

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

1. Abstract.....	3
2. Introduction	3
3. Problem Specification	3
4. Motivations.....	4
5. Objectives.....	4
6. Literature Review	5
7. Proposed Methodology	6
8. Datasets Description	6
9. Timeline of Second Semester Project (10 Weeks)	8
10. Expected Challenges.....	8
11. Reference	9

1. Abstract

The purpose of this project is to develop an intelligent job suggestion system on the basis of machine learning and natural language processing techniques.

The goal is to assist job applicants with personalized and appropriate job suggestions based on their qualifications, experience, and resume information. The motivation for this project stems from the hassle individuals face while working with large and unfiltered job listings, resulting in wastage of time and opportunities lost. By applying content-based filtering techniques, resumes are matched with job descriptions on the basis of text similarity measures by the system, enabling better and precision-based job matching.

The proposed approach is to pull out significant skills and experiences from job descriptions and resumes, present this information as numerical vectors, and compute similarity scores using cosine similarity. One of the key contributions of this project is the integration of different datasets, including Wuzzuf job postings, LinkedIn job postings, and categorized resume data, which provide a diverse and broad training base. This multi-source approach enhances the robustness of the model and allows broader generalization over diverse job fields.

The desired outcome is a lightweight and explainable job recommendation system that can be extended later to include hybrid approaches and explainable AI components.

The software tools used for this project include Python, Gensim, Scikit-learn, and NLP packages such as NLTK. The system is scalable and can be tailored to fit realistic job matching applications.

2. Introduction

In today's competitive labor market, it is difficult to get a suitable job due to the number of job postings and lack of personalized matching systems. A job recommendation system can ease this heavy load by preventing the necessity for manual matching via automation. These systems utilize AI and machine learning to process resume data and job postings.

However, the issues are data inconsistency, unstructured text, and the cold-start problem.

This report presents the motivation, methodology, and plan of action of a system that addresses these issues.

3. Problem Specification

The problem that this project aims to solve is the inefficiency and frustration that job seekers go through in trying to select the relevant job postings from thousands of job advertisements on the web. The absence of targeted filtering leads to lost matches and wasted time. This project proposes the development of a content-based job recommender system that automatically analyzes a user's resume and suggests the most appropriate job postings based on their interests, qualifications, and skills.

Formal Problem Statement, Having a user's resume in textual format and a set of job postings (also in text), the problem is to rank and recommend a list of jobs that best fit the user's

profile using machine learning and natural language processing techniques.

Scope and Boundaries The system will have content-based filtering only in its first version, with focus on resume-job matching on the basis of skill and text similarity.

No collaborative filtering methods, past application data, or user feedback will be utilized at this phase. Matching will only be based on textual characteristics of resumes and job descriptions. **Assumptions and Constraints** , Resumes and job descriptions are English and semi-structured or structured, Users have at least one resume with extractable experience and skills, Job descriptions contain relevant features such as title, skills, and summary. **Type of Machine Learning Task**, This is a ranking problem, where the goal is to compute resume vectors and job posting vectors similarity scores , and suggest the top-N highest ranked jobs. It is not classifying or regressing in the classical sense, but involves unsupervised feature extraction and similarity matching.

4. Motivations

In today's competitive job market, finding the right career opportunities that align with our interests and skills can be challenging. However, with the advancement in AI/ML technologies, we can leverage data-driven systems to provide personalized job recommendations based on user preferences and historical job data. I have been diligently trying to find a job in Riyadh for the past 11 months, but I have failed to get a satisfactory opportunity that fits my area of expertise. My experience in sales, maintenance & operations, and business development is not easily translatable to a formal job-matching process. In addition, most companies do not adequately define their employment needs, making it difficult to identify appropriate opportunities. This has motivated me to develop an AI-driven job recommendation system that can help job seekers identify the right jobs more efficiently

5. Objectives

This work will develop a content-based job recommendation system that matches user resumes against relevant job listings. The system obtains primary attributes such as skills and job titles with the assistance of NLP algorithms and ranks job listings based on cosine similarity. The system will be evaluated based on precision, recall, and F1-score. The system can be further extended to a hybrid recommendation system in the future to be used in real recruitment environments.

6. Literature Review

In paper [1], the authors proposed a content based recommendation model based on word2vec for text representation of resume and job description, then ranking by cosine similarity.

The model was trained on Stack Overflow developer survey and scraped job posting from the internet. Notwithstanding the system had a good F1-score of 0.59

precision broke down due to Word2Vec's unrelated skill expansions. Moreover, the model suffered from cold-start failures and lack of diversity.

