IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⪤ Impact Factor 7.39 ⪤ Vol. 11, Issue 5, May 2022

DOI: 10.17148/IJARCCE.2022.115166

RESUME SCREENING USING TF-IDF

Chandraghandi S[1], Shilpa S[2], Anamika P[3], Kamalakkannan R[4], Santhoshsivan N[5]

Assistant Professor, Department of Computer Science And Engineering, JCT College Of Engineering And Technology[1]

Computer Science And Engineering, JCT College Of Engineering And Technology, Coimbatore, India[2-5]

Abstract: The goal of resume screening is to find the best candidates for a position. In order to match and rate candidates in real-time, the software must employ natural language processing and machine learning. Our system is a resume ranking software that uses natural language processing (NLP) and machine learning. Input would be resumes and job descriptions, output would be a highly ranked candidate's resume. Output results are acquired instantly in real-time. We will be using Mong for string matching, Cosine Similarity, TF-IDF. The existing systems are simple and effective but are not robust in terms of accuracy, efficiency, and processing. Through the analysis of the works of literature on existing methods, it can be found that these are traditional systems that could lead to inaccurate assumptions and loss of human potential. We propose a web application that aims to order the resumes, by intelligently reading job descriptions as input and comparing the resumes which fall into the category of given Job Descriptions. In order to match and rate candidates in real-time, the software employs natural language processing It provides a ranking after filtering and recommends the better resume for a given textual job description. The Advantages of the proposed system are Secured, Interpretability, High accuracy, Lightweight model & fast processing. Real-time use cases. It could be used in MNC's where multiple resumes must be screened every single day for multiple jobs, government, and administrative offices.

INTRODUCTION

Recruitment is a 200-billion-dollar business. It deals with hiring the best-fit candidates having the relevant skills for a given job profile from an immensely large pool of candidates. If a company has any job opening for a position, scores of candidates mail their resumes to the company to apply for that opening. In the hiring process, the first task for any recruiter is to screen the resumes of all the job applicants. Any company having a job opening for a particular position will have their mail inboxes bombarded with thousands of emails from aspiring job applicants every single day. Selecting the prospective candidates for that job position from a large pool of candidates for any recruiter is very tedious. It is an extremely daunting task for the recruiters of a company to manually go through thousands of resumes and select the most appropriate candidates for the job. Out of those thousands of resumes submitted to the company for the given job posting, about 75% of them do not showcase the relevant skills that are required for the job profile.

Due to this, the recruiters quite often find it really arduous to narrow down the most appropriate candidates from a large applicant pool. In recent years, there have been more than 50,000 e-recruitment sites have been developed. The developers of these online recruitment sites have used various approaches to identify the prospective candidates for a given job profile of a company. Some of these, have managed to employ classification techniques that will classify the candidate resumes into various categories for every job posting given by every company. In these approaches, every candidate's resume is tried to match with every given job posting on the recruitment site. The aim of these recruitment sites is to throw up the results to the candidate to which they are the best fit. The techniques used by these sites have resulted in high accuracy and precision, but one of the major disadvantages is the factor of time complexity. If every candidate's resume is matched with every other job posting given on the online recruitment site, the time complexity for acquiring the results is very high.

The world of Artificial Intelligence [AI] and Machine Learning [ML] has grown significantly in recent times. The availability of large amounts of data brought about by advancements in technology which has made the internet cheap and accessible to previously inaccessible regions of the world has contributed to a great increase in the performance of the ML models in recent times. Software companies around the world exploit the advances in ML to drive automation and increase their productivity in areas that relied mostly on manual human labor. The approach discussed in this project is by using machine learning to train the dataset for a particular type of job position. It is also proposed to use section-based segmentation for data extraction using Natural language Processing (NLP). In order to improve the time efficiency of the web application, the candidate's resume will only be matched to those job openings where they are interested in and have applied to which will, in turn, reduce the time complexity. Besides, the results of the resume matching of all the candidates who have applied for the job opening will be visible only to the recruiter of that particular company. This is

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⵘ Impact Factor 7.39 ⵘ Vol. 11, Issue 5, May 2022

DOI: 10.17148/IJARCCE.2022.115166

done with the aim to aid the recruiters of any company from the long and tedious task of viewing and analyzing thousands of candidates' resumes. In this intelligent-based approach, they will be given the option to view the candidate's resume as well as they will get the results of the best candidates suitable for the required job position.

NLP

Natural Language Processing, or NLP for short, is broadly defined as the automatic manipulation of natural language, like speech and text, by software.

Step #1: Sentence Segmentation

Breaking the piece of text in various sentences.

Step #2: Word Tokenization

Breaking the sentence into individual words known as tokens. We can tokenize them whenever we encounter a space, we can train a model in that way. Even punctuations are considered as individual tokens as they have some meaning.

Step #3: Predicting Parts of Speech for each token

Predicting whether the word is a noun, verb, adjective, adverb, pronoun, etc. This will help to understand what the sentence is talking about. This can be achieved by feeding the tokens(and the words around it) to a pre-trained part-of-speech classification model. This model was fed a lot of English words with various parts of speech tagged to them so that it classifies the similar words it encounters in future in various parts of speech. Again, the models don't really understand the 'sense' of the words, it just classifies them on the basis of its previous experience. It's pure statistics.

