UNIVERSITI MALAYSIA PAHANG
AL-SULTAN ABDULLAH

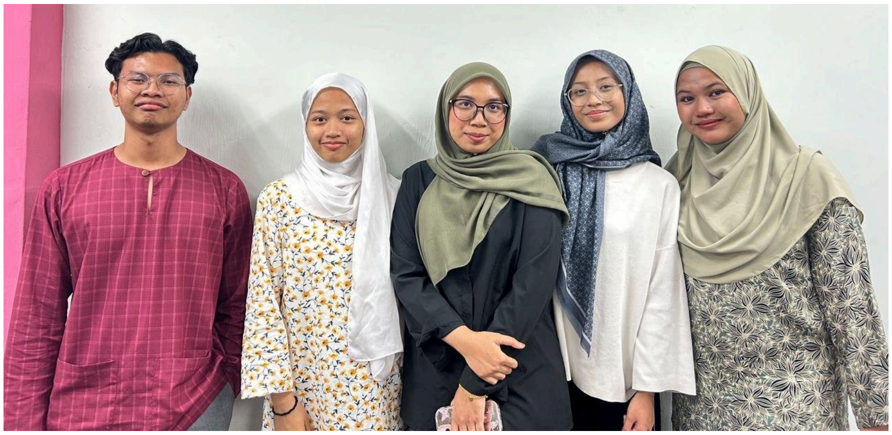
BSD2513
ARTIFICIAL INTELLIGENCE
SEMESTER II 23/24

GROUP PROJECT

GROUP : JESUMER

TITLE : RESUME ANALYZER

PREPARED FOR : DR KU MUHAMMAD NA'IM KU KHALIF



GROUP MEMBERS:

NAME	STUDENT ID
HAWA HUMAIRA BINTI HAMUZAN	SD22043
SITI MAISARAH BINTI SUHARDI	SD22006
NURUL SYAFIQAH NATASHA BINTI MOHD RAZI	SD22038
ABDUL HAZIQ AZIM BIN ABDUL MALIK	SD22025
NOR ADLIN SOFEA BINTI NOR HAIRUDIN	SD22024

TABLE OF CONTENT

1.0 Introduction

1.1 Project Description

In this project, we are going to use spacy for entity recognition on 200 Resume and experiment around various NLP tools for text analysis. The main purpose of this project is to help recruiters throw hundreds of applications within a few minutes. We have also added skills match features so that hiring managers can follow a metric that will help them to decide whether they should move to the interview stage or not. We will be using two datasets; the first contains resume texts and the second contains skills that we will use to create an entity ruler.This project is to cater to 3 objectives which are to create a screening resume system to reduce the time and energy spent by human recruiters. Next is to improve the quality acceptance of applicants based on the requirements stated and not being biased. Lastly, is to implement better decision-making with a more efficient and faultless hiring system.

1.2 Problem Statement

All companies encounter problems at one time or another. That is why companies need high quality employees on hand who can solve problems like dealing with changing client-needs to deal with the company's needs. Therefore, a company needs to hire competitive and reliable workers that suit the company needs. In this highly competitive era, companies are packed with a large number of new applicant resumes and Curriculum Vitae (CV) making the reviewing process longer and harder for the recruiter. Recruiter most of the time fail to capture bright applicant qualifications because the hundreds of resumes or CVs that need to be reviewed make the hiring process are not as effective as it needs to be.

Therefore, to overcome this problem, we were inspired to create a resume analyser where we speed up the recruitment process with a more accurate evaluation . This project can be done with the advancement of Natural Language Processing (NLP) and machine learning algorithms which can handle hundreds resumes or CV effectively like extracting relevant information that is related to requirement.The successful implementation of this Resume Analyzer will lead to a more efficient, accurate, and equitable recruitment process. It will enable recruiters to quickly identify the best candidates, reduce time-to-hire, and enhance the overall quality of the hiring process, ultimately contributing to the success of organizations in attracting and retaining top talent.

1.3 Basic Description Of The Data

Dataset Description: LiveCareer.com Resume Dataset

Overview

The LiveCareer. Description The Resume Dataset This dataset includes 2380 resume examples taken from the LiveCareer. com Resume Dataset com website. This template is useful if you want to categorize resumes or curriculum vitae, among other text analysis and natural language processing working with previously defined job categories.

Contents of the Dataset

The dataset is in CSV file format and contains the following features :

ID: A way to insulate each PDF resume with a unique identifier (file name).

- Resume_str: The resume textual content which is in form of string, to be able to embody all of the textual segment from the

resume.

Resume_html: The actual resume HTML as in the web scraping phase.

keeping the format and structure as it is.

Category: The category of the job for which a resume was used. This leaves us with 25 categories still in play.

Job Categories

The resumes are categorized into the following job sectors:

- HR
- Designer
- Information-Technology

- Teacher
- Advocate
- Business-Development
- Healthcare
- Fitness
- Agriculture
- BPO (Business Process Outsourcing)
- Sales
- Consultant
- Digital-Media
- Automobile
- Chef
- Finance
- Apparel
- Engineering
- Accountant
- Construction
- Public-Relations
- Banking
- Arts
- Aviation

Acknowledgements

The data was gathered by web-scraping individual resume examples from LiveCareer. com website. Source code used for scraping this data can be found in the author's GitHub repository. This dataset can be utilized for diverse cases like Text Classification, Resume Parsing, Job Matching Algorithms.

2.0 Summary Of Project

2.1 Project Objective

- Extracts key information from resumes using entity recognition.
- Matches candidate skills with job requirements to evaluate suitability.
- Reduces the time and effort required for initial resume screening.

2.2 Project Question

- How can natural language processing (NLP) tools be utilized to accurately extract key information such as personal details, education, work experience, and skills from resumes?
- What strategies can be used to effectively match the extracted skills from resumes with predefined job-specific skill requirements to evaluate candidate suitability?
- How can the integration of automated resume analysis and skills matching improve the efficiency and effectiveness of the recruitment process by reducing the time and effort involved in initial resume screening for hiring managers and recruiters?

2.3 Project Content

The objective of this project is to develop a recommendation system for resumes, analogous to Amazon's electronic goods recommendations. The project aims to leverage a curated dataset comprising diverse resumes, encompassing details such as job categories, skills, experiences, endorsements, and professional achievements. By harnessing advanced algorithms, our goal is to personalize job recommendations for users, tailoring suggestions based on their career preferences and aspirations.

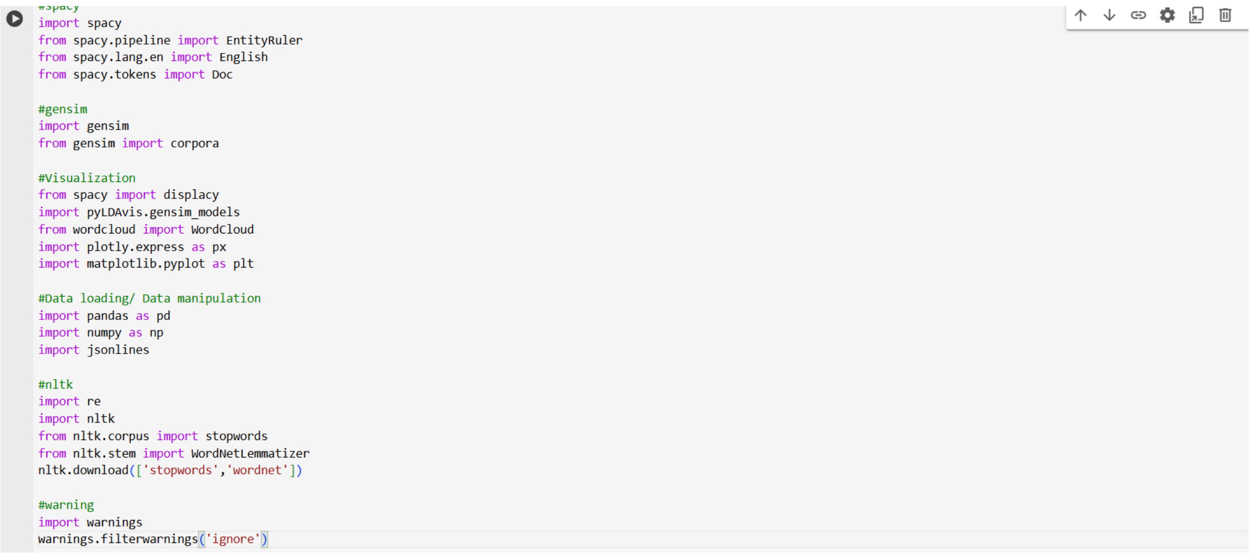
3.0 Methodology

3.1 Data Collection

We found this topic at GITHUB. The dataset that was used consists of resumes that were gathered in a CSV file format and given the name "Resume.csv” that we collected from Kaggle. This document contains a variety of information pertaining to jobs as well as written material taken from resumes.

3.2 Coding of project without GUI

The application that we used in this project is Google Collab. There are several libraries that were used in our coding. The libraries used are :



```
#spacy
import spacy
from spacy.pipeline import EntityRuler
from spacy.lang.en import English
from spacy.tokens import Doc

#gensim
import gensim
from gensim import corpora

#Visualization
from spacy import displacy
import pyLDAvis.gensim_models
from wordcloud import WordCloud
import plotly.express as px
import matplotlib.pyplot as plt

#Data loading/ Data manipulation
import pandas as pd
import numpy as np
import jsonlines

#nltk
import re
import nltk
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
nltk.download(['stopwords','wordnet'])

#warning
import warnings
warnings.filterwarnings('ignore')
```

spaCy

- `import spacy`: Imports the main spaCy library for NLP tasks.
- `from spacy.pipeline import EntityRuler`: Adds custom named entity recognition rules.
- `from spacy.lang.en import English`: Loads the English language model.
- `from spacy.tokens import Doc`: Defines the structure of a document in spaCy.

Gensim

- `import gensim`: Provides tools for topic modeling.
- `from gensim import corpora`: Manages the creation of dictionaries for LDA models.

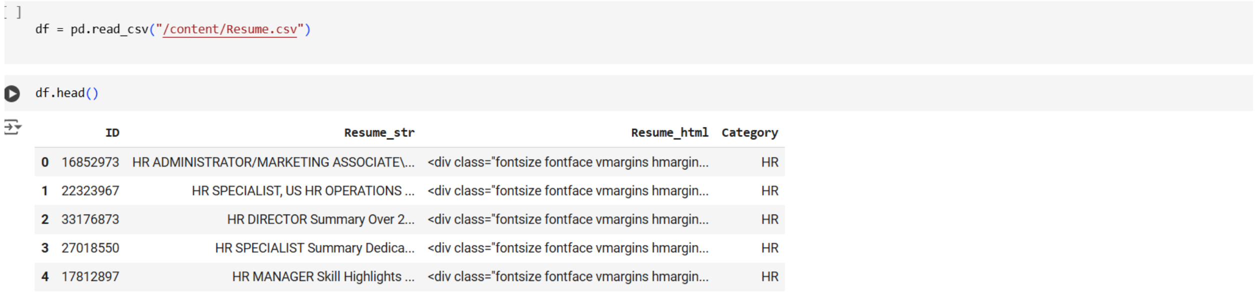
Visualization

- `from spacy import displacy`: Visualizes spaCy's parsed results.
- `import pyLDAvis.gensim_models`: Visualizes LDA topic models.
- `from wordcloud import WordCloud`: Creates word clouds for text data.
- `import plotly.express as px`: Generates interactive plots.
- `import matplotlib.pyplot as plt`: Plots static graphs.

Data Loading/Manipulation
- `import pandas as pd`: Handles data frames and data manipulation.
- `import numpy as np`: Provides numerical operations on arrays.
- `import jsonlines`: Reads and writes JSON Lines format.

NLTK
- `import re`: Uses regular expressions for text processing.
- `import nltk`: Provides NLP tools, including corpus and text processing functions.
- `from nltk.corpus import stopwords`: Loads stopwords for removal.
- `from nltk.stem import WordNetLemmatizer`: Lemmatizer words to their base form.
- `nltk.download(['stopwords','wordnet'])`: Downloads necessary NLTK data.

