International Conference on Recent Trends in Data Science and its Applications
DOI: rp-9788770040723.113

Natural Language Processing Based Screening for Applicant Tracking Systems

Arvind Jei
UG Student, Department of Data Science and Business Systems,
SRM Institute of Science and Technology
Kattankulathur, India
ad5245@srmist.edu.in

Ushashish Chandra
UG Student, Department of Data Science and Business Systems,
SRM Institute of Science and Technology
Kattankulathur, India
uc8189@srmist.edu.in

Anand Madasamy
Assistant Professor, Department of Data Science and Business Systems,
SRM Institute of Science and Technology
Kattankulathur, India
anandm4@srmist.edu.in

***Abstract*—Within a brief period, a typical online job posting receives a large number of applications. As it is highly inefficient in terms of money and time so hiring companies can't handle it, manually filtering out resumes isn't practical. Additionally, this screening procedure for resumes is unfair because many suitable profiles are not given the due consideration they deserve. This could mean that applicants who aren't the right fit for the job are chosen instead of those who are. Ideally, this paper develops a solution that improves the accuracy of resume screening. Relevant data like skills, education, and experience are extracted by our system using Natural Language Processing. From the unstructured resumes and, as a result, produces a form that summarizes each application. With the help of screening methods that eliminate unnecessary information, analysis of resume data is made easier. After the data pre-processing process, the data is compared with the job description and is analysed on similarity scales to rank each individual resumes. The resulting ranking scores can then be used to select candidates who are the most suitable for a given job opening.**

Keywords—Applicant tracking systems, resume screening, natural language processing, cosine similarity, term frequency-inverse document frequency

I. INTRODUCTION

All the major businesses' recruitment procedures have undergone major changes due to the increased internet connectivity. Recruiters are able to file an fill a wide range of positions by utilizing online job postings on various websites. Even though e-recruitment has made recruiting easier and saved money for both recruiters and applicants, there are new obstacles to overcome. Every day, Thousands of resumes are typically received by large businesses and recruitment agencies daily economic distress, when a lot of people are looking for work, this situation gets even worse because workers are more mobile.

It is inefficient and makes no sense for recruiters to individually screen each resume for such job postings because less than 5% of applicants will be selected [2]. The fact that these applicants use a variety of resume formats presents another challenge for the organizations. All People who apply to a job belong to different professional and personal backgrounds so every one of those applicants will have their own unique style to showcase his or her talents on their CV because they have had a variety of educational backgrounds and have worked on a variety of projects.

CV's are unorganized documents that can be stored in different file types (such as.pdf, .doc, .docx, .jpg,.txt, and so on). Likewise, there is no fixed or predefined formats or templates used in the writing of their content. Because it is difficult to read resumes, recruiters spend a lot of time sorting through them to find the best candidates.

In an effort to alleviate the difficulty of managing diverse and unstructured resumes, numerous job portals and external websites were established. Candidates are required to manually and orderly fill out an online form with all of their resume information to create candidate metadata. This method has the drawback of necessitating candidates to put in additional work, which often results in them not entering all of the required information into the online form.

These online sites use a general outline for all jobs which makes them inefficient in the case of specific job positions, making them unsuitable for all jobs. The employers then apply the keyword-based search to shortlist candidates using these templates. Insufficient to match candidates to the job description is this keyword-based search functionality This is due to the fact that it only requires certain required keywords to be present and has several disadvantages with regard to deriving information, such as avoiding natural language linguistics for example merged and compound words and the contextual meaning of the resume's content.

As a result, candidates who deserve to be considered for the shortlist are left out because these Boolean search techniques frequently produce results that are irrelevant.

II. RELATED WORKS

A. Manual Screening

Some of the people who will be recruiting for the company screen resumes manually. This means that each resume is looked at one by one to see if it fits the job description. If it does, the CV will be chosen. This might be dependent on the on the skills and qualities needed, the job experience of the candidates, among many other things to consider related to the description provided for the job. Issues with this method include:

- Time-inefficient - All CV's must be individually checked, which makes the whole screening process a lot slower.
- Even if all resumes are individually reviewed, recruiters are under a lot of pressure.
- Misallocated and resource waste: Instead of spending so much time reviewing resumes, recruiters could be working on other projects.