Step #4: Lemmatization Feeding the model with the root word.

For example – There's a Buffalo grazing in the field. There are Buffaloes grazing in the field. Here, both Buffalo and Buffaloes mean the same. But, the computer can confuse it as two different terms as it doesn't know anything. So we have to teach the computer that both terms mean the same. We have to tell a computer that both sentences are talking about the same concept. So we need to find out the most basic form or root form or lemma of the word and feed it to the model accordingly. In a similar fashion, we can use it for verbs too. 'Play' and 'Playing' should be considered as same.

Step #5: Identifying stop words

There are various words in the English language that are used very frequently like 'a', 'and', 'the' etc. These words make a lot of noise while doing statistical analysis. We can take these words out. Some NLP pipelines will categorize these words as stop words, they will be filtered out while doing some statistical analysis. Definitely, they are needed to understand the dependency between various tokens to get the exact sense of the sentence. The list of stop words varies and depends on what kind of output are you expecting.

Step 6.1: Dependency Parsing

This means finding out the relationship between the words in the sentence and how they are related to each other. We create a parse tree in dependency parsing, with root as the main verb in the sentence. If we talk about the first sentence in our example, then 'is' is the main verb and it will be the root of the parse tree. We can construct a parse tree of every sentence with one root word(main verb) associated with it. We can also identify the kind of relationship that exists between the two words. In our example, 'San Pedro' is the subject and 'island' is the attribute. Thus, the relationship between 'San Pedro' and 'is', and 'island' and 'is' can be established. Just like we trained a Machine Learning model to identify various parts of speech, we can train a model to identify the dependency between words by feeding many words. It's a complex task though. In 2016, Google released a new dependency parser Parsey McParseface which used a deep learning approach.

Step 6.2: Finding Noun Phrases

We can group the words that represent the same idea. For example – It is the second-largest town in the Belize District and largest in the Belize Rural South constituency. Here, tokens 'second', 'largest' and 'town' can be grouped together as they together represent the same thing 'Belize'. We can use the output of dependency parsing to combine such words.

Whether to do this step or not completely depends on the end goal, but it's always quick to do this if we don't want much information about which words are adjectives, rather focus on other important details.

Step #7: Named Entity Recognition(NER)

San Pedro is a town on the southern part of the island of Ambergris Caye in the 2. Belize District of the nation of Belize, in Central America. Here, the NER maps the words with the real-world places. The places that actually exist in the physical world. We can automatically extract the real-world places present in the document using NLP. If the above sentence is the input, NER will map it like this way: San Pedro - Geographic Entity Ambergris Caye - Geographic Entity Belize - Geographic Entity Central America - Geographic Entity NER systems look for how a word is placed in a sentence and make use of other statistical models to identify what kind of word actually it is. For example – 'Washington' can be a geographical location as well as the last name of any person. A good NER system can identify this. Kinds of objects that a typical NER system can tag: People's names. Company names. Geographical locations Product names. Date and time. Amount of money.

Step #8: Coreference Resolution

San Pedro is a town on the southern part of the island of Ambergris Caye in the Belize District of the nation of Belize, in Central America. According to 2015 mid-year estimates, the town has a population of about 16, 444. It is the second-largest town in the Belize District and largest in the Belize Rural South constituency. Here, we know that 'it' in sentence 6 stands for San Pedro, but for a computer, it isn't possible to understand that both the tokens are the same because it treats both the sentences as two different things while it's processing them. Pronouns are used with a high frequency in English literature and it becomes difficult for a computer to understand that both things are the same.

Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) is an approach to analyze the data using visual techniques. It is used to discover trends, patterns, or ti check assumptions with the help of statistical summary and graphical representations. EDA is the process of investigating the dataset to discover patterns, and anomalies (outliers), and form hypotheses based on our understanding of the dataset. EDA involves generating summary statistics for numerical data in the dataset and creating various graphical representations to understand the data better. EDA assists Data science professionals in various ways: -

1 Getting a better understanding of data
2 Identifying various data patterns
3 Getting a better understanding of the problem statement

EDA is primarily used to see what data can reveal beyond the formal modeling or hypothesis testing task and provides a better understanding of data set variables and the relationships between them. It can also help determine if the statistical techniques you are considering for data analysis are appropriate. The main purpose of EDA is to help look at data before making any assumptions. It can help identify obvious errors, as well as better understand patterns within the data, detect outliers or anomalous events, find interesting relations among the variables.

Data cleaning

This is the most important step in EDA involving removing duplicate rows/columns, filling the void entries with values like mean/median of the data, dropping various values, removing null entries

Data visualization

Data Visualization is the process of analyzing data in the form of graphs or maps, making it a lot easier to understand the trends or patterns in the data. There are various types of visualizations

- Univariate analysis: This type of data consists of only one variable. The analysis of univariate data is thus the simplest form of analysis since the information deals with only one quantity that changes. It does not deal with causes or relationships and the main purpose of the analysis is to describe the data and find patterns that exist within it.