Warnings
- `import warnings`: Suppresses warnings to clean up the output.



```
df = pd.read_csv("/content/Resume.csv")
```

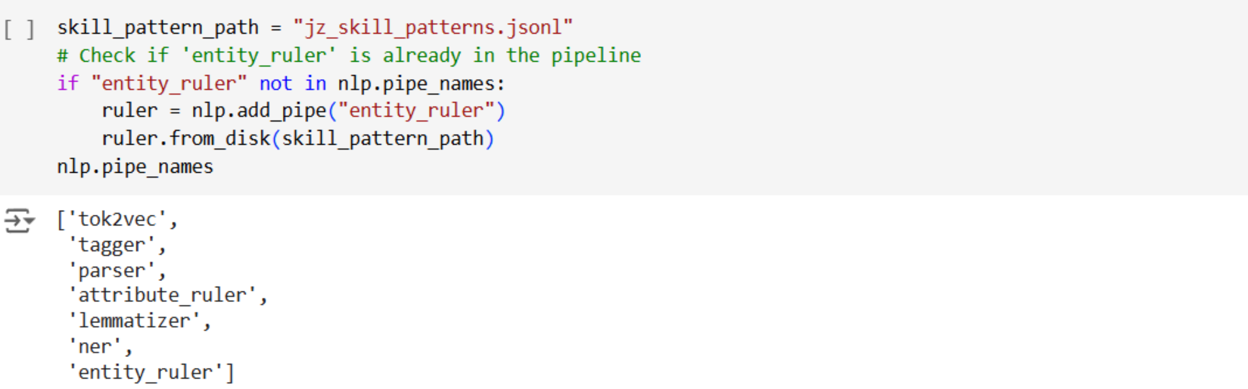
```
df.head()
```

	ID	Resume_str	Resume_html	Category
0	16852973	HR ADMINISTRATOR/MARKETING ASSOCIATE\...	<div class="fontsize fontface vmargins hmargin...	HR
1	22323967	HR SPECIALIST, US HR OPERATIONS ...	<div class="fontsize fontface vmargins hmargin...	HR
2	33176873	HR DIRECTOR Summary Over 2...	<div class="fontsize fontface vmargins hmargin...	HR
3	27018550	HR SPECIALIST Summary Dedica...	<div class="fontsize fontface vmargins hmargin...	HR
4	17812897	HR MANAGER Skill Highlights ...	<div class="fontsize fontface vmargins hmargin...	HR

This code reads the CSV file and stores the data in a pandas DataFrame called df. The head () function is used to display the first 5 rows of the DataFrame.

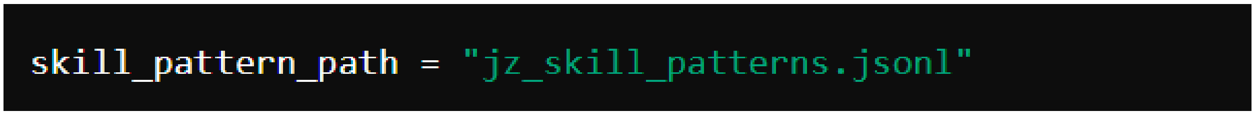
Entity Ruler

A pipeline must be added before the.json file containing the talents can be loaded into the entity ruler. We have successfully added a new pipeline entity_ruler, as you can see. In our scenario, the skills and job description categories are highlighted within the text with the use of extra rules that we may create using the entity ruler.



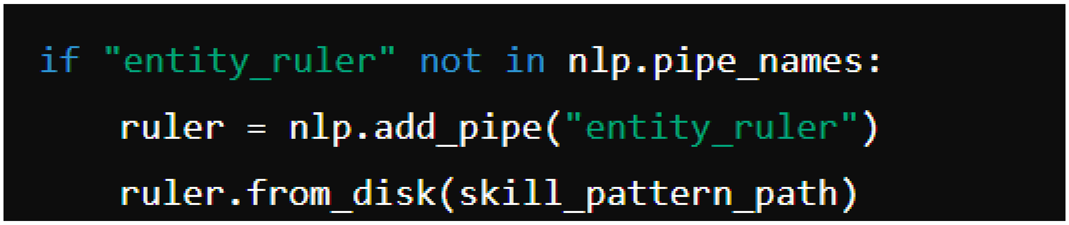
```
[ ] skill_pattern_path = "jz_skill_patterns.jsonl"
    # Check if 'entity_ruler' is already in the pipeline
    if "entity_ruler" not in nlp.pipe_names:
        ruler = nlp.add_pipe("entity_ruler")
        ruler.from_disk(skill_pattern_path)
    nlp.pipe_names
```

```
['tok2vec',
 'tagger',
 'parser',
 'attribute_ruler',
 'lemmatizer',
 'ner',
 'entity_ruler']
```



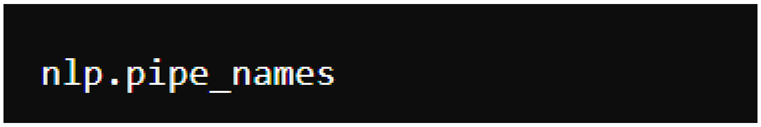
```
skill_pattern_path = "jz_skill_patterns.jsonl"
```

This line specifies the path of a JSON file containing patterns for text-based talent recognition to the variable skill_pattern_path.



```
if "entity_ruler" not in nlp.pipe_names:
    ruler = nlp.add_pipe("entity_ruler")
    ruler.from_disk(skill_pattern_path)
```

The entity_ruler component's presence in the nlp pipeline is verified by this block.
If not, it imports the skill patterns from the given JSON file using from_disk and adds the entity_ruler to the pipeline.



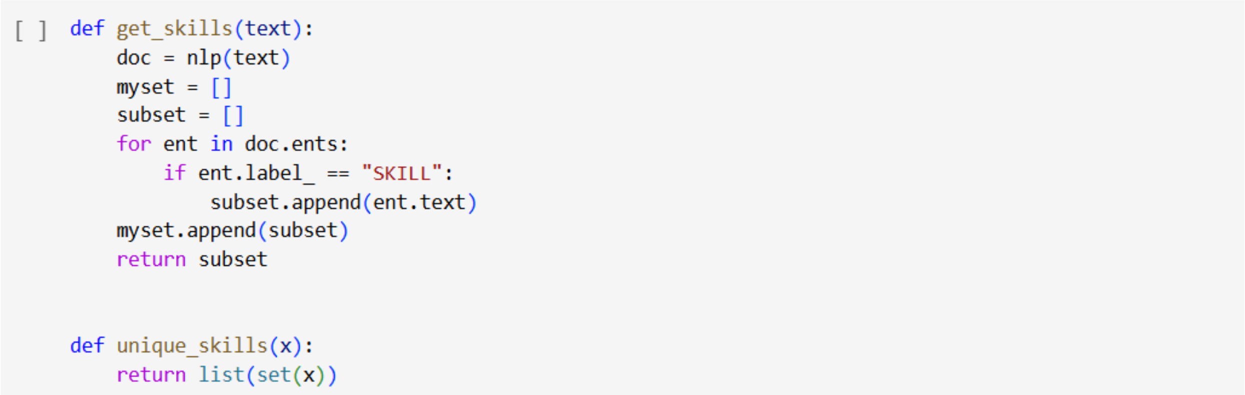
```
nlp.pipe_names
```

This command lists the components currently in the SpaCy pipeline. The output shows the following components:

- `'tok2vec'`
- `'tagger'`
- `'parser'`
- `'attribute_ruler'`
- `'lemmatizer'`
- `'ner'`
- `'Entity_ruler'`

The process to extract every skill from a CV and produce an array with every skill, we will write two Python methods. We'll use this function to our dataset later on to add a new feature we're calling skill. This will enable us to see patterns and trends in the dataset.

-`get_skills` is going to extract skills from a single text.
-`unique_skills` will remove duplicates.

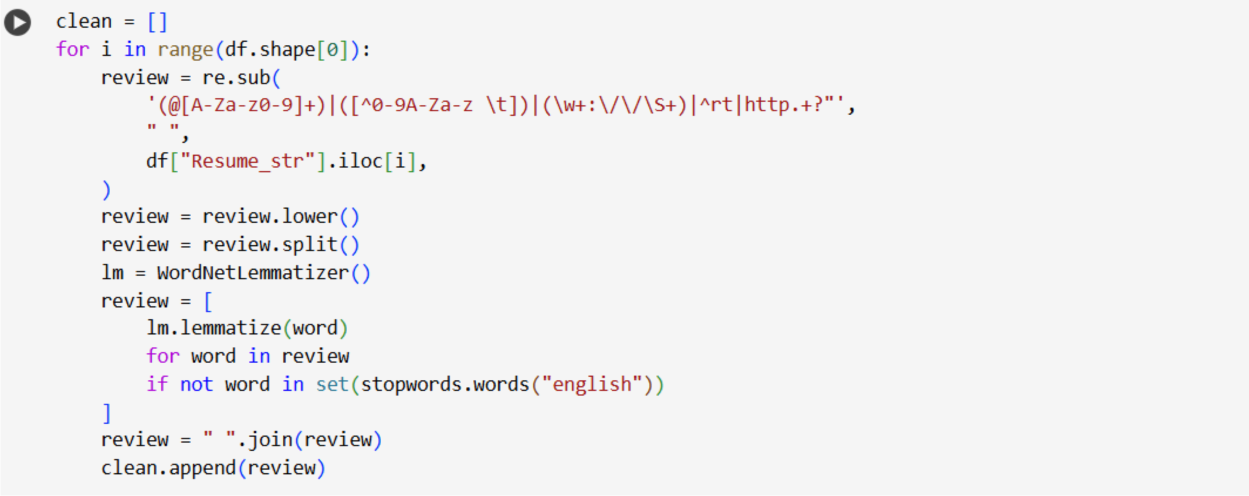


```
def get_skills(text):
    doc = nlp(text)
    myset = []
    subset = []
    for ent in doc.ents:
        if ent.label_ == "SKILL":
            subset.append(ent.text)
    myset.append(subset)
    return subset


def unique_skills(x):
    return list(set(x))
```

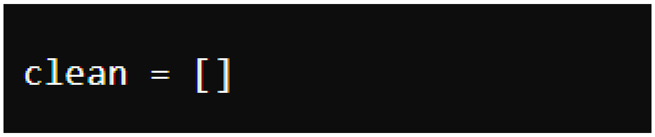
Cleaning Process :

In a few stages, we will utilize the nltk package to clean our dataset:

-Regex will be used to eliminate punctuation, special characters, and hyperlinks.
-Text lowering
-Splitting text into array based on space
-Lemmatizing text to its base form for normalizations
-Removing English stopwords
-Appending the results into an array.

