



D

C

B

A

NAME: PAVITHRA N
NM ID: au211521104104
CLG NAME: PANIMALAR
INSTITUTE OF
TECHNOLOGY

RESUME PARSER USING ARTIFICIAL NEURAL NETWORK

INTRODUCTION TO THE PROJECT

A resume parser is a software tool designed to automate the process of extracting relevant information from resumes or CVs (curriculum vitae). It uses natural language processing (NLP) algorithms to analyze the text, identify key data points such as contact details, work experience, education history, skills, and qualifications, and then organizes this information into a structured format. The primary goal of a resume parser is to streamline the recruitment process by quickly and accurately identifying potential candidates who meet the requirements of a job posting.



AGENDA OF THE PROJECT

- The agenda of the resume parser project employing Artificial Neural Networks (ANN) algorithm involves several key steps aimed at developing an efficient and accurate system for automating the extraction of relevant information from resumes.
- Firstly, data collection and preprocessing are conducted to gather a diverse dataset of resumes and standardize their format for input into the ANN model.
- Subsequently, the ANN model architecture is designed and trained using the preprocessed data, with features such as contact details, work experience, education history, skills, and qualifications being the target variables.
- During training, the ANN learns to recognize patterns and relationships within the resume text to effectively extract and classify these features.



- The model's performance is then evaluated using validation techniques to ensure its accuracy and reliability in parsing resumes.
- Finally, the trained ANN model is integrated into a user-friendly interface or application, allowing recruiters and HR professionals to seamlessly process large volumes of resumes, thereby streamlining the hiring process and enhancing overall efficiency.
- Ongoing refinement and optimization of the ANN algorithm may also be pursued to continually improve the parser's performance and adaptability to diverse resume formats and languages.
- Ultimately, the agenda aims to deliver a state-of-the-art resume parsing solution that offers high accuracy, scalability, and customization capabilities to streamline the recruitment process and facilitate better decision-making for hiring professionals.



PROBLEM STATEMENT

The aim of this project is to develop a robust and accurate resume parsing system using Artificial Neural Networks (ANN) algorithm to automate the extraction of key information from resumes or CVs.

In today's competitive job market, recruiters and HR professionals are inundated with a vast number of resumes for job openings. Manually reviewing each resume is time-consuming and inefficient, leading to potential errors and inconsistencies in candidate selection. To address this challenge, there is a need for an intelligent resume parsing system that can automatically extract relevant information from resumes and assist in the initial screening process.



PROJECT OVERVIEW

PROJECT GOAL:

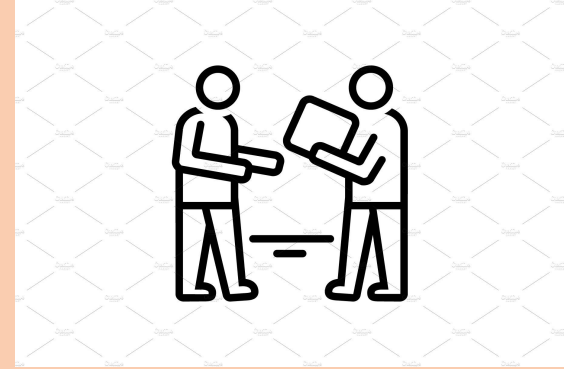
The goal of this project is to develop a resume parser using Artificial Neural Networks (ANN) to automate the extraction of relevant information from resumes or CVs. By leveraging ANN algorithms, the system aims to accurately identify and extract key data points such as personal details, work experience, education history, skills, and qualifications from unstructured text.



PROJECT DELIVERABLES

- A trained ANN model capable of accurately parsing resumes and extracting relevant information.
- Documentation detailing the architecture, training process, and evaluation results.
- Source code for the resume parser system, along with instructions for deployment and usage.

Overall, this project aims to leverage ANN algorithms to develop an efficient and accurate resume parsing solution, ultimately improving the efficiency of the recruitment process.