- Bi-Variate analysis: This type of data involves two different variables. The analysis of this type of data deals with causes and relationships and the analysis is done to find out the relationship among the two variables.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⚹ Impact Factor 7.39 ⚹ Vol. 11, Issue 5, May 2022

DOI: 10.17148/IJARCCE.2022.115166

- Multi-Variate analysis: When the data involves three or more variables, it is categorized under multivariate.

Some of the most common data science tools used to create an EDA include:

- **Python:** An interpreted, object-oriented programming language with dynamic semantics. Its high-level, built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for rapid application development, as well as for use as a scripting or glue language to connect existing components together. Python and EDA can be used together to identify missing values in a data set, which is important so you can decide how to handle missing values for machine learning.

- **R:** An open-source programming language and free software environment for statistical computing and graphics supported by the R Foundation for Statistical Computing. The R language is widely used among statisticians in data science in developing statistical observations and data analysis.

TF-IDF

It is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. It is often used as a weighting factor in searches of information retrieval, text mining, and user modeling. The tf–idf value increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word, which helps to adjust for the fact that some words appear more frequently in general. tf–idf is one of the most popular term-weighting schemes today.

TF-IDF (term frequency-inverse document frequency) is a statistical measure that evaluates how relevant a word is to a document in a collection of documents.

This is done by multiplying two metrics: how many times a word appears in a document, and the inverse document frequency of the word across a set of documents.

It has many uses, most importantly in automated text analysis, and is very useful for scoring words in machine learning algorithms for Natural Language Processing (NLP).

TF-IDF was invented for document search and information retrieval. It works by increasing proportionally to the number of times a word appears in a document but is offset by the number of documents that contain the word. So, words that are common in every document, such as this, what, and if, rank low even though they may appear many times since they don't mean much to that document in particular.

However, if the word *Bug* appears many times in a document, while not appearing many times in others, it probably means that it's very relevant. For example, if what we're doing is trying to find out which topics some NPS responses belong to, the word *Bug* would probably end up being tied to the topic Reliability, since most responses containing that word would be about that topic.

How is TF-IDF calculated?

TF-IDF for a word in a document is calculated by multiplying two different metrics:

- The **term frequency** of a word in a document. There are several ways of calculating this frequency, with the simplest being a raw count of instances a word appears in a document. Then, there are ways to adjust the frequency, by the length of a document, or by the raw frequency of the most frequent word in a document.

- The **inverse document frequency** of the word across a set of documents. This means, how common or rare a word is in the entire document set. The closer it is to 0, the more common a word is. This metric can be calculated by taking the total number of documents, dividing it by the number of documents that contain a word, and calculating the logarithm.

- So, if the word is very common and appears in many documents, this number will approach 0. Otherwise, it will approach 1.

Multiplying these two numbers results in the TF-IDF score of a word in a document. The higher the score, the more relevant that word is in that particular document.

Why is TF-IDF used in Machine Learning?

Machine learning with natural language is faced with one major hurdle – its algorithms usually deal with numbers, and natural language is, well, text. So, we need to transform that text into numbers, otherwise known as text vectorization. It's a fundamental step in the process of machine learning for analyzing data, and different vectorization algorithms will drastically affect end results, so you need to choose one that will deliver the results you're hoping for.

Once you've transformed words into numbers, in a way that's machine learning algorithms can understand, the TF-IDF score can be fed to algorithms such as Naive Bayes and Support Vector Machines, greatly improving the results of more basic methods like word counts.

Why does this work? Simply put, a word vector represents a document as a list of numbers, with one for each possible word of the corpus. Vectorizing a document is taking the text and creating one of these vectors, and the numbers of the vectors somehow represent the content of the text. TF-IDF enables us to gives us a way to associate each word in a document with a number that represents how relevant each word is in that document. Then, documents with similar, relevant words will have similar vectors, which is what we are looking for in a machine learning algorithm.

Applications of TF-IDF

Determining how relevant a word is to a document, or TD-IDF, is useful in many ways, for example:

- **Information retrieval**

TF-IDF was invented for document search and can be used to deliver results that are most relevant to what you're searching for. Imagine you have a search engine and somebody looks for LeBron. The results will be displayed in order of relevance. That's to say the most relevant sports articles will be ranked higher because TF-IDF gives the word LeBron a higher score. It's likely that every search engine you have ever encountered uses TF-IDF scores in its algorithm.

- **Keyword Extraction**

TF-IDF is also useful for extracting keywords from the text. How? The highest scoring words of a document are the most relevant to that document, and therefore they can be considered *keywords* for that document. Pretty straightforward.

Cosine Similarity

Cosine similarity measures the similarity between two vectors of an inner product space. It is measured by the cosine of the angle between two vectors and determines whether two vectors are pointing in roughly the same direction. It is often used to measure document similarity in text analysis. A document can be represented by thousands of attributes, each recording the frequency of a particular word (such as a keyword) or phrase in the document. Thus, each document is an object represented by what is called a *term-frequency vector*. Term-frequency vectors are typically very long and sparse (i.e., they have many 0 values). Applications using such structures include information retrieval, text document clustering, biological taxonomy, and gene feature mapping. The traditional distance measures that we have studied in this chapter do not work well for such sparse numeric data. For example, two term-frequency vectors may have many 0 values in common, meaning that the corresponding documents do not share many words, but this does not make them similar. We need a measure that will focus on the words that the two documents do have in common, and the occurrence frequency of such words. In other words, we need a measure for numeric data that ignores zero matches.