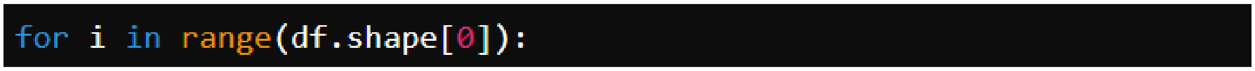


```python
clean = []
for i in range(df.shape[0]):
    review = re.sub(
        '(@[A-Za-z0-9]+)|([^0-9A-Za-z \t])|(\w+:\/\/\S+)|^rt|http.+?"',
        " ",
        df["Resume_str"].iloc[i],
    )
    review = review.lower()
    review = review.split()
    lm = WordNetLemmatizer()
    review = [
        lm.lemmatize(word)
        for word in review
        if not word in set(stopwords.words("english"))
    ]
    review = " ".join(review)
    clean.append(review)
```



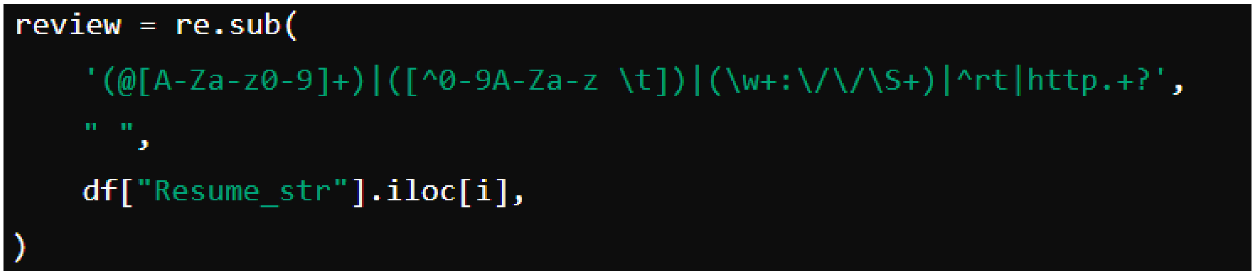
```python
clean = []
```

- The cleaned resume text is first stored in an empty list.



```python
for i in range(df.shape[0]):
```

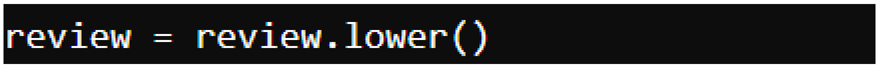
- The DataFrame is configured with a loop that iterates across each row (resume).



```python
review = re.sub(
    '(@[A-Za-z0-9]+)|([^0-9A-Za-z \t])|(\w+:\/\/\S+)|^rt|http.+?',
    " ",
    df["Resume_str"].iloc[i],
)
```

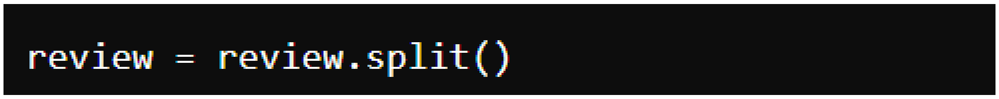
`‘re.sub’` is used to remove special characters, URLs, and unnecessary text from each resume string.
This pattern matches and replaces:

- Mentions `(@[A-Za-z0-9]+)`
- Non-alphanumeric characters except spaces and tabs `([^0-9A-Za-z \t])`
- URLs `(\w+:\/\/\S+)`
- Retweets `(^rt)`
- Any text starting with `‘http’`



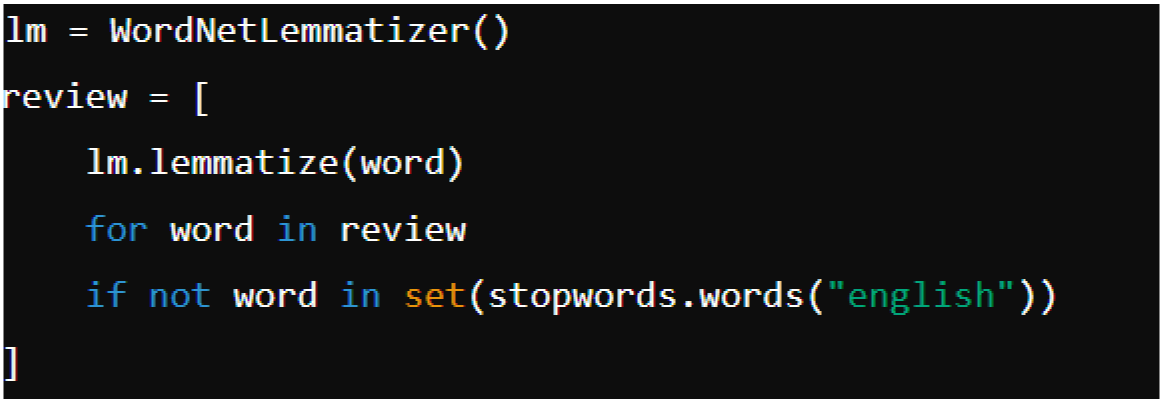
```
review = review.lower()
```

- Converts the cleaned text to lowercase.



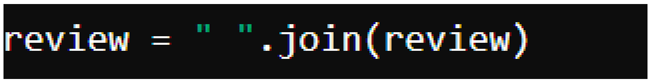
```
review = review.split()
```

- Splits the text into individual words.



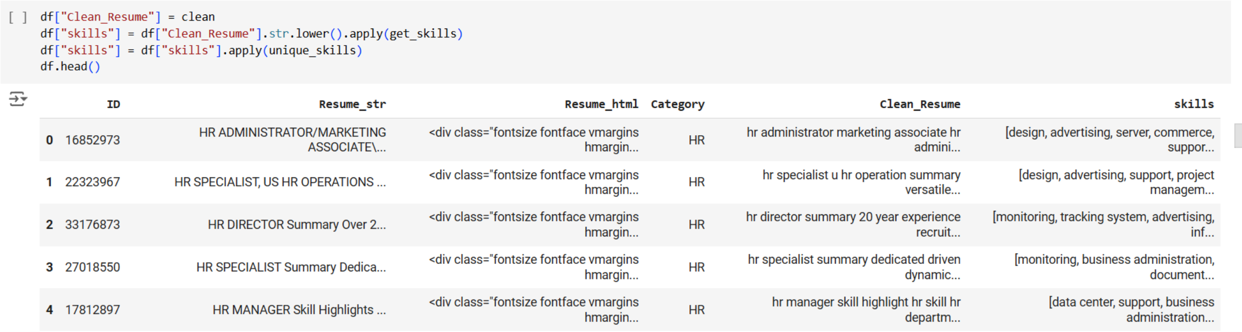
```
lm = WordNetLemmatizer()
review = [
    lm.lemmatize(word)
    for word in review
    if not word in set(stopwords.words("english"))
]
```

Initializes the `WordNetLemmatizer` to lemmatize words. Iterates through the list of words, lemmatizes each word, and removes stopwords.



```
review = " ".join(review)
```

Joins the cleaned and lemmatized words back into a single string.

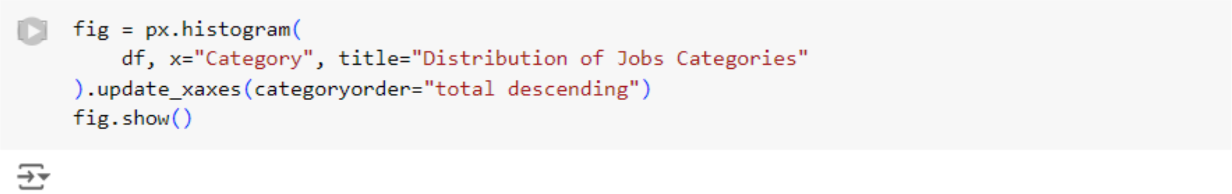


```
[ ] df["Clean_Resume"] = clean
    df["skills"] = df["Clean_Resume"].str.lower().apply(get_skills)
    df["skills"] = df["skills"].apply(unique_skills)
    df.head()
```

	ID	Resume_str	Resume_html	Category	Clean_Resume	skills
0	16852973	HR ADMINISTRATOR/MARKETING ASSOCIATE\...	<div class="fontsize fontface vmargins hmargin...	HR	hr administrator marketing associate hr admini...	[design, advertising, server, commerce, suppor...
1	22323967	HR SPECIALIST, US HR OPERATIONS ...	<div class="fontsize fontface vmargins hmargin...	HR	hr specialist u hr operation summary versatile...	[design, advertising, support, project managem...
2	33176873	HR DIRECTOR Summary Over 2...	<div class="fontsize fontface vmargins hmargin...	HR	hr director summary 20 year experience recruit...	[monitoring, tracking system, advertising, inf...
3	27018550	HR SPECIALIST Summary Dedica...	<div class="fontsize fontface vmargins hmargin...	HR	hr specialist summary dedicated driven dynamic...	[monitoring, business administration, document...
4	17812897	HR MANAGER Skill Highlights ...	<div class="fontsize fontface vmargins hmargin...	HR	hr manager skill highlight hr skill hr departm...	[data center, support, business administration...

Adds a new column `Clean_Resume` to the DataFrame containing the cleaned resume text from the `clean` list.Converts the text in the `Clean_Resume` column to lowercase.Applies the `get_skills` function to extract skills from the cleaned resume text.Adds the extracted skills to a new column `skills`.Applies the `unique_skills` function to the `skills` column to remove any duplicate skills.Displays the first few rows of the DataFrame to verify the results.

3.3 Exploratory Data Analysis (EDA)

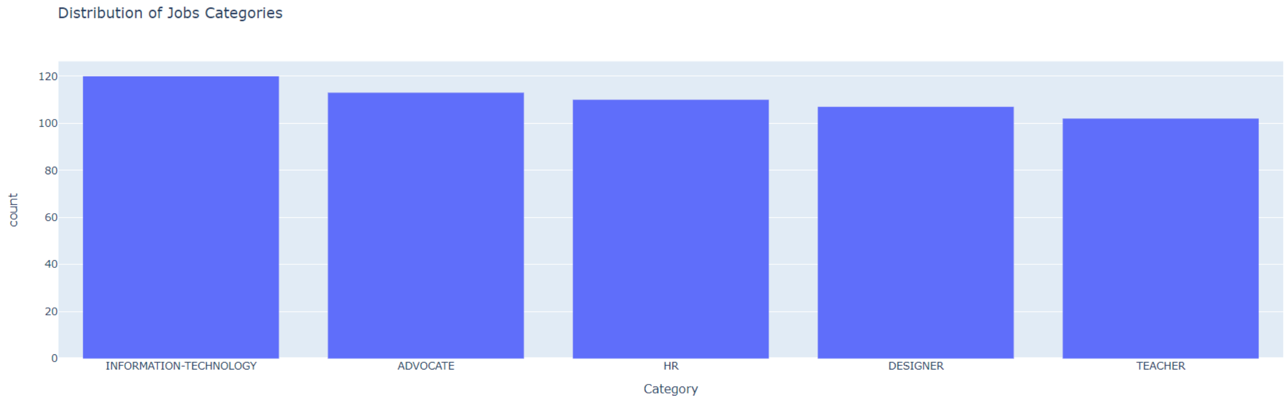


```
fig = px.histogram(
    df, x="Category", title="Distribution of Jobs Categories"
).update_xaxes(categoryorder="total descending")
fig.show()
```

Steps and Interpretation:

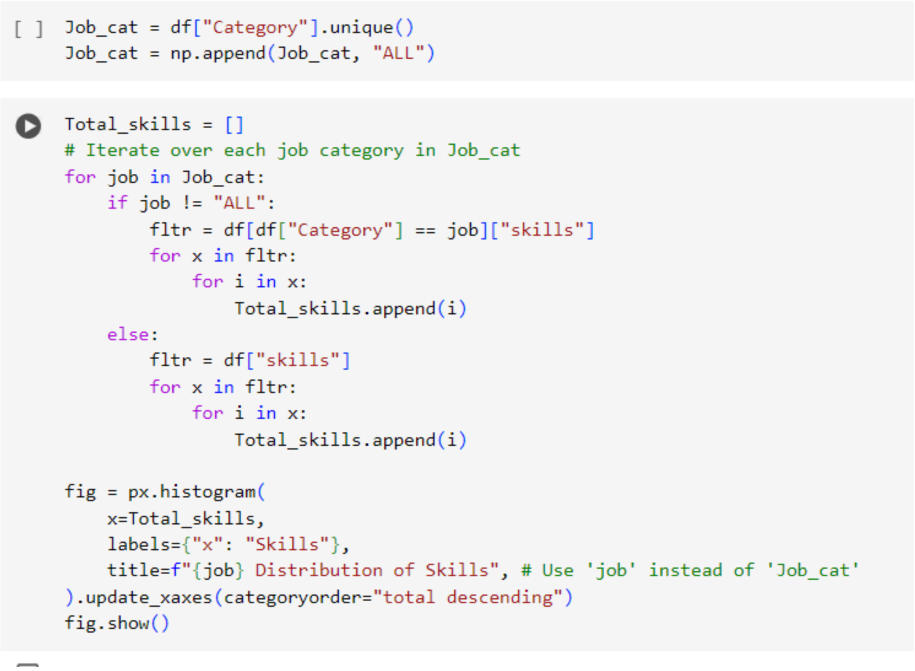
Create Histogram:

px.histogram is used to create a histogram with the DataFrame df. The x-axis (x) is set to the "Category" column, which contains different job categories.The title of the histogram is set to "Distribution of Jobs Categories". update_xaxes(categoryorder="total descending") sorts the x-axis categories in descending order based on their total count. fig.show() displays the histogram



Visualization Interpretation:

The histogram shows the count of resumes in each job category. The categories are sorted in descending order of their count, making it easy to see which job categories are most common in the dataset. The bars represent the number of resumes for each category, providing a clear visual distribution of job categories. For example, "INFORMATION-TECHNOLOGY" has the highest count, followed by "ADVOCATE", "HR", "DESIGNER", and "TEACHER".

