International Journal for Modern Trends in Science and Technology, 9(03): 39-45, 2023

ISSN: 2455-3778 online
DOI: https://doi.org/10.46501/IJMTST0903005
Available online at: http://www.ijmtst.com/vol9issue03.html

A Resume Keywords Prediction using Natural Language Processing

Mugada.Sri Lakshmi Vani[1] | A.Neeraja[2] | S. Sneha[2] | G. Sai Susmitha[2]

[1]Assistant Professor, Department of Information Technology, Andhra Loyola Institute of Engineering and Technology, Vijayawada, INDIA
[2]UG Students, Department of Information Technology, Andhra Loyola Institute of Engineering and Technology, Vijayawada, INDIA

To Cite this Article
Mugada.Sri Lakshmi Vani, A.Neeraja, S. Sneha and G. Sai Susmitha. A Resume Keywords Prediction using Natural Language Processing. International Journal for Modern Trends in Science and Technology 2023, 9(03), pp. 39-45. https://doi.org/10.46501/IJMTST0903005.

Article Info
Received: 06 February 2023; Accepted: 04 March 2023; Published: 07 March 2023.

ABSTRACT

This paper focuses majorly on the design of the system which will be used to screen resumes (Curriculum Vitae) for a particular job posting. In the proposed system will encourage the job applicant candidates as well as the recruiters to use it for job applications and screening of resumes. Recruitment is a tedious process wherein the first task for any recruiter is to screen the resumes. The proposed web application is designed in such a way that job applicant as well as recruiters can use it with ease for applying for job openings and screening respectively. The recruiters from various companies can post the details of the job openings available in their respective companies. The interactive system will allow the job applicants to submit their resume and apply for their job postings they may still be interested in. The resumes submitted by the candidates are then compared with the job profile requirement posted by the company recruiter by using techniques like machine learning and Natural Language Processing (NLP). Scores can then be given to the resumes and they can be ranked from highest match to lowest match.

Key Words: Curriculum Vitae, machine learning, Natural Language Processing

1. INTRODUCTION

Talent acquisition is an important, complex, and time-consuming function within Human Resources (HR). The sheer scale of Indias market is overwhelming. Not only is there a staggering one million people coming into the job market every month, but there is also huge turnover[1-3]. As per LinkedIn, India has the highest percentage of the workforce that is "actively seeking a new job". Clearly, this is an extremely liquid, massive market but one that also has many frustrating inefficiencies[4-6]. The most challenging part is the lack of a standard structure and format for resume which makes short listing of desired profiles for required roles very tedious and time-consuming [7].

Effective screening of resumes requires domain knowledge, to be able to understand the relevance and applicability of a profile for the job role. With a huge number of different job roles existing today along with the typically large number of applications received, short-listing poses a challenge for the human resource department [8]. Which is only further worsened by the lack of diverse skill and domain knowledge within the

HR department, required for effective screening? Being able to weed out non-relevant profiles as early as possible in the pipeline results in cost savings, both in terms of time as well as money. Today the industry face three major challenges: • Separating right candidates from the pack - India being a huge job market and with millions seeking jobs; it is humanly impossible to screen the CVs and find the right match. This makes the whole hiring process slow and inefficient costing resources to the companies [9-11].

Making sense of candidate CVs - Second challenges are posed by the fact that the CVs in the market are not standard practically every resume in the market has different structure and format [12-14]. HR has to manually go through the CVs to find the right match to the job description. This is resource intensive and prone to error whereby a right candidate for the job might get missed in the process [15]. Knowing that candidates can do the job before you hire them -The third and the major challenge is mapping the CV to the job description to understand if the candidate would be able to do the job for which she is being hired [16-17].

2. SYSTEM DESIGN AND ANALYSIS

All the existing methods are using either manual approach or face to face interviews for selecting the candidate profiles. But now the strategy is completely changed and hence a very accurate model is required which can identify the candidates very easily and effectively for giving job for them.

As there is lot of demand now days for several jobs, recruiters are filtering the resumes through online mode rather than manual approach. So in this case if the job seeker failed to prepare the resume with suitable keywords then there is a chance of losing the job due to missing of main keywords. Hence this motivated me to develop an application in which job seekers can able to check his resume before he upload for any job provider. Based on the job category role, our application will suggest best keywords and how much accuracy the current resume is holding.

