Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 08, August-2021

Resume Classification and Ranking using KNN and Cosine Similarity

Riza Tanaz Fareed
Information Science and Engineering
R V College of Engineering
Bangalore, India

Rajath V
Information Science and Engineering
R V College of Engineering
Bangalore, India

Sharadadevi Kaganurmath
Information Science and Engineering
R V College of Engineering
Bangalore, India

***Abstract*—One of the most important and crucial task for any company is to hire an ideal candidate for their job role. Traditional hiring practises are becoming ineffective as online recruitment grows in popularity. The traditional methods normally entail a time-consuming process of manually looking through all of the individuals who have applied, examining their resumes, and then establishing a shortlist of prospects who should be interviewed. Job seeking has grown both wiser and more accessible in our technological age. Companies receive a large number of resumes/CV's, many of which are not well-structured. There has been a great deal of effort put into the job search. The process of picking a candidate based on their résumé, on the other hand, has not been completely automated. KNN Algorithm is used to classify the resumes according to their respective categories and Cosine Similarity is used to find out how close the candidate's resume is to the job description and they are ranked accordingly.**

Keywords—Resumes/CV's, job, recruitment, Cosine Similarity, KNN Algorithm, NLP

I. INTRODUCTION

Recruiters must be able to properly screen 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 CV is known as resume screening. The importance of efficient and effective resume screening is at the heart of any strong recruitment strategy. The goal of resume screening is to find the best candidates for a position. In the current system, candidates must fill out a manual form with all of their resume information, which takes a long time, and then they are dissatisfied with the position that the current system prefers based on their qualifications. Our method will work in the same way as a handshake between two people. i.e. the employer prefers the best candidate available, and the candidate chooses the best position possible based on his or her talents and abilities. Our system is a resume ranking software that uses natural language processing (NLP) and machine learning. This AI-powered resume screening programme goes beyond keywords to contextually screen resumes. Following resume screening, the software rates prospects in real time depending on the recruiter's job needs. In order to match and rate candidates in real time, the software employs natural language processing and machine learning.

II. LITERATURE SURVEY AND RELATED WORK

Senthil Kumaran et al. [1] used an intelligent tool for ontology called EXPERT mapping-based candidate screening to create an automated system for intelligent screening of prospects for recruitment, enhancing the precision with which candidates are matched to the requirements of the job.

Jagan Mohan Reddy D et al. [2] suggested joining efficient candidates before resume selection, so that the entire process can be completed in a timely and cost-effective manner. Some characteristics, such as age and salary hike, cannot be used directly for classification due to substantial variations in values that must be transformed into bins.

Frank Färber et. al [3] suggested an Automated Recommendation Approach to Personnel Recruitment Selection which began by outlining the components of a matching method based on a probabilistic automated recommendation approach, before going on to demonstrate some promising results from using the algorithm on the given data. Because there was so little training data, it was thought that the approach's full potential was yet to be realised.

Chirag Daryania et. al [4] proposed an Automated Resume Screening System which used Natural Language Processing and Similarity : Vector Space Model to match each CV with the job description and then suggested an approach which uses a vectorisation model and cosine similarity. The calculated ranking scores could then be used to find the most suitable candidates for the job position.

Momin Adnan et. al [5] proposed a system for screening candidates for recruitment using Linear SVM classifier. To search for the resumes that are closest to the specified description of the job, the model used cosine similarity, KNN and content-based Recommendation,. The model only accepts CVs in CSV format, while most CVs are in.doc,.pdf, and other formats. When using the "gensim" package to create a summary, the implicit compression of the text due to summarising may have resulted in the loss of crucial information.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 08, August-2021

Arvind Kumar Sinha et. al [6] proposed a system for Resume Screening using Natural Language Processing and Machine Learning. PROSPECT, a decision assistance tool, was employed to assist these screeners in efficiently shortlisting resumes. Prospect will search the resumes for important parts of an applicant's profile, such as experience with each expertise, skills, education, knowledge and previous job experience. To help recruiters in the work of screening, derived data is given in the form of facets. The screeners can review considerably fewer resumes to shortlist a given number of candidates by ranking candidates based on their match to the job description and using the filters supplied based on various information extracted from the resume.