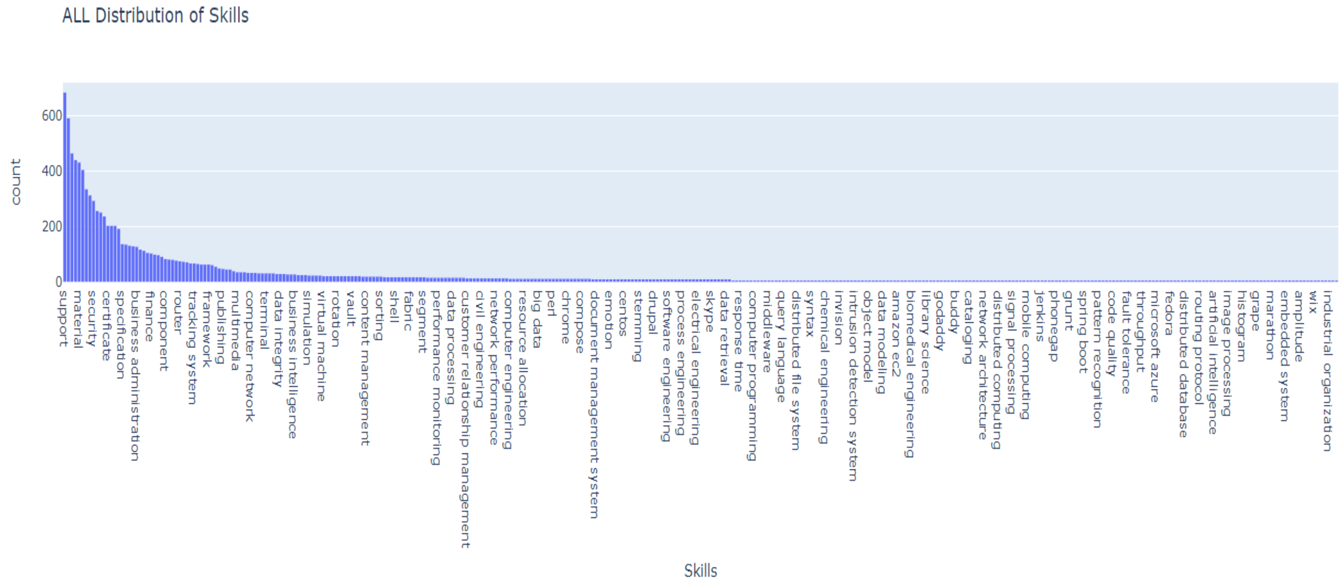


```
Job_cat = df["Category"].unique()
Job_cat = np.append(Job_cat, "ALL")
```

```
Total_skills = []
# Iterate over each job category in Job_cat
for job in Job_cat:
    if job != "ALL":
        fltr = df[df["Category"] == job]["skills"]
        for x in fltr:
            for i in x:
                Total_skills.append(i)
    else:
        fltr = df["skills"]
        for x in fltr:
            for i in x:
                Total_skills.append(i)

fig = px.histogram(
    x=Total_skills,
    labels={"x": "Skills"},
    title=f"{job} Distribution of Skills", # Use 'job' instead of 'Job_cat'
).update_xaxes(categoryorder="total descending")
fig.show()
```

Steps and Interpretation

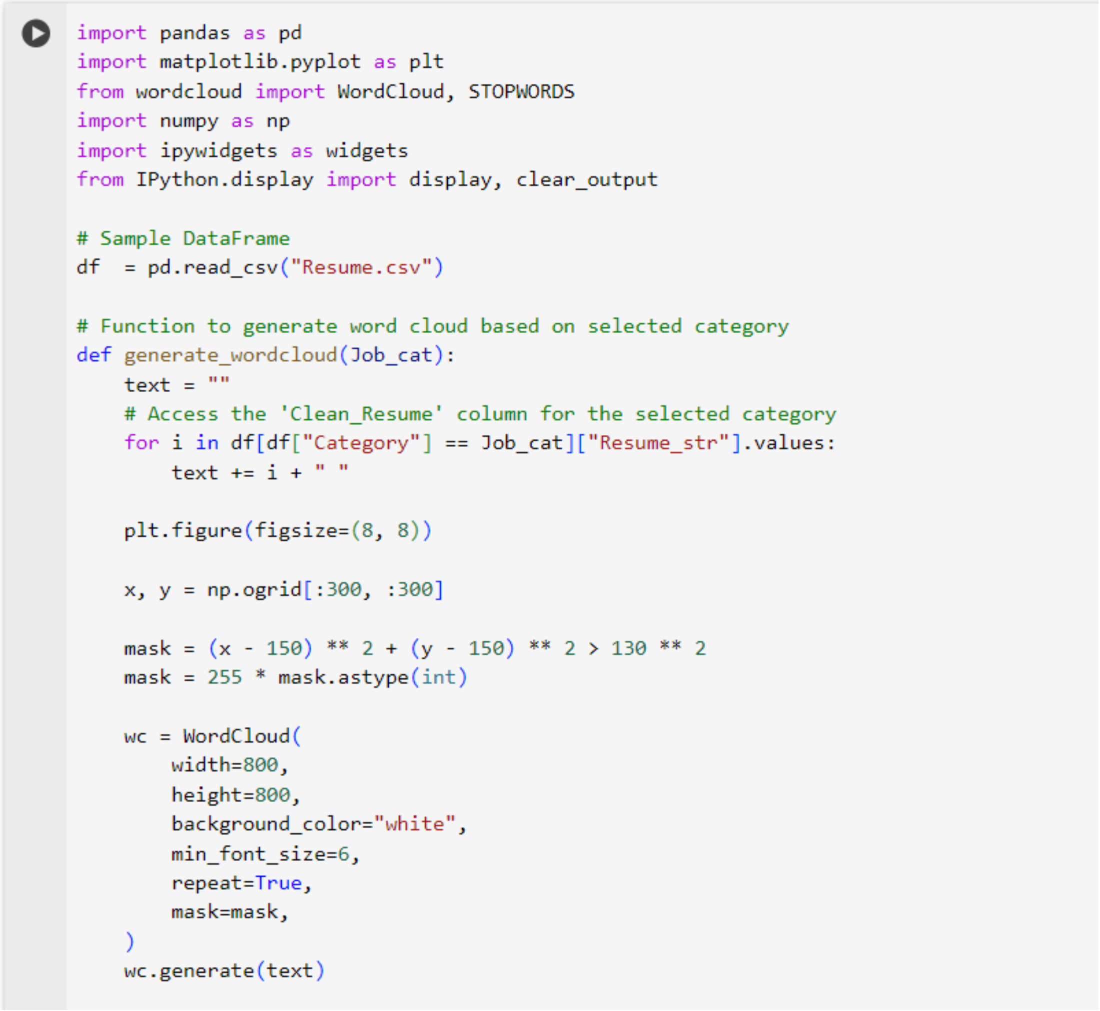
`df["Category"].unique()` extracts unique job categories from the DataFrame.`np.append(Job_cat, "ALL")` adds an "ALL" option to include all categories.The loop iterates through each job category in `Job_cat`. If the category is not "ALL", it filters resumes of that specific category and appends their skills to `Total_skills`.If the category is "ALL", it includes skills from all resumes.A histogram is created with `Total_skills` on the x-axis.The x-axis is labeled as "Skills".The histogram title is set dynamically based on the job category.The x-axis categories are ordered in descending order based on their total count. The histogram is displayed using `fig.show()`.



The visualization above shows the list of skills across different job scope where the most used skills will be on the left and go downward to least use skills. . The following skills are listed in descending order - Support DatabaseMarketing Business administration, management etc. are what we see most frequently. This proves that demand for these skills is high, and we found them showing up in a ton of resumes looked at.

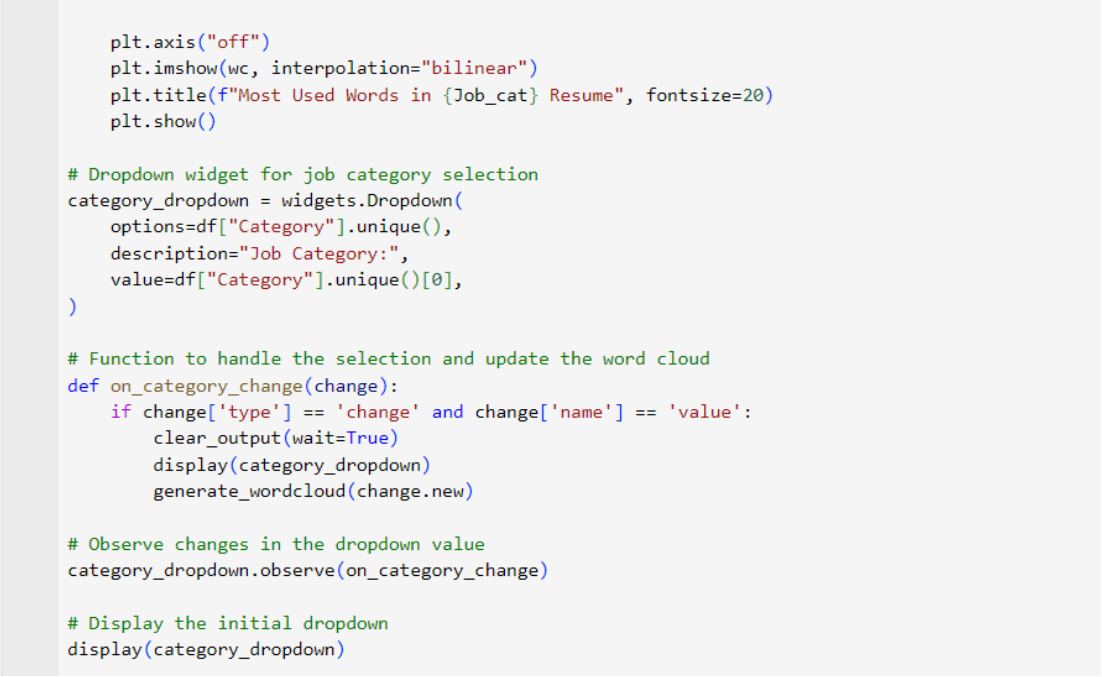
Lastly, the graphic above demonstrates a varying set of abilities, capturing the various competencies of the people in whose resumes were read. While the bulk of skills fall into those few common skills that are mentioned a lot, there is also a long tail of less common skills. This will also allow us to get a feel for the skill set in the job market, and so help manage talent and help match the right people to the right jobs.

Based on the graphic above shows the most important skills trend from the resume that might be helpful for companies and recruiters to make sure it align with the requirement needed. The highest common skills that are stated in the resume is the word ‘support’ where others applicant can shows the most common skills that many people are likely to have, as well as some less common skills that might be harder to find. This can help with hiring, training, and career growth, making sure that people with the right skills are put in the right jobs.



