**RESUME PARSER***\*Sambhavi Gajula - 100818281*

*\*Sai Charan Boda - 100824004*

*\*Vasavi Ravi - 100814974*

*\*Shashank Nandakumar – 100825494*

*AI\_680(Artificial Intelligence: Present and Future)*

*Prof. Deepesh Giri*

***ABSTRACT***

*This Article mainly focuses on the importance of resume parser, which plays a crucial role in the process of recruitment especially for HR professionals and technical team in any company. Screening the resumes of the applicants is one of the most challenging tasks for the recruiters, as it consumes a lot of time. To make it easy for the recruiters, most of the companies are now using advanced technologies for the resume parser. Technologies like “Natural Language Processing (NLP)”, “Machine Learning”, are being used to extract the desired information and parse the unstructured written language from the resume. This paper includes the review on screening of resume systematically and discussed about various techniques and approaches of machine learning for the scrutinizing of unstructured data in the resume. The evolution of resume parsing technology which includes manual resume parsing, automated resume parsing and the challenges associated in both the cases have been discussed in this paper. This paper explores the usage of various skills and technologies in python and their implications. Additionally, this paper also includes the future scope of resume parsing, importance of resume parsing in today’s world in terms of writing style, syntax of unstructured written language and word choice.*

***KEYWORDS***

*Resume Parser, Natural Language Processing, Machine Learning, Manual Resume Parsing, Automated Resume Parsing, Unstructured Written Language.*

**INTRODUCTION**

Nowadays, applying for jobs has become very easy. Most candidates are applying to one position with various resumes. Most recruiters receive large sets of resumes for various positions. The most challenging task for the recruiters is to scrutinize the best candidates by going through their resumes. As there are large numbers of resumes, it's hard for recruiters to select the best resumes and candidates. To tackle this challenge, many company recruiters are using parsing technologies, which include machine learning, natural language processing, and image processing techniques. By using these techniques, an automated system can convert unstructured data to structured data, eliminating the necessity for tiresome manual scrutiny.

Recruiters may now concentrate more on strategically important parts of the recruiting process because this automated system drastically cuts down on the time and effort they must spend. Furthermore, by standardizing the screening process and minimizing human bias, since this system follows strict rules and doesn’t have biases, it ensures a fair and unbiased evaluation of all applicants. This means companies are more likely to find the best candidates for the job, leading to better hiring decisions overall.

**EVOLUTION OF RESUME PARSING**

Earlier, resume parsing was entirely manual, where recruiters used to receive large sets of hard copies from candidates, and recruiters used to go through the entire resume of each candidate. It was very time-consuming for both recruiters and candidates as well. Later, recruiters decided to transition to online resume submission. Even though it is online, recruiters are facing challenges in scrutinizing worthy resumes as sending resumes online has become easier. Therefore, to tackle these challenges, recruiters are now using resume parsing technologies using machine learning and natural language processing techniques.

**METHODOLOGY**

Natural language processing and image processing are parts of artificial intelligence, where these techniques enable computers to understand and generate human language that is meaningful and contextually relevant. Python is a flexible language where we can implement all these techniques and achieve effective outcomes. Different concepts have been included in resume parsing with NLP, such as entity recognition, skill identification, keyword extraction, matching, structuring, and parsing, etc.

For setting an entity recognition, "entity\_ruler" is used which is a part of “Spacy”, which allows users to define customed named entities based on patterns. The code `nlp.add\_pipe("entity\_ruler”)sets up and adds a component called "entity\_ruler" to the spaCy NLP system. This component is responsible for recognizing specific patterns or rules in text. Next, ruler.from\_disk(skill\_path) loads the rules or patterns used by the "entity\_ruler" component from a file stored on disk. These rules likely define how to identify certain entities or concepts within text. Finally, `nlp.pipe\_names` retrieves the names of all components currently in the spaCy NLP system. This is helpful for checking if the "entity\_ruler" component has been successfully integrated into the pipeline.

By using NLP techniques, some entities can be extracted, such as name, contact details, work experience, education, skills, etc. Firstly, NLP converts unstructured data to structured data and extracts these entities. NLP models are being trained to recognize skills such as technical skills, domain-related skills, soft skills, etc. After identifying important information, NLP algorithms arrange the content of the resume into a standard layout. This includes putting the extracted details into appropriate sections like personal information, work history, education, skills, and accomplishments. NLP ensures that these sections are organized logically and consistently across all resumes.