Cosine similarity is a measure of similarity that can be used to compare documents or, say, give a ranking of documents with respect to a given vector of query words. 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 topmost similar resumes are suggested to the recruiter.

Overlap coefficient

The overlap coefficient is a similarity measure that measures the overlap between two finite sets. It is related to the Jaccard index and is defined as the size of the intersection divided by the smaller of the size of the two sets:

$$O(A,B) = \frac{|A \cap B|}{\min(|A|,|B|)}$$

DOI: 10.17148/IJARCCE.2022.115166

If set A is a subset of B or the converse then the overlap coefficient is equal to 1.

LITERATURE SURVEY

Recruitment of appropriate people for certain positions is critical for any company or organization. Manually screening to select appropriate candidates from large amounts of resumes can be exhausted and time-consuming. 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. The authors in [1] 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.

In [2], the objective was to create a résumé shortlisting system using natural language processing. In [4], The proposed system, JARO accelerates the interview process towards an unbiased decision-making process by proposing a chatbot that would conduct interviews by analyzing the candidate’s Curriculum Vitae (CV), based on which, it then prepares a set of questions to be asked to the candidate. The system will consist of features like resume analysis and automatic interview processes.

The manual process of screening resumes could stymie the team's efforts to locate the right individual at the right moment. The laborious screening may be greatly aided by an automated technique for screening and ranking applicants. In [5], the top applicants might be rated using content-based suggestion, which uses cosine similarity to find the curriculum vitae that are the most comparable to the job description supplied and KNN algorithm is used to pick and rank Curriculum Vitaes (CV) based on job descriptions in huge quantities. Experimental results indicate the performance of the proposed system as an average text parsing accuracy of 85% and a ranking accuracy of 92%. Pradeep Kumar Roy and Sarabhjit Singh [6] suggested that an automated way of “Resume Classification and Matching” could really ease the tedious process of fair screening and shortlisting, it would certainly expedite the candidate selection and decisionmaking process. This system could work with a large number of resumes for first classifying the right categories using different classifier, once classification has been done then as per the job description, top candidates could be ranked using Content-based Recommendation, using cosine similarity and by using k-NN to identify the CVs that are nearest to the provided job description.

The research work of authors in [7] presented a hybrid approach that employs conceptual-based classification of resumes and job postings and automatically ranks candidate resumes (that fall under each category) to their corresponding job offers. In this context, the exploit an integrated knowledge base for carrying out the classification task and experimentally demonstrate - using a real-world recruitment dataset- achieving promising precision results compared to conventional machine learning-based resume classification approaches. To address the issues associated with the previously highlighted techniques, what has been presented is a hybrid approach that employs conceptual-based classification of resumes and job postings and automatically ranks candidate resumes (that fall under each occupational category) to their corresponding job postings. The study work in [8] was conducted among 115 HR professionals at various IT sectors in Delhi/NCR region. A multiple regression method was used to test the hypothesis and confirmed a positive relationship between these two factors establishing about the increased use of AI at work results in better HR functional performance. However, AI has a significant relationship with innovativeness and also with the ease of use which reflects AI affects HR with innovations and ease of use.

The research work in [9] focuses on extracting data from resumes and performing the required analysis on the data to convert it into useful information for the recruiters. Thus, the Resume Parser would help the recruiters to select the best relevant candidates in a minimal amount of time, consequently saving their time and effort. The authors in [10] developed a method for automatic RQA. Since there is also no public dataset for model training and evaluation, we build a dataset for RQA by collecting around 10K resumes, which are provided by a private resume management company. By investigating the dataset, we identify some factors or features that could be useful to discriminate good resumes from bad ones, e.g., the consistency between different parts of a resume. Then a neural-network model is designed to predict the quality of each resume, where some text processing techniques are incorporated. To deal with the label deficiency issue in the dataset, they proposed several variants of the model by either utilizing the pair/triplet-based loss or introducing some semi-supervised learning technique to make use of the abundant unlabeled data.

[11] presents an overview of fairness definitions, methods, and tools as they relate to recruitment and establishes ethical considerations in the use of machine learning in the hiring space. Considering Deep Learning (DL) method recognize artificial Neural Network (NN) to nonlinear process, NLP tools become increasingly accurate and efficient that begin a debacle. Multi-Layer Neural Network obtaining the importance of the NLP for its capability including standard speed and resolute output. Hierarchical designs of data operate recurring processing layers to learn and with this arrangement of DL methods manage several practices. In [12], this resumed striving to reach a review of the tools and the necessary methodology to present a clear understanding of the association of NLP and DL for truly understanding in the training. Efficiency and execution both are improved in NLP by Part of speech tagging (POST), Morphological Analysis, Named Entity Recognition (NER), Semantic Role Labeling (SRL), Syntactic Parsing, and Coreference resolution. Artificial Neural Networks (ANN), Time Delay Neural Networks (TDNN), Recurrent Neural Network (RNN), Convolution Neural Networks (CNN), and Long-Short-Term-Memory (LSTM) dealings among Dense Vector (DV), Windows Approach (WA), and Multitask learning (MTL) as a characteristic of Deep Learning.

