Resume Parser

Perumbuduru Meher Shireen
IT dept. National Institute of Technology
Karnataka, Surathkal
perumbudurumehershireen
.221it049@nitk.edu.in

Prajna B Shettigar
IT dept.National Institute of Technology
Karnataka, Surathkal
prajnabshettigar.221it051@nitk.edu.in

Suveena Sadashiv
IT dept. National Institute of Technology
Karnataka, Surathkal
suveenasadashiv.221it070@nitk.edu.in

***Abstract*—Initially,the job recruitments of each company would be carried out by manually going through the resumes of all the candidates.Nowadays,with the world inclining towards technology,the trend of the recruitment process has also changed .Companies now have resume parsing tools to do the same.Hence an applicant is under pressure to provide a resume with the right format which the machine will be able to parse and analyse.Hence ,this project acts as an aid for the applicants for them to check whether their resume is parsable by the machine or not.Our project aims to revolutionize resume processing using Natural Language Processing (NLP). We're creating a system that can accurately extract key information from resumes, provided the resumes are in pdf format. This includes names, contact information, education, work history, skills, and more. The objective of the project is to aid a job seeker .The project helps the applicant to know whether his resume is parsable or not.In order to do this ,we are using multiple machine learning and deep learning techniques.**

KEYWORDS:natural language processing,named entity recognition,text classification,Multinomial naive bayes classifier,spacy libraries .

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

In today's fast-paced job market, the recruitment process is marked by a deluge of resumes inundating the desks of human resource professionals and hiring managers. This flood of candidate profiles necessitates efficient and effective strategies for resume evaluation, screening, and selection. To address this challenge, the field of ”Resume Parsing” has emerged as an indispensable tool .The use of Natural Language Processing (NLP) in resume parsing offers significant advantages and benefits that enhance the efficiency, accuracy, and effectiveness of the recruitment process. Natural Language Processing (NLP) is a field of artificial intelligence (AI) and computational linguistics that focuses on the interaction between computers and human language. It encompasses the development and application of algorithms and models to enable computers to understand, interpret, generate, and respond to human language in a valuable way. NLP powered resume parsing accelerates the initial screening process by quickly identifying candidates who best meet specific job criteria identifying top talents.. NLP-based resume parsers can be customized to match the specific requirements of different job roles, industries, or organizations. NLP empowers organizations to make more informed hiring decisions while improving the overall candidate experience. After signal/data preprocessing using tokenization and data extraction techniques like Named entity recognition(NER), NLP generates structured data by extracting key information from resumes, including skills, work experience,education, and contact details.ML (Machine learning) and DL(Deep learning) algorithms can then be used to extract additional features, such as sentiment analysis, to provide a deeper understanding of the candidate's qualifications.

Identify applicable funding agency here. If none, delete this.

II. Literature Review

A.

Maruprolu Naga Raju ReddyHarsha Vardhan Chirumamilla Tejus Paturu B.Surendiran worked on the following four algorithms in order achieve the resume classification.These methods are SVM Classifier,KNN Classifier,Decision Tree Classifier,Random Forest Classifier[1]. Random Forest is composed of multiple decision trees, forming a forest. Due to the involvement of several decision trees, Random Forest is considered to be more accurate and resilient. This technique does not suffer from overfitting issues and can handle missing values in the data. However, due to the presence of multiple trees, Random Forest generates predictions at a slower pace compared to a single decision tree. Additionally, interpreting the results of Random Forest is more complex than a decision tree due to the involvement of multiple decision trees. The decision tree classifier design resembles a flowchart, where tests on features are represented by internal nodes, class labels by leaves, and branching by the relationship between features and class labels. KNN works by classifying data based on the similarity between a data point and its nearest neighbours. This method can handle real-world data effectively because it does not make any assumptions about the input data set, making it a more attractive option. The Support Vector Machine (SVM) technique seeks to locate an effective hyperplane in an N-dimensional space for separating the data points. The size of the hyperplane is determined by the number of features in the data set.

Prof. Sanjay Mirchandani ,Jasmine Sawara,Chandni Megnani ,Vanshika Bajaj,Rishabh Bathija worked on different algorithms like K-nearest neighbors (KNN), Weighted KNN (WKNN), and Support Vector Machine KNN (SVM-KNN)[2]. The KNN method can suffer from overfitting, especially when k is too little and the algorithm is too sophisticated. When k is too little, the algorithm may try to fit the data noise, resulting in significant variance and overfitting. With an accuracy of 74KNN and SVM-KNN. WKNN also outperformed KNN and SVM-KNN in terms of accuracy, recall, and F1-score.

Raseek C Sreejith C Ayishathahira C H worked on combination of neural networks and conditional random fields for efficient resuming parsing.These include,e convolutional neural network (CNN), Bi-LSTM (Bidirectional Long Short Term Memory) and Conditional Random Field (CRF)[3]. They have used neural networks and CRF to segment and extract various information from resumes. CNN model is used for segmentation and compared with a Bi LSTM model. A CRF based model is chosen for information extraction and compared with a Bi-LSTM-CNN model.

