FINAL YEAR PROJECT 2021

Resume Analysis

Send resumes in any format

Parsing algorithm + Enrichment

Get relevant talent

MCS FINAL

PROJECT PARTICIPANTS:

ABEHA SHAHAB(EP19101001)

FAZEENA SEHAR(EP19101017)

PROJECT COORDINATOR :

SIR KHALID JAMAL

2/12/2021

Table of Contents

ABSTRACT:

In today's digital era, bulk recruitment and managing high applicant volumes can easily get handled with the help of a Resume Parser. Let's look at how HR teams make a definite and indelible impact on today's competitive job market with the help of technology.

I believe we all have been there when we were finding the perfect job opportunity, customizing our resume and cover letter, submitting our application online, and praying that it will pass the resume review test. Writing a perfect resume can be one of the most important things we do in our life. Unfortunately, nowadays, few applications are getting lost in the resume 'black hole,' and applicants are left wondering why the employer didn't look into their application.

Don't you feel that candidates always think that their job application is rejected every time they apply? What is missing in their application?

The answer is simple. Many job applicants don't realize that their job applications are rejected before they reach a recruiter's hands. It often first passes through the technology known as Applicant Tracking System/Resume Parsing.

Our project is based on a software house scenario to find a best deserving candidate for the available job by using NLP (Natural Language Processing) for Resume Parsing.

INTRODUCTION:

Would you be surprised if I told you that every organization nowadays involves a resume parser in their ATS/CRM to automate their screening task? I don't think so. Let's find out through this blog how important a resume parser is for ATS/CRM.

Why Does an ATS/CRM Software Need a CV/Resume Parser?

Structured Candidate Data

Resume parser will make candidate information more structured and make your candidate database more manageable. Now recruiters can easily filter candidates using keywords and tags instead of recollecting which candidate is the ideal hire.

Better Candidate Experience

When a resume parser is used on your application form, it will provide a positive candidate experience. Candidates only have to upload their resume while the fields will be populated automatically.

Email Inbox Parsing

Through resume parser, recruiters and HR professionals are allowed to fetch resumes from single or multiple emails..

Multi-lingual

A resume parser can automatically identify the region and languages used in the resume and parse the information accordingly.

Streamline the Process

When you have a resume parser along with ATS, the screening process becomes easy. All the screening tasks get streamlined with one software's help, starting from a custom application till sending the pre-assessment tests.

Save Time

Resume parser saves recruiters' time on sorting out resumes manually. It easily extracts candidate's data and saves into your ATS/CRM quickly in different fields. In a short time, recruiters can process and select the most relevant candidate that will be perfect for the organization.

Bulk Upload

Through this feature of resume parser, recruiters can parse multiple resumes in go.

Resume Parser can help an ATS/CRM to deliver a complete Recruitment Solution that will manage the workforce of both present and future. The truth behind using a parser in the recruitment process is that it will shorten the recruitment process for recruiters as well as for candidates.

DEFINITION:

What do you mean by Parsing?

Parsing, syntax analysis, or syntactic analysis is the process of analyzing a string of symbols, either in natural language, computer languages or data structures, conforming to the rules of a formal grammar. The term parsing comes from Latin pars (orationis), meaning part (of speech)

OR

Parsing means we are **analyzing an object specifically**. For example, when we enter some keywords in a search engine, they parse the keywords and give back results by searching for each word. So it is basically taking a string from the file and processing it to extract the information we want.

What is Machine Learning?

Machine Learning is developing a computer program that improves automatically with experience.

Why Machine Learning?

Artificial Intelligence is an emerging technology for upcoming times. Machine Learning is an application of Artificial Intelligence. There are many advantages of Artificial Intelligence.

The primary aim of the same is to allow software devices to learn automatically without any human interference.

By using Machine Learning, you are just trying to make your software device smart enough by feeding them the previous outcomes (e.g., Data), and based on the experience, your device will generate the output on its own.

Scope of Machine Learning:

Machine Learning has brought various scopes and improved strategies into various market fields and made us able to get more profit.