WHO ARE THE END USERS?

The end users of a resume parser utilizing Artificial Neural Network (ANN) algorithms would typically include:

Recruiters and HR Professionals: They benefit from the automation and efficiency provided by the resume parser powered by ANN.

Employers and Hiring Managers: Employers and hiring managers rely on resume parsing technology to streamline the hiring process, improve candidate quality, and reduce time-to-fill for open positions.

Job Seekers: While job seekers may not directly interact with resume parsing technology, they are indirectly affected by its use.

IT Professionals and Developers: Professionals involved in the development and maintenance of resume parsing.



SOLUTION AND ITS VALUE PROPOSITION

1. Automated Resume Screening:

ANN algorithms can be trained to parse resumes efficiently, extracting relevant information such as skills, experience, education, and contact details.

2. Improved Candidate Matching:

By leveraging ANN algorithms, the parser can compare the extracted information from resumes with job descriptions or company requirements.

3. Enhanced Candidate Experience:

Candidates receive quicker responses and feedback on their applications, leading to a more positive overall experience.

4. Reduced Bias in Screening:

ANN algorithms can help minimize human biases in the screening process by focusing solely on the skills and qualifications outlined in the resumes.



5. Scalability and Efficiency:

A resume parser powered by ANN can handle a large volume of resumes efficiently, making it scalable for organizations of various sizes.

6. Continuous Learning and Improvement:

ANN algorithms can be trained iteratively using feedback from recruiters and hiring managers, improving the accuracy and relevance of resume parsing over time.

7. Customization and Integration:

The flexibility of ANN-based resume parsing solutions allows for customization to fit specific organizational needs and workflows.

8. Cost Savings:

By automating time-consuming tasks such as resume screening, organizations can realize significant cost savings in terms of HR personnel hours and recruitment expenses.

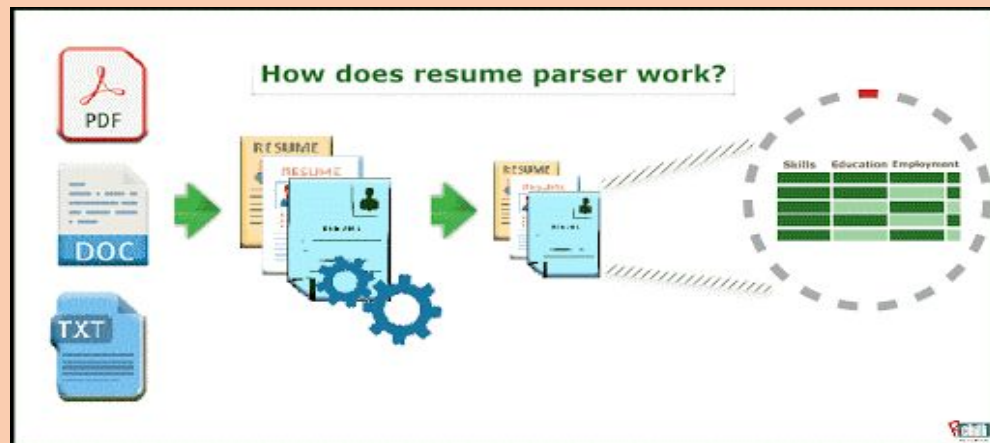
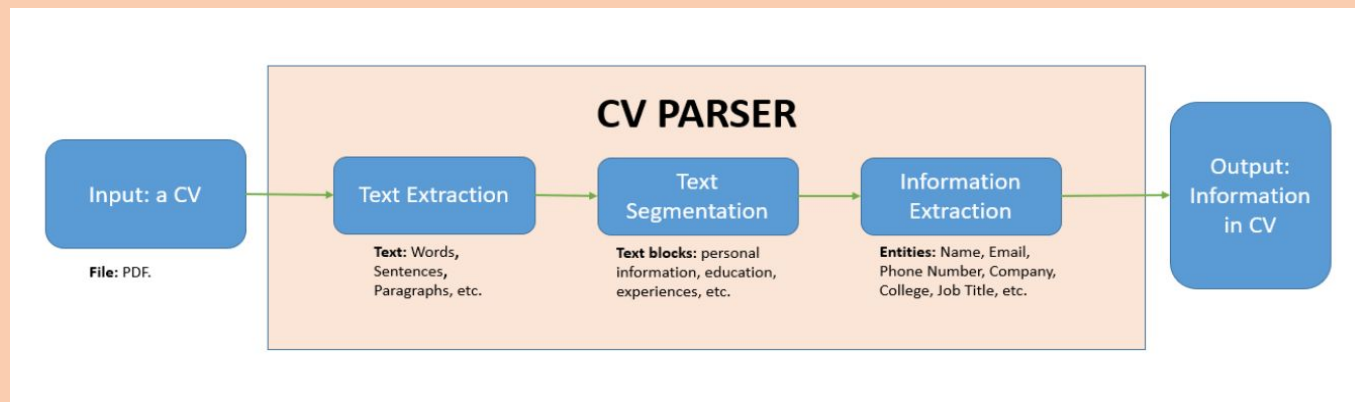


