A smart resume screening tool for customized shortlisting

Poonam Tijare[1], *Mohammed* Waseem[2], *Mohd Azaan* Sherani[3], *Kornipalli Sampath Kumargari* Sai Krishna[4], Kavitha P.[5]

[1]Assistant Professor and Research scholar, CSE (VTU RC), CMR Institute of Technology, Bangaluru, Email: poonam.v@cmrit.ac.in
[234]Department of Computer Science & Engg., CMR Institute of Technology, Bangaluru-560037
[5]Associate Professor, CSE (VTU RC), CMR Institute of Technology, Bangaluru, Email: kavitha.p@gmail.com

Abstract. Hundreds of resumes are received, processed, and managed by large companies and recruitment agencies. Furthermore, many people post their resumes on the internet. Organizations all across the world, on the other hand, are battling to locate the greatest resource. To complete this work, these organisations rely on industry expertise. Manual interaction is required in the resume screening process. Most of the current technologies search for keywords but do not consider semantics, resulting in many superfluous resumes being shortlisted. The goal of the proposed study is to create a smart resume screening algorithm that can automatically retrieve and process resumes. Name, phone / cell numbers, e-mail addresses, qualification, experience, skill sets, and other fields are mapped to the retrieved data. The proposed model uses AI and ML techniques to do so. The gathered data can be utilised to develop applicant profiles that meet the organization's recruitment needs. By applying multiple filters to the data, an efficient man-less screening process can be achieved. The model is applied to the resumes that the company receives. The model has performed with an average accuracy of more than 90%. The model can be enhanced to apply on the resumes written in languages other than English.

1 Introduction

Businesses nowadays receive thousands of applications for each available post in the offices. Every day, a large number of resumes are processed to find the suitable candidates. This figure soared during the recent COVID 19 incident, when the entire employment procedure was conducted online. It is impossible to personally review each and every application; in fact, attempting to do so could be a waste of human resources. This resume processing procedure should be automated. Human intelligence is required to determine whether an applicant is suitable for a particular job description. A variety of features, including as evaluating a candidate's soft skills, accounting for changing company expectations as market trends and requirements change, and analyzing the veracity of a candidate's resume, cannot yet be automated. However, we may automate a portion of the process to reduce the number of candidates who must be evaluated by a human [1].

Reading a candidate's resume and determining whether or not they are qualified for a specific job profile is an important aspect of the hiring process. This task entails Professional competence when reading and digesting information from a resume and then comparing it to a job description to determine the suitability of the application. This task, however, is extremely difficult for a machine to complete. Traditional computer systems are unable of recognizing the semantic significance of various resumes. However, current advances in machine learning and natural language processing (NLP) techniques enable this task to be completed without the need for human intervention and with high accuracy, saving the corporation employee hours.

The goal of the task at hand is to create an intelligent system that can extract all of the important information from varying resumes and convert them all to a common structured format that can then be ranked for a specific employment position. Name, email address, social accounts, personal websites, years of study, education experiences, positions and responsibilities, keywords, and the cluster of the CV are all parsed information (ex: computer science, human resource, etc.). This parsed data is then saved in a database to be used later. Resumes are more structured than other unstructured material (email body, web page contents, etc.). Resume information is maintained in discrete sets. Each set comprises information about the person's contact information, employment experience, and educational background [2].

Regardless, resumes are tough to decipher. This is due to differences in information types, arrangement, and writing style. They can also be written in a variety of formats. The most prevalent are '.txt,' '.pdf,' and '.doc'. The model cannot rely on the sequence of the data to effectively and efficiently interpret data from various types of resumes.

Our system's main goal is to automate the screening process in order to lower the cost of hiring and make the process more efficient. The methodology is designed to determine whether an applicant is qualified for a job by comparing the job requirements to the information on their resumes, such as education, abilities, and designation. The model does more than just classify data; it also summarises information in resumes and generates a candidate profile.

2 Literature survey

Building a smart model necessitates the use of technologies and an understanding of current research. The research findings are summarised in the table below, which includes the techniques as well as their benefits and drawbacks.

