IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
Volume 4, PP 01-04

Resume Ranker

Devraj Redkar[1], Sanjay Payiri[2], Durgesh Sawant[3], Vishal Prajapati[4], Nileema Pathak[5]

[1](Information Technology, Atharva College of Engineering/Mumbai University, India) [2](Information Technology, Atharva College of Engineering/Mumbai University, India) [3](Information Technology, Atharva College of Engineering/Mumbai University, India) [4](Information Technology, Atharva College of Engineering/Mumbai University, India) [5](Information Technology, Atharva College of Engineering/Mumbai University, India)

Abstract : *The proposed method is to understand the intelligence behind the hiring pattern and apply the machine learning to accommodate the identified intelligence. The proposed method offers the ranking system according to the hiring patterns. This paper proposes a highly trained models along with the traditional search method, predicts the ranking and sorting of resumes with high accuracy and simplifies the job of human resourcing efficiently.*

Keywords - *Docopt, Hiring Pattern, Human Resources, PyPDF2, Python, Resume.*

I. Introduction

Human Resource (HR) agencies use various head hunting tools and online search methods. These search methods connected with the database of millions of resumes. These are the simple search engines who parses the resumes against the given keywords and offers the best match results. The list of the searching keywords is usually prepared by the HR after reading the job description several times. The HR downloads these searched resumes and does the manual work by opening and reading the resumes. By this way, HR person tries to find the resumes which are best match to the Job description. This is a cumbersome process and requires reasonable time and multiple discussion with the candidate before offering the resume to the client. Usually, due to the complexity of the database, many efficient resumes missed out from the search results or not considered due to stringent timeline ofclosure.

A. Traditionalmethod

II. Literaturereview

Human Resource (HR) agencies use various head hunting tools and online search methods. These search methods connected with the database of millions of resumes. These are the simple search engines who parses the resumes against the given keywords and offers the best match results. The list of the searching keywords is usually prepared by the HR after reading the job description several times. The HR downloads these searched resumes and does the manual work by opening and reading the resumes. By this way, HR person tries to find the resumes which are best match to the Job Description. This is a cumbersome process and requires reasonable time and multiple discussion with the candidate before offering the resume to the client. Usually, due to the complexity of the database, many efficient resumes missed out from the search results or not considered due to stringent time lines ofclosure.[1]

B. ResumeRater

Various online portals like naukri.com, rezscore.com, resumegrade.com offer online resume screening and provide general optimal score for the uploaded resume.[4][5]

III. Methodologies

Our System consists of 3 modules:

1. List ofResumes.
2. ParsingCV.
3. List ofKeywords.

1. List ofResumes.

HR manager collects resumes which are to be ranked and creates a list of resumes. The resumes should be in given format, so that the algorithm can rank the resumes based on that format.[2]

2. Parsing CV:

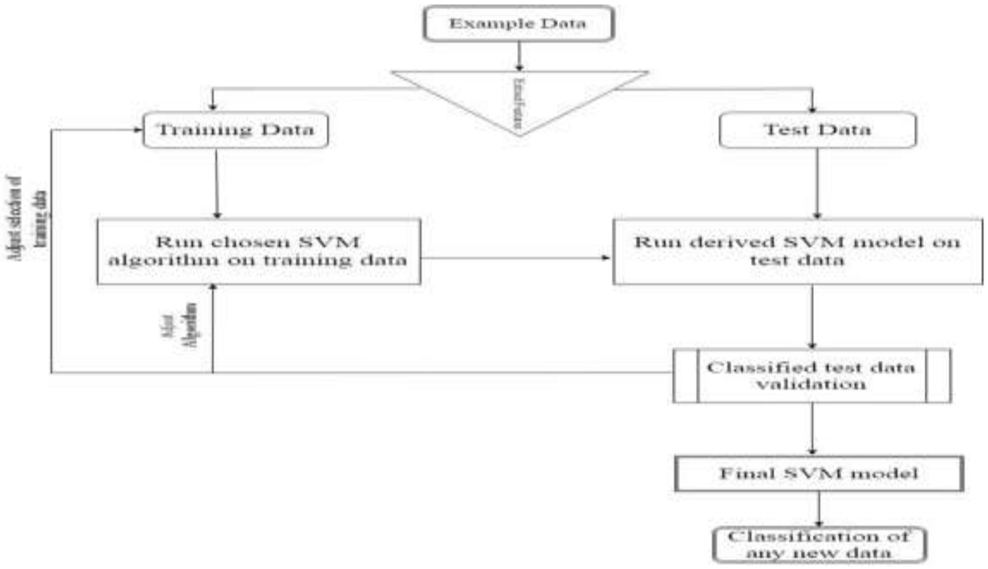
Resumes can be in any format such as TEXT, PDF & DOCX. So for pre-processing and parsing of keywords from the resumes for processing by the algorithm, we are using different packages for each file format.