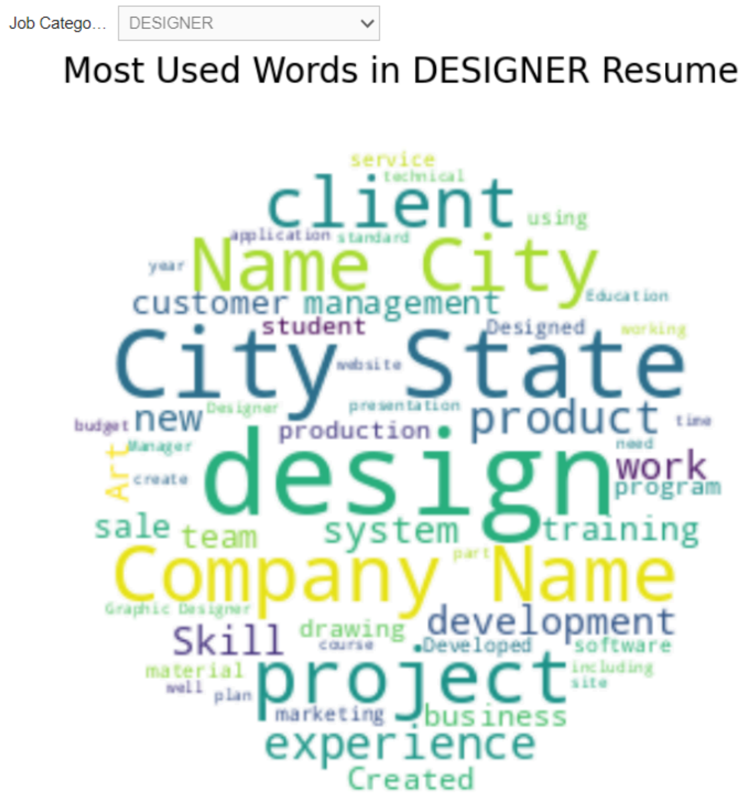
```
import pandas as pd
import matplotlib.pyplot as plt
from wordcloud import WordCloud, STOPWORDS
import numpy as np
import ipywidgets as widgets
from IPython.display import display, clear_output

# Sample DataFrame
df  = pd.read_csv("Resume.csv")

# Function to generate word cloud based on selected category
def generate_wordcloud(Job_cat):
    text = ""
    # Access the 'Clean_Resume' column for the selected category
    for i in df[df["Category"] == Job_cat]["Resume_str"].values:
        text += i + " "

    plt.figure(figsize=(8, 8))

    x, y = np.ogrid[:300, :300]

    mask = (x - 150) ** 2 + (y - 150) ** 2 > 130 ** 2
    mask = 255 * mask.astype(int)

    wc = WordCloud(
        width=800,
        height=800,
        background_color="white",
        min_font_size=6,
        repeat=True,
        mask=mask,
    )
    wc.generate(text)
```

These libraries are used for data manipulation (pandas), visualization (matplotlib, wordcloud), numerical operations (numpy), and interactive widgets (ipywidgets). The resume data is loaded into a pandas DataFrame from a CSV file. Concatenate Text: Initializes an empty string text and concatenates resume text from the Resume_str column for the selected job category (Job_cat). Concatenate Text function is to initializes an empty string text and concatenates resume text from the Resume_str column for the selected job category (Job_cat).WordCloud Object function is to configures the WordCloud object with specified dimensions, background color, minimum font size, repeat flag, and the circular mask.Generate Word Cloud: Generates the word cloud using the concatenated text.



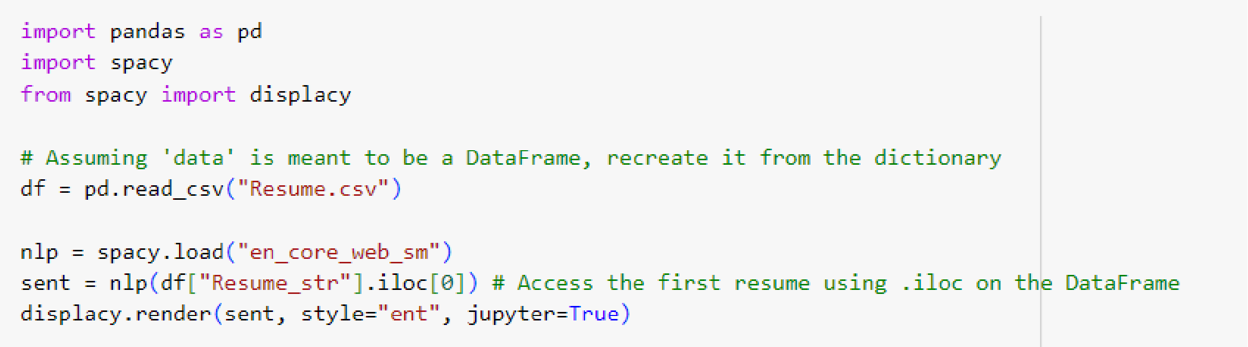
```
    plt.axis("off")
    plt.imshow(wc, interpolation="bilinear")
    plt.title(f"Most Used Words in {Job_cat} Resume", fontsize=20)
    plt.show()

# Dropdown widget for job category selection
category_dropdown = widgets.Dropdown(
    options=df["Category"].unique(),
    description="Job Category:",
    value=df["Category"].unique()[0],
)

# Function to handle the selection and update the word cloud
def on_category_change(change):
    if change['type'] == 'change' and change['name'] == 'value':
        clear_output(wait=True)
        display(category_dropdown)
        generate_wordcloud(change.new)

# Observe changes in the dropdown value
category_dropdown.observe(on_category_change)

# Display the initial dropdown
display(category_dropdown)
```

A dropdown widget (`category_dropdown`) allows users to select a job category. The `on_category_change` function handles the selection changes, updating and displaying the word cloud accordingly. This setup enables users to visualize the most frequently used words in resumes for different job categories interactively.



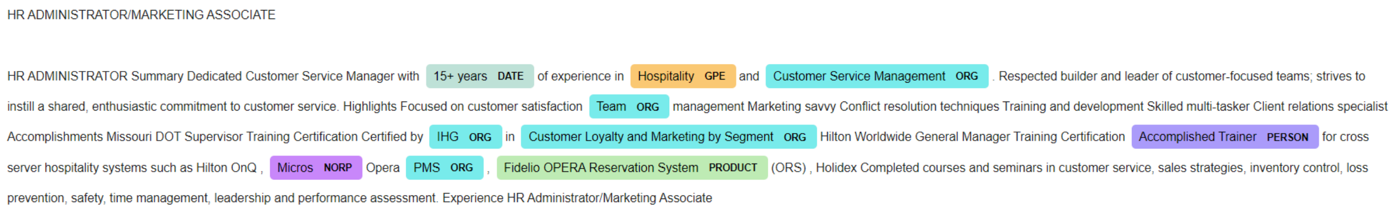
Most Used Words in DESIGNER Resume

The output is a word cloud generated from resumes for the "DESIGNER" job category. The most frequently mentioned words are larger and more prominently displayed. Key terms like "design," "project," "Company," "Name," "City," and "State" are highlighted, indicating their common occurrence in designer resumes. This visualization helps identify important themes and skills associated with the designer role, such as "client," "management," "development," and "experience." It provides insights into the essential elements and frequently discussed topics within designer resumes.

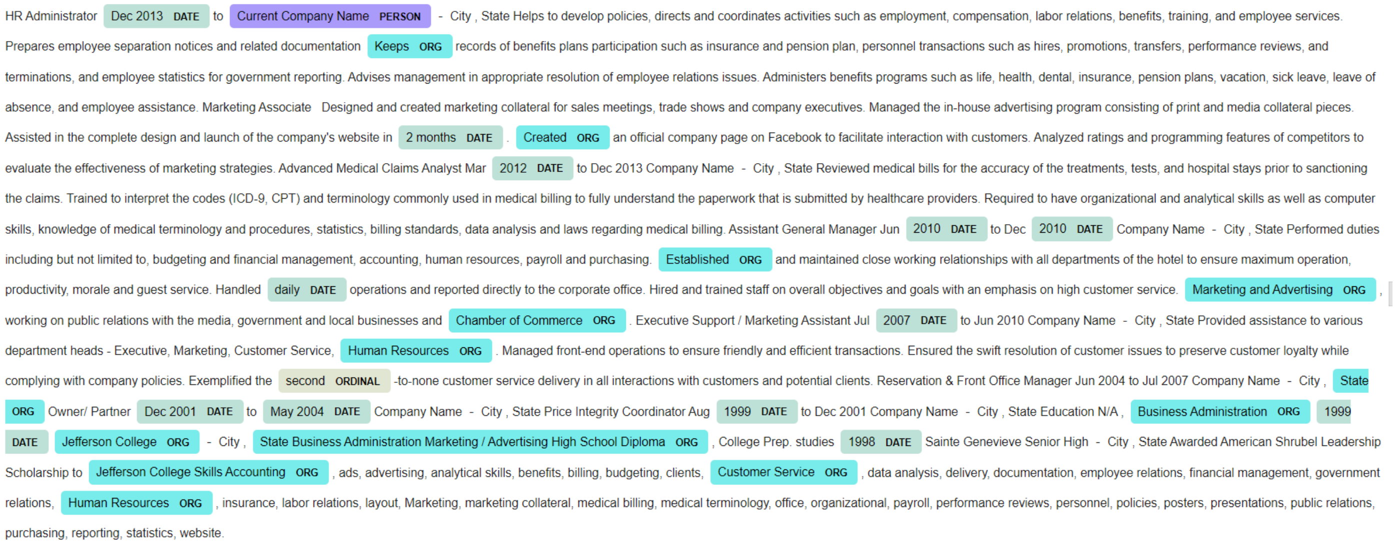


```
import pandas as pd
import spacy
from spacy import displacy

# Assuming 'data' is meant to be a DataFrame, recreate it from the dictionary
df = pd.read_csv("Resume.csv")

nlp = spacy.load("en_core_web_sm")
sent = nlp(df["Resume_str"].iloc[0]) # Access the first resume using .iloc on the DataFrame
displacy.render(sent, style="ent", jupyter=True)
```

The provided code snippet loads a resume dataset and uses spaCy to visualize named entities in the first resume. It begins by importing necessary libraries (`pandas` for data manipulation and `spacy` for NLP tasks). The dataset is loaded into a pandas DataFrame from a CSV file. The spaCy model `en_core_web_sm` is loaded to process the text. The code then extracts the text of the first resume from the DataFrame and processes it using the spaCy NLP pipeline. Finally, the named entities in the text are visualized using `displacy.render`, displaying them within a Jupyter notebook environment. This visualization helps in identifying key information such as names, dates, and organizations within the resume.

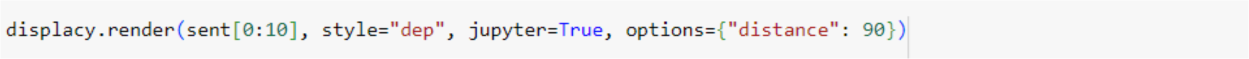


```
HR ADMINISTRATOR/MARKETING ASSOCIATE

HR ADMINISTRATOR Summary Dedicated Customer Service Manager with 15+ years DATE of experience in Hospitality GPE and Customer Service Management ORG . Respected builder and leader of customer-focused teams; strives to
instill a shared, enthusiastic commitment to customer service. Highlights Focused on customer satisfaction Team ORG management Marketing savvy Conflict resolution techniques Training and development Skilled multi-tasker Client relations specialist
Accomplishments Missouri DOT Supervisor Training Certification Certified by IHG ORG in Customer Loyalty and Marketing by Segment ORG Hilton Worldwide General Manager Training Certification Accomplished Trainer PERSON for cross
server hospitality systems such as Hilton OnQ , Micros NORP Opera PMS ORG , Fidelio OPERA Reservation System PRODUCT (ORS) , Holidex Completed courses and seminars in customer service, sales strategies, inventory control, loss
prevention, safety, time management, leadership and performance assessment. Experience HR Administrator/Marketing Associate
```