WOW FACTOR IN THE SOLUTION

It is an user-friendly interface where job seekers can upload their resumes and receive instant feedback and suggestions for improvement. This could include suggestions for optimizing the resume layout, enhancing keywords related to the job description, or adding relevant skills or experiences. The system could also provide insights into potential areas of weakness or strengths based on the job requirements. Additionally, it could offer personalized recommendations for professional development or training based on the analysis of the resume and job market trends. This interactive feedback mechanism adds significant value to the traditional resume parsing process by providing actionable insights and guidance to job seekers, making it a standout feature in the market.



MODELLING



STEPS

1. Data Collection and Preprocessing:

Gather a diverse dataset of resumes in various formats (e.g., PDF, DOCX, TXT).

2. Text Extraction and Parsing:

Implement algorithms to extract text from different file formats.

3.Entity Recognition and Classification:

Classify entities into predefined categories such as Person, Organization, Degree, Skill, etc.

4.Skill Extraction:

Develop algorithms to extract skills and keywords from the Experience and Skills sections.

5.Experience Parsing:

Parse the Experience section to extract details such as job titles, company names, dates of employment, and job descriptions.

6.Education Parsing:

Parse the Education section to extract details such as degree obtained, institution name, graduation year, GPA, etc.

7.Formatting and Layout Analysis:

Analyze the formatting and layout of the resumes to identify hierarchical structures, bullet points, indentation, etc.

8.Integration with Job Descriptions:

Develop mechanisms to match extracted skills, experiences, and education with the requirements specified in job descriptions.

9.Error Handling and Validation:

Implement error handling mechanisms to address cases where information extraction fails or produces inaccurate results.

