

Resume Screening using NLP and LSTM

S Bharadwaj¹
Department of CSE
GokarajuRangaraju Institute of
Engineering & Technology, Telangana,
India.
savanbharadwaj379@gmail.com

Rudra Varun²
Department of CSE
GokarajuRangaraju Institute of
Engineering & Technology, Telangana,
India.
varunrudra14@gmail.com

Potukuchi Sreeram Aditya³
Department of CSE
GokarajuRangaraju Institute of
Engineering & Technology, Telangana,
India.
sreeramaditya10@gmail.com

Macherla Nikhil⁴
Department of CSE
GokarajuRangaraju Institute of
Engineering & Technology, Telangana,
India.
nikhilmacherla23@gmail.com

G.Charles Babu⁵
Gokaraju Rangaraju Institute of
Engineering and Technology
(Autonomous)Hyderabad, Telagana
charlesbabu26@gmail.com

ABSTRACT

Resume screening is the process of determining whether a candidate is qualified for a position based on their education, experience, and other information contained on their resume. Only if the resume of an experienced employee/fresher matches the job description will they be called for an interview. Manual examination of resumes might be a burdensome task. Manually doing it would take a long time. Companies utilize tracking systems to shortlist personnel based on their abilities. The most prevalent reason for rejection is a mismatch between the job role and the applicant's skill set. It is vital for job seekers to understand which job categories they should apply to based on their skill set. This project intends to develop an application that will categorize CVs according to the skills they contain into various job options. As a result, these programs aid job seekers in evaluating what types of positions they are qualified for based on their resume skills. Upgrades in application innovation for misconduct will probably result in higher overall expenditure in this area. This work also strives to make grouping more important by combining multiple classes into larger groups. At last, the analysis has been presented with a wide range of classifiers and new approaches have been suggested.

Keywords – Resume Screening, NLP, LSTM, Neural network.

INTRODUCTION

'Resume' is a term you will hear a lot during the job-search process. What is a resume? A resume is a document that contains information about your education, work experience, talents, and accomplishments. Successful resumes attract the attention of potential employers. Potential candidates will be invited to interview. Almost all job applications require the submission of a resume.

Resumes have the following significance:

It includes critical information.

A potential employer can use this information to assess your suitability for a position.

Recruiters let candidates know if they wish to invite them to an interview.

Advantages of a Good Resume:

Summarize your vocation goal

To show your accomplishments

To sell your skills

To grab an interview

To Brand Yourself

Quick but lasting first impression

To show your achievements.

The Objective of our work is to:

save time

manage a large volume of candidates

have a clear idea about job-search

efficient screening of resumes

LITERATURE SURVEY

Resume screening has become a significant part of the selection process in today's world. In the olden days, recruiters used to go through the resumes manually. The current scenario is entirely different from the past days. In the current world, as there is massive competition between the people for a role, it has become difficult for the recruiters to go through the resumes manually. Companies do not want their recruiters to spend much time shortlisting resumes for a particular role. When resumes are shortlisted manually, there may be a chance that the recruiter might not see all the resumes when a recruiter is satisfied in finding a set of resumes from half of the resumes submitted. Thereby recruiters might miss the best resume from submitted resumes. The Application tracking system can solve these problems. It screens all submitted resumes, selects the best resumes, and matches the job description. They are happy to assign this work to a machine learning model or Application Tracking System as it selects the best resume and does not require human resources. At times, the recruiter may also make mistakes in shortlisting the resumes. They may not be accurate at times due to various reasons. However, Application Tracking systems will be accurate, and it selects a perfect resume for the given job description. Using Application tracking systems also helps companies select the best

persons aware of their knowledge and skills and can adequately present them. Having skills and proper knowledge does not help people when they cannot present them adequately. People may lose their chance when they cannot present their skills according to the job description.