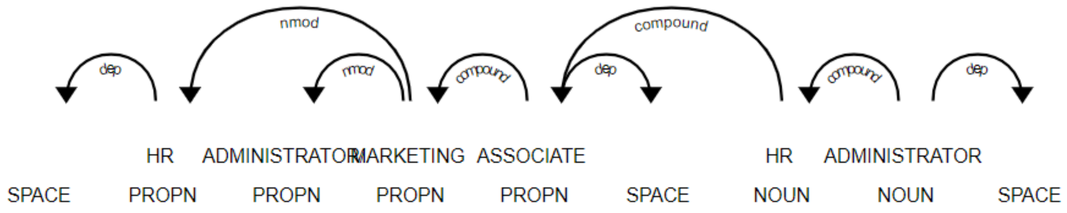
HR Administrator Dec 2013 DATE to Current Company Name PERSON - City , State Helps to develop policies, directs and coordinates activities such as employment, compensation, labor relations, benefits, training, and employee services. Prepares employee separation notices and related documentation Keeps ORG records of benefits plans participation such as insurance and pension plan, personnel transactions such as hires, promotions, transfers, performance reviews, and terminations, and employee statistics for government reporting. Advises management in appropriate resolution of employee relations issues. Administers benefits programs such as life, health, dental, insurance, pension plans, vacation, sick leave, leave of absence, and employee assistance. Marketing Associate Designed and created marketing collateral for sales meetings, trade shows and company executives. Managed the in-house advertising program consisting of print and media collateral pieces. Assisted in the complete design and launch of the company's website in 2 months DATE . Created ORG an official company page on Facebook to facilitate interaction with customers. Analyzed ratings and programming features of competitors to evaluate the effectiveness of marketing strategies. Advanced Medical Claims Analyst Mar 2012 DATE to Dec 2013 Company Name - City , State Reviewed medical bills for the accuracy of the treatments, tests, and hospital stays prior to sanctioning the claims. Trained to interpret the codes (ICD-9, CPT) and terminology commonly used in medical billing to fully understand the paperwork that is submitted by healthcare providers. Required to have organizational and analytical skills as well as computer skills, knowledge of medical terminology and procedures, statistics, billing standards, data analysis and laws regarding medical billing. Assistant General Manager Jun 2010 DATE to Dec 2010 DATE Company Name - City , State Performed duties including but not limited to, budgeting and financial management, accounting, human resources, payroll and purchasing. Established ORG and maintained close working relationships with all departments of the hotel to ensure maximum operation, productivity, morale and guest service. Handled daily DATE operations and reported directly to the corporate office. Hired and trained staff on overall objectives and goals with an emphasis on high customer service. Marketing and Advertising ORG , working on public relations with the media, government and local businesses and Chamber of Commerce ORG . Executive Support / Marketing Assistant Jul 2007 DATE to Jun 2010 Company Name - City , State Provided assistance to various department heads - Executive, Marketing, Customer Service, Human Resources ORG . Managed front-end operations to ensure friendly and efficient transactions. Ensured the swift resolution of customer issues to preserve customer loyalty while complying with company policies. Exemplified the second ORDINAL -to-none customer service delivery in all interactions with customers and potential clients. Reservation & Front Office Manager Jun 2004 to Jul 2007 Company Name - City , State ORG Owner/ Partner Dec 2001 DATE to May 2004 DATE Company Name - City , State Price Integrity Coordinator Aug 1999 DATE to Dec 2001 Company Name - City , State Education N/A , Business Administration ORG 1999 DATE Jefferson College ORG - City , State Business Administration Marketing / Advertising High School Diploma ORG , College Prep. studies 1998 DATE Sainte Genevieve Senior High - City , State Awarded American Shrubel Leadership Scholarship to Jefferson College Skills Accounting ORG , ads, advertising, analytical skills, benefits, billing, budgeting, clients, Customer Service ORG , data analysis, delivery, documentation, employee relations, financial management, government relations, Human Resources ORG , insurance, labor relations, layout, Marketing, marketing collateral, medical billing, medical terminology, office, organizational, payroll, performance reviews, personnel, policies, posters, presentations, public relations, purchasing, reporting, statistics, website.

The output from the visualization highlights various named entities extracted from a resume using spaCy's `display.render` method, which uses different color codes to identify and distinguish between them. DATE entities, such as "15+ years" and "Dec 2013," represent specific dates and periods. GPE (Geopolitical Entity), like "Hospitality," includes locations like countries, cities, and states. ORG (Organization) entities, such as "Customer Service Management" and "Team," denote company names and institutions. PERSON entities, like "Accomplished Trainer," refer to individual names. NORP (Nationalities or Religious/Political Groups), such as "Micros," identify groups of people by nationality, religion, or politics. PRODUCT entities, like "OPERA Reservation System," highlight products mentioned in the resume. This visualization aids in quickly understanding the candidate's background, key experiences, and affiliations.



```
displacy.render(sent[0:10], style="dep", jupyter=True, options={"distance": 90})
```

`displacy.render`: This line calls the displacy.render function to generate a dependency parse visualization.
`style`: This argument defines the type of visualization to be generated. In this case, it’s set to “`dep`” for dependency parsing.
`options`: This argument sets additional options for the visualization. Here, it sets the distance argument to 90, which likely controls the spacing between words in the parse visualization.



The image you sent shows a series of arrows pointing in different directions. The words "HR administrator/marketing associate" and "HR administrator" are written on the arrows, with "HR administrator" written in two different languages.

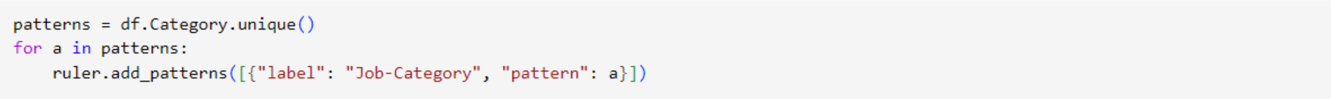
It appears to be a diagram illustrating that the role of "HR administrator" can encompass the tasks and responsibilities of "marketing associate". The arrows branching out from "HR administrator" could represent the various duties or skills required for the position.

Custom Entity Recognition

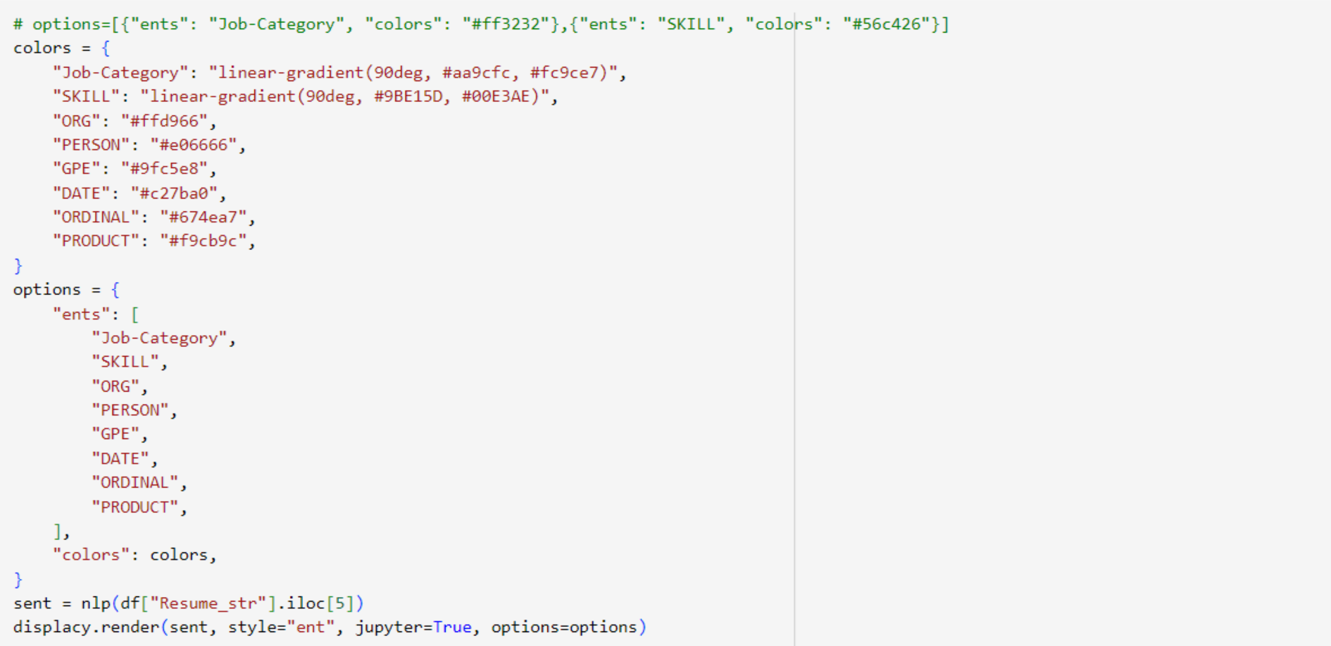
In our case, we have added a new entity called SKILL and is displayed in gray color. I was not impressed by colors and I also wanted to add another entity called Job Description so I started experimenting with various parameters within `displace.

- Adding Job-Category into entity ruler.
- Adding custom colors to all categories.
- Adding gradient colors to SKILL and Job-Category

You can see the result below as the new highlighted texts look beautiful.

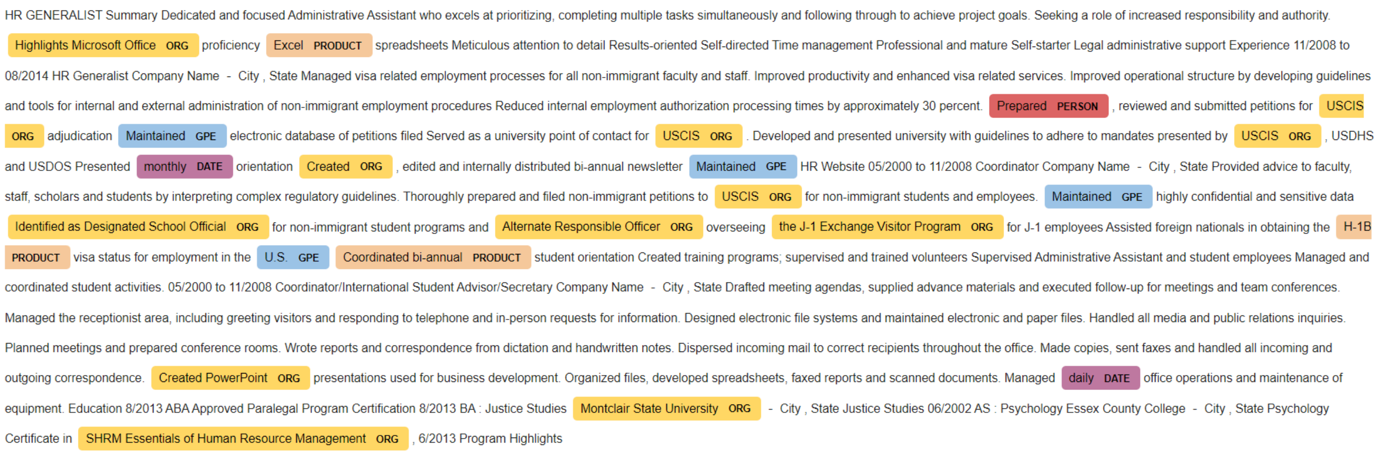


```
patterns = df.Category.unique()
for a in patterns:
    ruler.add_patterns([{"label": "Job-Category", "pattern": a}])
```

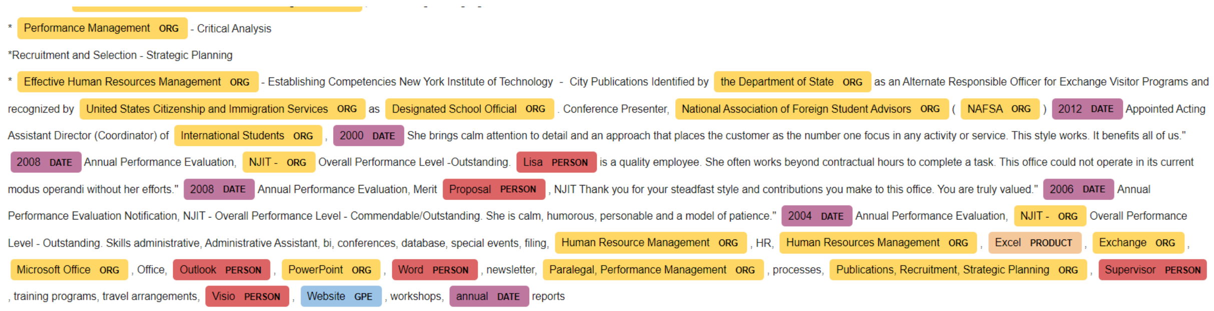


```
# options=[{"ents": "Job-Category", "colors": "#ff3232"},{"ents": "SKILL", "colors": "#56c426"}]
colors = {
    "Job-Category": "linear-gradient(90deg, #aa9cfc, #fc9ce7)",
    "SKILL": "linear-gradient(90deg, #9BE15D, #00E3AE)",
    "ORG": "#ffd966",
    "PERSON": "#e06666",
    "GPE": "#9fc5e8",
    "DATE": "#c27ba0",
    "ORDINAL": "#674ea7",
    "PRODUCT": "#f9cb9c",
}
options = {
    "ents": [
        "Job-Category",
        "SKILL",
        "ORG",
        "PERSON",
        "GPE",
        "DATE",
        "ORDINAL",
        "PRODUCT",
    ],
    "colors": colors,
}
sent = nlp(df["Resume_str"].iloc[5])
displacy.render(sent, style="ent", jupyter=True, options=options)
```

The provided code is designed to recognize and visualize specific entities in text data, such as job categories, skills, organizations, and more. It begins by extracting unique categories from a DataFrame column named "Category" using `patterns = df.Category.unique()`. For each unique pattern, it adds a new pattern to the ruler for entity recognition, labeling them as "Job-Category". An `options` dictionary is defined to specify which entity labels should be considered `(ents)` and their corresponding colors `(colors)`. The code then processes the 6th resume string (0-indexed, so index 5) in the DataFrame column "Resume_str" using the NLP pipeline with `sent = nlp(df["Resume_str"].iloc[5])`. Finally, it renders the processed text with entity annotations in a Jupyter notebook using `display.render(sent, style="ent", jupyter=True, options=options)`, applying the specified options for entity types and colors. This process enables the extraction and visualization of key information from resumes, making it easier to identify important details such as job categories, skills, and organizations.