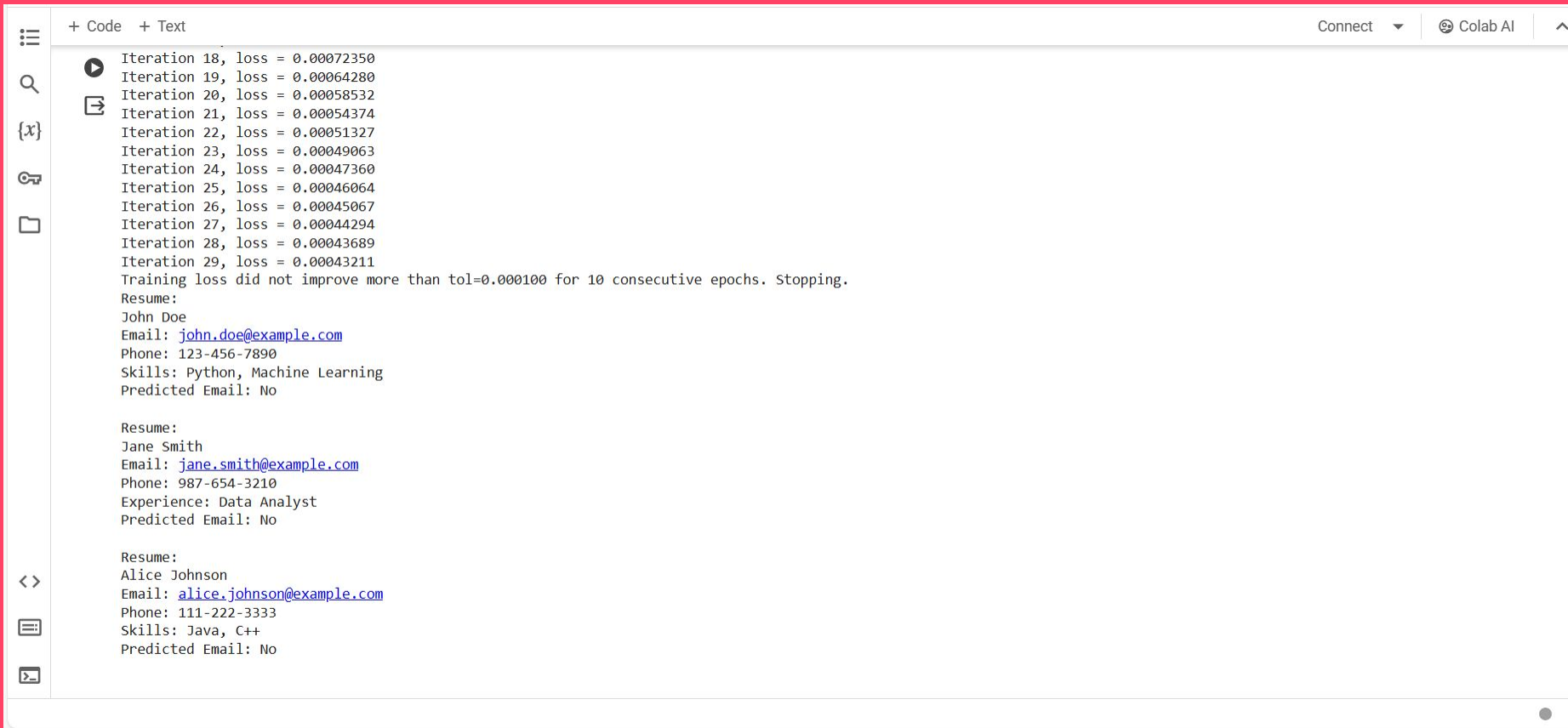
10.Feedback Mechanism:

Incorporate a feedback loop where users can provide corrections or feedback on the parsed information, allowing the system to improve over time.

11.Scalability and Performance Optimization:

Design the system to handle large volumes of resumes efficiently, considering factors like processing speed, memory usage, and scalability.

