International Journal for Modern Trends in Science and Technology, 9(05): 673-679, 2023

ISSN: 2455-3778 online
DOI: https://doi.org/10.46501/IJMTST0905115
Available online at: http://www.ijmtst.com/vol9issue05.html

Resume Screening using Natrual Language Processing

A Sunil Kumar, B. Siva Nandini, K. Jeevana, B. Kumuda Priya, D. Sai Varshini

Department of Comuter Sciecne Engineering, Narayana Engineering College, Nellore, Andhra Pradesh, India

To Cite this Article
A Sunil Kumar, B. Siva Nandini, K. Jeevana, B. Kumuda Priya, D. Sai Varshini. Resume Screening using Natrual Language Processing. International Journal for Modern Trends in Science and Technology 2023, 9(05), pp. 673-679. https://doi.org/10.46501/IJMTST0905115

Article Info
Received: 21 April 2023; Accepted: 18 May 2023; Published: 22 May 2023.

ABSTRACT

Resume Screening is the primary step in the hiring process. It evaluates the candidates' resumes and determines whether they are qualified for a role based on their education, skill sets, technical stuff, experience, and other information captured in their resume. To make it simple, it's a form of pattern that matches the job requirement and the candidate's qualifications based on their resume. It is a crucial step in the process of hiring. It is the step in which a decision is made to move the candidate to the next level or not. There are multiple processes to perform resume screening. Among all the processes, traditional resume or manual screening is the largest followed, even today. But usually, companies receive thousands of resumes for job applications, which consumes a lot of time and effort. In addition to this, many errors may arise due to human involvement. Multiple ways were introduced to cover all these cons to performing this resume screening process. Various technologies, including Artificial Intelligence and Machine Learning, were involved in searching for solutions. This paper contains a detailed survey report on various methodologies and techniques of resume screening.

KEYWORDS: Cosine Similarity, Resume Analyser, Ranking candidates, Resume parser,Vector space model

INTRODUCTION

Many large firms' recruitment processes have changed in recent years. Recruiters can attract a various range of applicants for his or her opportunities by using online job postings on numerous employment portals and websites. Talent acquisition is a vital, complex, and time-consuming function within Human Resources (HR). Applicants come from a spread of professions and are available from a spread of backgrounds. Each of them has had various kinds of education, has worked on various projects, and thus incorporates a distinct way of presenting his or her credentials within the resume. Resumes are unstructured documents that may be saved in a very form of file formats and are not produced using conventional forms or templates. As a result, reading resumes is difficult, and recruiters must devote a major amount of your time to sifting through resumes to pick out the most effective applicants. Effective screening of resumes requires domain knowledge, to be able to understand the relevance and applicability of a profile for the work role.

With the rapid increase in internet connectivity, there has been a change in the recruitment process of all major companies Though e-recruitment has provided convenience and savings for both the recruiters and the applicants, some new challenges arise. Large companies and recruitment agencies often receive thousands of resumes every day.

Our proposed solution will choose the best fitting candidates for a specific opportunity by relating the main features of the applicants' profile with the requirements defined in the job description. The system works in two main phases. In the first phase, all relevant candidate information like skills, work experience, years of education, certifications, etc. is extracted from the unstructured text in the resumes. The system uses Natural Language Processing to parse these relevant qualification details and then creates a summarised version of each resume (Allahyari, Mehdi, et al., 2017) irrespective of the order of content or the file format.

RELATED WORKS

A .Manual Screening

Screening of resumes is done by some of the company's employees who are going to recruit, i.e., every resume is checked individually, and if the resume is fit for the required job description, then the resume will be selected.

It might be done based on the capabilities they seek, the candidates' work experience, or other factors that are relevant to the job profile

B. Resume Screening using Artificial Intelligence

on AI, A Resume Screening Software is created using artificial intelligence, text mining, and processing algorithms. These algorithms filter and rank resumes based on specific keywords to identify which job applications recruiters should consider further.

Their main idea is to build a smart and automated Resume Evaluation System based on AI to overcome manual Resume Evaluation techniques for effectively classifying and shortlisting desired applicants.

The fundamental Working Procedure

The resume should be in PDF format to be viewed and read, and the text extracted. One resume at a time is given to the system. The excess material will be eliminated later, and the keywords, such as prerequisites, will be classified by area. The process is then repeated by calculating the scores for each region and sorting the results before generating the final scores. Finally, a pie chart will be displayed as an output based on the scores, helping recruiters select the required and eligible individuals for the offered job role.