- Not efficient:This approach also requires considerable time and resources that could be spent on other projects.
- Unnecessary bias:Moreover, recruiters may not review all resumes after identifying job requirements.

B. Screening Using Artificial Intelligence

To address these issues, the article "Resume Evaluation System based on AI"[4] proposes an AI-powered Resume Screening Software that filters and ranks resumes based on specific keywords to identify suitable applicants. This system aims to classify and shortlist desired candidates more efficiently than manual methods.

The system requires PDF-formatted resumes, which are processed one at a time by extracting text and eliminating excess material. Keywords such as prerequisites are then classified by area, and scores are calculated and sorted for each region. Finally, the system generates a pie chart displaying scores, which helps recruiters select eligible candidates for the job role.However, this approach has some drawbacks, such as:

- *Inefficient* – each curriculum vitae or resume requires exorbitant time to have each document reviewed in comparison to other available methods
- *Not user-friendly and tightly connected blocks*the software may not be user-friendly and may require disrupting the entire code for any changes.

C. Reviewing methods with Machine Learning

The article "Resume Screening using Machine Learning"[5] suggests that in order to efficiently screen resumes, they should be formatted in CSV. To begin the screening process, irrelevant or duplicated terms should be eliminated, leaving only relevant words. These words are then analyzed and assigned skill points based on their relevance to the job requirements. The points are then sorted in order of importance and candidates are selected based on their skill point total. A graph is then displayed to showcase the skill points and aid in the selection process of suitable candidates.

The included benefits are:

- Many resumes can be evaluated simultaneously.
- Easily identify suitable candidates for the job.
- Process completion time is reduced.

Issues and disadvantages with the above approach

- *Challenges with Resume Format*: Only resumes in CSV format are compatible, which may not always be feasible since many candidates submit resumes in Word or PDF.
- *Difficulty in Implementation*: Implementing these techniques requires a high level of expertise and may not be accessible to everyone.
- *Interdependent Blocks of Code*: Making changes to one block of code may require modifying the entire code, leading to disruptions in the workflow.
- *Removal of necessary information* – certain procedures may cause a loss of crucial information and therefore lower overall performance of the model

D. Deep Learning methods of Screening

The article on Resume Screening using Deep Learning (LSTM)[6] suggests a process that involves a dataset with two columns: Category and Resume. The Category column includes fields such as Devops, DBMS, Engineering, etc. The input is the Resume column, which is used to categorize the resume on how well it matches.

Performing a value count on categories, one can obtain and generate a distribution representing the frequencies of resumes that fall into each category.

- After obtaining the dataset, the first step is pre-processing, which involves eliminating irrelevant information from the resume.
- Stop words, which do not contribute to the information, must also be removed using nltk.
- Final preprocessing step involves data condensation, tokenizing features, and labels, and giving less weightage to the most frequent words and more importance to less frequent words. This ensures that unique words are more useful than concise words.
- The model is then trained and evaluated using test scores and accuracy, which generates graphs.

III. PROPOSED METHODOLOGY

A. Information Extraction

Natural Language Processing is used to extract information in the first phase of our proposed system. The resumes do not contain the information in a structured manner. The recruiters can't use the noise, inconsistencies, or irrelevant bits of data. The goal is extracting the main keywords that are similar from the resume's raw data without requiring human intervention. Tokenization, Stemming, POS Tagging, Named Entity Recognition, and other methods, Important job-related content, such as skills, education, and so on, is gathered by our system. From the resumes of candidates uploaded. Each resume is summarized in a JSON format as a result, making it simple to perform subsequent processing tasks in the resume screening system's subsequent phase. Each chapter should be given an appropriate title.