These resume screening systems are available to the companies but not the applicants. People are losing their chances even though they have relevant skills as they cannot present them properly. So a system has to be developed that can help applicants know the strength of their resumes before applying for a particular role. Jai Janyani et al. [2] built a web app to help employers analyze resumes and CVs, surfacing candidates that best match the position and filtering out those who did not. Roberto Salazar [3] proposed a python code to screen a candidate's resume and analyze his/her background. Venkat [4] proposed a python module `en_core_web_sm` (Spacy), which uses advanced natural language processing and machine learning. Team Jarvis [5] developed a model of python web application which screens the resumes of candidates and shortlists them for hr. An experienced candidate or fresher with skills in different domains can get roles in different technologies. All these application tracking systems use different technologies to find whether a resume has the potential to get shortlisted for a particular role or not. However, it is equally essential for a fresher or experienced candidate to know different roles that one may get based on their skills. The main objective of our project is to tell the users about the kind of roles that one may get based on their skills in their resume by displaying confidence percentages. These confidence percentages also help applicants find their resume strength, and thus they will have a clear idea about their resume shortlisting for a particular role.

METHODOLOGY

1. **Input Data:** The program's execution starts with taking of resume in the form of text from the user. Then the user clicks on the predict button, which is available in the window. After clicking on the predict button, the system goes through a process to determine the accurate results.
2. **Pre-processing Input Text:** The input text received from the user should be cleaned. A resume or input text should be free of unnecessary words, links, or details that will not help us find the results we are looking for. We remove a set of stop words from the input text. Commonly used words are known as stop words. Then the input text is passed an argument to the `clean Resume()` Function to remove Hashtags, URLs, links, which may not be included in the set of stop words. We make use of regular expressions to remove them.
3. **Tokenization:** After the resume is cleaned, the input text undergoes a tokenization process. Tokenization is the process of dividing the sentences into words known as tokens. We can also divide the word into tokens, where each letter after dividing is known as tokens. Tokens are part of the original text.

Tokenization is the primary step in Natural Language processing. By using tokenization meaning of the text can be simplified, which helps in further analysis.

4. **Vectorization:** In Vectorization, each word or token is mapped with a corresponding vector. Now the input text is in the form of vectors. These input vectors are given as input to the trained model.

5. **Model Training:** Before proceeding further, train the model to get accurate results. It is not a mandatory step to execute if you have a saved model already trained. In this phase, we build a neural network model. And train it with the Kaggle dataset, which contains approximately 1000 resumes. While training, the model also cleaned the resumes, followed and tokenization process then converted them into input vectors to build the model.

6. **Save Model:** After training the model, to prevent the redundant training of the model, we can use the saved method in Keras to save the trained model.

```
model.save("model name")
```

7. **Model Loading:** The saved model can be used again using the `load_model` method in Keras. There would be no data loss for saving and loading the model.

```
model=load_model("model name")
```

8. **Predicting the model output:** A sequence of steps is followed to convert the input text or resume into input vectors. The vectorized input is then used as input to the trained model. Using the `predict` method of the Keras library, roles can be predicted from the input text. Predictions can then be modified before being displayed to the user.

9. **Displaying Results:** Two possible processes may occur when the user clicks the predict button. If the user does not enter any input before clicking on predict button, the system responds with "Content is Empty .please enter the appropriate in formation." If there is content before clicking predict button, the input text or resume undergoes the above steps, and the result is predicted. The predicted output is modified before displaying. We take the top five roles in an array. Furthermore, we calculate the percentage of the roles by using probability.

IMPLEMENTATION

The architecture diagram represents the outline of the proposed software system. It provides overall summarization of the associations between the various components. It minimizes the total view of the software system into an understandable format. Firstly, the user interacts with the interface by entering the resume in the interface window in text form. There are buttons present on the window. They are predicting, reset, and quit buttons. When the user clicks on the predict button, the

system undergoes many processes. Firstly the pasted resume undergoes a data cleaning process.

In the data cleaning process, we import the nltk library and stop words from the nltk corpus, which helps us to remove stop words from the pasted resume. A function is defined to remove URLs, hashtags, mentions. After that, data tokenization happens where each word is assigned with a token. Then we load the trained neural network model, which will make predictions. The predictions made by the neural network are in terms of probabilities. After getting the results from the neural network, we are using those probabilities to make up results and get the probabilities we want to display.