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential. Three key considerations involved in the feasibility analysis are

A. HARDWARE REQUIREMENTS

Processor: Core I3
RAM: 4 GB (min)
Hard Disk: 100 GB

B. SOFTWARE REQUIREMENTS

Operating system: Windows7 (Min).
Coding Language: Python
Front-End: Google Collab
Dataset: ML Algorithms on two datasets

3.SYSTEM ARCHITECTURE

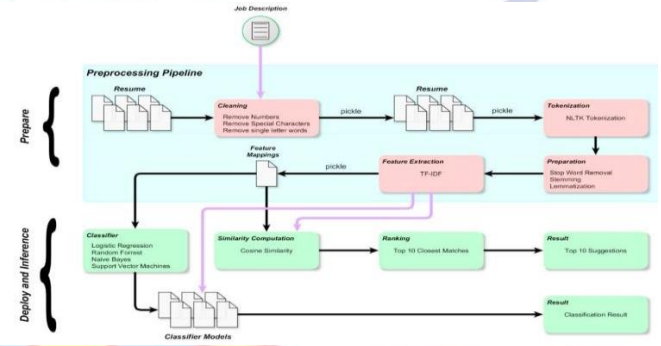


Fig. 1 System Architecture

3.1 MODULES

The modules are Gathering data, Pre-Processing, Processing, Interpretation.

The whole approach is depicted by the following flowchart.

DATA GATHERING
PRE-PROCESSING
PROCESSING
INTERPRETATION

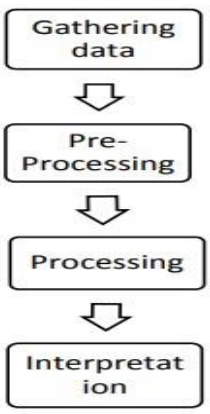


Fig. 2 Flowchart of the Technique

3.1.1 DATA GATHERING

Here we try to load the data set from two different sources one is from google and other is from LinkedIn profiles from Kaggle. Once dataset is downloaded, we try to load the dataset to the system for performing the operations.

3.1.2 PRE-PROCESSING

Data pre-processing is a technique that is used to convert raw data into a clean dataset. The data is gathered from different sources is in raw format which is not feasible for the analysis. Pre-processing for this approach takes 4 simple yet effective steps.

- **Attribute selection**: Some of the attributes in the initial dataset that was not pertinent (relevant) to the experiment goal were ignored.
- **Cleaning missing values:** In some cases the dataset contain missing values. We need to be equipped to handle the problem when we come across them. Obviously you could remove the entire line of data but what if you're inadvertently removing crucial information? after all we might not need to try to do that. one in every of the foremost common plan to handle the matter is to require a mean of all the values of the same column and have it to replace the missing data. The library used for the task is called Scikit Learn preprocessing. It contains a class called Imputer which will help us take care of the missing data.

- **Training and Test data:** Splitting the Dataset into Training set and Test Set Now the next step is to split our dataset into two. Training set and a Test set. We will train our machine learning models on our training set, i.e our machine learning models will try to understand any correlations in our training set and then we will test the models on our test set to examine how accurately it will predict. A general rule of the thumb is to assign 80% of the dataset to training set and therefore the remaining 20% to test set.

3.1.3 PROCESSING

Classification of data is a two-phase process. In phase one which is called training phase a classifier is built using training set of tuples. The second phase is the classification phase, where the testing set of tuples is used for validating the model and the performance of the model is analyzed. Here we try to classify both the datasets and make the system understand several keywords which are required for choosing job profile and how to classify the jobs based on suitable keywords.

3.1.4 INTERPRETATION

The data set used for is further spitted into two sets consisting of two third as training set and one third as testing set. Here we apply ML models such as N-Gram and WORDCLOUD for making the recommendation of job-related keywords for the end users based on suitable job type.

4. OUTPUT SCREENS

LOAD DATASET

IMPORT LIBRARIES


```
[ ]  import pandas as pd
     import numpy as np
     import matplotlib.pyplot as plt
     import seaborn as sns
     import warnings
     warnings.filterwarnings('ignore')
```

```
df=pd.read_csv('Resume_skills.csv')
df=df.drop(columns=['Unnamed: 0'])
df.head()
```

DISPLAY THE DATASET

	Category	Resume_skills
0	HR	{loyalty marketing, micros, office, mar, custo...
1	HR	{contractors, brochures, office, output, credi...
2	HR	{budget management, recruiting, office, turnov...
3	HR	{focus, completion, call quality, customer ser...
4	HR	{christmas, personnel records, organizational ...

```
import nltk
nltk.download("popular")
nltk.download('stopwords')
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer
lemmatizer = WordNetLemmatizer()
from bs4 import BeautifulSoup
import re
from nltk.corpus import stopwords # Import the stop word list
from wordcloud import WordCloud
# stopwords.words("english")
from nltk import word_tokenize
from nltk.util import ngrams
```

```
[nltk_data] Downloading collection 'popular'
[nltk_data]    |
```