Machine Learning Job Scope and Salary Trends

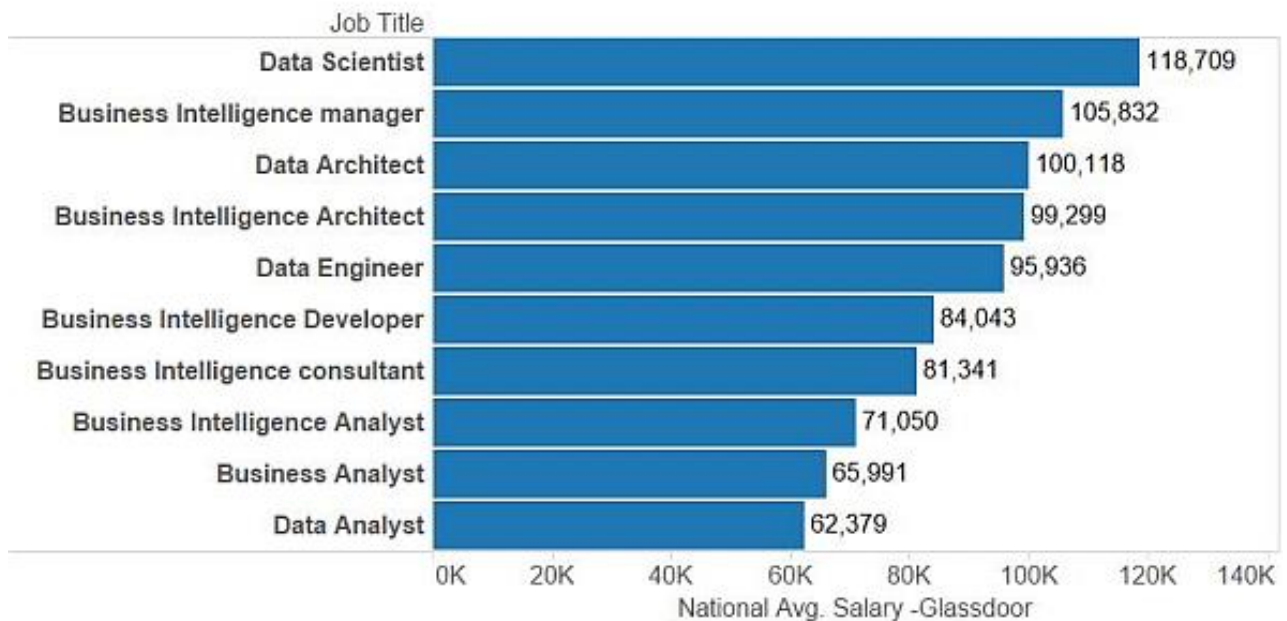Automotive Industry

Robotics

Quantum Computing

Computer Vision

Data Mining

Data Science

Artificial Intelligence

And many more.

Job Title	
Data Scientist	118,709
Business Intelligence manager	105,832
Data Architect	100,118
Business Intelligence Architect	99,299
Data Engineer	95,936
Business Intelligence Developer	84,043
Business Intelligence consultant	81,341
Business Intelligence Analyst	71,050
Business Analyst	65,991
Data Analyst	62,379

National Avg. Salary -Glassdoor

Natural Language Processing(NLP):

NLP is a powerful, machine-learning tool used to augment human teams and help organizations find an edge in a competitive world.

In a world of Google and other search engines, shoppers expect to enter a phrase, or even an idea, into a search box and to instantly see personalized recommendations that are clearly relevant to what it was they were meaning to discover.

Natural Language Processing Techniques:

NLP recognises, understands, summarises and analyses what we say in order to understand us. It does that so well, it can even help to generate language itself.

Algorithms, syntax and semantics help to give NLP its incredible powers of deduction.

- *Algorithms*

NLP uses algorithms to transform our diverse, unstructured, spontaneous communications into something a computer can understand and act upon.

Powered by these algorithms, NLP deciphers meaning from the jumble of sentences, colloquialisms, jargon and lingo we use everyday.

It picks through what we say and turns it into a base of data, converting our speak into a form computers can understand.

 Syntax and Semantics

Two key elements of NLP are syntactic and semantic analysis. Syntax determines what's being said, while semantics digs a little deeper into the meaning.

Syntax

divides up sentences and uses things like grammar rules or basic word forms to understand a piece of text.

Semantics

extracts the meaning behind it all. Using context, and tools like word categorization, or meaning databases, it discovers the intention behind using certain words. It's how a computer knows what someone really means.

 Everyday Natural Language Processing Examples:

Most of us have already come into contact with NLP. We connect to it via website search bars, virtual assistants like Alexa, or Siri on our smartphone.

