

Name: Ahmad Moneer Mzayek

Course Title: Graduation Project

Course Code: Thesis_AI900

Lecturer: Dr. Ashraf Elsayed

Hand-in Method: Online Submission via E-Learning.

University: Midocean

Assignment Report: Intelligent Job Recommendation System for Job Seekers Using Machine learning.

Table of Contents

Abstract	3
Introduction	3
Problem Specification	3
Motivations	4
Objectives.....	5
Literature Review	5
Proposed Methodology	6
Timeline of Second Semester Project (10 Weeks)	8
References.....	8

Abstract

The goal of this project is to develop an intelligent job recommendation system based on machine learning and natural language processing techniques. The objective is to give job applicants relevant and personalized job recommendations that depend on their experience, qualifications, and resume information. The reason for the project comes from the inconvenience and hassle individuals face when they have to deal with large and unfiltered job postings, and it results in wastage of time and lost opportunities. By applying content-based filtering techniques, resumes and job descriptions are matched by measures of text similarity, making job matching more accurate and efficient. The proposed technique is to identify relevant skills and experiences from job descriptions and resumes, express this data as numeric vectors, and measure similarity scores by cosine similarity. One of the greatest contributions of this work is combining diverse sets like Wuzzuf and LinkedIn job advertisements with categorized resume information, which can be used as a broad and heterogeneous training set. This multi-sourced approach enhances the robustness of the model and allows for broader generalization over diverse fields of jobs. The goal is an explainable and light-weighted job recommendation system that can be further expanded to include hybrid approaches and explainable AI modules in the future. The programming environments employed for this purpose are Python, Scikit-learn, Gensim, and NLP toolkits such as NLTK. The system is scalable in nature and can be tuned to fit realistic job-matching scenarios.

Introduction

In today's competitive workplace, obtaining a good job can be difficult due to the huge number of job posts and the lack of manual systems for matching. A job recommendation system can subsequently reduce this burden by eliminating the matching process with the help of automation. These systems use AI and machine learning algorithms to analyze resume details and job openings effectively. However, they are often beset with issues such as data inconsistency, unstructured text, and the cold-start problem for new users. This report presents the motivation, methodology, and action plan for a system which strives to address these issues.

Problem Specification

The core problem to be solved by this project is inefficiency and frustration for applicants in navigating hundreds of job advertisements online to identify appropriate opportunities. Inability to focus filtering leads to missed opportunities as well as wastage of time. This project proposes building a content-based job recommender system capable of automatically scanning for a user's

CV and suggesting the most appropriate job ads corresponding to their skills, qualifications, and interests.

Formal Problem Statement: With a user's resume in textual format and a collection of job ads, the issue is to rank and recommend a list of jobs most aligned with the user profile using machine learning and natural language processing techniques.

Scope and Boundaries: The initial deployment of the system will address only content-based filtering, emphasizing resume-to-job matching based on skills and text similarity. Collaborative filtering methods, usage patterns, and user feedback will not be utilized in the initial phase. Matching will be on the text features of resumes and job descriptions alone.

Assumptions and Constraints:

- Resumes and job descriptions are in English and are structured or semi-structured.
- Each user has at least one resume with extractable experience and skills.
- Applicable job descriptions like job title, required skills, and a summary.

Type of Machine Learning Task: This is mainly a ranking issue. It aims to calculate similarity scores between job posting vectors and resume vectors and suggest the top-N highest-ranked jobs. It is not a classic classification or regression but unsupervised feature extraction and matching by similarity.

Motivations

In the current competitive job market, it can be a huge challenge to discover career opportunities that match one's interests and skill sets. Nevertheless, with the developments in machine learning and AI, we are able to apply data-based systems to personalized job recommendations based on user history and interest. My own process of working diligently for the past 11 months for a job in Riyadh without being able to secure a fitting one that aligns with my experience in sales, maintenance, operations, and business development has been the greatest driving force. The difficulty in describing my experience as a structured employment-matching process and the fact that most of the employers don't describe their employment needs clearly encouraged me to design an AI job recommendation system that helps applicants find appropriate jobs more efficiently.