ELIMINATE DUPLICATE KEYWORDS FOR UNIQUENESS

```
[ ] df.Category.unique()
```

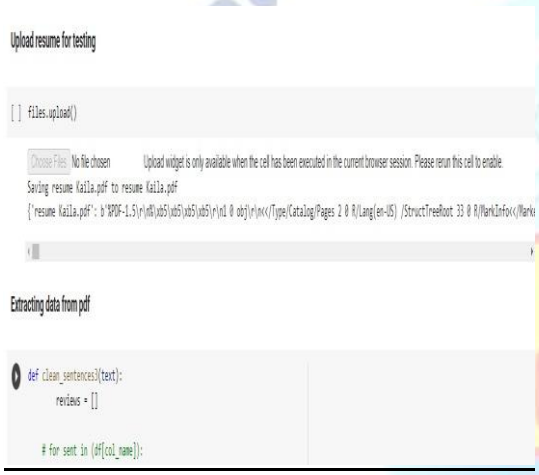
```
array(['HR', 'DESIGNER', 'INFORMATION-TECHNOLOGY', 'TEACHER', 'ADVOCATE',
       'BUSINESS-DEVELOPMENT', 'HEALTHCARE', 'FITNESS', 'AGRICULTURE',
       'BPO', 'SALES', 'CONSULTANT', 'DIGITAL-MEDIA', 'AUTOMOBILE',
       'CHEF', 'FINANCE', 'APPAREL', 'ENGINEERING', 'ACCOUNTANT',
       'CONSTRUCTION', 'PUBLIC-RELATIONS', 'BANKING', 'ARTS', 'AVIATION',
       'Data Science', 'Advocate', 'Arts', 'Web Designing',
       'Mechanical Engineer', 'Sales', 'Health and fitness',
       'Civil Engineer', 'Java Developer', 'Business Analyst',
       'SAP Developer', 'Automation Testing', 'Electrical Engineering',
       'Operations Manager', 'Python Developer', 'DevOps Engineer',
       'Network Security Engineer', 'PMO', 'Database', 'Hadoop',
       'ETL Developer', 'DotNet Developer', 'Blockchain', 'Testing'],
      dtype=object)
```

WORD CLOUD

APPLY NLP

```
from matplotlib.gridspec import GridSpec

target_count=df['Category'].value_counts()
target_label=df['Category'].unique()

plt.figure(1, figsize=(50,70))
grid=GridSpec(2,2)

cmap=plt.get_cmap('coolwarm')
colors=[cmap(i) for i in np.linspace(0,1,3)]
plt.subplot(grid[0,1],aspect=1, title='Category Distribution')

source_pie=plt.pie(target_count,labels=target_label,autopct='%1.1f%%', shadow=True, colors=colors)
```

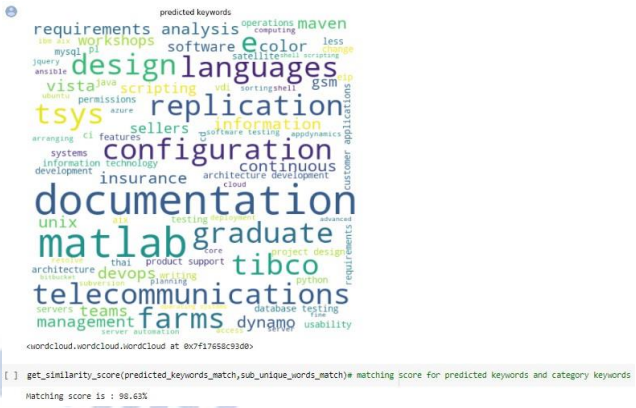
CHOOSE CATEGORY AND UPLOAD RESUME



1.

LIST OF WORDS

PREDICT THE ACCURACY AND RECOMMEND KEYWORDS



5. SYSTEM TESTING

TEST CASES

Negative Test Cases

Test Case 1: Dataset Selection	Priority(H.L):High
Test Objective: to check Dataset Selection success or fail	
22252Test Description: In this HOME screen, when the user selects the resume_skills dataset	
Requirement Verified: Yes	
Test Environment: System connected with the dataset.	
Actions	Expected Results
◎ when the user selects the dataset by clicking upload button.	• Uploading dataset fail • Select valid Dataset
Pass: no **Condition Pass**: No **Fail:** Yes	
Problems/Issues: Yes	
Notes: Selection is fail	

Test Case 2: Execute inception V3 Algorithms	Priority(H.L):High
Test Objective: has to check whether algorithms are working	
52Test Description: In this home page user dataset and after that he will check resume_skills dataset	
Requirement Verified: No	
Test Environment: System connected with the dataset.	
Actions	Expected Results

◎ Run ML Algorithms by clicking Run button	• Comparing fail • Select valid dataset and algorithms
Pass: no **Condition Pass**: No **Fail:** Yes	
Problems/Issues: Yes	
Notes: Algorithm fail	