V.V.Dixit et. al [7] proposed a system for Resume Sorting using Artificial Intelligence. This system sorts all resumes according to the company's requirements and sends them to the HR for further consideration. The required resume is chosen from a pool of applicants, and the others are discarded. This sorts all resumes according to the company's needs and forwards them to the appropriate HR department for further review. The required resume is chosen from a pool of candidates, with the rest being eliminated.

Dr K Sateesh et. al [8] formulated a system which helps the recruiters in selecting the resumes based on job-description in a short duration of time. It helps in an easy and efficient hiring process by extracting the requirements automatically.

The rate of unemployment is growing continuously; a large number of people are applying for many jobs, many of which are appropriate to the position being advertised. It is a significant challenge for job recruiters, who must select the most qualified profile/resume from a large pool of candidates [9]-[11].

As the profile of the candidate needed for a particular role, the way the applicants resumes are matched is very similar to a recommender system. Resnick and Varian pioneered the recommendation mechanism [12].

III. METHODOLOGY

Fig. 1 depicts the entire framework for the suggested approach.

A. Preprocessing

The resume's provided as input would be shortlisted in this procedure to remove any special or garbage characters from the resumes. All unique characters, numerals, and words with only single letters are eliminated during cleaning. After these processes, we had a clean dataset with no unique characters, numerals, or single letter words. NLTK tokenizers are used to break the dataset into tokens. Stop word removal, lemmatization and vectorization are among the preprocessing operations performed on the tokenized dataset. The data is masked in the following ways:

- Masking the strings such as \w
- Masking the escape letters like \n
- Masking all the numbers
- By substituting an empty string for all single-letter words
- Stop words are removed
- Lemmatization is performed

Removing Stop Words: Stop words such as and, the, was, and others appear very often in words and limit the process which determines prediction, thus they are removed. Filtering the Stop Words consists of the following steps:

1. The input words are tokenized into individual tokens and saved in an array.

2. Each word now corresponds to the Stop Words list in the NLTK library:

   (a) import stopwords from nltk.corpus

   (b) SW[] = set(stopwords.words('english'))

   (c) It returns 180 stop words, which may be confirmed using the (len(StopWords)) function and displayed using the print (StopWords) function.

3. When the words appear in the StopWords list, they are removed from the main sentence array.

4. Repeat the above sequence of steps until the tokenized array's last entry is not matched.

5. There are no stop words in the resultant array.

Lemmatization: Lemmatization reduces derived phrases to make entirely sure that the underlying word is accurately associated with the language. The routine phases of lemmatization are as follows:

- Convert the text corpus into a list of words.
- Make a corpus concordance, which includes all of the word list entries as they appear in the corpus.
- Based on the concordance, link the word-forms to their lemmas.

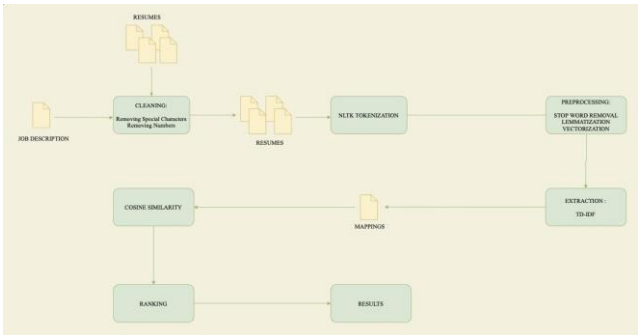

Fig. 1. Architecture of the proposed model

The extraction of features is the next phase. We used the Tf-Idf (Term Frequency, Inverse Document Frequency) to extract features from a preprocessed dataset. The cleansed data was transferred, and Tf-Idf was used to extract features. Taking the input as a numerical vector, processing a machine learning-based classification model or learning algorithms takes place. The input text with varying length was not processed by ML-based classifiers. As a result, during the preparation procedures, the texts are changed to the required equal length vector form. There are several methods for extracting

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 08, August-2021

characteristics, including tf-idf, and others. Using the scikit learn library function, we generated tf-idf for each term.