C. Resume Sorting using Artificial Intelligence

According to the article Resume Sorting using Artificial Intelligence, A Database is created to store job applicants' resumes. This system is trained using Artificial Intelligence to recognize the words separately from a resume. Some important keywords that fit the job description, like skills, education qualification, and so on, are given to the system separately through some other files. The system is trained to scan the resumes and search for the separately given keywords. By matching the keywords in the resume, the system will shortlist the candidates according to the requirement. The system will select all the resumes that maximum reaches the job description requirement, and the remaining resumes will be rejected.

The entire project is designed to streamline or automate some parts of the recruiting workflow, especially repetitive high-volume tasks.

METHODOLOGY

In this section, we describe the concepts that facilitate the construction of the proposed Automated Resume Screening System. The system works in two phases as described below.

A. Information extraction

The first phase of our proposed system involves information extraction using Natural Language Processing. The information in the resumes is not present in a structured format. There are noises, inconsistencies and irrelevant bits of data which is of no use to the recruiters. The objective is to derive relevant keywords from the unstructured textual data in the resume without any need of human crawling efforts. Using techniques like Tokenization, Stemming, POS Tagging, Named Entity Recognition, etc., our system obtains important job-related content (skills, experience, education, etc.) from the uploaded candidate resumes. The result is a summarised version of each resume in a JSON format which can be easily used for further processing tasks in the next phase of this resume screening system.

B. Tokenization

After converting the various resume formats (.docx, .pdf, .jpg, .rtf, etc.) into text, we begin the tokenization process to identify terms or words that form up a character sequence. This is important as through these words, we will be able to derive meaning from the original text sequence. Tokenization

involves dividing big chunks of text into smaller parts called tokens. This is done by removing or isolating characters like whitespaces and punctuation characters. Tokens are sentences initially (when tokenized out of paragraphs) and then are further split into individual words. By performing Tokenization, we can derive information like the number of words in a text, frequency of a particular word in the text and much more. The tokenization can be performed in multiple ways such as using Natural Language Toolkit [NLTK], the spaCy library, etc. Tokenization is a mandatory step for further text processing such as removal of stop words, stemming and lemmatization.

C. Stemming and lemmatization

It is frequently seen that a single word of the English language is used in various different forms in different sentences according to its grammatical rules. For example -implement, implemented and implementing are just different tenses of the same verb. This situation results in the need to reduce all the altered or derived forms of a word to their central stem or base so that these derivationally related words with similar meanings are not considered to be different from each other. Both Stemming and lemmatization have the same objective but differ in their approach.

"Stemming is the mechanism of reducing inflected or derived words to their word root, or stem. It is a crude heuristic process that involves chopping off the ends of words to achieve this objective, and often includes the removal of derivational affixes" (Jivani, A.G., 2011). These are rule-based algorithms in which a particular word is tested on a range of conditions and then based on a list of known suffixes, decides how to cut it down. It is noteworthy that the root derived after stemming may not be identical to the morphological root of the word. Due to the heuristic-based approach of stemming, it suffers from issues such as under-stemming and over-stemming. Some common stemming algorithms used are Porter-Stemmer, Snowball stemmer, and Lancaster stemmer. On the other hand, lemmatization is the process of utilising a language dictionary to perform an accurate reduction to root words. Unlike Stemming which simply cuts off tokens by simple pattern matching, lemmatization is a more careful approach that uses language vocabulary and morphological analysis of words to give linguistically correct lemmas. This means lemmatization utilises the knowledge of context and therefore can differentiate between words that have different meanings based on parts of speech. For the English language, our system uses the WordNet Lemmatizer (based on WordNew Database) provided by the NLTK python package.

D. Parts of speech (POS) tagging

It is a process of assigning grammatical information to a word based on its context and its relationship with other words in the sentence (Gelbukh, 2014). The part-of-speech tag specifies whether the word is a noun, pronoun, verb, adjective, etc. according to its usage in the sentence. It is important to assign these tags so as to understand the correct meaning of a sentence and for building knowledge graphs for named entity recognition. This process is not as simple as mapping a word to their corresponding part of speech tags. This is so as a particular word may have a different part of speech based on different contexts in which it is used. For example: In the sentence "I am building a software", building is a Verb, but in the sentence "I work in the tallest building of that street", building is a Noun. Also called grammatical tagging or word-category disambiguation, it is a supervised learning solution that analyses the features such as the preceding word, following word, first letter capitalized or not, etc. to label the words after tokenization. Rule-Based POS tagging, Stochastic POS tagging, and Transformation based tagging are mostly used (Hasan, 2006).

