International Journal of Modern Developments in Engineering and Science
Volume 1, Issue 3, March 2022
https://www.ijmdes.com | ISSN (Online): 2583-3138

Machine Learning Based Resume Recommendation System

Anushka Lad[1*], Siddhi Ghosalkar[2], Balkrishna Bane[3], Krutika Pagade[4], Anupama Chaurasia[5]
[1,2,3,4]*UG Scholar, Department of Electronics and Telecommunication Engineering, K. C. College of Engineering and Management Studies and Research, Thane, India*
[5]*Assistant Professor, Department of Electronics and Telecommunication Engineering, K. C. College of Engineering and Management Studies and Research, Thane, India*

***Abstract*: Filtering resumes out of bulk is more difficult and time intense task for recruiters. Since corporations received resume in immense quantity and typically it usually has tangential and unnecessary data. With the assistance of machine learning, a correct and quicker system are often created which might save days for recruiters to scan every resume manually. KNN Algorithm is used to classify the resumes according to their respective categories. Our model can facilitate the recruiters to scan the resume based on the requirements they have entered. Basic workflow is that the recruiters upload a job description and bunch of resumes received to the tool. It ranks resumes on the basis of job description given.**

***Keywords*: K-NN, Machine Learning, NLP.**

1. Introduction

Recruiters must be able to properly rank resumes in order to hire the right individual at the right time. The process of deciding whether a candidate is qualified for a position based on his or her qualification, education, work-experience, and other information from their resume is known as resume screening. The importance of resumes screening is that it must provide best candidates for the particular job role. The proposed system uses NLP and Machine Learning to for ranking the resumes based on the skills. K-NN algorithm which is one of the simplest machine learning algorithms that is based on supervised learning. It is used for classification and regression but it is widely used for classification purposes-NN assumes the similarity between new data and available data. K-NN is called as the lazy learner algorithm as it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset. Using K-NN and Machine Learning to rank the resumes based on the job description, the system ranks the pdf and document format. The system determines whether a candidate is qualified for a position based on the skills listed on his or her resume. If the candidate meets the company's technical requirements, he will be shortlisted for the next round.

2. Related Work

The quick development of modern information technology in the past few years has caused an increase in number of people turning to the jobs in web development. Many businesses hire employees using online knowledge management systems or artificial intelligence (AI). These are known as e-recruitment systems, and they automate the process of receiving resumes and ranking them based on the skills listed-recruitment systems have grown rapidly in recent years, allowing Human Resources (HR) departments to reach a large number of people at a low cost. Automating the process of analyzing the applicant profiles to determine the ones that fit the positions specifications could lead to an increased efficiency.

1) This project uses automated techniques to identify, extract, and exploit information from resumes to find the most appropriate one for a given post. Our work focuses on resumes ranking.
2) The resumes fitted in the system will be retrieve using textract which is a library in python. The resumes will be in different format like .doc or .pdf, by parsing the resumes we will convert to same format which will be easy to analyze.
3) Many approaches can be applied to automate the e-recruitment process combining techniques of NLP and machine learning. In this work we have implemented an e-recruitment system that ranks the candidate resumes based on the job description selected by the recruiter.
4) The scores are evaluated by the system itself. If the candidate meets the company's technical requirements, he will be shortlisted for the next round.

3. Proposed System

The resumes provided in the system will be ranked based on the job description selected by the recruiter. The k-NN algorithm is used to find the resumes that are closest to the specified job description. To begin, we utilized an open-source tool called "gensim" to scale the job description and resumes. The package used provides a summary of the given text within the word limit. To get the job description and resumes to the same word scale, this library was used to build a summary of