HR GENERALIST Summary Dedicated and focused Administrative Assistant who excels at prioritizing, completing multiple tasks simultaneously and following through to achieve project goals. Seeking a role of increased responsibility and authority. Highlights Microsoft Office ORG proficiency Excel PRODUCT spreadsheets Meticulous attention to detail Results-oriented Self-directed Time management Professional and mature Self-starter Legal administrative support Experience 11/2008 to 08/2014 HR Generalist Company Name - City , State Managed visa related employment processes for all non-immigrant faculty and staff. Improved productivity and enhanced visa related services. Improved operational structure by developing guidelines and tools for internal and external administration of non-immigrant employment procedures Reduced internal employment authorization processing times by approximately 30 percent. Prepared PERSON , reviewed and submitted petitions for USCIS ORG adjudication Maintained GPE electronic database of petitions filed Served as a university point of contact for USCIS ORG . Developed and presented university with guidelines to adhere to mandates presented by USCIS ORG , USDHS and USDOS Presented monthly DATE orientation Created ORG , edited and internally distributed bi-annual newsletter Maintained GPE HR Website 05/2000 to 11/2008 Coordinator Company Name - City , State Provided advice to faculty, staff, scholars and students by interpreting complex regulatory guidelines. Thoroughly prepared and filed non-immigrant petitions to USCIS ORG for non-immigrant students and employees. Maintained GPE highly confidential and sensitive data Identified as Designated School Official ORG for non-immigrant student programs and Alternate Responsible Officer ORG overseeing the J-1 Exchange Visitor Program ORG for J-1 employees Assisted foreign nationals in obtaining the H-1B PRODUCT visa status for employment in the U.S. GPE Coordinated bi-annual PRODUCT student orientation Created training programs; supervised and trained volunteers Supervised Administrative Assistant and student employees Managed and coordinated student activities. 05/2000 to 11/2008 Coordinator/International Student Advisor/Secretary Company Name - City , State Drafted meeting agendas, supplied advance materials and executed follow-up for meetings and team conferences. Managed the receptionist area, including greeting visitors and responding to telephone and in-person requests for information. Designed electronic file systems and maintained electronic and paper files. Handled all media and public relations inquiries. Planned meetings and prepared conference rooms. Wrote reports and correspondence from dictation and handwritten notes. Dispersed incoming mail to correct recipients throughout the office. Made copies, sent faxes and handled all incoming and outgoing correspondence. Created PowerPoint ORG presentations used for business development. Organized files, developed spreadsheets, faxed reports and scanned documents. Managed daily DATE office operations and maintenance of equipment. Education 8/2013 ABA Approved Paralegal Program Certification 8/2013 BA : Justice Studies Montclair State University ORG - City , State Justice Studies 06/2002 AS : Psychology Essex County College - City , State Psychology Certificate in SHRM Essentials of Human Resource Management ORG , 6/2013 Program Highlights



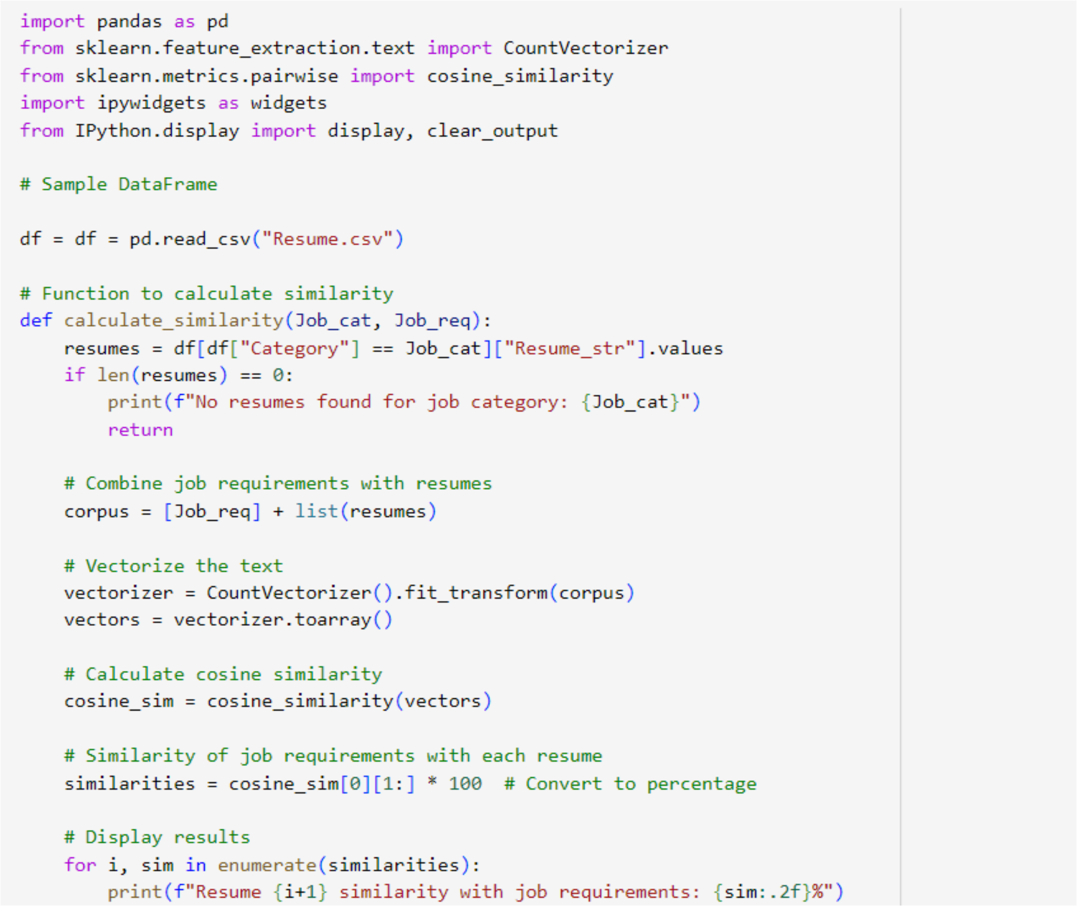
* Performance Management ORG - Critical Analysis

*Recruitment and Selection - Strategic Planning

* Effective Human Resources Management ORG - Establishing Competencies New York Institute of Technology - City Publications Identified by the Department of State ORG as an Alternate Responsible Officer for Exchange Visitor Programs and recognized by United States Citizenship and Immigration Services ORG as Designated School Official ORG . Conference Presenter, National Association of Foreign Student Advisors ORG (NAFSA ORG) 2012 DATE Appointed Acting Assistant Director (Coordinator) of International Students ORG , 2000 DATE She brings calm attention to detail and an approach that places the customer as the number one focus in any activity or service. This style works. It benefits all of us." 2008 DATE Annual Performance Evaluation, NJIT - ORG Overall Performance Level -Outstanding. Lisa PERSON is a quality employee. She often works beyond contractual hours to complete a task. This office could not operate in its current modus operandi without her efforts." 2008 DATE Annual Performance Evaluation, Merit Proposal PERSON , NJIT Thank you for your steadfast style and contributions you make to this office. You are truly valued." 2006 DATE Annual Performance Evaluation Notification, NJIT - Overall Performance Level - Commendable/Outstanding. She is calm, humorous, personable and a model of patience." 2004 DATE Annual Performance Evaluation, NJIT - ORG Overall Performance Level - Outstanding. Skills administrative, Administrative Assistant, bi, conferences, database, special events, filing, Human Resource Management ORG , HR, Human Resources Management ORG , Excel PRODUCT , Exchange ORG , Microsoft Office ORG , Office, Outlook PERSON , PowerPoint ORG , Word PERSON , newsletter, Paralegal, Performance Management ORG , processes, Publications, Recruitment, Strategic Planning ORG , Supervisor PERSON , training programs, travel arrangements, Visio PERSON , Website GPE , workshops, annual DATE reports

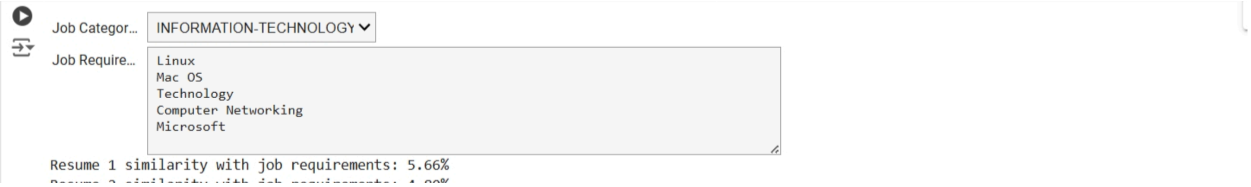
The provided image displays a processed resume text using Named Entity Recognition (NER) with various entities highlighted in different colors, each indicating a specific category. This visual representation is part of an NLP pipeline designed to facilitate the extraction of structured information from resumes. The categories include organizations (ORG), highlighted in yellow, such as "USCIS", "Company Name", and "Montclair State University". Products (PRODUCT) are highlighted in peach, including terms like "Excel" and "PowerPoint". Persons (PERSON) are highlighted in red, though there appears to be a misclassification with the term "Prepared" being incorrectly tagged as a person. Geopolitical entities (GPE) are highlighted in blue, exemplified by "City", "State", and "U.S.". Dates (DATE) are marked in purple, with examples like "monthly" and "daily". Additionally, while skills (SKILL) are intended to be highlighted in a green gradient, this specific category isn't explicitly visible in the provided image.