Table 1. Approaches used for automated resume parsing

Approach	Advantages	Shortcomings
Named Entity Recognition (NER) Spacy[3]	Fastest and most accurate syntactic analysis. Fits best for the smaller dataset.	Due to the application on small datasets, the model's accuracy is not thoroughly tested.
GraphIE [4]	Used un-localized words for NER rather character-level or adjacent word level	The model doesn't count the global entities into consideration for NER.
Continual Learning for NER[5]	For new entities in a new domain, there is no need to retrain the entire NER system.	Requires high computing Resources.
NER-BERT [6]	Instead of greatly compressing the categories into four types, the authors only merged the most fine-grained categories to lower the number of entity categories.	The model needs to be scaled and tested on various NER systems.
Knowledge Discovery from CVs [7]	Uses probabilistic approach rather than model based approach.	Dataset provided is small.
NER as Dependency Parsing[8]	Lead to substantial improvements for both nested and flat NER.	Suffers with obscurity and abbreviations issue.
ML approach: Resume Recommendation system[9]	Model works on structured data. Uses NLP approaches to recommend CV	Model accepts Job applications in CSV format, but in fact, such applications are in .doc or .pdf formats.
ML approach using KNN (K-nearest neighbours)[10]	Model discovers matching resumes using KNN algorithm, System employs ranking of the resumes	System lists the resume files based on the rank. Each resume then need to be opened and checked.
Resume classifier system with ML approach[11]	Model built tested on nine ML approaches. SVM (Support Vector Machine) outperforms compared to other models, resulting into 96% accuracy	Model only able to classify resumes, does not rank them.
Convolution Neural Network(CNN) based approach applied to screen candidate job profiles[12]	CNN based model with NLP techniques developed to rank resumes based on the similarity in the fields	Model works with 74% accuracy applied on the resumes of job portal

The findings of the table 1 states that NER model is the most preferred model by the researchers. Due to the probable difficulty of acquiring personal information of the candidates, the proposed models are not exposed to the various datasets. The study also looks into semantic analysis issues. This indicates that for more relevant and accurate findings, the smart resume screening tool should combine ML and AI techniques.

3 Proposed model

The proposed model works by using the various ML and AI based techniques. The model works by taking the resume files in .doc, .docx, and .pdf formats. The model works in phases. The Text Extractor receives the.pdf or.docx version of the resume or CV. The collected text is subsequently sent to the Text pre-processor for further processing. The Text pre-processor produces cleaned text as its output. The relevant entities are extracted from the cleaned text.

An Education Details Extractor and a Basic Details Extractor receive the cleaned text. The Education Details Extractor returns the candidate's educational information. To extract Indian names, the Bidirectional Encoder Representations from Transformers(BERT) model is employed. The Basic Details Extractor gathers information such as phone numbers, email addresses, and social network profiles.

The Noun Chunks Extractor takes the cleaned text and extracts the nouns. The Entity Tagger takes these nouns and classifies them as a skill, a role, or a degree. The nouns that aren't relevant are ignored. The database stores all of the extracted information.

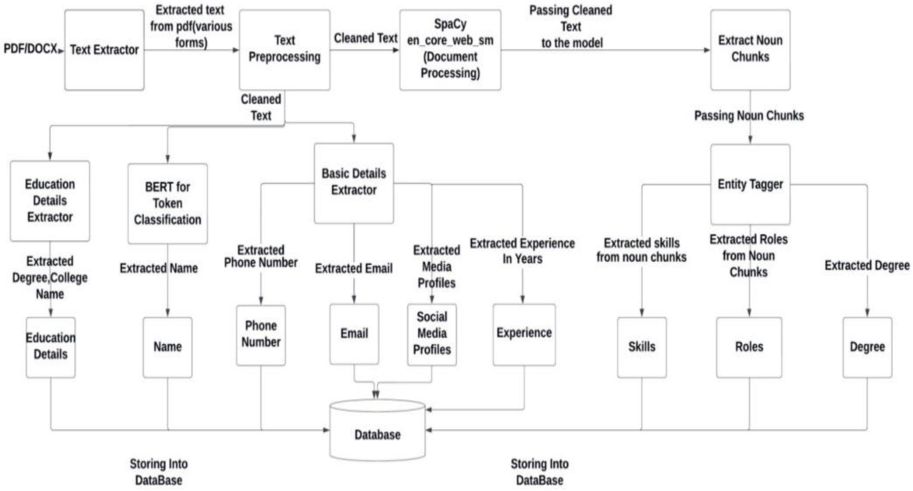

Fig. 1. Block diagram of proposed model

3.1 Text Extraction and Preprocessing

The resume text is extracted with the Python-based pdfminer program. It works with PDF and DOCX documents. The pdfminer automatically analyses the layout and extracts the text from the file. The pdfminer uses a lazy parsing mechanism, which means it only decodes information as needed.