[13] proposes a system for resume parsing using deep learning models such as the convolutional neural network (CNN), Bi-LSTM (Bidirectional Long Short-Term Memory), and Conditional Random Field (CRF). CNN Model is used for classifying different segments in a resume. CRF and Bi-LSTM-CNN models were used for sequence labeling in order to tag different entities. The pre-trained Glove model is used for word embedding. The proposed system could classify a resume into three segments and extract 23 fields. At an industry sector level such as Information Technology or across such different industry sectors (such as retail, insurance, health care), mining and recommending the most relevant career paths for a user is still an unsolved research challenge. Towards addressing this problem, [14] proposes a system that leverages the notion of skills to construct skill graphs that can form the basis for career path recommendations. Skills are more amenable for career path standardizations across the organizations. This system ingests a user’s profile (in a pdf, word format, or other public and shared data sources) and leverages an Open IE pipeline to extract education and experiences. Subsequently, the extracted entities are mapped as specific skills that are expressed in the form of a novel unified skill graph.

[16] presents a survey of e-recruiting process and existing recommendation approaches for building personalized recommender systems for candidates/job matching.

Every job advertisement receives a significant number of applications, many of which are related to the listed position. Because they must identify the most qualified profile/resume from a broad pool of prospects, job recruiters encounter substantial challenges. Because it is the profile of the applicant recommended for a specific role, the method of matching the candidate CV with the job description is similar to a recommender system. Otaibi et al. [16] investigated the utilization of employment referral services in-depth and discussed the measures that must be taken during the hiring process for any organization and also described how the organization benefits from the e-recruitment portal, what candidate criteria might lead to the selection, and a variety of other essential recruiting approaches. To produce employment suggestions, Malinowski et al. used an Expectation-Maximization (EM) algorithm that took into account both the candidate's resume and the employer's job description. Golec et al. [17] suggested a fuzzy-based method for determining candidate relevance to the job description

One of more useful and efficient retrieval model [19] is the vector space model. It uses term weighting tf-idf for assigning a score to a query/document pair, in order to rank the documents. This method in [19] is based on the bag-of-words model and does not capture position of terms in document and semantics. Keeping this approach, here we propose a new ranking measure that combines the vector space measure denoted tf-idf and new factor deducted from association rules technique based on dominance relations. This approach will help the user to get the most relevant documents at the beginning. The experiments on TREC collection show that the proposed method is giving better results compared to the baselines and the semantic ranking ESA.

Using NLP(Natural Language Processing) and ML(Machine Learning) to rank the resumes according to the given constraint, this intelligent system ranks the resume of any format according to the given constraints or the following requirement provided by the client company. We will basically take the bulk of input resume from the client company and that client company will also provide the requirement and the constraints according to which the resume should be ranked by our system. Besides the information provided by the resume, we are going to read the candidate’s social profiles (like LinkedIn, GitHub, etc.) which will give us more genuine information about that candidate.

METHODOLOGY AND SYSTEM DESIGN

Recruitment in the IT sector has been on the rise in recent times. Software companies are on the hunt to recruit raw talent right from the colleges through job fairs. The process of allotment of projects to the new recruits is a manual affair, usually carried out by the Human Resources department of the organization. This process of project allotment to the new recruits is a costly affair for the organization as it relies mostly on human effort. In recent times, software companies around the world are leveraging the advances in machine learning and Artificial intelligence, in general, to automate routine tasks in the enterprise to increase productivity. The existing system is a traditional Machine Learning based system that gives lower rates of accuracy. It has lower efficiency and gives lower inaccurate results. This system may lead to loss of human potential.

The major objective of our system is to take the current resume ranking system to another level and makes it more flexible for both the entity.

1) Candidates, who have been hired.