Objectives

This project will develop a content-based job recommendation system that will suggest relevant job postings based on resumes. The system will use NLP algorithms to harvest major features, such as skills and job titles, and rank job postings according to cosine similarity scores. The system performance will be evaluated using precision, recall, and F1-score metrics. Future extension will be to scale this system to a hybrid recommendation model in actual recruitment environments.

Literature Review

The authors in [1] proposed a content-based recommendation model through Word2Vec-based resume and job description text representation and then ranking using cosine similarity. Although the system attained a good F1-score of 0.59, precision was impacted by Word2Vec's propensity to create unrelated skill expansions. The model also struggled with cold-start and diversity issues. Researchers in paper [2] contrasted different models like Content-Based Filtering (CBF), Collaborative Filtering (CF), and hybrid setups. Their work proved that the optimal performance (Recall = 63.97%) was delivered by a hybrid model which combined a basic CBF with an improved CF utilizing Word2Vec. Skill generalization was also proven to contribute significantly towards reducing dimensionality. The system's performance, however, declined while testing the system on short application histories of the users. Paper [3] focused on user profiling in terms of the jobs requested before. It enriched user skill vectors with Word2Vec's top-10 similar skills, and accuracy decreased (Recall = 30.03%) as semantic noise was incorporated. This indicated that naive enrichment with Word2Vec without filtering context would decrease recommendation quality. Authors in paper [4] employed a hybrid approach with CBF and CF for job suggestion enhancement. The model used a Pearson Correlation Coefficient (PCC) with a k-NN algorithm to calculate similarity. Their results showed that the best performance was given by a hybrid model using simple CBF and CF with Word2Vec augmentation (Recall = 63.97%, F1-score = 15.71%), confirming that hybrid models outperform a single recommender model when sufficient interaction data is available. Collectively, these works inform our approach. We acknowledge the limitations of Word2Vec of [1] and [3], and [2] and [4] demonstrate superiority with hybrid models. We will therefore begin with a content-based system because of its simplicity and interpretability, with a straightforward path for upgrading in the future using hybrid models and explainable AI systems.

Proposed Methodology

The method employed in this project is content-based filtering, which relies on interpreting the textual data of user resumes and job descriptions to determine their similarity. This was employed for its simplicity, interpretability, and performing well on datasets lacking user interaction history. The system primarily utilizes cosine similarity, a measure that calculates the cosine angle between the vector representations of the job descriptions and resumes. We use Word2Vec, a popular word embedding model, to map words into dense numerical representations based on their semantic context. This allows us to have resumes and jobs represented in the same space and effectively capture the semantic relationships between job requirements and skills.

A few critical steps constitute the preprocessing pipeline:

- **Text Processing:** Removal of punctuation and lowercase conversion.
- **Tokenization:** Splitting of text into words (tokens).
- **Normalization:** Applying stop-word removal and lemmatization to lowercase words to their stem base.
- **Feature Extraction:** Extracting and identifying qualifications from job postings and resumes.

Once the data preprocessing is done, Word2Vec model is learned to generate embeddings. The system then calculates similarity scores between a job posting and all the resumes to recommend the most similar top-N jobs. The following tools and libraries would be employed in the project:

- **Python:** For general implementation.
- **Gensim:** For Word2Vec modeling.
- **Scikit-learn:** For the calculation of similarity measures and evaluation.
- **NLTK:** For text preprocessing operations.

1- Datasets Description

Wuzzuf Job Listings (2025):

- **Source:** Open-source dataset from Kaggle.
- **Size & Format:** Over 8,000 job postings from Egypt in CSV format.
- **Features:** Includes job title, required skills, company, location, and job summary.
- **Preprocessing:** Removing duplicates, cleaning HTML tags, and normalizing text.

Wuzzuf Job Listings (FEB - 2025):

- **Source:** Open-source dataset from Kaggle.
- **Size & Format:** 3451 job postings from Egypt in CSV format.
- **Features:** Includes job title, company, location, and Level.
- **Preprocessing:** Removing duplicates, cleaning HTML tags, and normalizing text.

Resume Dataset – Gaurav Dutta:

- **Source:** Publicly available on Kaggle.
- **Size & Format:** 893 resumes in PDF & TXT formats, categorized by job roles.
- **Features:** Raw resume text with job domain labels (e.g., HR, Data Science, Developer).
- **Preprocessing:** Converting files to plain text, removing non-text elements, and labeling.