In paper [2], the authors experimented and compared a number of models including CBF, CF, and some hybrid setups. Experiments showed that combining basic CBF

and improved CF with Word2Vec achieved the best performance (Recall = 63.97%). The

paper also highlighted the importance of skill generalization (e.g., grouping frameworks under parent languages like PHP) to reduce dimensionality. However, despite strong recall, the system's performance decreased when tested on users with limited application history.

Paper [3], focused on user profiling from the jobs they applied to earlier. In Word2Vec, user skill vectors were enriched by incorporating top-10 similar skills.

Although it was meant for more relevance, it resulted in lower accuracy (Recall = 30.03%) because semantic noise could be added.

The study demonstrated that naive Word2Vec enrichment without context filtering can decrease the quality of recommendations, emphasizing coverage vs. accuracy trade-off.

in paper [4], the researchers have used a hybrid recommendation approach with content-based filtering (CBF) and collaborative filtering (CF) for improving job suggestion quality.

The system utilized user-job interaction data to generate a user-job matrix, while job and skill relationships were demonstrated via skill mapping. Pearson Correlation Coefficient (PCC) together with k-NN algorithm was used to compute similarity.

Experiments were also carried out to compare three configurations of hybrids: (1) basic CBF + CF, (2) CBF and CF both written in Word2Vec, and (3) basic CBF + CF using

Word2Vec. Setup 3 gave the highest performance where recall was 63.97% and F1-score was 15.71. While accuracy and diversity were improved, the

paper highlighted common problems such as cold-start and noise generated by Word2Vec skill expansion. This paper confirms the assumption that hybrid

models would be better than individual recommenders if there is enough interaction data available.

Collectively, these papers direct our project approach. We know the pros and cons of using Word2Vec in content-based filtering from [1] and [3]. Paper [2] illustrates the benefit of hybrid models to perform this for greater recall, while [4] affirms that merging CBF and CF can yield better overall performance—especially with Word2Vec used in controlled settings. Thus, we begin with a content-based system due to its interpretability and simplicity of implementation, with future scope for improvement through hybrid models and explainable AI frameworks.

7. Proposed Methodology

Methodology for this project is content-based filtering, which is based on analysis of the textual content of user resumes and job postings to determine similarity.

The reason for this is its ease, interpretability, and usability on datasets with no user interaction history.

The system primarily uses cosine similarity, which is a metric that computes the angle between vector representations of resumes and job descriptions. We employ Word2Vec, a popular word embedding model, to map words to dense numerical vectors based on their semantic context, and utilize it to represent jobs and resumes in the same vector space.

We used this strategy since it is effective and can yield semantic relationships between skills and job descriptions. It also has fairly fast computation and scalability.

The preprocessing pipeline consists of a number of steps:

- Processing of text: removing punctuation and case folding.
- Tokenization: splitting text into words.
- Stop word removal and lemmatization to reduce word forms.
- Characterization of skills from resumes and job posts.

Once the data is preprocessed, we train the Word2Vec model and generate embeddings. The system then calculates similarity scores between each resume and job posting, and recommends the top-N most similar jobs.

The project will make use of tools and libraries such as:

- Python for implementation.
- Gensim for Word2Vec modeling.
- Scikit-learn for similarity measures and evaluation.
- NLTK for text preprocessing.

8. Datasets Description

1. Wuzzuf Job Listings (2025):

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

2. Resume Dataset – Gaurav Dutta:

- Source: Publicly available on Kaggle.
- Size & Format: 893 text resumes categorized by job roles (PDF & TXT formats).
- Features: Raw resume text with job domain labels (HR, Data Science, Developer).
- Preprocessing: Converting to plain text, removing non-text elements, labeling.
- Challenges: Limited resume count, some variation in formatting.

3. 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, detailed description.
- Preprocessing: Standardizing titles, removing duplicates, cleaning text.
- Challenges: Text inconsistency, some jobs missing skill fields.

4. Resume Dataset – Snehaan Bhawal:

- Source: Open-source structured dataset from Kaggle.
- Size & Format: Over 600 resumes in CSV format, labeled by profession.
- Features: Resume text, labeled with job categories (Developer, Analyst, HR).
- Preprocessing: Text normalization, tokenization, removal of stop words.
- Challenges: Lack of detailed metadata (years of experience, location), but good structure.

9. 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 main report and prepare presentation
Week 10: Final review and submission

10.Expected Challenges

The following challenges are anticipated during the implementation of the JRS:

- Extracting structured skills from free-text resumes is difficult due to variations in writing style.
- Using Word2Vec may introduce noise by expanding user profiles with unrelated skills.
- Cold-start problem for new users with limited historical data.
- Lack of real-time user feedback to validate recommendations dynamically.

11.Reference

- 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>
Ahmed Hazem Elabady.
- 6- <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.