2) Client company, who is hiring the candidates

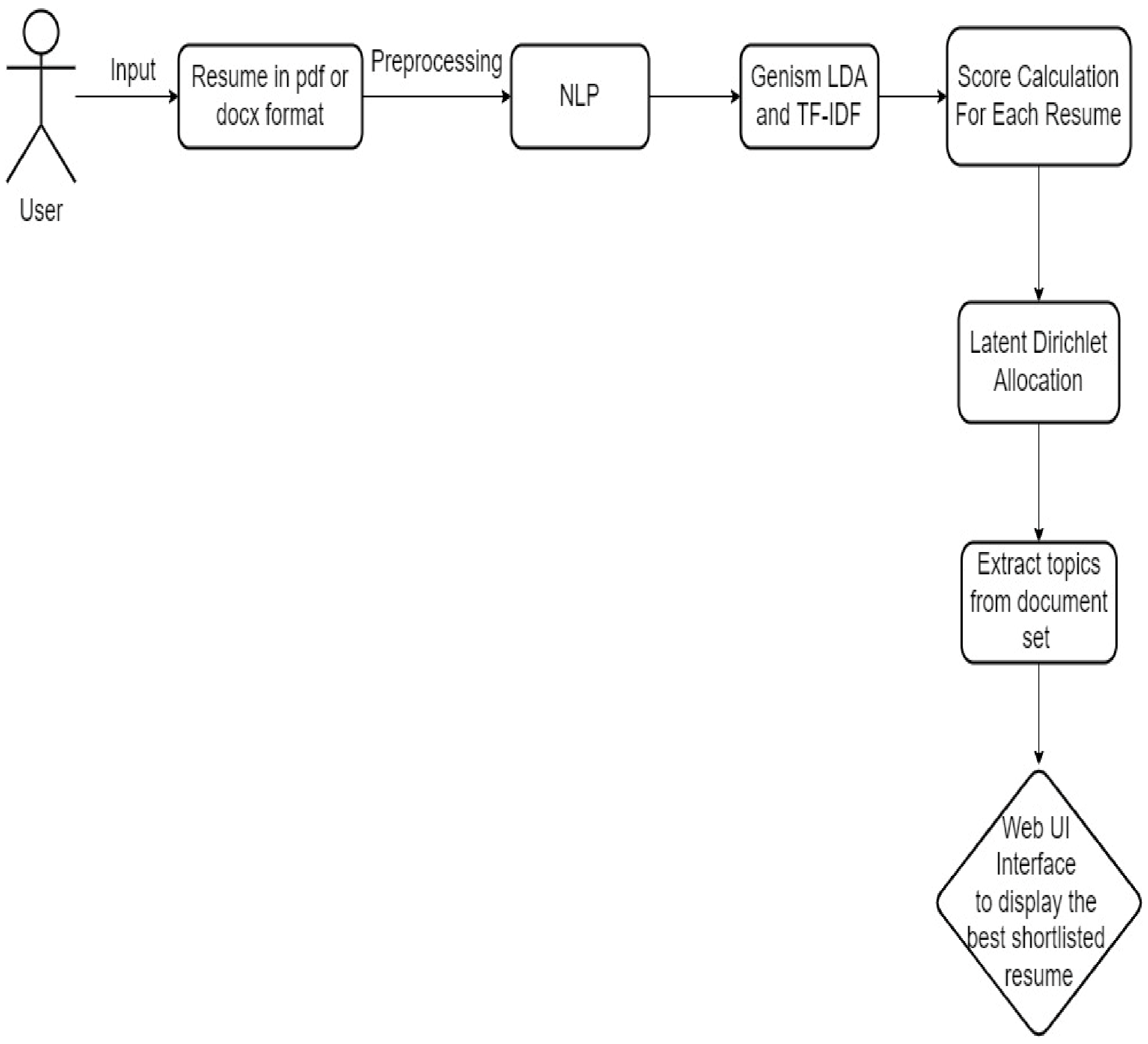
The report also aims to propose an algorithm that provides a list of applicants with appropriate experience and then presents the high points of each selected resume, unlike the conventional way of applying filters and manually scanning resumes to identify the most qualified candidates for a certain vacancy. The model is also designed with an aim to automate tasks that the human resources can do and recognize and identify human faces

We propose a web application that aims to order the resumes, by intelligently reading job descriptions as input and comparing the resumes which fall into the category of given Job Descriptions. It provides a ranking after filtering and recommends the better resume for a given textual job description. Our system is a resume ranking software that uses natural language processing (NLP) and machine learning. This AI-powered resume screening program 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. Unlike generic processes, this app utilizes Mong for string matching, Cosine Similarity, Overlapping coefficient Natural Language.

Our work takes a different approach as it focuses mainly on the content of the resumes where we perform the extraction of skills and related parameters to match candidates with the job descriptions. The interactive web application will allow the job applicants to submit their resumes and apply for 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. This ranking is made visible only to the company recruiter who is interested to select the best candidates from a large pool of candidates. This is done with the aim to aid the recruiters of any company from the long and tedious task of viewing and analyzing thousands of candidates' resumes. The calculated ranking scores can then be utilized to determine best-fitting candidates for that particular job opening. Since the dynamic model leverages NLP, it gives the output instantly. While going through all these pipelines, it will score each resume and give out accurate output with higher efficiency, precision, and accuracy.

Input is the job applicant's resume which will be further preprocessed 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'll have a clean dataset with no unique characters, numerals, or single letter words. The web application returns a highly ranked candidate's resume that is most similar to 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 application makes use of Natural Language Processing (NLP) which helps in data training and feature extraction of the text data. NLP is an analysis of natural languages so that computers can understand them. Natural language, whether spoken, written, or typed, is the most natural means of communication between humans, and the mode of expression of choice for most of the documents they produce. Using NLP methods, semi-structured text data is converted to a structured format with required extracted features. 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 topmost similar resumes are suggested to the recruiter.