E. Chunking

Chunking is a process that aims to add more structure to sentences by grouping short phrases with parts of speech tags. Because parts of speech tags alone cannot give information about the structure of the sentence or the actual meaning of the text, chunking combines parts of speech tags with regular expressions to give a result as a set of chunk tags like Noun Phrase (NP), Verb Phrase (VP), etc. Also called Shallow Parsing, it involves the construction of a parse tree that can have a maximum one level of information from roots to leaves. This ensures there is more information than just part of speech of the word without needing to create a full parse tree. Chunking segments and labels multi-token sequences (Bird,

Klein and Loper, 2009), mostly making groups of "noun phrases" that are used for finding named entities.

F. Named entity recognition

Named Entity Recognition is an information extraction technique which extracts relevant information by classifying chunks of unorganized text into predefined categories like names of persons, companies, contact info, educational credentials, and skills. After classifying the unstructured resume data into such different sets of categories, our aim is to use a similarity model to determine the similarity between the categorized resume data and the requirements provided by the recruiters. There are many approaches to implement the Named Entity Recognition (Mansouri, A., Affendey, L.S. and Mamat, A., 2008) in order to derive relevant categories from unstructured data. These include the Rule-Based approach in which we define our own algorithms according to the required domain. We can also use regular expressions, which finds patterns in a string to detect the named entities. Another approach is using Bidirectional-LSTM with the Conditional Random Field algorithm for named entity recognition as a sequence labelling problem (Huang, Z., Xu, W. and Yu, K., 2015).

G. Cosine Similarity

A Similarity measure is a metric that determines how much the two objects are alike. Cosine similarity (Sidorov, Grigori, et al., 2014) is a measure to find how similar the two documents are regardless of their size. It represents the orientation of the documents when plotted on an N-dimensional space, where each dimension depicts the features of the object. It's a symmetrical algorithm, which implies that the results from computing the similarity of item X to item Y is equal to computing the similarity of item Y to item X. Mathematically, we can

represent it as shown below (Sidorov, Grigori, et al., 2014) in equation (4).

$cos(\theta)=a\rightharpoonup.b\rightharpoonup/||a\rightharpoonup||||b\rightharpoonup||=\Sigma aibini/=1\sqrt{\Sigma}\ ai2ni=1\sqrt{\Sigma}bi2ni=1$ (4)

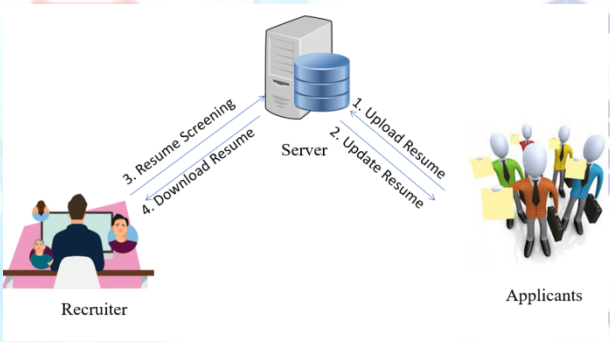
Here, $\mathbf{a}.\rightharpoonup\mathbf{b}\rightharpoonup=\Sigma aibin1=a1\ b1+a2b2+...+an$

is the dot product of the two vectors. Using this formula, we calculate the cosine similarity between all pairs of elements. It can then be used to rank the resume documents with respect to a given vector of query words. Using this formula, we calculate the cosine similarity between all pairs of elements. It can then be used to rank the resume documents with respect to a given vector of query words.

SYSTEM DESIGN

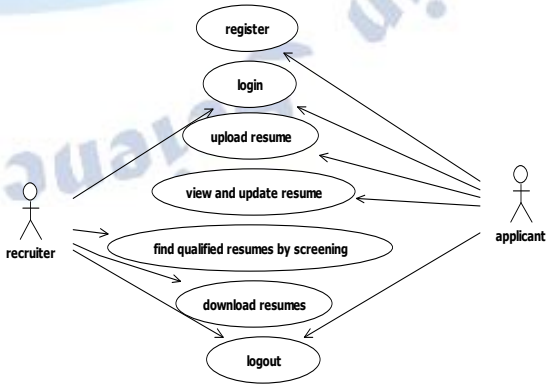
A. Architecture Diagram

Our project architecture is composed of four main components: resume uploading by applicant, applicants giving the details required to highlight, the resume screening with the job description given by the recruiter, downloading the resume of the desired applicant. Firstly, the applicant will register with required details. And then he will login in to his account and then uploads the resume. And the recruiter will login and gives the job description and downloads the desired resume.