The email spam box or voicemail transcripts on our phone, even Google Translate, all are examples of NLP technology in action. In business, there are many applications.
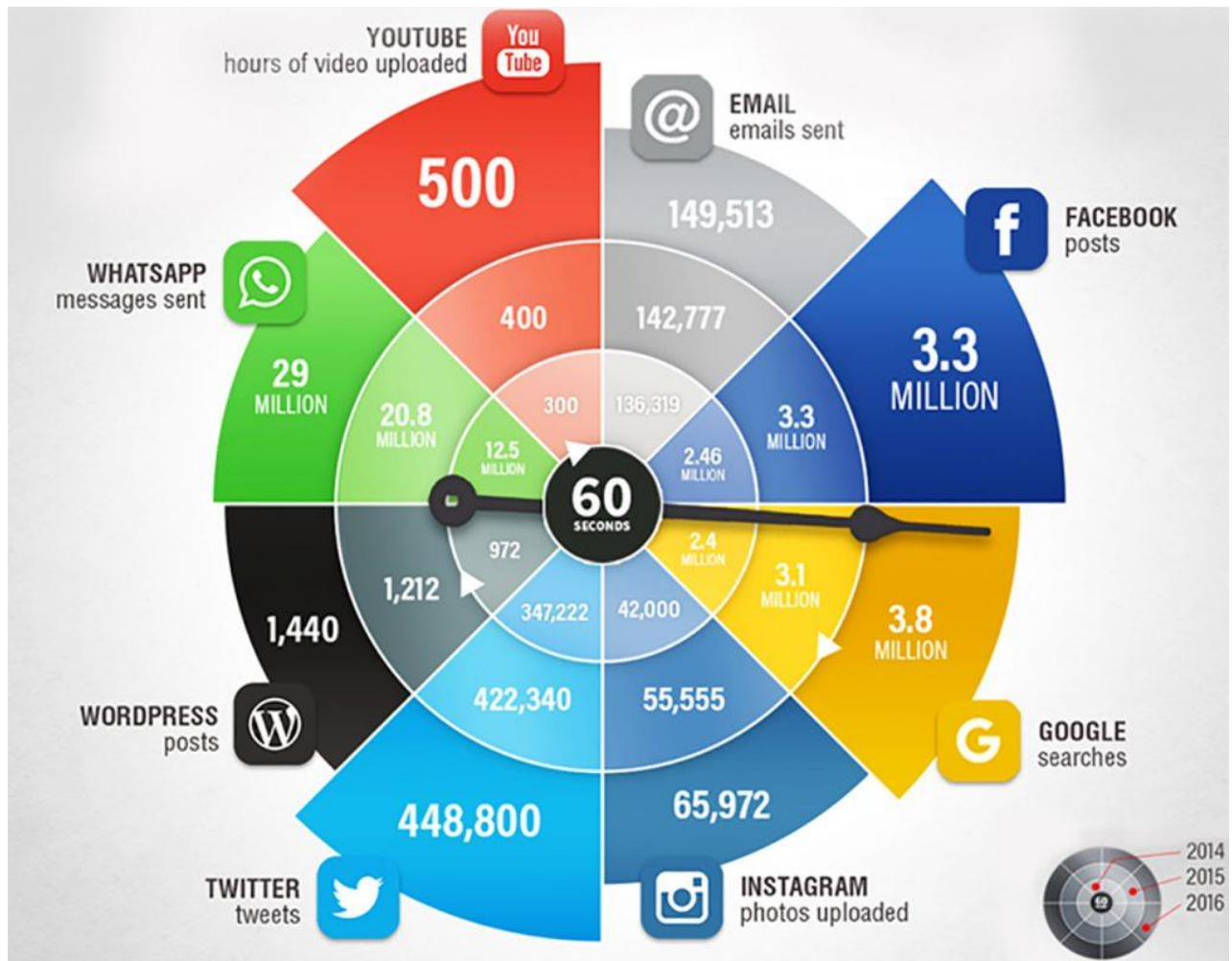
Semantic Based Search