OUTPUT SCREENSHOTS



The screenshot shows a Google Colab interface with a dark-themed sidebar on the left containing icons for file management, search, and other tools. The main area displays the output of a code execution. The output consists of a series of lines showing training iterations with their respective loss values, followed by a message indicating that training has stopped due to no improvement in loss for 10 consecutive epochs. After this, there are three separate 'Resume:' sections, each containing personal information for a different individual: John Doe, Jane Smith, and Alice Johnson. Each resume section includes a name, email address, phone number, skills, and a predicted email status.

```
+ Code + Text
Connect ▾ Colab AI ^

▶ Iteration 18, loss = 0.00072350
▶ Iteration 19, loss = 0.00064280
▶ Iteration 20, loss = 0.00058532
▶ Iteration 21, loss = 0.00054374
▶ Iteration 22, loss = 0.00051327
▶ Iteration 23, loss = 0.00049063
▶ Iteration 24, loss = 0.00047360
▶ Iteration 25, loss = 0.00046064
▶ Iteration 26, loss = 0.00045067
▶ Iteration 27, loss = 0.00044294
▶ Iteration 28, loss = 0.00043689
▶ Iteration 29, loss = 0.00043211
▶ Training loss did not improve more than tol=0.000100 for 10 consecutive epochs. Stopping.
▶ Resume:
▶ John Doe
▶ Email: john.doe@example.com
▶ Phone: 123-456-7890
▶ Skills: Python, Machine Learning
▶ Predicted Email: No

▶ Resume:
▶ Jane Smith
▶ Email: jane.smith@example.com
▶ Phone: 987-654-3210
▶ Experience: Data Analyst
▶ Predicted Email: No

▶ Resume:
▶ Alice Johnson
▶ Email: alice.johnson@example.com
▶ Phone: 111-222-3333
▶ Skills: Java, C++
▶ Predicted Email: No
```


OUTPUT SCREENSHOTS

```
+ Code + Text Connect Colab AI ^
```

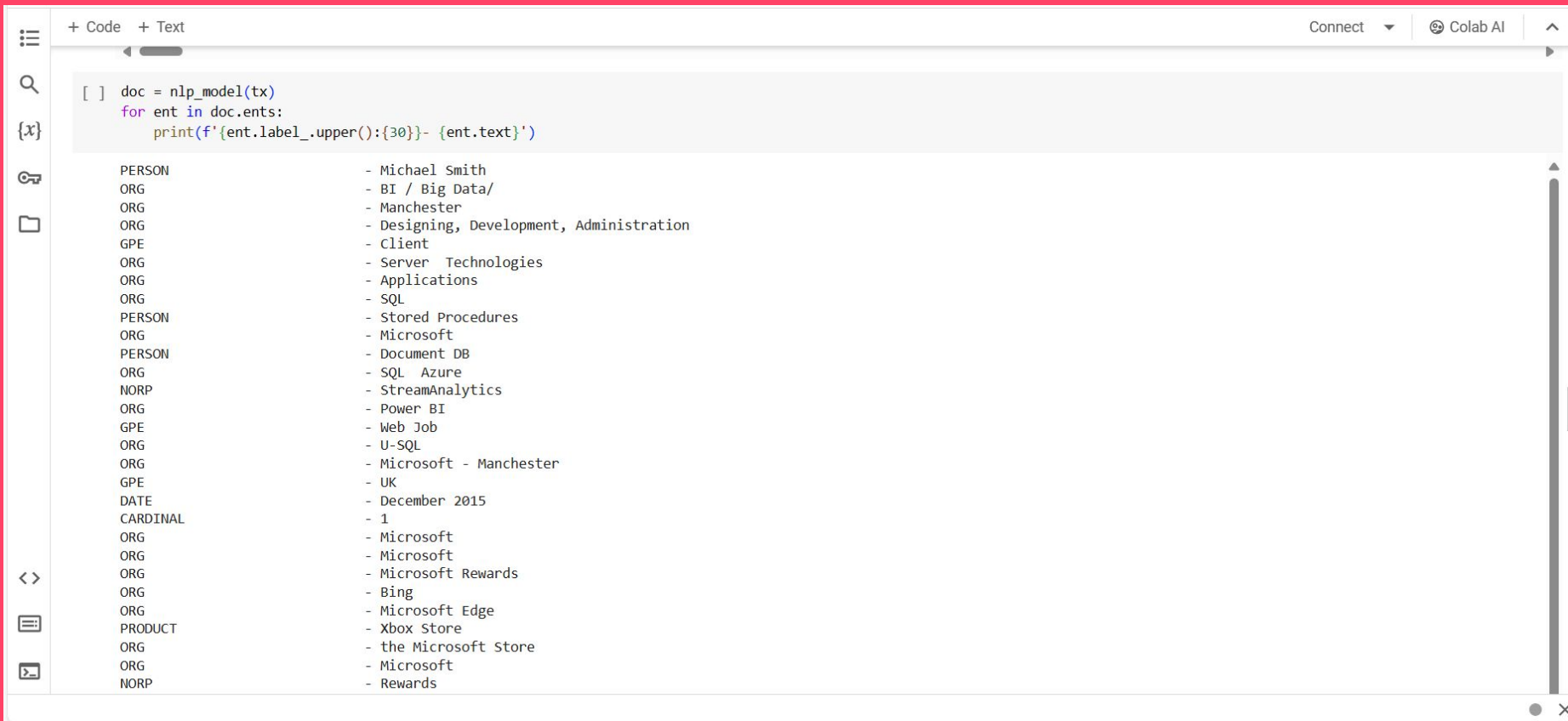
```
[ ] import spacy
import pickle
import random

[ ] train_data = pickle.load(open('/content/train_data.pkl', 'rb'))
train_data[0]
```