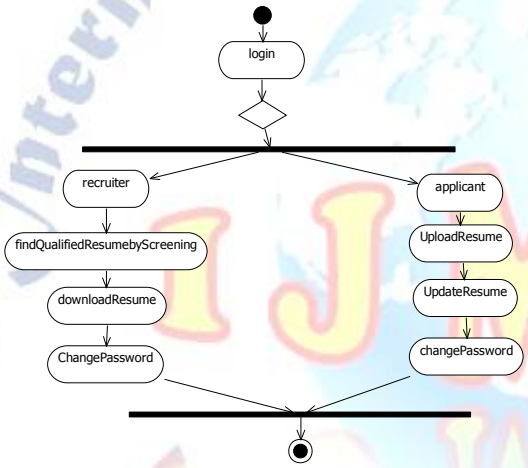
B. Use Case Diagram

In this model there are two actors: Recruiter and Applicant. The use cases of the Resume screening using NLP are: register, login, upload resume, view and update resume, find qualified resumes by screening, download resumes, logout.



C. Activity Diagram

- The actions in the activity diagram are:
- Recruiter loginbutton,
- Applicant Loginbutton
- Screening Resumes.
- In recruiter side actions are: finqualifiedresumesbyscreening, downloadresume, changepassword, logout.
- In the applicant side actions: Uploadresume, Updateresume, Changepassword.



C. Class Diagram

There are three classes in this model:Recruiter, Applicant, Data base:

Class 1:
The attributes are

- User name
- Password

The operations are

- Login()
- Findqualifiedresumes()
- Downloadresume()
- Logout()
- Changepassword()

Class 2:
The attributes are:

- Username
- Password

The operations are:

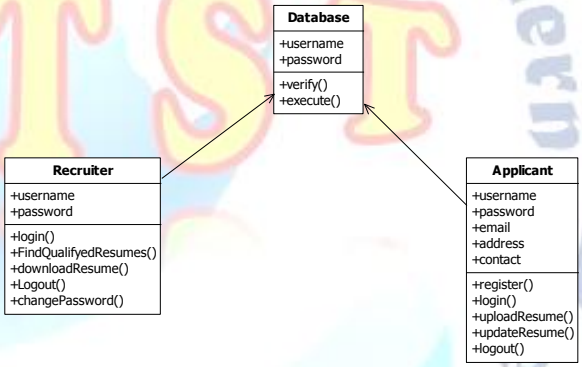
- Verify()
- Execute()

Class 3:
The attributes are:

- Username
- Password
- Email
- Address
- contact

The operations are:

- Register()
- Login()
- Uploadresume()
- Updtaeresume()
- Logout()



V.RESULTS

%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html;

charset=UTF-8">

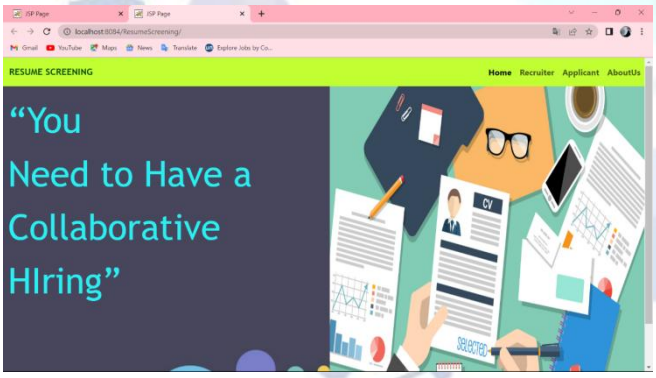
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha2/dist/css/bootstrap.min.css" rel="stylesheet">

<title>JSP Page</title>

```
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js"></script>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha2/dist/js/bootstrap.min.js"></script>

<style>
   body{
     background: url('images/bg1.jpg');
     background-size: cover;
     background-repeat: no-repeat;
   }
   .bujji
   {
     margin: 50px 120px;
     width:80%;
     height: 500px;
     border: 1px solid #fff;
     border-radius: 20px;
     //background: rgba(2, 188, 154, 0.5);
   }
```



CONCLUSION

In this project, we presented an automated resume screening system that simplifies the e-recruitment process by eliminating the various problems faced by the recruiters as they relied on manual shortlisting of applicants for a given job position. Our system works on two fronts. Firstly, it uses Natural Language Processing to extract relevant information from the unstructured and wide-ranging formats of the resumes. It creates a summarised version of each resume which has only the entities that are pertinent to the selection process. With all the insignificant information removed, the task of the screening officials is simplified, and they can better analyse each resume with better efficiency. On the other front, our system provides the provision of ranking the applicants by using a content-based recommendation that uses the Vector Space Model and similarity to match the extracted resume features with the requirements in the job description. It calculates the similarity score value for each resume and thus creates a ranked list of top-N recommended candidates that best fit the particular job opening.