Positive Test Cases

Test Case 1: Dataset Selection	Priority(H.L):High
Test Objective: to check dataset Selection success or fail	
22252Test Description: In this HOME screen, when the user selects the input dataset it display path label	
Requirement Verified: Yes	
Test Environment: System connected with the dataset.	
Actions	Expected Results
◎ when the user selects the input by clicking upload button.	◎ Dataset path Label can be shown
Pass: yes **Condition Pass**: No **Fail:** No	
Problems/Issues: No	
Notes: Dataset Selection successfully completed	

Test Case 2: Running inception V3 Algorithm	Priority(H.L):High
Test Objective: has to check ML algorithms related to trained Dataset	
22252Test Description: In this home page user will choose resume_skills dataset	
Requirement Verified: No	
Test Environment: System connected with the dataset.	
Actions	Expected Results
◎ Run VGG-19 algorithm by clicking Run button	◎ Display Features of resume skills dataset
Pass: yes **Condition Pass**: No **Fail:** Yes	
Problems/Issues: No	
Notes: NLP Model is successful.	

6. CONCLUSION

Huge number of applications received by the organization for every job post. Finding the relevant candidate's application from the pool of resumes is a tedious task for any organization nowadays. The process of classifying the candidate's resume is manual, time consuming, and waste of resources. To overcome this issue, we have proposed an automated machine learning based model which recommends suitable candidate's resume to the HR based on given job description. The proposed model worked in two phases: first, classify the resume into different categories. Second, recommends resume based on the similarity index with the given job description. The proposed approach effectively captures the resume insights, their semantics and yielded an accuracy of 78.53% with Linear SVM classifier. The performance of the model may enhance by utilizing the deep learning models like: Convolutional Neural Network, Recurrent Neural Network, or Long-Short Term Memory and others. If an Industry provides a large number of resume, then Industry specific model can be developed by utilizing the proposed approach. By involving the domain experts like HR professional would help to build a more accurate model, feedback of the HR professional helps to improve the model iteratively.

Conflict of interest statement

Authors declare that they do not have any conflict of interest.

REFERENCES

[1] Al-Otaibi, S.T., Ykhlef, M., 2012. A survey of job recommender systems. International Journal of Physical Sciences 7, 5127–5142.

[2] Breaugh, J.A., 2009. The use of biodata for employee selection: Past research and future directions. Human Resource Management Review 19, 219–231.

[3] Breiman, L., 2001. Random forests. Machine learning 45, 5–32.

[4] Carrer-Neto, W., Hernandez-Alcaraz, M.L., Valencia-Garc ´ ´ıa, R., Garc´ıa-Sanchez, F., 2012. Social knowledge-based recommender system. ´ application to the movies domain. Expert Systems with applications 39, 10990–11000.

[5] Celma, O., 2010. Music recommendation, in: Music recommendation and discovery. Springer, pp. 43–85.

[6] Das, A.S., Datar, M., Garg, A., Rajaram, S., 2007. Google news personalization: scalable online collaborative filtering, in: Proceedings of the 16th international conference on World Wide Web, ACM. pp. 271–280.

[7] Diao, Q., Qiu, M., Wu, C.Y., Smola, A.J., Jiang, J., Wang, C., 2014. Jointly modeling aspects, ratings and sentiments for movie recommendation (jmars), in: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM. pp. 193–202.

[8] Farber, F., Weitzel, T., Keim, T., 2003. An automated recommendation approach to selection in personnel recruitment. AMCIS 2003 proceedings , 302.

[9] Golec, A., Kahya, E., 2007. A fuzzy model for competency-based employee evaluation and selection. Computers & Industrial Engineering 52, 143–161.

[10] Howard, J.L., Ferris, G.R., 1996. The employment interview context: Social and situational influences on interviewer decisions 1. Journal of applied social psychology 26, 112–136.

[11] Lin, Y., Lei, H., Addo, P.C., Li, X., 2016. Machine learned resume-job matching solution. arXiv preprint arXiv:1607.07657 , 1–8.

[12] Loper, E., Bird, S., 2002. Nltk: the natural language toolkit. arXiv preprint cs/0205028 .

[13] Lu, J., Wu, D., Mao, M., Wang, W., Zhang, G., 2015. Recommender system application developments: a survey. Decision Support Systems 74, 12–32.

[14] Maheshwary, S., Misra, H., 2018. Matching resumes to jobs via deep siamese network, in: Companion Proceedings of the The Web Conference 2018, International World Wide Web Conferences Steering Committee. pp. 87–88.

[15] Malinowski, J., Keim, T., Wendt, O., Weitzel, T., 2006. Matching people and jobs: A bilateral recommendation approach, in: Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06), IEEE. pp. 137c–137c.

[16] Mooney, R.J., Roy, L., 2000. Content-based book recommending using learning for text categorization, in: Proceedings of the fifth ACM conference on Digital libraries, ACM. pp. 195– 204.