T.Mallikarjuna ,P.Divya Sai,V. V. SatyanarayanaTallapragada worked on contextual meaning extraction using bert.They used Bidirectional Encoder Representations from Transformer (BERT) vectorization to identify the text contextually[4]. BERT uses a deep learning architecture based on transformers. This lets it handle longer text sequences and understand more complex relationships between words.

Rajagopalan V ,Supriya Mandal,Gunaseelan B built multiple classification models using various techniques, such as K-Nearest Neighbors (KNN), Support Vector Machine(SVM) with RBF kernel and ensembling techniques like Bagging, Random Forest (RB), Gradient Boosting (GB), Extreme Gradient Boosting (XGB), Ada boost and LightGBM.[5]

Hira Sajid, Javeri Kanwal, Saad Ali and few others built a resume parser for E-recruitment using framework which includes different methods like text extraction, preprocessing, text block classification using bnb,NER, bert models, and other ontology techniques. The Boolean Naive Bayes algorithm used here has both machine and deep learning models, where deep learning model has more accuracy which helps in classifying text more precisely which helps in further steps like resume facts identification and skill ontologyfor data enrichment.[6]

Shubham Bhor, Harish Shinde, Vivek Gupta, Vishak Nair, and Prof. Manasi Kulkarni, used Optical Character Recognition to convert the resume to a single text format.Then Lexial, Syntactic, Semantic analysis followed by NER are done. NER is used to tune the NLP model as the jargons and words that mean something for that company's domain and may mean something else in general, this hindrance is overcome in this Named Entity Recognisation model [7]

Varsha Tiwari, Prof. Sapna Jain Choudhary,built a Parser which generates the parse tree with the help of syntactic analysis. A parse tree or parsing tree is an ordered, rooted tree that represents the syntactic structure of a string according to some context free grammer.Each candidate will be scored based on the skillset, experience and project which will also be influenced by his Github and linkedIn profile. And used precision, recall and F-measure metrics for performance evaluation. [8]

Stephan Gunawardana, created a website where a candidate can upload his/her resume in .md form. Resumes are organisedn in a way that they are titled by headers followed by more precise language such as details about each section. The values for skills are taken into database. A library called "vue-markdown" which is used to convert the ".md" file directly to html. Therefore, with vue being a dynamic framework, this directly displays the rsult in website.[9]

Chirag Daryani, Gurneet Singh Chhabra, Harsh Patel worked on a resume parsing tool that does tokenisation uses cosine similarity, and vector space model so it transform these terms (part of our created vocabulary) into a numeric form that the machine can understand. Finally the most relevant features in the candidate resume that are needed for the job are selected and thus creates the candidate profile. [10]

In the paper Saswat et al [11] with their objective being to properly study resumes to hire the finest candidate . The résumé submitted with the application is used as the input and is cleaned and tokenized during preprocessing. In order to compute the text similarity and assess our model, the keyword is extractedusing name entity recognition, recovered using XGBoost, and then classified in accordance with preset elements.They discovered that XGB is more precise and has a lower standard deviation after subjecting it to testing and training along with machine learning strategies including KNN, SVM, Decision Trees, and Naive Bayes.The use of machine learning algorithms for resume parsing has altered and made the hiring processmore effective and efficient.

In the paper Narendra et al [12] the proposed methodology assists recruiters in quickly reviewing candidates based on the job description. It automates the recruiting process by extracting the necessary entities using NER. This paper proposed a system that takes up a resume and performs Named Entity Recognition (NER) on it and summarizes the content. It also ranks the resumes based on company requirements from a group of resumes. Tokenization, Parts of Speech tagging, text categorization, Named Entity Recognition, and other features are offered by spaCy, Python-based open-source framework that does sophisticated natural language processing.

In paper Nimriti et al[13] to ensure a good performance of web application for Resume Parsing they proposed a hybrid methodology including the best of both techniques of machine learning and deep learning ensuring an optimum output.A long with Tokenization ,stop words Removal,removal of punctuation using nltk in Preprocessing, Segmentation process is carried out by using Fuzzy String Matching that in turn uses bigrams or ngrams .Further they have also adopted techniqus like Regular Expression,Phase Matcher for rule based and key word based Extraction.The hybrid approach using transformer based models like BERT with NLP functions of Spacy significantly improved the performance .

In the paper Bhushan et al [14] the proposed model is made up of 2 tecniques.The first one using either K-Nearest Neighbour or Support Vector Machine which will help us to get the prediction of amount of resumes, and rank them according towhat kind of job role our resume is best fit for .The second one uses cosine similarity which will check the user's input of what job role they want, what the model predicted on that basis the Recommendation system and strong. This Paper deals with multiple methods to detect,identify and classify various resumes using multiple machine learning and Neural Network models likeJanuary 2019, SVM, KNN, Word2Vec, Cosine similarity, etc.

In the paper Satyaki et al [15] they used the following constraints of NLP to parse the information from the resumes:I Lexical Analysis II. Syntactic Analysis III. Semantic Analysis.The Lexical analyzer pre-processes the data and tokenizes them. The Syntactic analyzer takes t okens , finds the structure in it. The parse tree diagrammat ically represents the syntactic structure in the form of a tree. The Semantic analyzer studies the structure of the data to find their language-independent meaning.In the preprocessing of data in json format they used many techniques like Tokenization , Stemming ,Lemmatization,POS Tagging,Chunking etc. To obtain suitable data from Resume summarization.