- PyPDF2: **PyPDF2** is a pure-python PDF toolkit originating from the PyPdf project. It is currently maintained by Phaseit, Inc. PyPDF2 can extract data from PDF files, or manipulate existing PDFs to produce a newfile.PyPDF2iscompatiblewithPythonversions2.6,2.7,and3.2-3.5.
- Docopt: **docopt** is based on conventions that have been used for decades in help messages and man pages for describing a program's interface. An interface description in **docopt** is such a help message, but formalized.
- Python-docx: python-docx is a Python library for creating and updating Microsoft Word (.docx)files.

IV. List Ofkeywords:

The company will provide a list of skills required for a particular post. The company will also give weight- age of each skill to be assigned in the keyword file. The algorithm will rank the resumes based on the keyword file.

ALGORITHM:


Example Data
Extract Features
Training Data
Test Data
Adjust selection of training data
Run chosen SVM algorithm on training data
Run derived SVM model on test data
Adjust Algorithm
Classified test data validation
Final SVM model
Classification of any new data

Support Vector Machine (SVM) is a machine learning technique to separate data which tries to maximize the gap between the categories (a.k.a Margin).[3]

Maximum Margin Classifieris the simple way to divide the data if it is linearly (or via a plane in multi-dimensionalspace)separable.Butduetoobviousreasonsitcan’tbeappliedtothedatawherenoclearlinear boundary is possible.

Support Vector Classifier is an extension to maximum margin classifier where some misclassifications are allowed to happen.

Support Vector Machine is a further extension to SVC to accommodate non-linear boundaries. Though there is a clear distinction between various definitions but people prefer to call all of them as SVM to avoid any complications.

SVM is a further extension of SVC that results from enlarging the feature space in a specific way using “KERNELS”. The main idea is that the system may want to enlarge its feature space in order to accommodate a non-linear boundary between the classes. Dealing with high dimensional data or enlarging the feature space requires high computational power. The Kernel approach is simply an efficient computational approach for the same. Solution to the Support Vector Classifier problem involves only the inner – product of the observations and not the variables themselves. This is the basic idea behind using a Kernel technique. A Kernel is a function that quantifies the similarity of two observations. When a Support Vector Classifier is combined with a non-linear Kernel, the resulting classifier is known as SVM. Depending on the data one has to select the Kernel which best classifies it. There is no direct rule or formula which can choose a Kernel, most of the timesit will be trialanderror.Therearevariouscoefficientslike‘gamma’, ‘degree’, ‘coef’ whichonecantuneinordertoreacha bestclassification.

