

A WEB BASED APPLICATION FOR RESUME PARSER USING NATURAL LANGUAGE PROCESSING TECHNIQUES

Kanchan A Purohit¹, Anagha V², Chaitra Hasyagar³, Deeksha R⁴, Deesha B Raj⁵

¹Assistant Professor, Computer Science & Engineering, Bangalore Institute of Technology, Karnataka, India

^{2,3,4,5}B.E Student, Computer Science & Engineering, Bangalore Institute of Technology, Karnataka, India

Abstract - Resume is a powerful tool in a job search which outlines one's knowledge, skills, experience, expertise and accomplishments. An excellent resume attracts potential companies. It is very hard to find the right talent for a particular job. In order to make recruiters job easy, we have proposed a model which extracts details like education, projects, experience, phone no., email ID etc. and ranks the resumes based on company's specifications using Natural Language Processing (NLP) techniques. We have built a job portal where employees and applicants can upload resumes for a particular job, the required information will be parsed, structured resumes will be generated and the resumes of the employees will be ranked based on the company's skill set and employees skills mentioned in the resume.

Key Words: Parsing, Ranking, Natural Language Processing, K Nearest Neighbor, Vectorization, Cosine Similarity

1. INTRODUCTION

Earlier recruiters used to spend most of their time in manually screening resumes. In this process, organizations would lose out on quality candidates and recruiters waste their time and effort. To avoid this, recruiters use automated Resume Parsing Systems in order to cut down on monotonous tasks and make recruitment easy. Resume Parsing System, lets recruiters find highly qualified candidates for a certain position. We have designed a model in which a parser will extract certain keywords like education, skills, experience, name, phone no., email ID etc. from the resume that is fed to it and ranks it accordingly for making the job of HR manager easier to select candidates for recruitment process.

A set of resumes from employees in .pdf format is taken as input and the employer gets a list of candidates that has been ranked according to parser and classifier. Applicants receive mail regarding status identified with their resumes.

In this proposed methodology, we are using Natural Language Processing technique for parsing the resume according to the particular companies. We use Named Entity recognizer (NER) which labels the data according to the type of their attribute and displays the text with the label. One of the platforms of NER is spaCy, an open source

named entity visualizer that is used for information extraction. KNN algorithm is used for classification.

2. SYSTEM ARCHITECTURE

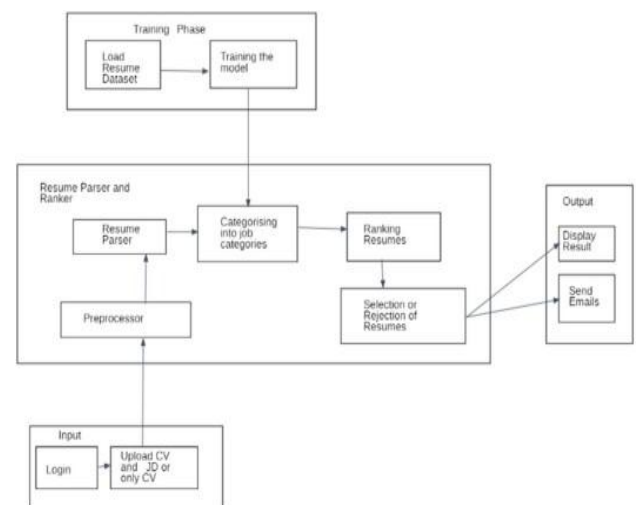


Fig -1: Architecture of the proposed system

- Training module: Loading dataset is a part of the training phase where we load our dataset to the model to test its accuracy. Training of the model is the next part of this phase where we train the model against the given dataset.
- Input: The candidate and recruiter upload the resumes in the required format.
- Processing: Each resume is parsed and the resumes are ranked according to their category.
- Output: The resumes along with their ranks are displayed and an option to send an email is given.

2.1 MODULE DESCRIPTION

- Login: In this component job seekers should register with our application and then they can login. Also in this component the recruiter can register or login to our application.

- **Input Job Availability and Resume Uploader:** The candidates upload their resume in pdf format. The recruiters can upload single or multiple resumes and can add the vacancy details and required skills.
- **Resume Preprocessor:** In this component, we remove any unnecessary information from resumes like stop words, hashtags, and special characters.
- **Resume Parser:** We are extracting the resume contents like name, email, education, skills, etc using the spacy parser function.
- **Loading the dataset:** In this component we load the labeled dataset into the system for testing against the untrained data.
- **Building KNN classifier:** In this component we use sklearn class to train a model to predict the suitable job title for their skills which is extracted from the seekers resume.
- **Categorizing into jobs categories:** Uses the trained model from the previous component to categorize each resume into corresponding job category.
- **Resume Ranker:** In this component system will compare the cosine similarity between the job seekers skills and recruiter’s requirement skills for the corresponding job using NLP. Based on the similarity score resumes are ranked.
- **Resume selection and rejection:** Depending on the number of vacancies the resumes will get selected or rejected.
- **Results are displayed:** The list of ranked resumes is displayed to the recruiter.
- **Emails are sent:** Status emails regarding selection or rejection of their application are sent.

3. METHODOLOGY

The user registers or login to the system and uploads the resume. The system preprocesses, categorizes and ranks the resume and displays the output. The steps performed by the system are explained in the further sections.

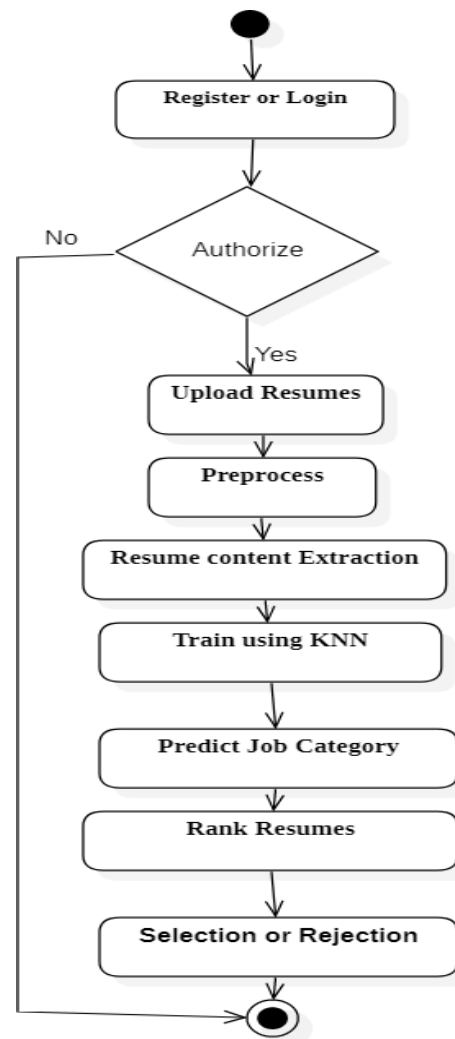


Fig -2: Activity diagram of the resume ranker application

3.1 PREPROCESSING

The natural language expressed in the resumes are not suitable for application of any machine learning or Natural language processing techniques. Hence the primary purpose of this preprocessing step is data cleaning. We have used spacy, an open-source natural language processing library to perform preprocessing of the resumes.

The first step in preprocessing is sentence detection. Each sentence is identified and isolated. The next step in preprocessing is tokenization. Tokenization is extracting the words and reducing them into their basic form. Example: extracting “organization” and reducing it into “organize”. The reduced form of the word is called as a token.

In the next step, the punctuations and stop words are removed. Stop words are frequently used words that do