*Corresponding author: anushkalad1031@gmail.com

the job description and resumes, and then K-NN was used to locate resumes that closely matched the given job description. To extract characteristics from a pre-processed resume, we employed the Tf- Idf (Term Frequency, Inverse Document Frequency) method. After transferring the cleansed data, Tf-Idf was used to extract features. Term frequency–inverse document frequency is a numerical measure meant to reflect the importance of a word in a collection or corpus of documents. The term frequency-inverse document frequency (TF-IDF) is a statistical measure that determines how relevant a word is to a document in a set of documents. This is accomplished by multiplying two metrics: the number of times a word appears in a document and the word's inverse document frequency over a set of documents. It has numerous applications, the most notable of which is automatic text analysis, and it is quite useful for scoring words in machine learning algorithms for Natural Language Processing (NLP). The TF-IDF format was developed for document search and retrieval. It works by increasing proportionally to the number of times a word appears in a document, but this is countered by the number of papers containing the word. So, terms that appear frequently in every document, such as this, what, and if, rank low even though they appear frequently, because they don't mean much to that paper in particular. However, if the word Bug appears frequently in one document but not frequently in others, it is likely that it is highly relevant. We will retrieve the resume using textract which is a library in python. Textract is a one-stop shop with a single interface for extracting text from a wide range of file types. The resumes will be in different format like .doc or .pdf, by parsing the resumes we will convert to same format which will be easy to analyze. Stop words are words that have no meaning in the context of the project. Hence the stop words will be removed the word embeddings will be made using the list from the pre-processing module. It is an algorithm which will match the company's requirement to the skills mentioned in the resume by the candidate. Based on the skills mentioned in the resumes the candidate will be asked to give an online test. The scores are evaluated by the system itself. The top candidates are shortlisted for the interview process. We have programming languages like Python and Libraries like textract, gensim, pdfminer are used.

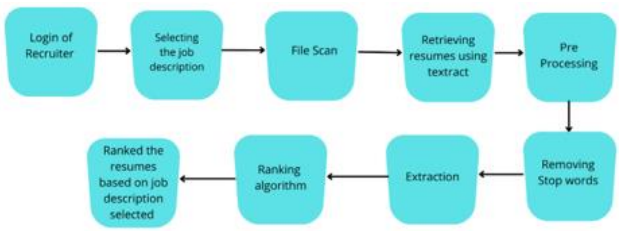

Fig. 1. Block diagram

4. Flowchart

The system provides option to the recruiter to choose the job description for particular job role. Then, the system converts the resume into text, so finding the desired key-words. Those students having the mandatory skills get elite by the unit of time for his or her positions. The system additionally displays the proportion match of the student's skills to the desired description. As shown in fig. 2.

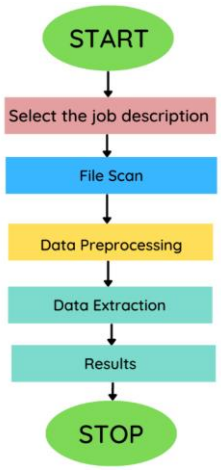

Fig. 2. Flowchart

5. K-NN Algorithm

Nearest neighbors is a supervised learning model for classification and regression analysis. But K-NN does not include any learning, i.e., there are no parameters that can be changed to improve performance, nor is there an objective function that can be optimized. This is a significant distinction from most supervised learning techniques. KNN is a rule that can be used in production to classify instances based on their neighbors. Computing neighbors and vector distances does not necessitate the usage of a class label, which is required to make a classification choice. It features three alternative algorithms for computing its nearest neighbors, and the best one is chosen to implement based on the amount of the training data. The distances of each vector from the ideal vector, which in our instance is the job description provided by the recruiter, are calculated using the brute force, KD tree, and Ball tree methods. If the distance is short, they are more suited for the job.

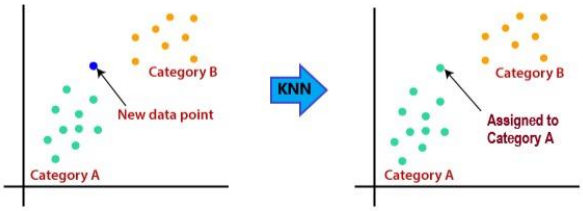

Fig. 3. K-NN algorithm

A. Algorithm

1) Start.
2) Initialize, define K.
3) Compute the distance between input sample and the training samples.
4) Sort the distance.
5) Take K nearest neighbors.
6) Apply simple majority.
7) End.

In the initial analysis, a KNN was used to rank the resumes based on their vector distance from the ideal vector. The KNN was chosen because it is regarded as one of the finest initial classification algorithms and is less complex than a CNN. The KNN is based on a search for nearest neighbors. As a result, it can be defined as: given a collection S of points in a space M and a query point q M, locate the k points in S that are closest to q. In this scenario, the query point is our vector of job descriptions. The k closest points are vectors that are eligible for that specific job description and are then sorted based on their distances. The square Euclidean distance is the distance between two vectors.