III. Proposed Framework

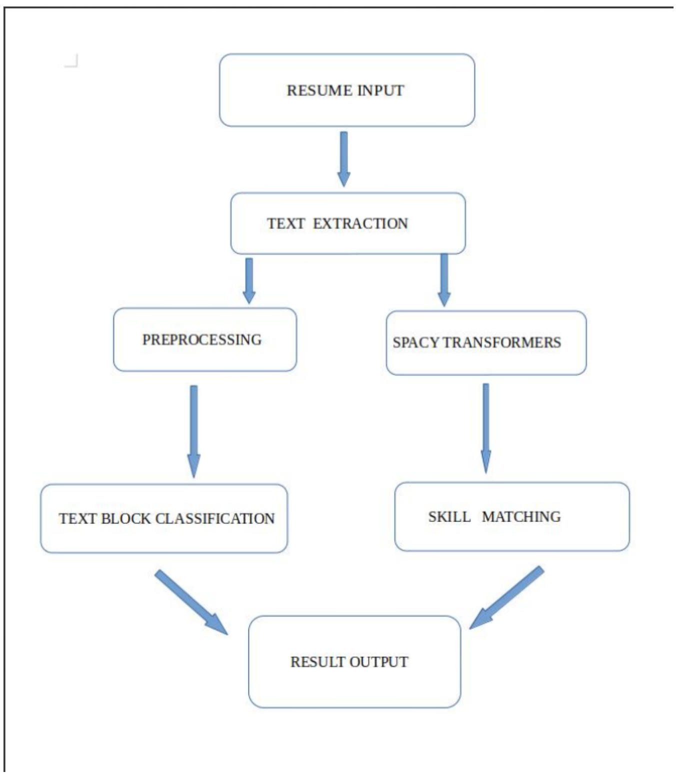


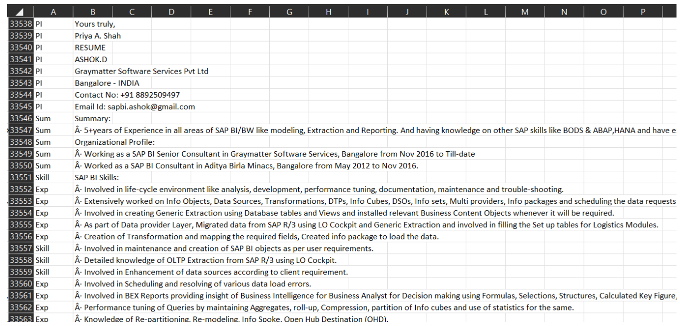
Fig. 1. Flow Chart

This section covers the input, output and detailed description of the proposed resume parsing framework. Fig. 1 shows the pipeline of resume parsing.

A. Input and Output This pipeline focuses to extract information from resumes without considering the font styles, font sizes or standard structure of the document.

1. *Input:* Input of the resume parser is an unstructured document in pdf form of any format.

2. *Output:* Output is the structured format that includes all the facts and summary about a candidate, extracted from the resume. Gives score to the candidate based on his/her skills matching with the job description, aiding the candidate to evaluate his/her own resume.



	A	B
33538	PI	Yours truly,
33539	PI	Priya A. Shah
33540	PI	RESUME
33541	PI	ASHOK.D
33542	PI	Graymatter Software Services Pvt Ltd
33543	PI	Bangalore - INDIA
33544	PI	Contact No: +91 8892509497
33545	PI	Email Id: sapbi.ashok@gmail.com
33546	Sum	Summary:
33547	Sum	Â· 5+years of Experience in all areas of SAP BI/BW like modeling, Extraction and Reporting. And having knowledge on other SAP skills like BODS & ABAP,HANA and have e
33548	Sum	Organizational Profile:
33549	Sum	Â· Working as a SAP BI Senior Consultant in Graymatter Software Services, Bangalore from Nov 2016 to Till-date
33550	Sum	Â· Worked as a SAP BI Consultant in Aditya Birla Minacs, Bangalore from May 2012 to Nov 2016.
33551	Skill	SAP BI Skills:
33552	Exp	Â· Involved in life-cycle environment like analysis, development, performance tuning, documentation, maintenance and trouble-shooting.
33553	Exp	Â· Extensively worked on Info Objects, Data Sources, Transformations, DTPs, Info Cubes, DSOs, Info sets, Multi providers, Info packages and scheduling the data requests
33554	Exp	Â· Involved in creating Generic Extraction using Database tables and Views and installed relevant Business Content Objects whenever it will be required.
33555	Exp	Â· As part of Data provider Layer, Migrated data from SAP R/3 using LO Cockpit and Generic Extraction and involved in filling the Set up tables for Logistics Modules.
33556	Exp	Â· Creation of Transformation and mapping the required fields, Created info package to load the data.
33557	Skill	Â· Involved in maintenance and creation of SAP BI objects as per user requirements.
33558	Skill	Â· Detailed knowledge of OLTP Extraction from SAP R/3 using LO Cockpit.
33559	Skill	Â· Involved in Enhancement of data sources according to client requirement.
33560	Exp	Â· Involved in Scheduling and resolving of various data load errors.
33561	Exp	Â· Involved in BEX Reports providing insight of Business Intelligence for Business Analyst for Decision making using Formulas, Selections, Structures, Calculated Key Figure,
33562	Exp	Â· Performance tuning of Queries by maintaining Aggregates, roll-up, Compression, partition of Info cubes and use of statistics for the same.
33563	Exp	Â· Knowledge of Re-partitioning, Re-modeling, Info Spoke, Open Hub Destination (OHD).

