters need an automated system to efficiently rank resumes based on job relevance.

Objectives

1. Extract resume content from PDF files
2. Apply NLP preprocessing techniques
3. Compute similarity between resumes and job description
4. Rank resumes automatically

Technology Stack

Python, NLTK, Scikit-learn, TF-IDF Vectorizer, Cosine Similarity, PyPDF2, Pandas

System Architecture

Resume PDFs → Text Extraction → Text Preprocessing → TF-IDF Vectorization → Cosine Similarity → Resume Ranking

Algorithm

1. Input resumes and job description
2. Extract and preprocess text
3. Convert text into TF-IDF vectors
4. Calculate cosine similarity
5. Sort resumes based on similarity score

Implementation (Source Code)

```
import os
import nltk
import PyPDF2
import string
import pandas as pd
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity

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

def preprocess_text(text):
    text = text.lower()
    text = text.translate(str.maketrans('', '', string.punctuation))
    tokens = word_tokenize(text)
    tokens = [word for word in tokens if word not in stopwords.words('english')]
    return " ".join(tokens)

def extract_text_from_pdf(file_path):
    text = ""
    with open(file_path, 'rb') as file:
        reader = PyPDF2.PdfReader(file)
        for page in reader.pages:
            text += page.extract_text()
    return text

resume_folder = "resumes"
resumes = []
resume_names = []

for file in os.listdir(resume_folder):
    if file.endswith(".pdf"):
        text = extract_text_from_pdf(os.path.join(resume_folder, file))
        text = preprocess_text(text)
        resumes.append(text)
        resume_names.append(file)

job_description = preprocess_text(
    "Looking for a Python developer with ML, NLP and data analysis skills"
)

documents = resumes + [job_description]
vectorizer = TfidfVectorizer()
tfidf_matrix = vectorizer.fit_transform(documents)

scores = cosine_similarity(tfidf_matrix[-1], tfidf_matrix[:-1])[0]

results = pd.DataFrame({
    "Resume": resume_names,
    "Score": scores
}).sort_values(by="Score", ascending=False)

print(results)
```

Advantages

1. Reduces human effort
2. Unbiased resume evaluation
3. Fast and scalable solution

Applications

1. HR departments
2. Recruitment platforms
3. Campus placement systems

Conclusion

The Resume Screening System using AI effectively automates resume shortlisting by leveraging NLP and machine learning techniques, making recruitment efficient and unbiased.