LinkedIn Job Postings (2023–2024):

- **Source:** Open-source data collected via scraping (Kaggle).
- **Size & Format:** 124,000 international job postings in CSV format.
- **Features:** Job title, company name, location, and detailed description.
- **Preprocessing:** Standardizing titles, removing duplicates, and cleaning text.

Resume Dataset – Snehaan Bhawal:

- **Source:** Open-source structured dataset from Kaggle.
 - **Size & Format:** Over 2400 resumes in CSV format, labeled by profession.
 - **Features:** Resume text labeled with job categories (e.g., Developer, Analyst, HR).
 - **Preprocessing:** Text normalization, tokenization, and removal of stop words.
-

Timeline of Second Semester Project (10 Weeks)

- **Week 1:** Dataset collection and environment setup
 - **Week 2:** Data cleaning and preprocessing
 - **Week 3:** Feature extraction and vectorization
 - **Week 4:** Baseline recommendation system implementation
 - **Week 5:** Implement main model
 - **Week 6:** Fine-tune hyperparameters
 - **Week 7:** Evaluation and metrics collection
 - **Week 8:** Comparative analysis
 - **Week 9:** Write final report and prepare presentation
 - **Week 10:** Final review and submission
-

References

- 1- <https://ijisrt.com/a-deep-learning-approach-to-job-recommendation-analysis-with-nlp>
Uday Jain¹, Daksh Jain², Aditya Raj Varshney³, A Deep Learning Approach to Job Recommendation Analysis with NLP, Volume 8, Issue 11, November – 2023, ISSN No:-2456-2165, November – 2023.
- 2- https://www.researchgate.net/publication/382508947_Job_Recommendation_System_Combining_Collaborative_Filtering_and_Content_Based_Filtering
Muhammad Thoriq ,Imam Sayyid Rahman
- 3- <https://www.ijscce.org/wp-content/uploads/papers/v9i3/C3266099319.pdf>
Juhi Dhameliya, Nikita Desai, Job Recommendation System using Content and Collaborative Filtering based Techniques, ISSN: 2231-2307, Volume-9, Issue-3, September 2019.

- 4- https://www.researchgate.net/publication/360418721_Job_Recommendation_System_Using_Hybrid_Filtering
Job Recommendation System Using Hybrid Filtering Aneesh Mulay,Shriyash Sutar, JitenPatel ,AditiChhabria,SnehalMumbaikar, 44,02002 (2022)
- 5- <https://www.kaggle.com/datasets/ahmedhazemelabady/wuzzuf-job-listings-dataset-egypt-january-2025>
Wuzzuf Job Listings Dataset (Egypt),Ahmed Hazem Elabady, Collected on January 2025
- 6- <https://www.kaggle.com/datasets/ahmedmansour343/wuzzuf-jobs-listing-in-egypt>
Wuzzuf Jobs Listing in Egypt,Ahmed Mansour343, Collected on 28th February 2025
<https://www.kaggle.com/datasets/gauravduttakiit/resume-dataset>
Gaurav Dutta.
- 7- <https://www.kaggle.com/datasets/arshkon/linkedin-job-postings>
Arsh Koneru and Zoey Yu Zou.
- 8- <https://www.kaggle.com/datasets/snehaanbhawal/resume-dataset>
Snehaan Bhawal.

Note

I found another sources they tried the idea

<https://www.kaggle.com/code/sonawanelalitsunil/resume-job-description-matching-ml-98-50/notebook#Title:-Resume%E2%80%93Job-Description-Matching-Dataset>

The difference is that it uses TF-IDF + machine Learning classification instead of Word2Vec + Cosine Similarity. Let's analyze it step by step so that you understand exactly what Shu be doing First the general idea It calculates the similarity between the text of the resume and the job description It then classifies the result into 3 levels: • Low → weak match • Medium → medium match • High → strong match Therefore, it is not a direct recommendation system 'It is a rating system for assessing how well a resume matches a job.

<https://www.kaggle.com/code/siddigantm/task-resume-matching-with-job-descriptions>

The project uses data from the "resume and job_description" dataset on Kaggle. Kaggle +1 The code contains steps such as word processing, extracting a similarity between a resume and a job description. Kaggle The goal: matching a resume with a job description, is not to select a list of multiple recommendations, but to find out the extent of the match.