Key to making every search a fruitful one is to incorporate semantic-based search.
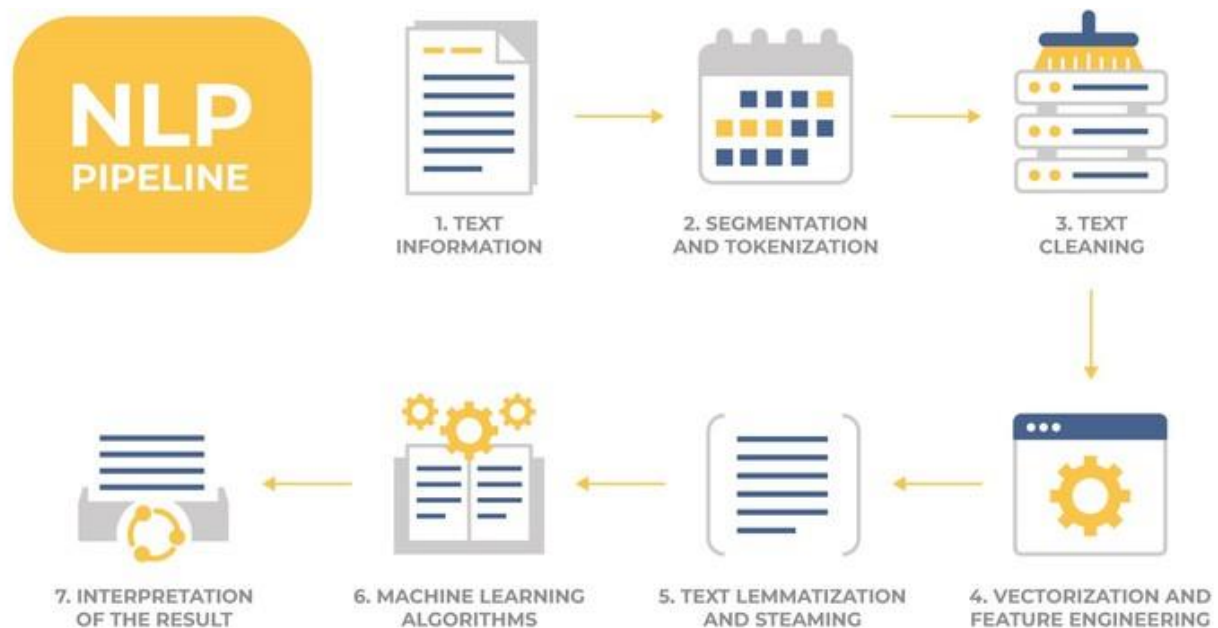
Semantic-based search is so intuitive that shoppers still get relevant results, even when using their own unique search queries.

It figures out intent, and brings out products located deep in a merchant's online product catalog in the lease amount of time. And the numbers prove it works.

Social media listening

has become an important tool for e-retailers who want to understand consumer shopping habits, predict product demand, or monitor trends to target marketing messages.

NLP PIPELINE

1. TEXT INFORMATION
2. SEGMENTATION AND TOKENIZATION
3. TEXT CLEANING
4. VECTORIZATION AND FEATURE ENGINEERING
5. TEXT LEMMATIZATION AND STEAMING
6. MACHINE LEARNING ALGORITHMS
7. INTERPRETATION OF THE RESULT

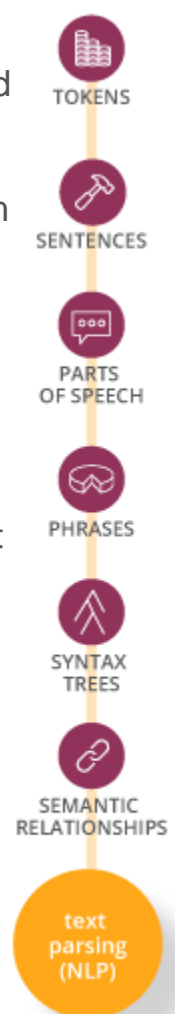## Hybrid Machine Learning Systems for NLP

Our text analysis functions are based on patterns and rules. Each time we add a new language, we begin by coding in the patterns and rules that the language follows. Then our supervised and unsupervised machine learning models keep those rules in mind when developing their classifiers. We apply variations on this system for low-, mid-, and high-level text functions.

Low-level text functions are the initial processes through which you run any text input. These functions are the first step in turning unstructured text into structured data. They form the base layer of information that our mid-level functions draw on. Mid-level text analytics functions involve extracting the real content of a document of text. This means who is speaking, what they are saying, and what they are talking about.