FUTURE WORK

Future work for this system includes mining social networking data (e.g. Facebook, LinkedIn, GitHub profiles) of the candidates and utilising this social behaviour information in combination with resume content to make even more improved recommendations. Another possibility is using a collaborative filtering based approach that can match the current applicant with a job according to how well other similar candidates (neighbours) are rated for it. Another scope of future work lies in the use of Latent Semantic Analysis (Berry, M., 2001) in the calculation of semantic similarity between the documents .

ACKNOWLEDGEMENTS

This research was supported by Narayana Engineering College, Nellore. We would like to sincerely thank , our guides, MR. R NAVA TEJA REDDY (Assoiciate professor), and MR. S SURESH BABU(Assistant professor) for h constant support and guidance We would also like to thank all anonymous reviewers for their valuable feedback.

Conflict of interest statement

Authors declare that they do not have any conflict of interest.

REFERENCES

[1] Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E.D., Gutierrez, J.B. and Kochut, K., 2017. A brief survey of text mining: Classification, clustering and extraction techniques. arXiv preprint arXiv:1707.02919.

[2] Arguello, J., 2013. Vector space model. Information Retrieval September, 25.
[3] Berry, M., 2001. Computational Information Retrieval. Philadelphia: Society for Industrial and Applied Mathematics,121-144.
[4] Bird, S., Klein, E. and Loper, E., 2009. Natural Language Processing With Python. Bejing: O'Reilly, 264.
[5] Faliagka, E., Ramantas, K., Tsakalidis, A. and Tzimas, G., 2012, May. Application of machine learning algorithms to an online recruitment system. In Proc. International Conference on Internet and Web Applications and Services.
[6] Gelbukh, A., 2014. Computational Linguistics And Intelligent Text Processing. Berlin, Heidelberg: Springer Berlin Heidelberg.
[7] Guo, X., Jerbi, H. and O'Mahony, M.P., 2014, September. An analysis framework for content-based job recommendation. In 22nd International Conference on Case-Based Reasoning (ICCBR), Cork, Ireland, 29 September-01 October 2014.
[8] Hasan, F.M., 2006. Comparison of different POS tagging techniques for some South Asian languages (Doctoral dissertation, BRAC University).
[9] Huang, A., 2008, April. Similarity measures for text document clustering. In Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand, 4, 9- 56.
[10] Huang, Z., Xu, W. and Yu, K., 2015. Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991.
[11] Jabri, S., Dahbi, A., Gadi, T. and Bassir, A., 2018, April. Ranking of text documents using TF-IDF weighting and association rules mining. In 2018 4th International Conference on Optimization and Applications (ICOA), 1-6. IEEE.
[12] Jivani, A.G., 2011. A comparative study of stemming algorithms. Int. J. Comp. Tech. Appl, 2(6), 1930-1938.
[13] umaran, V.S. and Sankar, A., 2013. Towards an automated system for intelligent screening of candidates for recruitment using techontology mapping (EXPERT).
[14] Rutuja Patil, Pratiksha Sarvade, Ajinkya Patil, Yash Bhosale, "Resume Evaluation System Based on AI," International Research Journal of Engineering and Technology, vol. 7, no. 7, 2020.
[15] V. V. Dixit, Trisha Patel, Nidhi Deshpande, Kamini Sonawane, "Resume Sorting using Artificial Intelligence," International Journal of Research in Engineering, Science and Management, vol. 2, no. 4, 2019.
[16] Pradeep Kumar Roy, Sarabjeet Singh Chowdhary, Rocky Bhatia, "A Machine Learning Approach for Automation of Resume Recommendation System, Procedia Computer Science, vol. 167, pp. 2318-2327, 2020.
[17] Gaurav Dutta, Resume Screening Using Machine Learning, 2021.
[18] "Resume Screening Using Deep Learning on Cainvas," AI Technologies and Systems, 2021. Tumula Mani Harsha et al. / IJCSE, 9(4), 14-22, 2022 20