Some concepts such as tokenization, lemmatizer, stop words, NER, etc., have been used. Tokenization is the process where new tokens can be created by breaking down the texts, phrases, or words into smaller units. There are some types of tokenization such as word tokenization, sentence tokenization, character tokenization, etc. In word tokenization, sentences or paragraphs are split into individual words. In sentence tokenization, paragraphs are split into individual sentences. Text is split into individual characters in character tokenization.

Lemmatization is the tool that can be used in NLP, where it reduces words into bases, or the standardized representation of the word known as a lemma. The lemma represents the dictionary form of a word, which is typically the form found in dictionaries and lexicons.

During text preprocessing, stop words will be removed, which are common words that are usually filtered out in NLP. The noise in the text should be reduced because some stop words may appear frequently in text, which does not have any meaning. To improve performance and accuracy, stop words should be removed during text preprocessing.

The code snippet imports a collection of common English stop words from spaCy. Stop words are words like "the," "is," and "and" that are typically removed during text processing because they don't provide significant meaning. It then converts these stop words into a list called `stopwords`, which will be used to filter out these words from the input text. The cleaned tokens are joined together into a single string with spaces in between, representing the processed sentence where stop words and certain types of tokens have been eliminated.

Named Entity Recognition (NER) in NLP extracts entities within texts and categorizes them into some formats. Named Entity Recognition aims to pull out details from unorganized text and allocate them into suitable groups. This helps computers grasp the meaning and organization of written information better.

**FINDINGS**

Recruiters are using machine learning and natural language processing techniques to scrutinize resumes.

* First, the resume is loaded, and then the spaCy library, which allows the provision of a list of words, automatically creates a pattern.
* Afterward, an "entity\_ruler" is set up and connected to pipes such as 'tok2vec', 'parser', 'lemmatizer', 'ner', and 'entity\_ruler'.
* Stop words are imported from spaCy, and through tokenization, these stop words can be removed. This helps reduce noise and potential issues during model training.
* The NLP model compares the candidate's skills with the job requirements to determine if they meet the criteria. Soft skills, achievements, and education can all be extracted using NLP techniques.
* In the process of resume parsing first, preprocessing is the text data by removing stop words, symbols, extra spaces.
* The function `get\_skills` is responsible for finding skills in the cleaned resume text by using spaCy's named entity recognition (NER) feature.
* It looks for entities tagged as 'SKILL' and collects them into a list. Later, the code ensures that only unique skills are kept for each resume with the `unique\_skills` function.
* There's an effort to visualize the skills extracted from resumes using matplotlib, but it seems like there's a missing variable called `counting` which should be defined before plotting.
* For visualizing named entities, particularly skills, the code employs spaCy's `displacy` module. However, it seems that the variable `doc` is not defined before rendering the visualization.

By strategically restructuring content, resumes can effectively highlight key competencies and experiences, aligning them more closely with employers' requirements. Furthermore, paraphrasing aids in mitigating issues of plagiarism or overly generic content, which are common pitfalls in resume writing. Additionally, findings suggest that paraphrasing promotes readability and clarity, allowing recruiters to quickly grasp the candidate's qualifications and achievements.

However, challenges such as maintaining the original meaning and tone while paraphrasing necessitate the use of advanced linguistic and NLP techniques to achieve optimal results.

As organizations increasingly rely on automated parsing and screening processes, the findings underscore the importance of incorporating effective paraphrasing strategies to enhance resume quality and maximize candidates' chances of success in the competitive job market.

**RESULTS AND DISCUSSION**

The results of resume parsing studies reveal significant improvements in recruitment efficiency and candidate matching accuracy. Through the application of Natural Language Processing (NLP) techniques, automated parsing systems demonstrate the ability to swiftly extract and categorize key information from resumes, such as work experience, skills, and qualifications.