B. Tokenisation

The initial step in identifying the bits of data that constitute a sequence of individual letters or words is to convert various resume formats such as word documents, pdfs, rich text format’s, etc. into a common format. Tokenization is the subsequent process, which involves breaking up large text chunks into smaller tokens. This allows us to analyze the original text sequence through these words. Tokenization involves removing or isolating whitespace and punctuation characters to break up sentences into singular tokens. After following the tokenization process, one continues the processtoextract analytics such as

DOI: rp-9788770040723.113
the total word count, word frequency, etc. There are several ways to perform tokenization, including with tools like the NLTK (Natural Language Toolkit), spaCy library, etc.

C. Stemming and Lemmatization

According to its grammatical rules, it is frequently observed that a single English word is used in numerous different forms in various sentences. For instance, "implement," "implement," and "implement" all refer to the same verb but in different tenses. educing all to their original stems from their variations of a word and its bases is important to avoid distinguishing similar meanings of derivationally related words. This goal is achieved through stemming and lemmatization, which have different methods but share the same objective.

Lemmatization is a more precise method for reducing words to their root forms. It involves using a language dictionary and morphological analysis to provide linguistically correct lemmas, rather than simply applying pattern matching rules to remove affixes like in stemming. This means that lemmatization takes into account the context of the word and its part of speech, allowing it to accurately determine the root form. In addition, unlike stemming, lemmatization always returns a valid word that exists in the language dictionary. The NLTK library provides a WordNetLemmatizer that is based on the WordNet database and is commonly used for English language processing.

The WordNetLemmatizer is a part of the NLTK Python package, which is a widely used tool for natural language processing tasks. It is based on the WordNet lexical database, which is a large electronic lexical database of English words and their meanings. The WordNetLemmatizertakes into account the context in which a word appears to identify the correct lemma, which is the base or dictionary form of a word. It uses different techniques such as part-of-speech tagging to identify the correct lemma for a given word. Compared to stemming, lemmatization provides more accurate and meaningful results for natural language processing tasks.

D. Chunking

Chunking is a method for organizing short phrases with parts of speech tags to provide categorizing phrases or components of a sentence. Combining regular expressions with POS tags, some chunk tags like Noun Phrase (NP) and Verb Phrase (VP) can be generated. This is necessary because POS tagging alone do not provide significant amounts of valid data for sentence integrity or derive any meaningful context from the corpus. Shallow parsing is the process of creating a parse tree with a singular information level, from the base of the tree, that is the root, to the leaves. The overall process then guarantees that more information can be available than the POS (Parts of Speech). Chunking is useful for segmenting and labeling multi-token sequences, with the primary purpose of identifying groups of "noun phrases" that can be used to enhance the next process, which is Entity Recognition.

E. NER Process (Named Entity Recognition)

One of the most useful techniques that aid in the data processing stage is to extract useful information from unstructured text data by identifying and categorizing it into predefined categories. NER successfully accomplishes this objective.Our objective is to use a similarity model to determine how closely the categorized resume data matches the requirements of recruiters, after categorizing the unstructured resume data into these diverse categories. There are several ways to use Named Entity Recognition [1] to extract relevant categories from unstructured data. One approach uses rules and syntax as a base, which involves developing our own algorithms that are tailored for use in domains that are specific to the employer. Regular expressions are contextualized to be utilized to identify named entities by searching for patterns in a string. Another method to approach NER as a sequence labeling problem is to use Bidirectional-LSTM and the Conditional Random Field algorithm [3].

F. Process of vectorising input data

Vector space is a geometric structure composed of a collection of elements known as vectors, which can be multiplied and combined by scalars. The overall process, which is popularly used in the domains of mining textual data, natural language processingandretrieval and processing of data, vectorisation is an algebraic model that represents documents as numerical vectors. Many models, especially those in the context of machine learning, sometimes involve vectors to be considered as the input data rather than UNICODE or ASCII data, making the process of vectorization significantly important. One common method of vectorizing data is to convert each word or string into a certain value within a data range, with each word occupying unique vector positions in the array/vector. Each corresponding value at every index represents the total count or frequency distribution of corresponding word occurring in the text. Since the result array is often smaller than the original corpus and its range of vocabulary, a vectorization strategy is almost necessary to be employed to account for this.