Fig. 2. Dataset, (csv file)

3. *Data set:* A csv file containing sentences taken from nearly 200 resumes labelled into the 7 categories. And The dataset used for section using NLP model with Spacy Transformer is a json file containing data from 200 Resumes which are in labelled format. The dataset used has the resumes in text format as well as the annotated data with labels like Name, Designation, Companies worked at ,College Name ,Skills ,Email , Phone number etc. The text Annotation of dataset was done using online NER Annotation tools like doccan, prodigy etc.

Fig. 3. PDF to Raw text

1) B. Text Extraction: The resumes vary in their format in multiple ways i.e. layout, writing style, font size, font style and structure of the document. These variations in resumes occur due to the absence of standard format for the resume writing. The text in these resumes needs to be extracted in raw format to perform further text mining techniques. In the proposed methodology, PyPDF2, PyMuPDF (MuPDF) libraries are used to extract raw text from resumes present in PDF format. Using modules in these libraries, it reads a PDF file, iterates through each page, extracts the text content from each page, and concatenates it into a single string . After the loop completes, the string would contain the text content of the entire PDF document.

C. Data Preprocessing

The extracted data is preprocessed to exclude the noise brought with PDFbox i.e. non-ascii characters, unnecessary punctuation marks, missing and extra spaces, and newlines.

a) Non-Ascii characters: Extraction of raw text from pdf generates some non-ascii characters in the output text during text extraction in response to bullet points, emoticons or

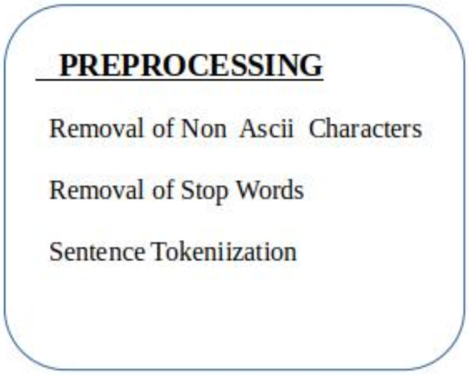


Fig. 4. Preprocessing before Text Block Classification

icons. All non-ascii characters present in the extracted text are removed before the training of the text block classification.

b) Stop-Words : Removing stop-words before text block classification improves model efficiency and accuracy by eliminating common words which are frequently occurring in a language (e.g., "and," "the," "is") that often do not carry significant meaning in the context of a specific task and contribute less to the distinctive features of text blocks, allowing the classifier to focus on more informative content words for better decision-making.

c) Sentence tokenization : Tokenizing at the sentence level is done as different segments of the text may belong to different categories or when the context within sentences varies significantly. It allows the classifier to make predictions based on more localized context, making the overall classification process more precise and contextually aware.

D. Text Block Classification

Text Block classification is an important step in resume parsing as further levels depend on its output. These text blocks are classified into multiple categories such as personal in- formation, experience, education, skills, qualifications, objectives, additional points.

1) Training : The dataset is loaded from a CSV file containing labelled text samples and corresponding categories. NaN values in the columns are replaced with empty strings. The dataset is split into training and testing sets using train_test_split. The text data in the training set is transformed into a numerical representation using CountVectorizer, creating a bag-of-words model. This output is further transformed using the TF-IDF (Term Frequency-Inverse Document Frequency) transformer. A Multinomial Naive Bayes classifier is then trained on the transformed training set with corresponding category labels. The trained model is applied to the test set to predict category labels . The classification accuracy is calculated using scikit-learn's accuracy_score function, comparing predicted labels with true labels from the test set. The resulting accuracy score provides insight into how well the model generalizes to unseen data.

2) Multinomial Naïve Bayes Algorithm : The Multinomial Naive Bayes (MNB) algorithm is a probabilistic classification method based on Bayes' theorem. It is specifically designed for classification tasks where the features are discrete and represent the frequency of occurrences of events. In the context of MNB, we assume that the features are conditionally independent given the class, which is a naive assumption but simplifies the computation.

$$\hat{\theta}_{yi} = \frac{N_{yi} + \alpha}{N_y + \alpha n}$$

Fig. 5. MNB

3) Classification of uploaded resume : During testing, each sentence from tokenized from the extracted text is processed individually through the trained text classification model. The process involves transforming the sentence into a numerical representation using the same vectorization and TF-IDF transformations that were applied during training. The trained classifier then predicts the category label for each sentence. The resulting predictions are stored, representing the assigned categories for each segment in the input text. This allows for the identification and categorization of different segments within the document based on the patterns learned during training.