DOI: 10.17148/IJARCCE.2022.115166


Input
Resume in pdf or docx format
Preprocessing
NLP
Genism LDA and TF-IDF
Score Calculation For Each Resume
User
Latent Dirichlet Allocation
Extract topics from document set
Web UI Interface to display the best shortlisted resume

SYSTEM ARCHITECTURE

Natural Language Processing [NLP] and EDA

This application makes use of Natural Language Processing (NLP) which helps in data training and feature extraction of the text data. NLP is an analysis of natural languages so that computers can understand them. Natural language, whether spoken, written, or typed, is the most natural means of communication between humans, and the mode of expression of choice for most of the documents they produce. Using NLP methods, semi-structured text data is converted to a structured format with required extracted features. The profiles of the new recruits are fed as input to the NLPP by the HR team of the organization. The module is responsible for eliminating the unnecessary information from the resume and providing only the required data in the form of tokens which could aid in the process of allotment of projects to the classification module. Resume collection is being performed and folder Structure creation is being done. Data Cleaning such as removing clutter and unnecessary punctuation would be taken off, Feature Engineering would be performed for enhancement. This includes removing stop words, punctuation, and stemming. This process will construct a graph with sentences as the vertices. Importing necessary libraries is performed. This library creates a summary of the supplied information within the word limit. With the help of Natural language processing techniques, the application which extracts the names of people, places, and other entities from text, the main goal is to get a reduced version of it that retains the most important information.

There are several different methods of statistical analysis and data visualization techniques in the dataset that can be used to explore the data to identify the appropriate data cleaning operations to be conducted. Also known as E.D.A, exploratory data analysis is a very important phase in researching and investigating various data sets and summarizing their significant characteristics. The application of EDA can assist in uncovering hidden patterns in datasets and how important is it in data science. Exploratory data analysis (EDA) is often a necessary task in uncovering hidden patterns, detecting outliers, and identifying important variables and any anomalies in data. The data is cleaned and pre-processed at this stage, where missing and null value records are dropped. The main purpose of preprocessing is to identify and drop or substitute the missing values in the dataset which occupy a very small part of the whole data, to ensure an accurate result.

Term frequency-inverse document frequency (TF-IDF)

The TF-IDF method is the most frequently used method for determining word frequencies. This is an abbreviation for "Term Frequency – Inverse Document" Frequency, one of the criteria used to determine the final score for each word. TF-IDF is word frequency scores that aim to emphasize phrases that are more interesting, e.g., common in a text but not across texts, without delving into the arithmetic. The TF-IDF Vectorizer tokenizes texts, learns vocabulary, inverts frequency weightings, and allows encoding new ones. At this stage, a dynamic Script for the Tf-Idf approach is written. Term frequency-inverse document frequency is a numerical statistic that is intended to reflect how important a word is to a document in a collection. TF-IDF is word frequency scores that aim to emphasize phrases that are more interesting, e.g., common in a text but not across texts, without delving into the arithmetic. The TF-IDF Vectorizer tokenizes texts, learns vocabulary, inverts frequency weightings, and allows encoding new ones.

$$TF - IDF(t,d) = TF(t,d) * IDF(t,d) \quad (1)$$

$$\text{TF}\ (\text{t},\ \text{d}) = \frac{\text{freq}\ (\text{t},\ \text{d})}{\sum^{\text{freq}}\ (\text{ti},\ \text{d})} \quad (2)$$

$$\text{IDF}\ (\text{t}) = \log\left(\frac{N}{count(t)}\right) \quad (3)$$

Where freq (t, d) is the count of the instances of the term t in document d.

TF (t, d) is the proportion of the count of term t in document d

N is the number of distinct terms in document d.

It provides information on a word frequency in the documents. Higher the TF- IDF score of a term which is computed using the above equations represents more relevance in a document. It provides information on a word frequency in the documents.

Higher the TF- IDF score of a term which is computed using the above equations represents more relevance in a document. In our system, we modeled the CVs and JD into a vector space. This is accomplished by compiling a glossary of terms found in the papers and converting them. Each phrase corresponds to a vector space dimension. Using the Count Vectorizer and the TF- IDF matrix, we generated the TF- IDF matrix for the CVs and the job query.

Latent Dirichlet Allocation (LDA)

A tool and technique for Topic Modeling, Latent Dirichlet Allocation (LDA) classifies or categorizes the text into a document and the words per topic, these are modeled based on the Dirichlet distributions and processes. Latent Dirichlet Allocation has been used in the application for the following functions-

- Discovering the hidden themes in the data.
- Classifying the data into the discovered themes.
- Using the classification to organize/summarize/search the documents.

The application then deals with the calculation of the score for a candidate's resume according to the job posting they have applied for. According to the score each candidate's resume receives, a rank list will be made with the candidate receiving a higher score placed higher as compared to the candidate receiving a lower score. 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. Customized options for job description in our web application is shown. Slider options are present in our app for a better user experience. The web application would also have a Bar Chart that shows the stats. The Resume Screening System replaces ineffective manual screening, ensuring that no candidate is overlooked

CONCLUSION

The results from the model are encouraging. The resume classifier application is successful in automating the manual task of project allocation to the new recruits of the organization based on the interests, work experience, and expertise mentioned by the candidate in the profile. The Resume Screening System replaces ineffective manual screening, ensuring that no candidate is overlooked. The need for efficient and effective resume screening is at the heart of every excellent recruitment strategy. The system will be able to accept or reject a job applicant based on two factors: the company's requirements must match the skills listed in the applicant's resume, and the test evaluation will be based on the applicant's skills, ensuring that the resumes uploaded by the applicant are genuine and the applicant is truly knowledgeable about the skills. Using NLP(Natural Language Processing) and ML(Machine Learning) to rank the resumes according to the given constraint, this intelligent system ranks the resume of any format according to the given constraints or the following requirement provided by the client company. We will basically take the bulk of input resume from the client company and that client company will also provide the requirement and the constraints according to which the resume should be ranked by our system. Besides the information provided by the resume, we are going to read the candidate's social profiles (like LinkedIn, GitHub, etc.) which will give us more genuine information about that candidate. The application automates the task of project allocation, thereby eliminating the tedious and redundant affair of opening and analyzing the resumes manually by the HR team of the organization.