G. TF-IDF

“Term Frequency – Inverse Document Frequency" is the acronym for TF-IDF. TF-IDF was developed for document search and information retrieval. Since there are many terms that may not be important or relevant to be used for the similarity measure, therefore terms need to be measured relatively in terms of significance and th the overall weight they carry in terms of relevance in a corpus. Hence TF-IDF measures based on the weight of a word in terms of relevance as well as the frequency. The number of documents containing a word offsets the importance, which increases in proportion to its frequency in the document.

Therefore, terms like "this and "what," "who," "the," "if," and so on are frequently used in all documents. rank low, despite their numerous appearances, because they have little significance for that particular document. As shown in the first equation below, one can calculate the TF-IDF value for a word picked out from a file by the product of two of two variables or measures:

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

Number of occurrences of each Term: It determines how frequently a term is repeated in a file document. You must modify or standardize this frequency because a certain term can be repeated more with much larger documents, when compared to documents that are smaller in length. The number of times a word appears in a file is divided by the total number of words in that file to get a normalized term frequency. It can be written [7]as shown in equation (2) below.

$$TF(t,d) = \frac{freq(t,d)}{\sum_i^n freq(t,d)} \qquad (2)$$

In context, the total number of files in the collection where the term t exists is referred to as count(t) and the unique document count is represented as N.

The TFIDF score of a word in a document is the product of these two metrics, which are represented by equation (2). If the TF-IDF metric of a term in a file is high it indicates the greater importance of that term. The CV's and the work interpretation were modelled into a vector space by our system.

We can use various similarity metrics to generate and rank scores between job designations and the documents. Cosine similarity is commonly used in text mining and information retrieval tasks. The metric aims to measure the cosine of the angle between two vectors in a high-dimensional space, which is a measure of similarity between the two vectors. The similarity score ranges from -1 to 1, with 1 indicating that the two vectors are identical and -1 indicating that they are completely dissimilar.

H. Cosine Similarity

A metric that measures how similar two things are is known as a similarity measure. Cosine similarity is a measure across the sine curve that measures the similarity across the sine. Because it is a symmetrical algorithm, we can mathematically represent it, as below:

$$\cos(\theta) = \frac{\vec{a}.\vec{b}}{\|\vec{a}\|\|\vec{b}\|} = \frac{\sum_{i=1}^n a_i b_i}{\sqrt{\sum_{i=1}^n a_i^2}\sqrt{\sum_{i=1}^n b_i^2}} \qquad (3)$$

The dot product of the two vectors is +anbn. All pairs of elements' cosine similarity can be determined using this equation(3). The resume documents can then be ranked according to a particular query word vector. Cosine similarity, on the other hand, only looks at features that are related to the words in the text and will produce results that are less accurate. By including semantic information, similarity measures can be made more effective.

IV. RESULT AND ANALYSIS

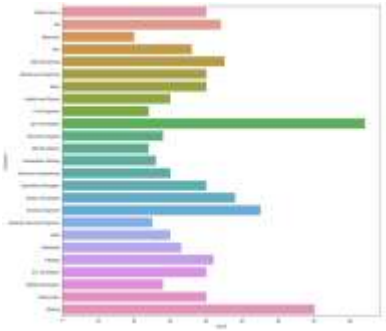

Fig 1. Categorization based on domain and skillsets.

The dataset was created by scraping across data from online job application sites, and was the condensed into a feature set of eight to nine features that would be relevant towards the use case. The model consists of ten domains that each are sub divided into job positions that require significantly different skill sets, such as Hadoop Engineers, .Net development, etc, and each resume is then categorized and plotted against a frequency distribution as Fig 1.