To calculate a tf-idf vector,we use TfidfVectorizer:

1) Sub-linear df is set to True to utilise a logarithmic form for frequency.
2) Min df is the minimal number of documents in which a word must appear in order to be saved.
3) The norm is set to l2 to ensure that all feature vectors have the same euclidean norm.
4) The gramme range is set to (n1, n2), with n1 equaling 1 and n2 equaling 2. It means that both unigrams and bigrams are taken into account.

The model takes a job description and a list of CVs as input and returns a list of CVs that are most similar to the job description.

Given that this is a situation of document similarity detection, we've chosen the Cosine Similarity Algorithm, in which the employer's Job Description is matched against the content of resumes in the space, and the top most similar resumes are suggested to the recruiter. The algorithm merges the cleaned resume data and job description into an unified set of data before computing the cosine similarity between both the job description and CVs.

The k-NN model is used in this model to find the resumes that are closest to the specified job description. To begin, we utilised an open source tool called "gensim" to scale the JD and CVs. The package used gives a summary of the given text within the limit of words that is provided. To get the JD and resumes to the same word scale, this library was used to build a summary of the JD and CVs, and then k-NN was used to locate CVs that closely matched the given JD.

B. Inference

The tokenized resume data and the job descriptions would be compared in this process, and the model would generate resumes that were relevant to the job description.

IV. RESULTS AND DISCUSSIONS

By displaying a resume list in order of relevance to the position, the technique ranks CVs according to their match with the job description, making it easy for recruiters. This would allow the recruiter to categorise the resumes according to the job requirements and quickly locate the CVs that best match the job description. The approach would aid the recruiter in expediting profile shortlisting while also ensuring the shortlisting process's authenticity, since they would be able to examine a large number of resumes in a short period of time, also with the proper fit, which a human would not be able to perform in near real time. This would help to make the process of recruiting individuals more efficient and successful in terms of selecting the best candidates. This would also assist the recruiter in reducing the time and resources required in locating the best candidates, making the process more cost-effective.

To evaluate the KNN model for classification of resumes, 5 metrics were used namely - Accuracy, Kappa Statistics, Precision, Recall and f1-score. The results were as follows:

TABLE I. PERFORMANCE EVALUATION

Metrics	Score
Accuracy	0.9896
Kappa Statistics	0.9890
Precision	0.9910
Recall	0.9896
f1-score	0.9895

Table II consists of the precision, recall and f1 score for each category of resumes :

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 08, August-2021

TABLE II. CLASSIFICATION REPORT

Category	Precision	Recall	F1-Score	Number of Samples
Data Science	1.00	1.00	1.00	3
HR	1.00	1.00	1.00	3
Advocate	1.00	0.80	0.89	5
Arts	1.00	1.00	1.00	9
Web Designing	1.00	1.00	1.00	6
Mechanical Engineering	0.83	1.00	0.91	5
Sales	1.00	1.00	1.00	9
Health and Fitness	1.00	1.00	1.00	7
Civil Engineer	1.00	0.91	0.95	11
Java Developer	1.00	1.00	1.00	9
Business Analyst	1.00	1.00	1.00	8
SAP Developer	0.90	1.00	0.95	9
Automation Testing	1.00	1.00	1.00	5
Electrical Engineering	1.00	1.00	1.00	9
Operations Manager	1.00	1.00	1.00	7
Python Developer	1.00	1.00	1.00	19
DevOps Engineer	1.00	1.00	1.00	3
Network Security Engineer	1.00	1.00	1.00	4
PMO	1.00	1.00	1.00	6
Database	1.00	1.00	1.00	6
Hadoop	1.00	1.00	1.00	11
ETL Developer	1.00	1.00	1.00	4
DoNet Developer	1.00	1.00	1.00	13
Blockchain	1.00	1.00	1.00	15
Testing	1.00	1.00	1.00	8