The high-level function of sentiment analysis is the last step, determining and applying sentiment on the entity, theme, and document levels.

Low-Level

- **Tokenization**: ML + Rules
- **PoS Tagging**: Machine Learning
- **Chunking**: Rules
- **Sentence Boundaries**: ML + Rules
- **Syntax Analysis**: ML + Rules

TOKENS

SENTENCES

PARTS OF SPEECH

PHRASES

SYNTAX TREES

SEMANTIC RELATIONSHIPS

text parsing (NLP)

The algorithm is mainly divided into:

- Training Phase
- Testing phase

Training Phase:

You take a randomly selected specimen of a candidate (**training data**), make a table of all the characteristics of a candidate, like Name, Age, Address, Qualification, Skills, etc (**features**), along with the departments/categories, Web Development, Game Development, Web Design, Network Engineer, Machine Learning, AI, and Database Engineer (**output variables**). You feed this data to the machine learning algorithm (**classification/regression**), and it learns a model of the correlation between a candidate and its belonging field.

Testing Phase:

Next time when you will run the code it will automatically select the candidate based on its characteristics saved as a trained data.

Code Explanations:

Programming language used in our project:
Python:

Python Libraries Used in project:

- **PyPDF2**
- **os**
- **pandas**
- **collections**
- **en-core-web-sm**
- **spacy**

PyPDF2:

A Pure-Python library built as a PDF toolkit. It is capable of:

- extracting document information (title, author, …)
- splitting documents page by page
- merging documents page by page
- cropping pages
- merging multiple pages into a single page
- encrypting and decrypting PDF files
- and more!

fileleReader = PyPDF2.PdfFileReader(open(file,'rb'))

By being Pure-Python, it should run on any Python platform without any dependencies on external libraries. It can also work entirely on StringIO objects rather than file streams, allowing

for PDF manipulation in memory. It is therefore a useful tool for websites that manage or manipulate PDFs.

os:

This module provides a portable way of using operating system dependent functionality. If you just want to read or write a file see `open()`, if you want to manipulate paths, see the

`os.path` module is sub-module of OS module in Python used for common pathname manipulation.

mypath='/content/resume_final'

os.listdir() method in python is used to get the list of all files and directories in the specified directory. If we don't specify any directory, then list of files and directories in the current working directory will be returned.

os.path.join() method in Python join one or more path components intelligently. This method concatenates various path components with exactly one directory separator ('/') following each non-empty part except the last path component. If the last path component to be joined is empty then a directory seperator ('/') is put at the end.

onlyfiles = [os.path.join(mypath, f) for f in os.listdir(mypath) if os.path.isfile(os.path.join(mypath, f))]

StringIO module an in-memory file-like object. This object can be used as input or output to the most function that would expect a standard file object. When the StringIO object is created it is initialized by passing a string to the constructer. If no string is passed the StringIO will start empty. In both cases, the initial cursor on the file starts at zero.

rule_id = nlp.vocab.strings[match_id]

Pandas is an open-source, BSD-licensed Python library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Counter A counter is a dictionary-like object designed to keep tallies. With a counter, the key is the item to be counted and value is the count. You could certainly use a regular dictionary to keep a count, but a counter provides much more control.

keywords = "\n".join(f'{i[0]} {i[1]} ({j})' for i,j in Counter(d).items())

en_core_web_sm is a small English pipeline trained on written web text (blogs, news, comments), that includes vocabulary, vectors, syntax and entities.

en_core_web_sm.load() get the code to load it from within spaCy and an example to test it.

PhraseMatcher The spaCy library comes with Matcher tool that can be used to specify custom rules for phrase matching. The process to use the Matcher tool is pretty straight forward. The first thing you have to do is define the patterns that you want to match. Next, you have to add the patterns to the Matcher tool and finally, you have to apply the Matcher tool to the document that you want to match your rules with.