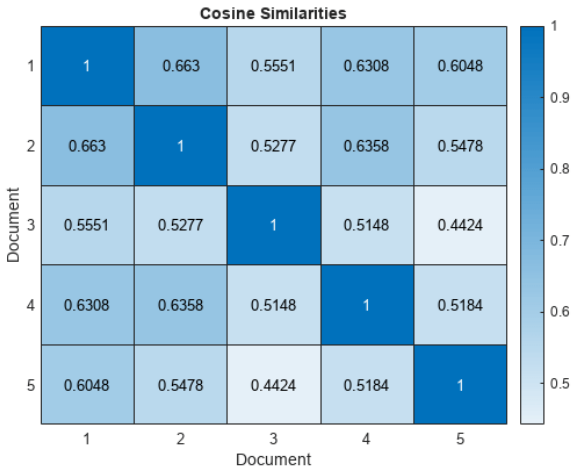

Fig 2. Cosine similarity measures against each resume and job description

The system model can then effectively categorise resumes based on the expected skill sets from each job group from the corpus and help determine what is the most suitable job for each candidate based on the position they applied to.The methodology used was successful in extracting the relevant skills from each resume as well as the personal details as well.

Using classifier measurements generated results that created strong precision scores for each skill classification as can be seen and cosine similarity metrics were able to evaluate the similarity of a resume against a job description, as demonstrated in the similarity matrix in Fig 2, the value closest to 1 indicating the highest similarity, and 0 indicating no similarity. Based on these metrics, the resumes are ranked.

V. CONCLUSION

Most organizations receive a large number of applications for each job posting. In today's world, it can be time-consuming for any organization to sort through the

plethora of resumes to find the most relevant application for a position. The candidate's resume must be manually classified, which takes a long time and wastes resources.

Therefore, a proposed automated machine learning-based model that, based on the job description, recommends suitable candidates' resumes to HR. The proposed model is expected to be effective in two stages: to begin with, characterize the resume into various classes. Second, suggests a resume based on how closely it matches the job description.

With help of a LinearSVM classifier, there is an expectancy was able to accurately capture the resume insights and the semantics with an higher accuracy. Using deep learning models like these could improve the model's performance: LSTM, Recurrent Neural Networks, Convolutional Neural Network (CNN), and others. An integration with LinkedIn and GitHub APIs will expand use cases. The developed method can thereafter be utilized to develop an industry model if the industry is plagued by large number of applications. By including domain experts like HR professionals, a more accurate model can be constructed, and HR professionals' feedback aids in iterative model improvement.

REFERENCES

[1] Dhanabalan, S. S., Sitharthan, R., Madurakavi, K., Thirumurugan, A., Rajesh, M., Avaninathan, S. R., & Carrasco, M. F. (2022). Flexible compact system for wearable health monitoring applications.Computers and Electrical Engineering, 102, 108130.

[2] P. Brown and K. Austin, "Applied HRM Research,"Appl. Phys. Letters, vol. 8, no. 2, pp. 51-62, 85X, 2503–2504, 2004.

[3] Huang, Zhiheng, Wei Xu and Kai Yu. "Bidirectional LSTM-CRF Models for Sequence Tagging," 2015, ArXiv abs/1508.01991, n. pag.

[4] Pazhani. A, A. J., Gunasekaran, P., Shanmuganathan, V., Lim, S., Madasamy, K., Manoharan, R., &Verma, A. (2022).Peer–Peer Communication Using Novel Slice Handover Algorithm for 5G Wireless Networks.Journal of Sensor and Actuator Networks, 11(4), 82.

[5] Bhushan Kinge, Shrinivas Mandhare, Pranali Chavan, and S. M. Chaware, "Resume Screening using MachineLearning and NLP: A proposed system," InternationalJournal of Scientific Research in ComputerScience,EngineeringandInformation Technology(IJSRCSEIT), ISSN : 2456-3307, vol. 8, issue 2, pp.253-258, March-April 2022.

[6] Navale Sakshi, Doke Samiksha, Mule Divya, Prof Said S. K. Resume Screening using LSTM

[7] Jabri, Siham, Dahbi, Azzeddine, Gadi, Taoufiq, Bassir, and Abdelhak,"Ranking of text documents using TF-IDF weighting and association rules mining," pp. 1-6, 2018, 10.1109/ICOA.2018.8370597.