For vectors v1(w1, x1, y1, z1) and v2(w2, x2, y2, z2)

Squared Euclidean distance = $(w1-w2)^2 + (x1-x2)^2 + (y1-y2)^2 + (z1-z2)$

The best algorithm for locating nearest neighbors is determined mostly by four factors:

a) Dataset size
b) Data structure employed
c) Number of neighbors k sought for a query point
d) Total number of query points.

The KNN technique is based on feature similarity, which decides how we classify a particular data point based on how closely our out-of-sample features resemble our training set. KNN may be computationally expensive, i.e., the cost of distance calculation may be large due to the expense of forming tree data structures. It necessitates a large amount of memory. And, as a result of these characteristics, the prediction time may be slow.

B. Brute Force Approach

Only when the training dataset is N<30 in size can the brute force approach algorithm be employed. For nearest neighbors, it is the most naïve approach. This entails calculating distances between two vectors using brute force. The time required for a brute force query grows as O[DN], where N is the number of samples and D is the dimensionality of the samples, i.e., the features extracted. If either of the data structures is employed, the brute force query time will be the same. It will be unaffected by the value of k as well.

C. KD Tree Method

A type of tree-based data structure has been devised to avoid the inefficiencies of the brute-force approach. By efficiently encoding aggregate distance information for the sample, these structures attempt to reduce the number of distance computations required. The essential principle is that if position A is extremely far away from point B and point B is very close to point C, we know that points A and C are very far apart without having to explicitly compute their distance; hence, if we eliminate B, we can also eliminate C. The entire cost of a nearest neighbors search can thus be decreased to O[DNlog(N)] or less. For large datasets, this is a significant improvement over brute-force. If the value of k is similarly huge, the query time for the KD tree technique will be long.

D. Ball Tree Method

Ball trees, like KD trees, are tree-based data structures that were developed in response to the inefficiency of the KD tree technique when working with vectors of greater dimensions. Ball trees divide data in a succession of nesting hyper-spheres, whereas KD trees divide data along Cartesian axes. This makes ball tree creation more expensive than KD tree construction, but the results on well-organized data, even in extremely high dimensions, are very efficient. The query time for a ball tree is given by O[DNlog(N)]. Similar to the KD tree, the query time for the ball tree will be long if the value of k is large.

6. Advantages

- Cost-effective: As the application made is a web page so it is cost effective and there is no involvement of money because the web page will be completely open and anyone can use it without having to pay any amount of money or subscription so there.
- Time management: As there are plenty of screening and ranking happening at the same time it reduces a lot of man work or physical work and the system is automated entirely which makes the task easy and time saving for the recruitment team. Work load is reduced and a bulk of resumes are processed at the same time.
- Quality management: This way of approach makes the system easy and effective and the overall quality of work is improved for the recruitment head and the industry which is producing good results without costing a lot of money.

7. Application

- Machine Learning resume recommendation system is most valuable for high volume recruitment such as retail sales or customer service roles.
- Machine Learning resume recommendation system can handle massive volumes of data. In fact, AI requires a lot of data in order to make accurate recommendations about which candidates to move forward to the next stage.

8. Results and Discussion

Fig. 4. Homepage

The fig. 4. is the home page of our web page and it displays the name of our software.

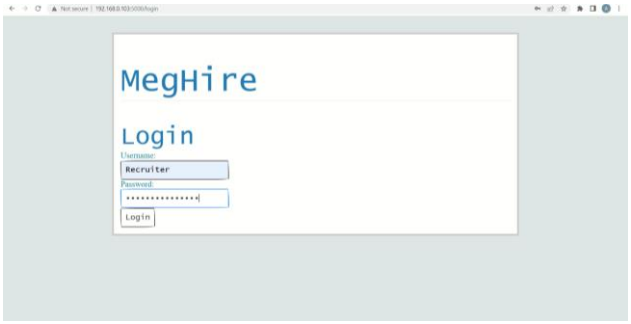

Fig. 5. Login page

Login page will take credentials from the recruiter as shown in fig. 5.

Fig. 6. Job description selection page

The fig. 6 enables the recruiter to select the job description for the job role he wants to hire the candidates.