E. Building parser with Spacy V3

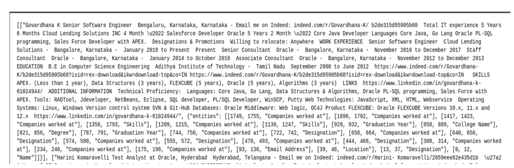

[["Govardhana K Senior Software Engineer Bengaluru, Karnataka, Karnataka - Email me on Indeed: indeed.com/r/Govardhana-K/ b2de315d95905b68 Total IT experience 5 Years 6 Months Cloud Lending Solutions INC 4 Month \u2022 Salesforce Developer Oracle 5 Years 2 Month \u2022 Core Java Developer Languages Core Java, Go Lang Oracle PL-SQL programming, Sales Force Developer with APEX. Designations & Promotions Willing to relocate: Anywhere WORK EXPERIENCE Senior Software Engineer Cloud Lending Solutions - Bangalore, Karnataka - January 2018 to Present Present Senior Consultant Oracle - Bangalore, Karnataka - November 2016 to December 2017 Staff Consultant Oracle - Bangalore, Karnataka - January 2014 to October 2016 Associate Consultant Oracle - Bangalore, Karnataka - November 2012 to December 2013 EDUCATION B.E in Computer Science Engineering Adithya Institute of Technology - Tamil Nadu September 2008 to June 2012 https://www.indeed.com/r/Govardhana-K/b2de315d95905b68?isid=rex-download&ikw=download-top&co=IN https://www.indeed.com/r/Govardhana-K/b2de315d95905b68?isid=rex-download&ikw=download-top&co=IN   SKILLS APEX. (Less than 1 year), Data Structures (3 years), FLEXCUBE (5 years), Oracle (5 years), Algorithms (3 years) LINKS https://www.linkedin.com/in/govardhana-k-61024944/ ADDITIONAL INFORMATION Technical Proficiency: Languages: Core Java, Go Lang, Data Structures & Algorithms, Oracle PL-SQL programming, Sales Force with APEX. Tools: RADTool, Jdeveloper, NetBeans, Eclipse, SQL developer, PL/SQL Developer, WinSCP, Putty Web Technologies: JavaScript, XML, HTML, Webservice Operating Systems: Linux, Windows Version control system SVN & Git-Hub Databases: Oracle Middleware: Web logic, OC4J Product FLEXCUBE: Oracle FLEXCUBE Versions 10.x, 11.x and 12.x https://www.linkedin.com/in/govardhana-k-61024944/", {"entities": [[1749, 1755, "Companies worked at"], [1696, 1702, "Companies worked at"], [1417, 1423, "Companies worked at"], [1356, 1793, "Skills"], [1209, 1215, "Companies worked at"], [1136, 1247, "Skills"], [928, 932, "Graduation Year"], [858, 889, "College Name"], [821, 856, "Degree"], [787, 791, "Graduation Year"], [744, 750, "Companies worked at"], [722, 742, "Designation"], [658, 664, "Companies worked at"], [640, 656, "Designation"], [574, 580, "Companies worked at"], [555, 572, "Designation"], [470, 493, "Companies worked at"], [444, 468, "Designation"], [308, 314, "Companies worked at"], [234, 240, "Companies worked at"], [175, 198, "Companies worked at"], [93, 136, "Email Address"], [39, 48, "Location"], [13, 37, "Designation"], [0, 12, "Name"]]}], ["Harini Komaravelli Test Analyst at Oracle, Hyderabad Hyderabad, Telangana - Email me on Indeed: indeed.com/r/Harini- Komaravelli/2659eee82e435d1b \u27a2

Fig. 6. dataset, (json)

Processing of Training Dataset: The function 'spac_doc' essentially processes annotated text data, extracts entities based on the annotations, handles overlapping entities, creates spaCy Span objects representing entities in each doc object , and stores these annotated documents in a DocBin object for further processing or storage. Later this dataset is split Train and Test dataset (ratio:0.3) .

SPACY TRANSFORMER: In our NLP model we are using a config.cfg configuration file obtained from spacy.io Quickstart section that include all the settings and hyperparameters. The main objective here is to perform Named Entity Recognition(NER) for given entities .

Training Pipeline: The settings define the training pipeline, including optimizer configurations(Adam), batch sizes, learning rate schedules, and other parameters to train and optimize the NER model.

Data Handling: Components also consist configurations for initialization, loading vectors, tokenizers, and other components before training.The file also outlines how training and development corpora are handled, including settings for data readers, paths, and pre-processing in the Corpora Configuration component.

Tokenization and Vectorization: The configurations specify how tokenization and vectorization of text data will be performed using transformer models or other architectures. In this section we used tok2vec .The tok2vec transformation aims to convert individual tokens into fixed-sized vector representations that capture rich contextual and semantic information, often leveraging pre-trained transformer models. These token representations serve as essential features for downstream tasks like Named Entity Recognition (NER), where contextual information is crucial for accurately identifying named entities within text.