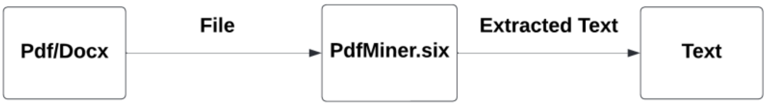

Fig. 2. Text Extraction

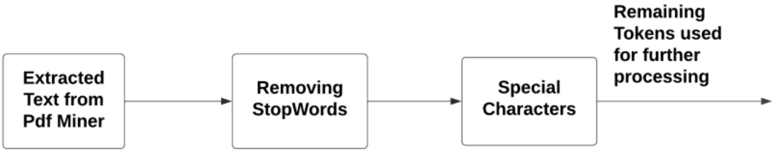
Extracted Text from Pdf Miner
Removing StopWords
Special Characters
Remaining Tokens used for further processing

Fig. 3. Text Preprocessing

Unrelated and special characters are then removed from the retrieved text. Any characters that aren't ASCII are eliminated. The process is shown in figure 3. Now the extracted text is converted to clean tokens which will act as a input to the subsequent phases.

3.2 Basic Details Extractor

The Cleaned text is then passed to the Basic Details Extractor. Phone numbers and emails are extracted using regular expressions and pattern matching techniques. Profile links such as LinkedIn are extracted using an URL extractor from the resumes. The Experience Extractor extracts the number of years of experience of the candidate.

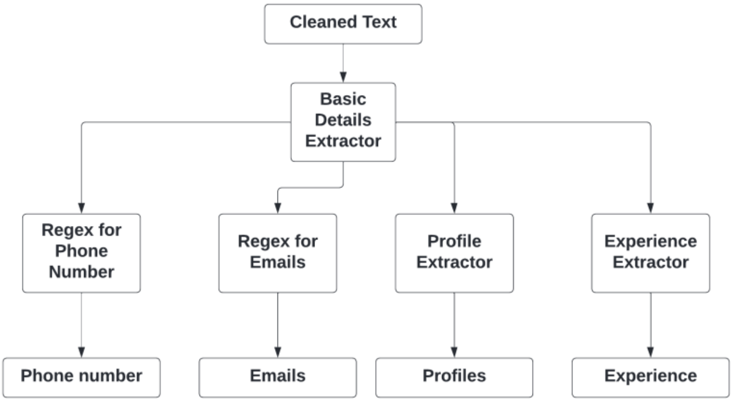

Fig. 4. Basic Details Extraction

3.3 Education Details Extraction

To extract education details, first noun chunks are compared with available list of colleges and college name is identified and extracted. Similar process is used for extracting the name of the degree. For extracting CGPA, the position of extracted college name is taken as a reference and search space is reduced to the lines of text surrounding the college name using string slicing. Pattern matching is done to identify the CGPA in the reduced text space and CGPA is extracted.

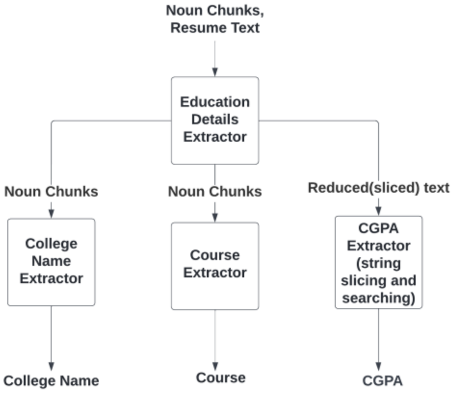

Fig. 5. Education Detail Extraction

3.4 Bert Model for Name Extraction

The task of extracting names was difficult because all nouns cannot be names. This conflict develops because some college names appear on people's names. To solve this problem a dataset comprising 14846 male and 15383 female names was used. The model is designed to distinguish the applicant's name from any other names on the CV. The POS-Tagging sequence was padded to satisfy BERT's input after all of the names and tagged names were tokenized. Over token categorization, the model was trained for 5 epochs.