Below are the outcomes obtained after processing a sample dataset:

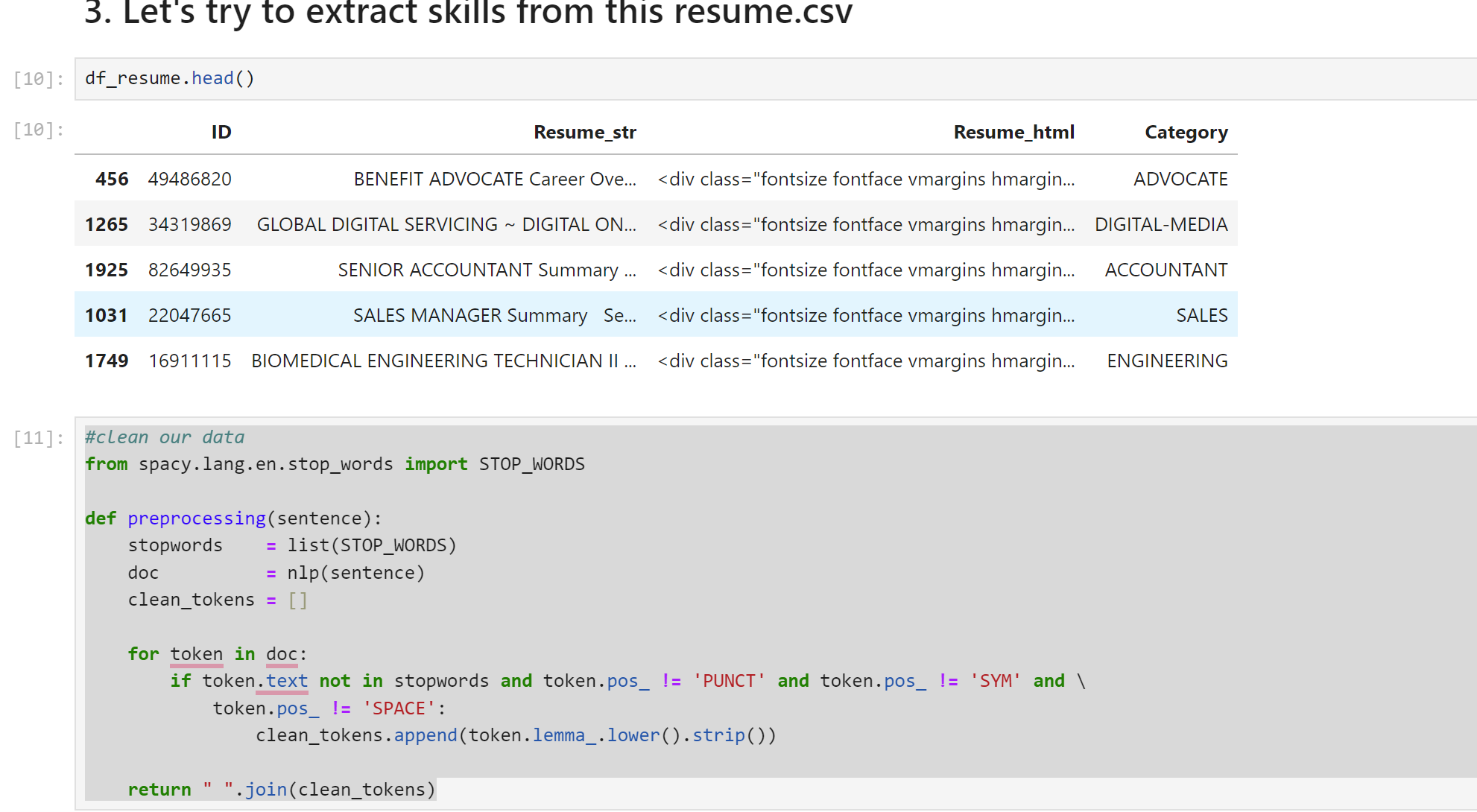
Fig 1.1 setting entity\_ruler

**A screenshot of a computer

Description automatically generated**

From the above Fig (1.1) , spaCy is used which is a natural language processing library, to load a pre-trained model and perform named entity recognition (NER) on a sample text. The code initializes a spaCy pipeline by loading the English language model (en\_core\_web\_md), which is pre-trained on a large corpus of text data and includes word vectors for enhanced linguistic processing. After adding the entity ruler to the pipeline, the code prints the names of all pipeline components using nlp.pipe\_names. This allows for verification that the entity ruler component has been successfully added to the pipeline.

The code processes a sample text ("Chaky loves ajax.") using the initialized spaCy pipeline (nlp). The doc object represents the processed text, including tokenization, part-of-speech tagging, and named entity recognition. The code demonstrates the use of spaCy for NLP tasks, including loading pre-trained models, adding custom entity recognition rules, and extracting named entities from text data. However, the effectiveness of entity recognition depends on the quality of the pre-trained model and the specificity of entity rules defined in the provided file.