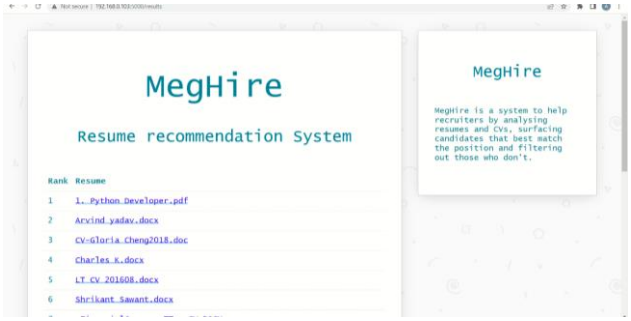

Fig. 7. Resume ranking page

The fig. 7 shows the ranking of the resumes based on the job description selected by the recruiter.

Thus, the system ranks the resumes based on the job description and displays the output.

9. Conclusion and Future Scope

The proposed system filters the resumes from the bulk of resumes and according to the skills requirements entered by the recruiter it ranks them. As a result, the proposed system will show the ranking of the all the resumes. The project can include interview scheduling the companies can shortlist candidates and send them a schedule of the interview. Offer management where the companies to which the user has applied to for a job and the responses of the companies will be displayed. The future work in this project can be done by providing database connection this will ensure smooth login for users and will allow HR personals to upload multiple resumes at one click. An application based on this software can be developed.

References

[1] J. Chen, Z Niu, H. Fu, "A Novel Knowledge Extraction Framework for Resumes Based on Text Classifier", Proceedings of the International Conference on Web-Age Information Management. Springer International Publishing, pp. 540-543, 2015.

[2] C. Hauff, G. Gousios, "Matching GitHub developer profiles to job advertisements." Proceedings of the 12th Working Conf. on Mining Software Repositories, pp. 362-366, 2015.

[3] T. Schmitt, P. Caillou, M. Sebag, "Matching Jobs and Resumes: A Deep Collaborative Filtering Task," Proc. of the 2nd Global Conf. on Artificial Intelligence, pp. 1-14, 2016.

[4] S. Mehta, R. Pimplikar, A Singh, L. R. Varshney and K. Visweswariah, "Efficient multifaceted screening of job applicants," Proceedings of the 16th International Conference on Extending Database Technology. ACM, pp. 661–671, 2013.

[5] S. Al-Otaibi and M. Ykhlef, "Job Recommendation Systems for Enhancing E-recruitment Process", in Proceedings of the International Conference on Information and Knowledge Engineering (IKE), Las Vegas Nevada, USA, pp. 433-439, 2012.

[6] A. Kmail, M. Maree, and M. Belkhatir, "Matching Sem: Online recruitment system based on multiple semantic resources," Proceedings of the 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), IEEE, pp. 2654- 2659, 2015.

[7] W. Hong, S. Zheng, H. Wang, J. Shi, "A Job Recommender System Based on User Clustering," Journal of Computers, vol. 8(8), pp. 1960-1967, 2013.

[8] V. S. Kumaran, and A. Sankar, "Towards an automated system for intelligent screening of candidates for recruitment using ontology mapping EXPERT," Int. J. Metadata Semantics. Ontologies, vol. 8(1), pp. 5664, 2013.

[9] J. Zhou and G. Salvendy, Human aspects of IT for the aged population, Cham: Springer, pp. 557-568.

[10] R. G. Vishruth, R. Sunitha, K. S. Varuna, N. Varshini, Prasad B. Honnavalli, "Resume Scanning and Emotion Recognition System based on Machine Learning Algorithms," IEEE 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA), pp. 1127–1132, 2020.

[11] K. Satheesh, A. Jahnavi, L. Iswarya, K. Ayesha, G. Bhanusekhar, K. Hanisha, "Resume Ranking Based on Job Description using SpaCy NER model," 2020 International Research Journal of Engineering and Technology.

[12] E. Faliagka, L. Iliadis, I. Karydis, M. Rigou, S. Sioutas, A. Tsakalidis, and G Tzimas, "On-line consistent ranking on e-recruitment: seeking the truth behind a well-formed CV," The Artificial Intelligence Review, 42(3), 515, 2014.

[13] R. G. Vishruth, R. Sunitha, K. S. Varuna, N. Varshini, Prasad B. Honnavalli, "Resume Scanning and Emotion Recognition System based on Machine Learning Algorithms," IEEE 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA), pp. 1127–1132, 2020.