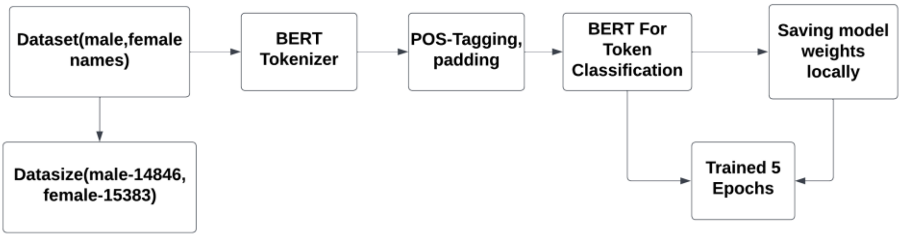
Dataset(male,female names)
BERT Tokenizer
POS-Tagging, padding
BERT For Token Classification
Saving model weights locally
Datasize(male-14846, female-15383)
Trained 5 Epochs

Fig. 6. Name Extraction

3.5 Entity Tagger

The Entity Tagger accepts noun chunks and identifies them as skill, degree, or role. The extracted noun chunks are first run via a unigram builder, which generates all possible unigrams, followed by a bigram and trigram builder, which generates all possible bigrams and trigrams. After that, all of the viable words are gathered into a single container. After that, the nouns are sent through three taggers: Skills Tagger, Degree Tagger, and Role Tagger. Each tagger takes a noun and assigns it to one of three categories: skill, degree, or role. The retrieved information is subsequently placed in the appropriate containers.

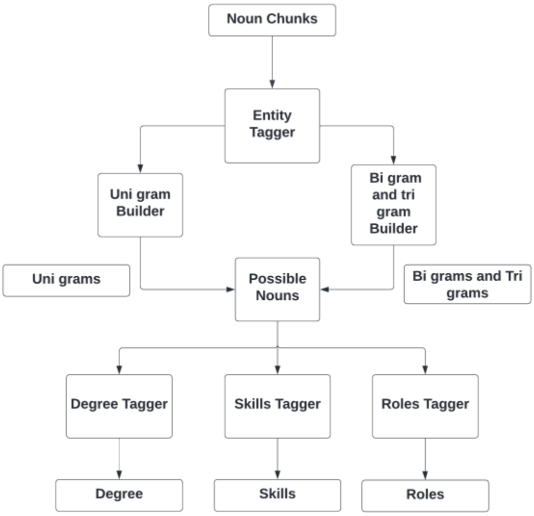

Fig. 7. Entity Tagger

4 Results

These models' investigations are carried out on resumes obtained by the CMRIT institution. Because of the inherent complications in finding the proper word, associated semantics, and diverse formats used by the candidates, the outcomes of each module vary. By weeding out the difficult process of segregating resumes for each branch, the model produces a result that can be further explored by applying simple queries or filters. Organizations might use skill sets, positions, or percentages to determine their selection criteria.

Name extraction model based on BERT had been trained with 5 epochs. The results of the name extraction model are as follows: validation loss returned 0.056, the accuracy as 0.98, and the F1-Score as 0.70. The model is explicitly designed and enhanced to deal with the name recognition in the Indian continent. Model works with more than 90% accuracy on the names of American and European continent. Basic Detail extractor works on parsing the information related to Email, Phone number, Email ID, Profile statement and experience from the applicant resume. Basic Details Extractor returns result with an accuracy of 100% for phone numbers and Email ID. For profile statement and experience, model works with the rate of 91% accuracy. Figure 8 shows the result from Basic detail extractor.

Entity Tagger modules returns degree information, skill, roles and responsibilities related details. The module was applied on the sample resumes received. This module returns results with an accuracy rate of 92%. Figure 9 shows the results of entity tagger. Education Details Extractor returns information related to the college, degree and CGPA or equivalent percentage. This module has an accuracy rate of 94%. Module works by using the concept of pattern matching. Figure 10 shows the results returned by this module.

This model is designed by looking at the requirements by the organisation. The parses resume efficiently over an average accuracy more than 90%. The results are then extracted by the potential recruiters by applying search constrains on the roles, percentages, degree or any of the parameter which was extracted. The results of the proposed model can be enhanced by training it on the resumes from various continents and on bigger sample size.