  
Fig 1.2 cleaning the data using tokenization concept

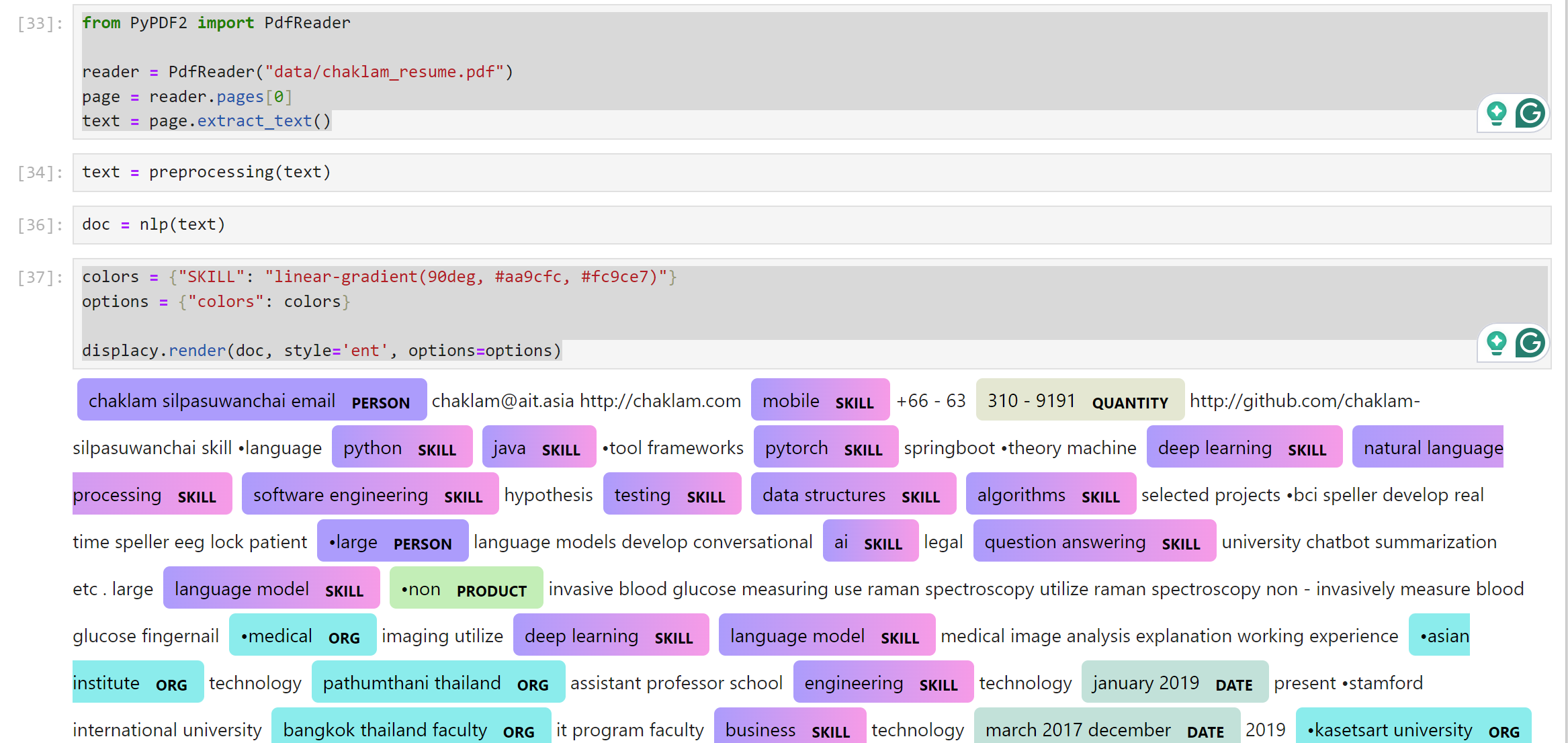
In the above Fig (1.2), The code imports a set of stop words from the English language module of spaCy, which are common words typically filtered out during text preprocessing as they carry little semantic meaning. The preprocessing function takes a single argument sentence, representing the text data to be cleaned. Within the function, the input sentence is processed using spaCy's NLP pipeline (nlp). This includes tokenization, which breaks the text into individual tokens (words), and lemmatization, which converts words to their base or canonical form.