matcher = PhraseMatcher(nlp.vocab)

matcher.add('Game Development', None,*Game_Development_words)

matcher.add('Web Development', None,*Web_Development_words)
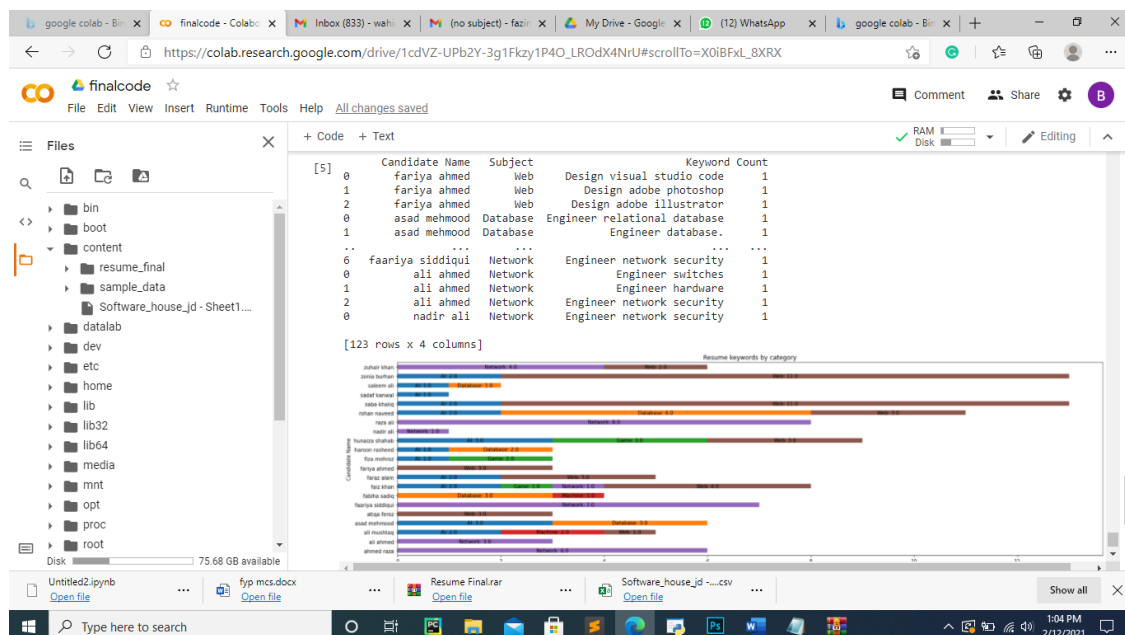
matcher.add('Web Design', None,*Web_Design_words)

matcher.add('Network Engineer', None,* Network_Engineer_words)

matcher.add('Machine Learning', None,*Machine_Learning_words)

matcher.add('AI', None,*AI_words)

matcher.add('Database Engineer', None,*Database_Engineer_words)

OUTPUT:

finalcode

File Edit View Insert Runtime Tools Help   All changes saved

Comment   Share

+ Code   + Text   RAM / Disk   Editing

```
     Candidate Name    Subject                        Keyword  Count
0       fariya ahmed        Web    Design visual studio code      1
1       fariya ahmed        Web        Design adobe photoshop      1
2       fariya ahmed        Web        Design adobe illustrator    1
0       asad mehmood   Database  Engineer relational database      1
1       asad mehmood   Database             Engineer database.      1
..               ...        ...                           ...    ...
6   faariya siddiqui    Network      Engineer network security      1
0         ali ahmed    Network               Engineer switches      1
1         ali ahmed    Network               Engineer hardware      1
2         ali ahmed    Network      Engineer network security      1
0         nadir ali    Network      Engineer network security      1

[123 rows x 4 columns]
```


Resume keywords by category

finalcode.ipynb
Open file   Show all

---

finalcode

File Edit View Insert Runtime Tools Help   All changes saved

Comment   Share

+ Code   + Text   RAM / Disk   Editing

[5] [123 rows x 4 columns]

Resume keywords by category

Future Scope:

Our aim is to further enhance the code for editing and updating the training data from GUI interface. Which will make it easy for HR departments of organizational users to update their required matching keywords from GUI in case they will have more departments' data to analyze and find out the best fit candidate for their organization.

Conclusion:

By using our project of Resume Parsing any organization can reduce their required time and physical efforts in finding best deserving candidate for the job application. This project is not only useful for Resume Parsing but also with further enhancement also educational institutes can find best students' assignments from bulk number of students' assignments and many more like Google and other Social media platforms gain the training data from users' searching keywords and suggest best required output .

Bibliography / references:

Why Every ATS Needs A Resume Parser? (rchilli.com)

https://stackoverflow.com/

https://www.geeksforgeeks.org/

https://www.tutorialspoint.com/

https://www.edureka.co/