EXISTING APPROACHES

Currently, resumes are classified according to the following methods:

1. Application tracking system:

The popularity of online recruitment has made conventional methods ineffective for screening resumes. As more and more resumes are being submitted, application tracking systems (ATS) are becoming increasingly popular. ATS makes shortlisting resumes easier for recruiters and hiring managers.

An applicant tracking system is primarily responsible for:

- Resumes are stored for review.
- Searching for specific keywords within resumes.

Selection of candidates for interviews.

ATS has a significant drawback: they are recruiter-oriented, and candidates cannot know if their resume matches the job role.

2. Dictionary with critical terms

Recruiters use these methods to screen resumes for specific keywords and then categorize them to determine which job applications should be further reviewed.

A standardized, automated screening process is necessary to categorize qualified candidates from unqualified ones based on their skill set.

The dictionary is created using key terms relevant to the roles.

Example:

```
terms={'Data analytics':['analytics', 'API', 'aws', 'big data', 'business intelligence', 'PMI', 'PMP', 'project', 'risk', 'schedule', 'data mining', 'data science', 'deep learning', 'hadoop', 'hypothesis test', 'iot', 'internet', 'machine learning', 'modelling', 'nosql', 'nlp', 'predictive', 'programming', 'python', 'r', 'sql', 'tableau', 'text mining', 'visualization']}
```

In the above example, it can be seen that a dictionary is created, and essential keywords are listed for a particular job role. When a resume contains these keywords, it qualifies for the next round. This has the significant advantage of not having a machine learning algorithm, preventing a suitable candidate from getting selected.

3. Resume Screening with Linear SVC

SVC (Support Vector Classifier) is a supervised algorithm under Support Vector machines. It is a learning algorithm that analyses data for classifications and regressions. A "best fit" hyper plane is created by fitting the data provided and then categorizing the data.

The significant disadvantages of Linear SVC are:

1. Large datasets are not suitable for SVM.
2. Data with more noise do not perform well with SVM.
3. There can be no probabilistic explanation for the classification of support vector machines since they use data points above and below the classifying hyper plane.

4. Resume Screening with K-Nearest Neighbour Classifier

K-nearest Neighbours (KNN) is a supervised Machine Learning algorithm for regression and classification problems.

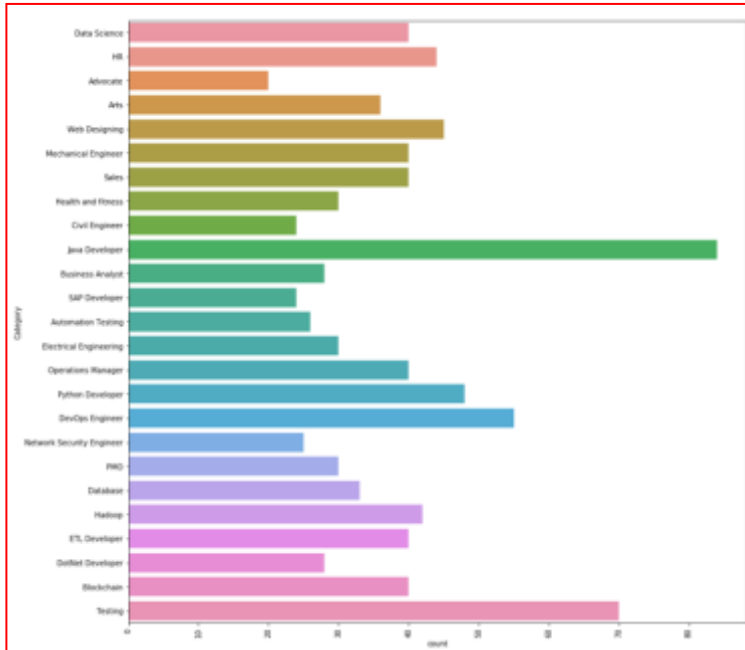
KNN algorithms classify new data points based on similarity by calculating the distance between two points in the dataset.

The algorithm determines classification based on the majority vote.