NER Model Training: The configuration file defines various components and their settings required for training a custom NER model, potentially leveraging transformer-based architectures like ”roberta-base” for tokenization and feature extraction.In the NER Component of the file we have used “spacy.TransitionBased Parser” that uses maxout activation function with dropout regularization technique.

Bert Model :

The ”roberta-base” model referenced in the provided configuration file is a variant of the RoBERTa (Robustly optimized BERT pretraining approach) model. RoBERTa is a transformer-based language representation model developed by Facebook AI that builds upon the BERT (Bidirectional Encoder Representations from Transformers) architecture with modifications and enhancements for better performance.During training, the RoBERTa base model is typically used to generate contextual embeddings for the input tokens. These embeddings capture the contextual meaning of each token in the context of the entire sentence or document.

FeatureExtraction: The contextual embeddings obtained from RoBERTa serve as essential features for the downstream task of Named Entity Recognition (NER). These embeddings provide a rich representation of words, allowing the NER model to understand the context surrounding each word, which is crucial for accurate entity recognition.

Transfer Learning:By using a pre-trained RoBERTa base model, the NER model benefits from transfer learning. The pre-trained RoBERTa model has learned general language patterns from a large corpus, and this knowledge is transferred to the NER model, which helps improve its performance, especially when training data is limited.

We trained this model using the command:

python -m spacy https://spacy.io/api/cli#traintrain config.cfg –output ./output –paths.train ./train.spacy –paths.dev ./dev.spacy

We load the text int this model using command: spacy.load(’model-best’)

We trained the model in Google Colab in GPU runtime about an hour and obtained a ’model-best’ having about 60% accuracy.It is expected that for longer training hours the accuracy may increase above 70%.

F. Skill Set Matching

```
[ ] 2023-11-10 06:46:18.840782: E tensorflow/compiler/xla/stream_executor/cuda/cuda_blas.cc:1518] Unable to register cuBLAS factory: Attempting to register factory for plugin cuBLAS when one has already been
    2023-11-10 06:46:20.498712: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not find TensorRT
    ℹ Saving to output directory:
    /content/drive/MyDrive/Resumes/Spacy3Resumeparser/model/output2
    ℹ Using GPU: 0

    =========================== Initializing pipeline ===========================
    Downloading (…)lve/main/config.json: 100% 481/481 [00:00<00:00, 2.53MB/s]
    Downloading (…)olve/main/vocab.json: 100% 899k/899k [00:00<00:00, 6.82MB/s]
    Downloading (…)olve/main/merges.txt: 100% 456k/456k [00:00<00:00, 6.97MB/s]
    Downloading (…)/main/tokenizer.json: 100% 1.36M/1.36M [00:01<00:00, 1.20MB/s]
    Downloading model.safetensors: 100% 499M/499M [00:02<00:00, 237MB/s]
    Some weights of RobertaModel were not initialized from the model checkpoint at roberta-base and are newly initialized: ['roberta.pooler.dense.bias', 'roberta.pooler.dense.weight']
    You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
    ✔ Initialized pipeline

    ============================= Training pipeline =============================
    ℹ Pipeline: ['transformer', 'ner']
    ℹ Initial learn rate: 0.0
    E    #       LOSS TRANS...  LOSS NER  ENTS_F  ENTS_P  ENTS_R  SCORE
    ---  ------  -------------  --------  ------  ------  ------  ------
      0       0        2923.28   1593.73    0.38    0.20    7.05    0.00
      3     200       89263.99  61881.70   30.94   41.80   24.56    0.31
      7     400        9881.59  21430.59   53.17   59.14   48.30    0.53
     11     600        4936.71  18682.19   51.99   62.44   44.54    0.52
     15     800        2127.37  17065.46   49.15   42.00   59.22    0.49
     19    1000        1372.84  15868.03   56.29   63.16   50.76    0.56
     23    1200        1244.81  14809.02   53.66   58.45   49.59    0.54
     27    1400        1744.65  14092.09   48.31   62.22   39.48    0.48
     31    1600       19105.09  14087.15   51.05   63.57   42.66    0.51
     35    1800        8459.20  13465.84   53.24   61.80   46.77    0.53
     39    2000        9589.44  12915.83   55.29   54.44   56.17    0.55
     43    2200       20564.63  12344.92   54.37   52.28   56.64    0.54
     47    2400         252.89  11499.11   56.34   53.92   58.99    0.56
     50    2600       49953.08  11028.88   57.13   59.02   55.35    0.57
     54    2800         140.70   9646.14   56.51   63.01   51.23    0.57
     58    3000         337.47   8998.23   54.93   64.50   47.83    0.55
     62    3200        5699.32   8272.30   55.73   59.65   52.29    0.56
     66    3400         216.18   6980.41   56.87   61.36   53.00    0.57
     70    3600         116.90   5830.24   56.09   56.97   55.23    0.56
     74    3800       33989.67   4871.93   54.40   57.98   51.23    0.54
     78    4000        7012.04   3788.75   57.04   60.37   54.05    0.57
     82    4200        4343.48   2762.71   58.39   64.14   53.58    0.58
     86    4400        3052.95   1884.60   55.88   59.68   52.53    0.56
     90    4600         173.80   1170.37   55.49   62.06   50.18    0.55
     94    4800         351.67    744.55   58.92   61.81   56.29    0.59
     98    5000         341.44    511.87   55.74   64.02   49.35    0.56
    101    5200         203.97    313.29   58.61   63.85   54.17    0.59
    105    5400         109.52    146.39   57.23   63.21   52.29    0.57
    109    5600          77.56     95.73   56.42   61.41   52.17    0.56

    Aborted.
```