V. Figures

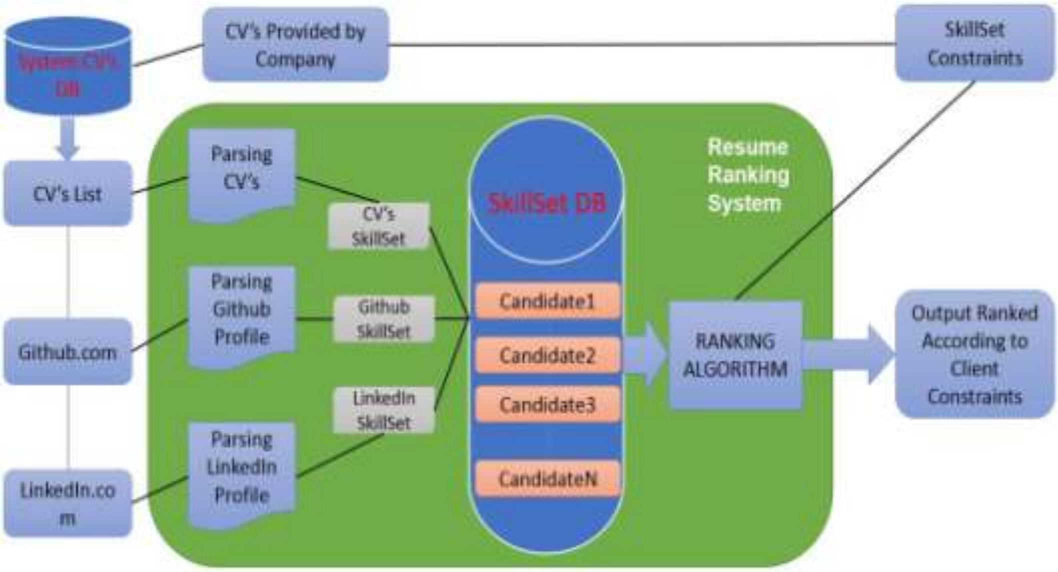


Figure 1: System Block Diagram

HR manager collects resumes which are to be ranked and creates a list of resumes. Resumes are also collected from Github and LinkedIn. Resumes can be in any format such as TEXT, PDF & DOCX. The company will provide a list of skills required for a particular post. The company will also give weight-age of each skill to be assigned in the keyword file. The algorithm will rank the resumes based on the keyword file.

VI. Expected Output

Resumes are ranked according to the keywords provided by the user and algorithm ranks the resumes from 1 to x, with rank one having 100 percentile. Here's the format of how the output will be displayed: (File_Format/Percentile/ [count] /Resume_name). A file_format is a type of the file which can be a .pdf or .docx or .txt. The percentile is calculated using a function (percentile = self.get_percentile(d, self.files)), where "d" is the method of parsing. Depending on the keywords and multiplier of that keywords assigned by the user, a count will be generated for a particular resume which will be displayed in square brackets. The formula to calculate count is (count = count + (keyword * multiplier)). Resume_name is the name of the file. Example: (PDF / 68.87% / [8] / DevrajRedkarResume). The formula for ranking of these resumes is (rank = rank + word_percentage).

VII. Future Scope

As future work we plan to expand the application functionalities by using Support Vector Machine (SVM) for large data sets especially for large organizations. Also for processing large amount of data based on text input we can improve the performance by using Natural LanguageProcessing.

VIII. Conclusion

A benchmarking is done to observe the results obtained by the manual process and the resumes recommended by the proposed method through Machine Learning. The results obtained by the proposed method are mathematically and practically much better than traditional methods. Ranking and Re-ranking based on the Hiring Pattern are very useful for next generation Head Huntingsolution.

References

[1]. "Machine Learning Methods for Solving Complex Ranking and Sorting Issues in HumanResourcing", Institute of Electrical and Electronics Engineers (IEEE) 2017

[2]. "IntelligentHiringwithResumeParserandRankingusingNaturalLanguageProcessingandMachine Learning", International Journal of Innovative Research in Computer and Communication Engineering(IJIRCCE)

[3]. Swapnil Sonar, Resume Parsing with Named Entity Clustering Algorithm, IEEE Research, may2012, http://www.slideshare.net/swapnilmsonar/resume-parsing-with-named-entity-clustering-algorithm

[4]. Sovren Resume/CV Parser,http://www.sovren.com

[5]. Connectifier,http://www.connectifier.com

[6]. Rchillies,http://www.rchillies.com