REFERENCES

[1] Rajath V; Riza Tanaz Fareed; Sharadadevi Kaganurmath," Resume Classification And Ranking Using KNN And Cosine Similarity",international JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY, Volume 10, Issue 08, AUGUST 2021
[2] Akshay Kulkarni; Adarsha Shivananda; Anoosh Kulkarni," Creating a Résumé Parsing, Screening and Shortlisting System",
Natural Language Processing Projects pp 125-155, December 2021
[3] Sujit Amin; Nikita Jayakar; Sonia Sunny; Pheba Babu; M. Kiruthika; Ambarish Gurjar, "Web Application for Screening Resume",
2019 International Conference on Nascent Technologies in Engineering (ICNTE), January 2020
[4] Jitendra Purohit; Aditya Bagwe; Rishabh Mehta; Ojaswini Mangaonkar; Elizabeth George, "Natural Language Processing based Jaro-The Interviewing Chatbot", 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), August 2019
[5] Tejaswini K; Umadevi V; Shashank M Kadiwal; Sanjay Revanna," Design and Development of Machine Learning based Resume Ranking System", Global Transitions Proceedings, October 2021
[6] Pradeep Kumar Roy; Sarabjeet Singh Chowdhary; Rocky Bhatia," A Machine Learning approach for automation of Resume Recommendation system", Procedia Computer Science Volume 167, Pages 2318-2327, 2020
[7] Abeer Zaroor; Mohammed Maree; Muath Sabha," A Hybrid Approach to Conceptual Classification and Ranking of Resumes and Their Corresponding Job Post", Smart Innovation, Systems and Technologies book series (SIST, volume 72), 2017
[8] Garima Bhardwaj; S. Vikram Singh; Vinay Kumar," An Empirical Study of Artificial Intelligence and its Impact on Human Resource Functions", International Conference on Computation, Automation and Knowledge Management (ICCAKM), 2020
[9] Anushka Sharma; Smiti Singhal; Dhara Ajudia," Intelligent Recruitment System Using NLP", International Conference on Artificial Intelligence and Machine Vision (AIMV), 2021
[10] Yong Luo; Huaizheng Zhang; Yongjie Wang; Yonggang Wen; Xinwen Zhang," ResumeNet: A Learning-Based Framework for Automatic Resume Quality Assessment", 2018 IEEE International Conference on Data Mining (ICDM), December 2018
[11] Mujtaba, Dena F., and Nihar R. Mahapatra. "Ethical Considerations in AI-Based Recruitment." 2019 IEEE International Symposium on Technology and Society (ISTAS). IEEE, 2019.
[12] Fahad, SK Ahmed, and Abdul Samad Ebrahim Yahya. "Inflectional review of deep learning on natural language processing." 2018 International Conference on Smart Computing and Electronic Enterprise (ICSEE). IEEE, 2018
[13] Ayisha Thahir, C. H., C. Sreejith, and C. Rasik. "Combination of Neural Networks and Conditional Random Fields for Efficient Resume Parsing." 2018 International CET Conference on Control, Communication, and Computing (IC4). IEEE, 2018.

DOI: 10.17148/IJARCCE.2022.115166

[14] Gugnani, Akshay, Vinay Kumar Reddy Kasireddy, and Karthikeyan Ponnalagu. "Generating unified candidate skill graph for career path recommendation." 2018 IEEE International Conference on Data Mining Workshops (ICDMW). IEEE, 2018.
[15] Naim, M.I. Tanveer, D. Gildea and E. Hoque, "Automated analysis and prediction of sjob interview performance", IEEE Transactions on Affective Computing, 2016.
[16] Shaha T. Al-Otaibi, Mourad Ykhlef ,"A survey of job recommender systems", International Journal of Physical Sciences, 7 (29) (2012), pp. 5127-5142
[17] Adem Golec, Esra Kahya ,"A fuzzy model for competency-based employee evaluation and selection", Computers & Industrial Engineering, 52 (1) (2007), pp. 143-161
[18] Siham Jabri, Azzeddine Dahbi, Taoufiq Gadi, Abdelhak Bassir ,"Ranking of text documents using TF-IDF weighting and association rules mining", 2018 4th international conference on optimization and applications (ICOA), IEEE (2018), pp. 1-6
[19] Jyothis Joseph, Jaimy Sunny, R Raveena, BlessyElzaByju, KC Laya, ,"Resume Analyser: Automated Resume Ranking Software", International Journal for Research in Applied Science & Engineering Technology (IJRASET), 8 (7) (2020), pp. 896-899