Fig. 7.   Output

Given a job description and a candidate resume, the code aims to match the skills mentioned in the candidate's resume with the required skills in the job description.Extract skills from the job description and candidate's resume.Compare the extracted skills from the candidate's resume with the required skills from the job description.Assign a score to each matched skill . Calculate the total score based on matched skills. Determine the skill level of the candidate based on predefined threshold values (low, intermediate, high).

SpaCy's PhraseMatcher is used to extract skills from a given text. It follows the following steps to do this.

Create a PhraseMatcher object using spaCy's vocabulary. Define patterns for skills using the spaCy's nlp function. Add these patterns to the Phrase Matcher. Process the input text with spaCy. Find matches of the defined skills within the processed text using the PhraseMatcher.

Finally the matching of the candidate skills with the job requirements is done and a total score is calculated. The output includes the job required skills and the candidate skills and the score along with the normalized total score which is calculated by dividing the score the 6(which in case is the total number of skills required for the job).Finally ,there is determination of the skill levels (low, high and intermediate) based on the total score.

Results and Conclusion

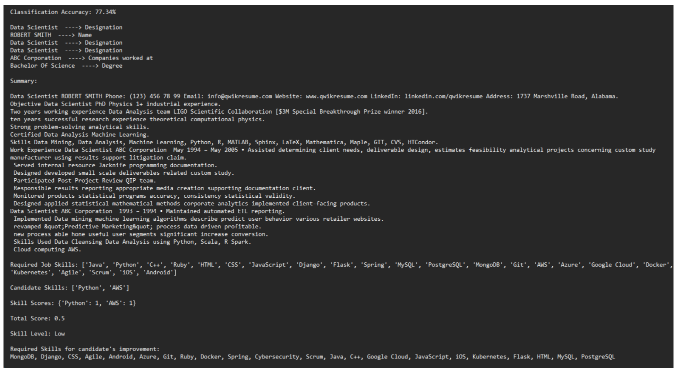
The output aims to provide a concise and informative summary of the candidate's qualifications, with a focus on technical skills and their relevance to the job requirements.

The resume is broken down into sections such as Personal Information, Education, Skills, Work Experience. The system extracts and presents key information about the candidate, including their name, designation, location, degree, and contact information. The candidate's skills are listed, providing an overview of their technical proficiency and expertise. The skills include programming languages, tools, and methodologies. Details about the candidate's work experience are included, highlighting positions held, responsibilities, and achievements are shown.

A comparison is made between the skills required for the job and the skills possessed by the candidate. This comparison is likely used to assess the candidate's suitability for the role.

Skill scores are assigned based on the presence of specific skills in the candidate's profile. A total score is calculated, and a skill level is determined. The system identifies areas where the candidate could improve their skills to better align with the job requirements. This could be valuable feedback for both the candidate and the hiring team.

This output has a classification accuracy of 77.34% using NLP models mentioned above. Upon training with more number of accurate datasets the accuracy of the model can be enhanced.



```
Classification Accuracy: 77.34%

Data Scientist  ----> Designation
ROBERT SMITH  ----> Name
Data Scientist  ----> Designation
Data Scientist  ----> Designation
ABC Corporation  ----> Companies worked at
Bachelor Of Science  ----> Degree

Summary:

Data Scientist ROBERT SMITH Phone: (123) 456 78 99 Email: info@qwikresume.com Website: www.qwikresume.com LinkedIn: linkedin.com/qwikresume Address: 1737 Marshville Road, Alabama.
Objective Data Scientist PhD Physics 1+ industrial experience.
Two years working experience Data Analysis team LIGO Scientific Collaboration [$3M Special Breakthrough Prize winner 2016].
ten years successful research experience theoretical computational physics.
Strong problem-solving analytical skills.
Certified Data Analysis Machine Learning.
Skills Data Mining, Data Analysis, Machine Learning, Python, R, MATLAB, Sphinx, LaTeX, Mathematica, Maple, GIT, CVS, HTCondor.
Work Experience Data Scientist ABC Corporation  May 1994 - May 2005 • Assisted determining client needs, deliverable design, estimates feasibility analytical projects concerning custom study
manufacturer using results support litigation claim.
 Served internal resource Jacknife programming documentation.
 Designed developed small scale deliverables related custom study.
 Participated Post Project Review QIP team.
 Responsible results reporting appropriate media creation supporting documentation client.
 Monitored products statistical programs accuracy, consistency statistical validity.
 Designed applied statistical mathematical methods corporate analytics implemented client-facing products.
Data Scientist ABC Corporation  1993 - 1994 • Maintained automated ETL reporting.
 Implemented Data mining machine learning algorithms describe predict user behavior various retailer websites.
 revamped &quot;Predictive Marketing&quot; process data driven profitable.
 new process able hone useful user segments significant increase conversion.
 Skills Used Data Cleansing Data Analysis using Python, Scala, R Spark.
 Cloud computing AWS.

Required Job Skills: ['Java', 'Python', 'C++', 'Ruby', 'HTML', 'CSS', 'JavaScript', 'Django', 'Flask', 'Spring', 'MySQL', 'PostgreSQL', 'MongoDB', 'Git', 'AWS', 'Azure', 'Google Cloud', 'Docker',
'Kubernetes', 'Agile', 'Scrum', 'iOS', 'Android']

Candidate Skills: ['Python', 'AWS']

Skill Scores: {'Python': 1, 'AWS': 1}

Total Score: 0.5

Skill Level: Low

Required Skills for candidate's improvement:
MongoDB, Django, CSS, Agile, Android, Azure, Git, Ruby, Docker, Spring, Cybersecurity, Scrum, Java, C++, Google Cloud, JavaScript, iOS, Kubernetes, Flask, HTML, MySQL, PostgreSQL
```