1. This algorithm degrades the algorithm's performance when working with large datasets: calculating the distance between a new point and each existing point is exceptionally costly in large datasets.
2. It is difficult for the given algorithm to calculate the distance within each dimension when there are many dimensions.
3. Sensitive to noisy data, missing values, and outliers.

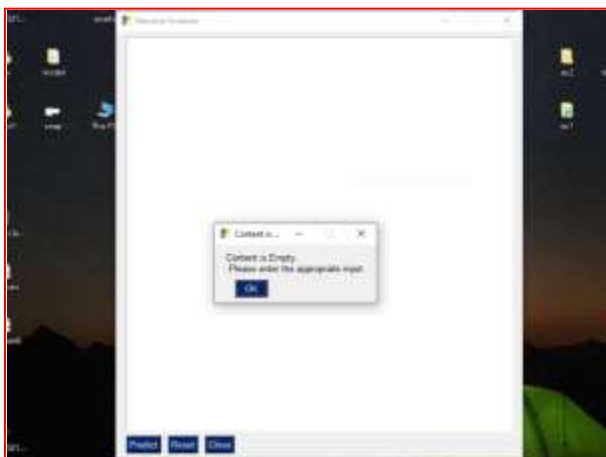
DATA SET

A total of 962 resumes and 25 job categories comprise the utilized dataset. All resumes in the dataset are full-size, detailed resumes. The dataset is a balanced dataset. Categories of job roles include:

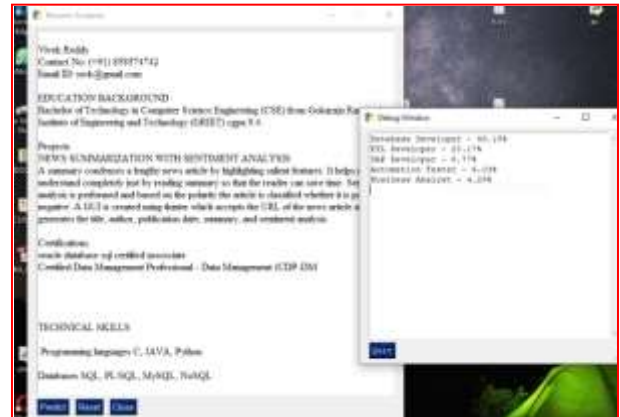


EXPERIMENTAL RESULTS

After installing the python environment and the required modules, the application is ready for execution and results. Firstly when we execute the application file, A window opens with three buttons. They are predicted, reset, and closed if the user clicks on the predict button without entering any text. The system responds with information saying, “Content is empty. Please enter the appropriate input”. The corresponding picture is shown in the diagram.



If the user clicks on the predict button after giving the input text, the system goes through a process to give results. The system is expected to give the output that makes people like recruiters and applicants choose their employees and career. The Below Image shows an output, where we can see that after entering the candidate's resume, the output is displayed as database developer along with the confidence percentage. The top 5 job categories for the resume are shown.



CONCLUSION AND FUTURE SCOPE

The proposed project used a natural language processing toolkit to achieve high accuracy. It has a straightforward interface, So it is easy for naive users to access it. In the future, it can be upgraded such that the system does 90% of the selection process of resumes. We can extract the candidate's skills and works from the LinkedIn GitHub profiles mentioned. Thereby, it will be an easy recruiter to select a candidate. We can also develop a system that can extract the information when uploading a resume in different formats.

REFERENCES

- [1] “Three-step Resume Screening Process” by [Fresh works](#), founded in October 2010.
- [2]- “Automated-Resume-Screening-System” built by [jai janyani](#) and pushed into github, 2018.
- [3] – “Resume-Screening” by [Roberto Salazar](#) on github, May 2020.
- [4] – “Resume-Screening-NLP” by [Venkat](#) using [spaCy](#), Jan 2019
- [5] – “resume-screening” proposed by [Team jarvis](#), Everest Hackathon, Jul 2019.
- [6]-Practical Natural Language Processing: A Comprehensive Guide to Building Real-World NLP Systems by HarshitSurana
- [7]-Resume Classification using KNN and Cosine Similarity Authors: Rajath V, RizaTanazFareed, SharadadeviKaganumath Paper ID : IJERT V10I1S080057