Stop words are filtered out from the processed tokens based on whether they are present in the list of stop words (stopwords) and their part-of-speech tag is not punctuation, symbol, or white space. This step helps eliminate common words that do not contribute to the semantic meaning of the text. The lemmatized tokens that pass the stop word check are converted to lowercase and stripped of any leading or trailing whitespace. The function returns the cleaned tokens joined into a single string with spaces, representing the processed sentence where stop words and certain types of tokens (punctuation, symbols, spaces) have been removed.

A graph with blue squares

Description automatically generated with medium confidence  
Fig (1.3), bar plot showing resume submission of different roles.

The analysis depicted in Figure 1.3 indicates varying resume requirements based on different job roles. The code produces a bar plot with the specified figure size (15 inches wide and 3 inches tall) using Matplotlib's plt.figure(figsize=(15, 3)) function. This ensures that the plot has the desired dimensions for effective visualization. The plt.xticks(rotation=45) command rotates the x-axis labels by 45 degrees to prevent overcrowding and improve readability, especially when dealing with long or overlapping labels. The plt.bar(counting.keys(), counting.values()) function generates the bar plot using the keys and values from the counting dictionary. Each key-value pair represents a category (x-axis) and its corresponding count (y-axis) to be plotted as a bar.

  
Fig(1.4), Scrutinizing of entities in resume

The code reads the content of the PDF file using the PdfReader class from the PyPDF2 library. It extracts text from the first page of the PDF and assigns it to the variable text. The extracted text undergoes preprocessing using the previously defined preprocessing function. This function removes stop words, punctuation, and other non-essential elements from the text to prepare it for further analysis. The preprocessed text is processed using spaCy's NLP pipeline (nlp). This includes tokenization, part-of-speech tagging, and named entity recognition (NER) to identify entities such as skills mentioned in the resume. The recognized entities, specifically those labeled as "SKILL," are visualized using spaCy's displacy module. The entities are highlighted in the text with a specified color gradient ("linear-gradient(90deg, #aa9cfc, #fc9ce7)") to distinguish them from the rest of the text.

It automatically segregates entities such as name, skills, organization etc and finally the resume has been parsed.

Moreover, findings indicate that NLP-based parsing reduces the burden on recruiters by automating tedious tasks, allowing them to focus on higher-value activities such as interviewing and relationship-building. Discussions surrounding resume parsing underscore its potential to revolutionize recruitment processes, particularly in industries with high volumes of job applications. However, challenges persist, including the accurate parsing of unstructured data, handling of diverse resume formats, and ensuring privacy compliance.

Future research directions may focus on refining NLP algorithms to better handle nuanced information and improving integration with Applicant Tracking Systems (ATS) for seamless workflow automation. Overall, the results and discussions highlight the transformative impact of resume parsing with NLP on modern recruitment practices, paving the way for more efficient and effective talent acquisition processes.

**CONCLUSION**

This paper underscores the importance of resume parsing in the hiring process and emphasizes the crucial role played by advanced technologies such as machine learning and natural language processing (NLP). These tools automate the extraction of vital data from resumes, streamlining the screening process and saving recruiters valuable time and effort.

The transition from manual to automated resume processing highlights the necessity of embracing new technologies to meet the evolving demands of contemporary hiring practices. The methods section delves into the application of NLP techniques like entity recognition, tokenization, and stop word removal, showcasing their effectiveness in resume parsing.

The findings demonstrate how machine learning algorithms can accurately extract pertinent information and skills from resumes, leading to improved candidate matching. Visual aids facilitate understanding of the distribution of CV submissions across various job titles. Nonetheless, challenges such as maintaining the original meaning and tone during paraphrasing underscore the ongoing need for research and advancement in this field.

Overall, resume parsing with NLP holds promise for enhancing hiring decisions and recruitment efficiency in today's job market.

**REFERENCES**

1. <https://www.slideshare.net/irjetjournal/resume-parser-with-natural-language-processing>
2. Shicheng Zu, Xiulai Wang and Seth Darren “Resume Information Extraction with A Novel Text Block Segmentation Algorithm”.
3. Bhatia, V., Rawat, P., Kumar, A. & Shah, R. R., 2019. End-to-End Resume Parsing and Finding Candidates for a Job Description using BERT. Arxiv’
4. Chen, J., Gao,. L. & Tang, Z., 2016. Information Extraction from Resume Documents in PDF. Research Gate. 64
5. Kopparapu, S.. K., 2015. Automatic Extraction of Usable Information from Unstructured Resumes to Aid Search. ieeexplore.
6. <https://www.ijraset.com/research-paper/resume-parser-analysis-using-machine-learning>