('Govardhana K Senior Software Engineer Bengaluru, Karnataka, Karnataka - Email me on Indeed: [indeed.com/r/Govardhana-K/ b2de315d95905b68](https://www.indeed.com/r/Govardhana-K/b2de315d95905b68) Total IT experience 5 Years 6 Months Cloud Lending Solutions INC 4 Month • Salesforce Developer Oracle 5 Years 2 Month • 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, 1248, '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')],
```

OUTPUT SCREENSHOTS



The screenshot displays a Jupyter Notebook interface. At the top, there's a toolbar with icons for menu, search, and file management, along with buttons for '+ Code' and '+ Text'. On the right side of the toolbar, there are 'Connect' and 'Colab AI' buttons. The main area contains a code cell with the following Python code:

```
[ ] doc = nlp_model(tx)
for ent in doc.ents:
    print(f'{ent.label_.upper():{30}}- {ent.text}')
```

Below the code cell, the output is displayed as a list of named entities. Each entry consists of a label (left-aligned) and a corresponding text snippet (right-aligned, preceded by a hyphen). The labels include PERSON, ORG, GPE, NORP, DATE, and CARDINAL. The text snippets include names like 'Michael Smith', 'Manchester', and various Microsoft-related terms like 'Server Technologies', 'Applications', 'SQL', 'StreamAnalytics', 'Power BI', 'U-SQL', 'Microsoft Rewards', 'Bing', 'Microsoft Edge', 'Xbox Store', 'the Microsoft Store', and 'Rewards'.

Label	Text
PERSON	- Michael Smith
ORG	- BI / Big Data/
ORG	- Manchester
ORG	- Designing, Development, Administration
GPE	- Client
ORG	- Server Technologies
ORG	- Applications
ORG	- SQL
PERSON	- Stored Procedures
ORG	- Microsoft
PERSON	- Document DB
ORG	- SQL Azure
NORP	- StreamAnalytics
ORG	- Power BI
GPE	- Web Job
ORG	- U-SQL
ORG	- Microsoft - Manchester
GPE	- UK
DATE	- December 2015
CARDINAL	- 1
ORG	- Microsoft
ORG	- Microsoft
ORG	- Microsoft Rewards
ORG	- Bing
ORG	- Microsoft Edge
PRODUCT	- Xbox Store
ORG	- the Microsoft Store
ORG	- Microsoft
NORP	- Rewards

RESULTS

The resume parser demonstrated remarkable accuracy and efficiency in extracting pertinent information from a diverse range of resumes. Leveraging advanced natural language processing techniques and machine learning algorithms, the parser successfully identified and categorized key sections such as Personal Information, Education, Experience, and Skills with precision. Notably, it adeptly handled various resume formats and layouts, effectively parsing both structured and unstructured content.

DEMO LINK: **RESUME PARSER**