V. LIMITATIONS

There are currently a few limits to the model architecture, however they can be overcome with sufficient data to train the model with. The model's present limitations are as follows: i) The model accepts resumes in CSV format, however in the real world, resumes of candidates are typically in .docx, .pdf, or other formats. The library can read a variety of file formats and convert them to a single format that the model may utilise11 as input. ii) The implicit compression of the text due to summarization may have resulted in the loss of vital information when creating a summary using the "gensim" package. This summarising technique can be fine-tuned to ensure minimal information loss, for example, critical data elements such as candidate expertise and experience are not lost.

VI. CONCLUSION

The organisation receives a large number of applications for each employment opening. Finding the right candidate's application from a sea of resumes is a time-consuming endeavour for any company these days. The classification of a candidate's resume is a laborious, time-consuming, and resource-intensive process. To address this problem, we created an automated machine learning-based algorithm that recommends acceptable applicant resumes to HR based on the job description provided. The suggested methodology had two stages: first, it classified resumes into various groups. Second, it suggests resumes based on their resemblance to the job description. If an industry produces a high number of resumes, the proposed approach can be used to create an

Industry-specific model. By engaging domain experts such as HR professionals, a more accurate model may be built, and HR professionals' feedback can be used to iteratively enhance the model.

REFERENCES

[1] Sankar, A. (2013). “Towards an automated system for intelligent screening of candidates for recruitment using ontology mapping (EXPERT)”. International Journal of Metadata, Semantics and Ontologies, 8(1), 56. https://doi.org/10.1504/ijmso.2013.054184

[2] Jagan Mohan Reddy D, Sirisha Regella., “Recruitment Prediction using Machine Learning”, IEEE Xplore, 2020.

[3] Färber,F., Weitzel, T.,Keim, T., 2003. “An automated recommendation approach to selection in personnel recruitment”. AMCIS 2003 proceedings , 302.

[4] Chirag Daryania, Gurneet Singh Chhabrab, Harsh Patel, Indrajeet Kaur Chhabrad, Ruchi Patel., “An Automated Resume Screening System using Natural Language Processing and Similarity”. (2020). Topics In Intelligent Computing And Industry Design.

[5] Momin Adnan, Gunduka Rakesh, Juneja Afza, Rakesh Narsayya Godavari, Gunduka and Zainul Abideen Mohd Sadiq Naseem., “Resume Ranking using NLP and Machine Learning”, (2016b). Institutional Repository of the Anjuman-I-Islam’s Kalsekar Technical Campus. https://core.ac.uk/display/55305289

[6] Arvind Kumar Sinha, Ashwani Kumar, Md. Amir Khusru Akhtar., “Resume Screening using Natural Language Processing and Machine
Learning A Systematic Review”, (2019)., Machine Learning and Information Processing : Proceedings of ICMLIP.

[7] V. V. Dixit , Trisha Patel , Nidhi Deshpande , Kamini Sonawane, “Resume Sorting using Artificial Intelligence”. (2019). International Journalof Research in Engineering, Science and Management Volume-2, Issue-4.

[8] Dr.K.Satheesh, A.Jahnavi, L Aishwarya, K.Ayesha, G Bhanu Shekhar, K.Hanisha, “Resume Ranking based on Job Description using SpaCy NER model”. (2020). International Research Journal of Engineering and Technology.

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

[10] Zhang, L.,Fei, W. ,Wang ,L.,2015.Pj matching model of knowledge workers.Procedia Computer Science 60,1128–1137.

[11] Roy, P.K., Singh, J.P., Baabdullah, A.M., Kizgin, H., Rana, N.P., 2018a. Identifying reputation collectors in community question answering (cqa) sites: Exploring the dark side of social media. International Journal of Information Management 42, 25–35.

[12] Resnick,P.,Varian,H.R.,1997.Recommender Systems.Communications of the ACM40, 56–59.