Fig. 8. Final Output

Future Scope :

-In addition to the existing models,we can inculcate APIs inorder to lead the candidate to the required platform according to his or her shortcomings .An example for this would be ,the advancement will be able to trace any particular required skill that the candidate would be lacking and hence the API will lead the candidate to the required website or a an online source that would help him/her in that particular field.

-A second addition to this would be dynamic generation of required job skills instead of static job description.

-A third addition to this would be to unroll the methodology to various fields and domains across all the sectors .

REFERENCES

1)Maruprolu Naga Raju ReddyHarsha Vardhan Chirumamilla Tejus Paturu B.Surendiran,Resume Classification Using ML Techniques,2023 International Conference on Signal Processing, Computation, Electronics, Power and Telecommunication (IConSCEPT)

2)Prof. Sanjay Mirchandani ,Jasmine Sawara,Chandni Megnani ,Vanshika Bajaj,Rishabh Bathija,Revolutionizing Recruitment: A Comparative Study Of KNN, Weighted KNN, and SVM - KNN for Resume Screening,Proceedings of the 8th International Conference on Communication and Electronics Systems (ICCES 2023)

3)Raseek C Sreejith C Ayishathahira C H,Combination of Neural Networks and Conditional Random Fields for Efficient Resume Parsing,2018 International CET Conference on Control, Communication, and Computing (IC4) — July 05 – 07, 2018 — Trivandrum

4)T.Mallikarjuna ,P.Divya Sai,V. V. SatyanarayanaTallapragada,Improved Resume Parsing based on Contextual Meaning Extraction using BERT,Proceedings of the 7th International Conference on Intelligent Computing and Control Systems (ICICCS-2023)

5)Rajagopalan V ,Supriya Mandal,Gunaseelan B,Automatic Extraction of Segments from Resumes using Machine Learning,2020 IEEE 17th India Council International Conference (INDICON).

6)Hira Sajid, Javeri Kanwal, Saad Ali ,"Resume Parsing Framework for E-recruitment"

7)Shubham Bhor, Harish Shinde, Vivek Gupta, Vishak Nair, and Prof. Manasi Kulkarni,Resume Parser Using Natural Language Processing Techniques

8)Varsha Tiwari, Prof. Sapna Jain Choudhary,Intelligent Hiring with Resume Parser and Ranking Using Machine Learning and Natural Language Processing

9)Chirag Daryania, Gurneet Singh Chhabrab, Harsh Patelc , Indrajeet Kaur Chhabrad, Ruchi Patele,AN AUTOMATED RESUME SCREENING SYSTEM USING NATURAL LANGUAGE PROCESSING AND SIMILARITY

10)Stephan Gunawardana,Resume Parser and Job Search

11)Saswat Mohanty,Anshuman Behera,Sushruta Mishra,Ahmed Alkhayyat,Deepak Gupta "Resumate: A Prototype to Enhance Recruitment Process with NLP based Resume Parsing" 4th International Conference on Intelligent Engineering and Management (ICIEM 2023)

12)Narendra G O ,Hashwanth S "Named Entity Recognition based Resume Parser and Summarizer" International Journal of Advanced Research in Science, Communication and Technology (IJARSCT) March 2022

13)Nirmiti Bhoir , Mrunmayee Jakate , Snehal Lavangare , Aarushi Das , and Sujata Kolhe "Resume Parser using hybrid approach to enhance the efficiency of Automated Recruitment Processes" Datta Meghe College of Engineering, April 2023

14)Bhushan Kinge, Shrinivas Mandhare, Pranali Chavan, S. M. Chaware "Resume Screening Using Machine Learning and NLP : A Proposed System" IJSRCSEIT, April 2022

15)Satyaki Sanyal, Souvik Hazra , Sou myashree Adhikary , Neelanjan Ghosh "Resume Parser with Natural Language Processing" IJESC 2017