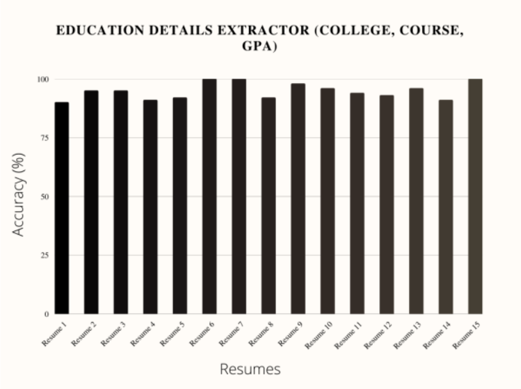

Fig. 8. Basic Details Extractor Accuracy

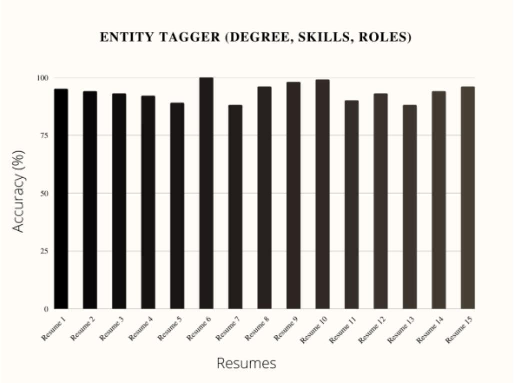

Fig. 9. Entity Tagger Accuracy

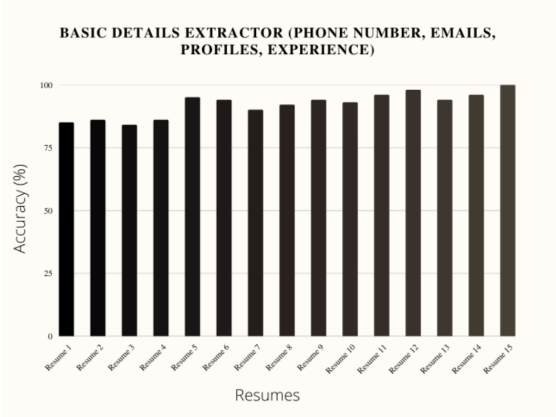

Fig. 10. Education Details Extractor Accuracy

5 Conclusion and Future Work

The proposed resume screening model is capable of extracting critical information from resumes and CVs. Our strategy is to make the recruitment process easier and more efficient for both enterprises and individuals. The procedure will reduce the need for physical labour and human resources, making it more efficient. As a result, the hiring process takes less time and costs less money. The information obtained can be utilised to create applicant profiles that meet the organization's needs. The model can parse the details of English resumes. The extraction could be expanded to more languages in the future. The model must be enhanced in order to achieve more accuracy in areas such as the candidate's experience.

References

1. D. Cer, Y. Yang,S. Y. Kong, N. Hua,N. Limtiaco, R. S. John, N. Constant,M. Guajardo-Cespedes, S. Yuan, C. Tar, Y. H. Sung,*Universal sentence encoder* (arXiv preprint arXiv:1803.11175, 2018).
2. S. Bhor, H. Shinde,V. Gupta, V. Nair, M. Kulkarni,*Resume parser using natural language processing techniques,* Int J Res Eng Sci (IJRES), (2021)
3. S. Sonar, B. Bankar,*Resume parsing with named entity clustering algorithm*. SVPM COE,(2012).
4. S. Sanyal, S. Hazra,S. Adhikary, N. Ghosh,*Resume parser with natural language processing*, International Journal of Engineering Science, (2017)
5. A. Schiller. *Knowledge Discovery from CVs: A Topic Modeling Procedure*, (2019)
6. N. Monaikul, G. Castellucci, S. Filice, O. Rokhlenko,*Continual learning for named entity recognition,* InProceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence, (2021)
7. Z. Liu, F. Jiang, Y. Hu,C. Shi, P. Fung,*NER-BERT: A Pre-trained Model for Low-Resource Entity Tagging*, arXiv preprint arXiv:2112.00405.,(2021)
8. J. Yu, B. Bohnet, M. Poesio,*Named entity recognition as dependency parsing*, arXiv preprint arXiv:2005.07150., (2020)
9. P. K. Roy, S. S. Chowdhary, R. Bhatia, *A Machine Learning approach for automation of Resume Recommendation system*, Procedia Computer Science, (2020)
10. A. Lad, S. Ghosalkar, B. Bane, K. Pagade and A. Chaurasia,*Machine Learning Based Resume Recommendation System,* International Journal of Modern Developments in Engineering and Science, (2022)
11. I. Ali, N. Mughal, Z. H, Khand, J. Ahmed, G. Mujtaba,*Resume classification system using natural language processing and machine learning techniques*, Mehran University Research Journal Of Engineering & Technology,(2022)
12. M. F. Mridha, R. Basri,M. M. Monowar, M. A. Hamid, *A Machine Learning Approach for Screening Individual's Job Profile Using Convolutional Neural Network*. IEEE International Conference on Science & Contemporary Technologies (ICSCT) (2021)