This annotated text benefits recruiters as well as it results in easy identification of the data points such as skills, experiences, organizations and dates which helps the recruiter in his line of work. Yet, there are some errors or, in other words, misclassifications, which show the weakness or inaccuracy of the chosen NER model. For NLP practitioners, it demonstrates how to adjust the way NER visualizations are done through the use of different colors for different kinds of entities; this would be handy when one is checking for problems with the model or even when explaining the results. In sum, the use of the visualization enhances the efficiency of grasping important resume information as a whole.



```python
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics.pairwise import cosine_similarity
import ipywidgets as widgets
from IPython.display import display, clear_output

# Sample DataFrame

df = df = pd.read_csv("Resume.csv")

# Function to calculate similarity
def calculate_similarity(Job_cat, Job_req):
    resumes = df[df["Category"] == Job_cat]["Resume_str"].values
    if len(resumes) == 0:
        print(f"No resumes found for job category: {Job_cat}")
        return

    # Combine job requirements with resumes
    corpus = [Job_req] + list(resumes)

    # Vectorize the text
    vectorizer = CountVectorizer().fit_transform(corpus)
    vectors = vectorizer.toarray()

    # Calculate cosine similarity
    cosine_sim = cosine_similarity(vectors)

    # Similarity of job requirements with each resume
    similarities = cosine_sim[0][1:] * 100  # Convert to percentage

    # Display results
    for i, sim in enumerate(similarities):
        print(f"Resume {i+1} similarity with job requirements: {sim:.2f}%")
```

The code given below works with the given DataFrame that consists of resume information; it selects out the unique job types present in the “Category” column in the DataFrame `df. Category`. unique(). They are then incorporated into the NLP ruler for entity recognition and identified as “Job-Category “each next unique category. Sixth resume string (indexed at 5) of the “Resume_str” column processing is done using the `nlp` function which evaluates the processed text and define particular entities like job category, skills, and organizations. Different color codings are assigned to any type of entities, for example, the type may refer to skills or organizations and each type will have a different color. Using a display. `render` function, the processed text with entity annotations is rendered and the output is shifted in a Jupyter notebook with emphasis on the recognized entities in terms of their colors.



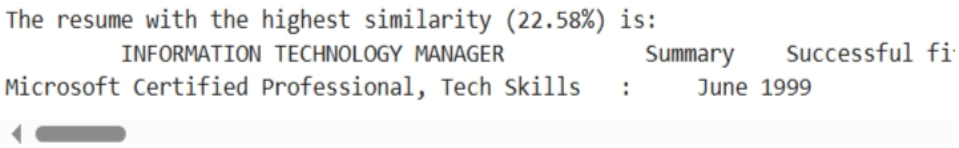
Job Categor... INFORMATION-TECHNOLOGY

Job Require...

```
Linux
Mac OS
Technology
Computer Networking
Microsoft
```

```
Resume 1 similarity with job requirements: 5.66%
```

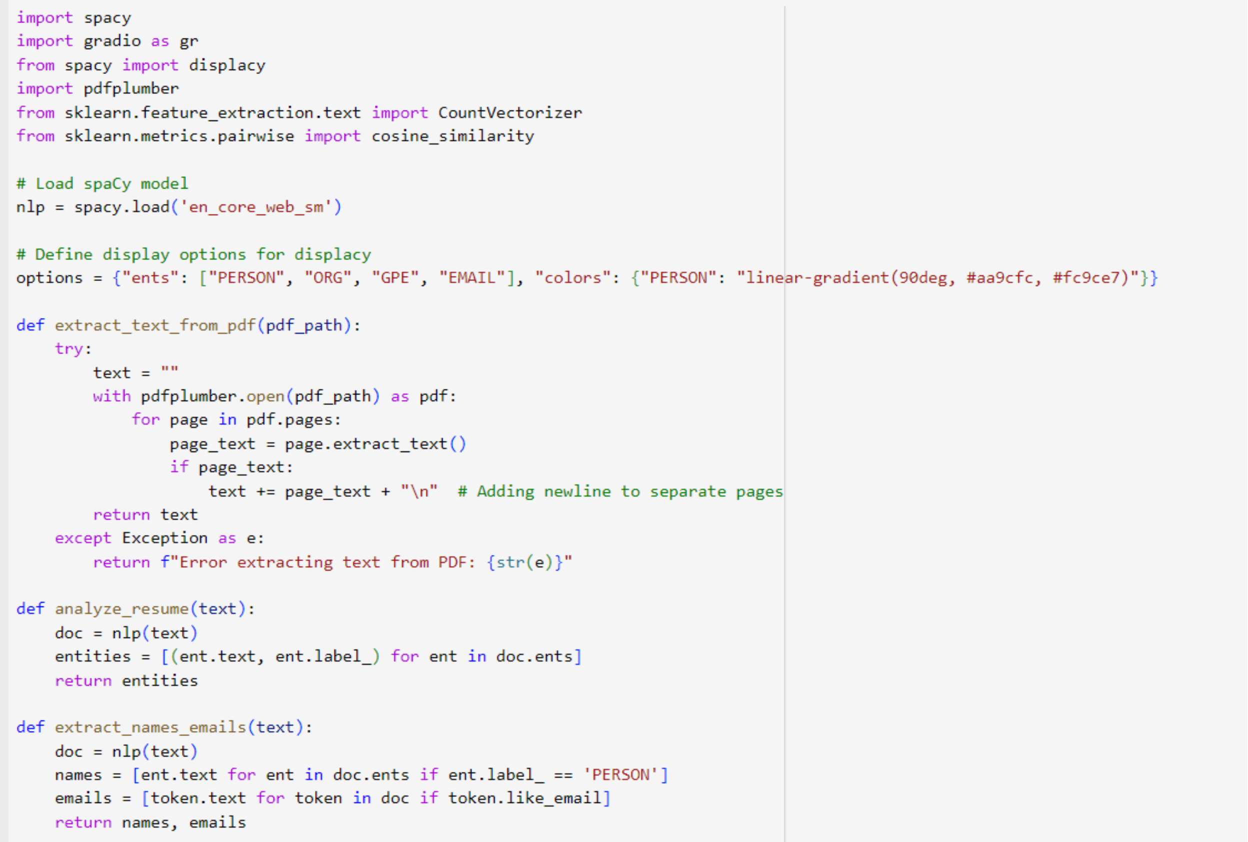
The program compared 26 resumes to the job listing and found resume number 15 to be the best match with a similarity score of 22.58%. However, it has limitations: it may not accurately assess all candidate skills, overlook important intangibles like soft skills and experience, and could be biased against candidates with non-traditional work experience or employment gaps. While helpful for screening, it should not be the sole criterion for hiring decisions.



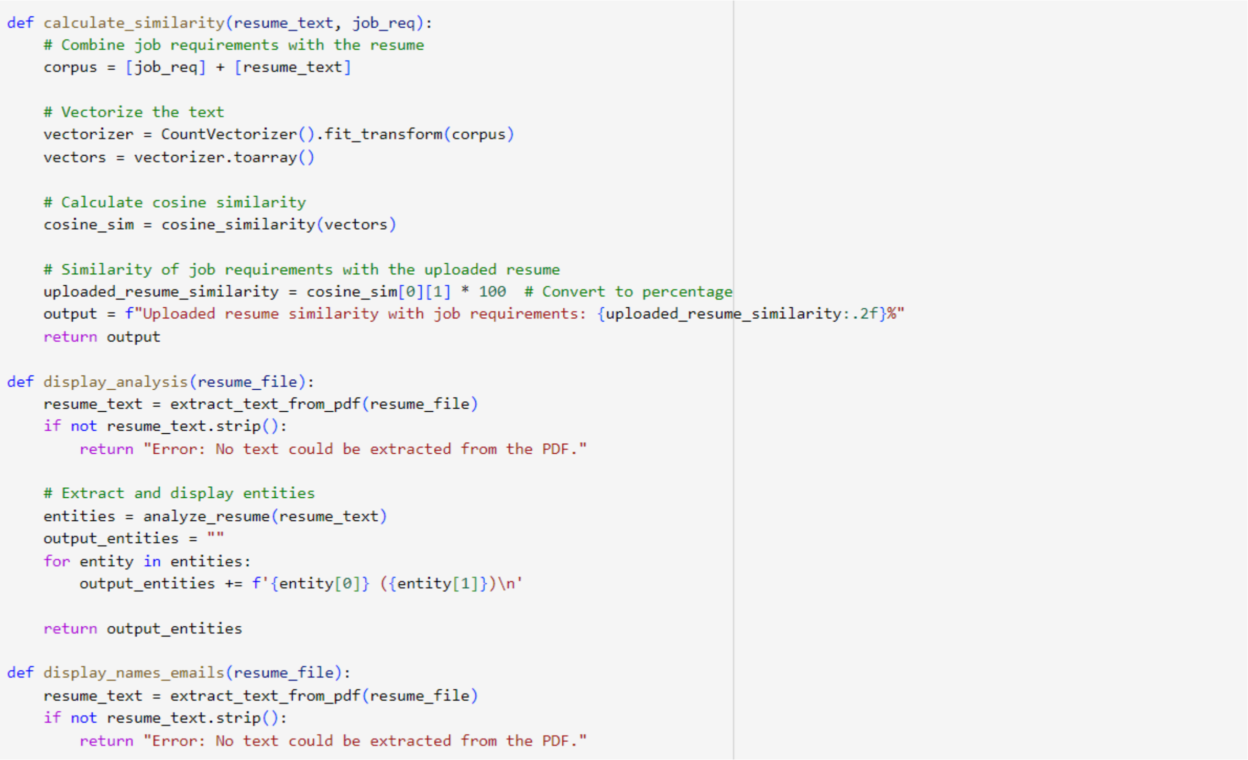
```
The resume with the highest similarity (22.58%) is:
        INFORMATION TECHNOLOGY MANAGER          Summary    Successful fi
Microsoft Certified Professional, Tech Skills   :     June 1999
```

This appears to show a program analyzing resumes for similarity to a job description. Resume 15 is the closest match according to the test with a value of 22. 58% score on resumes compared out of the 26 resume. However, these tools have limitations: they may not cover all activities required to assess a candidate fully, fail to incorporate factors such as soft skills, lose sight of some candidates. , though helpful for the purpose of the first filter still cannot be applied as the sole filter to select candidates.

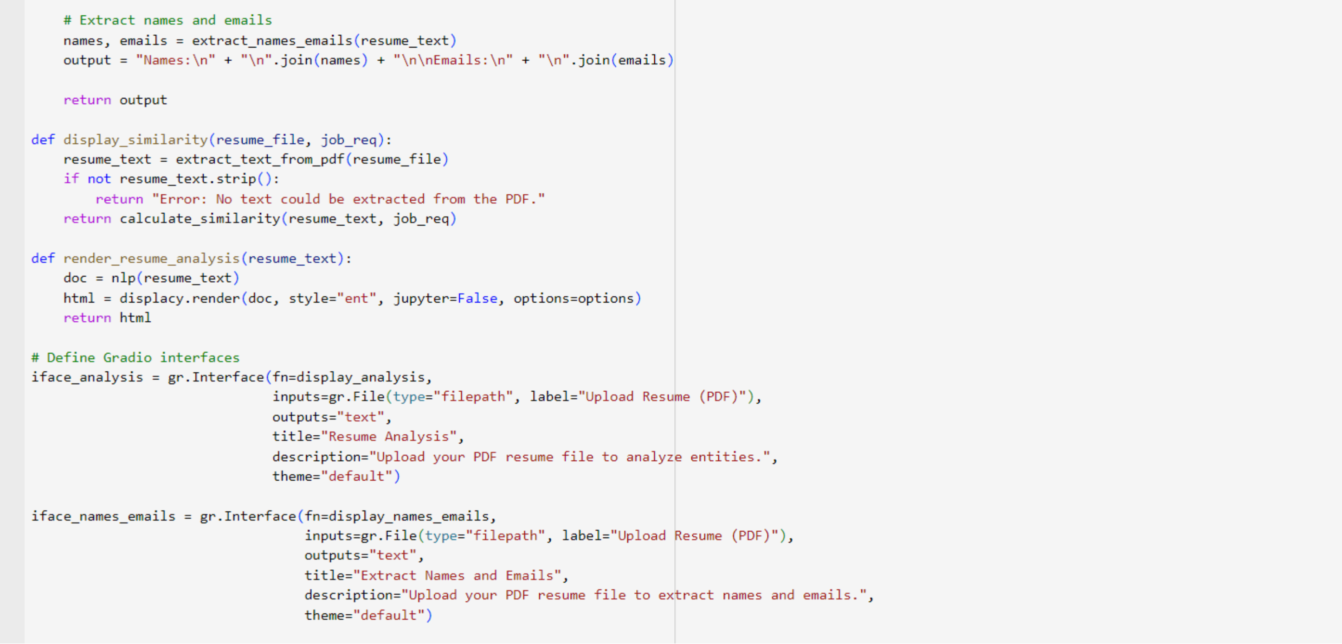
3.3 Coding of project with GUI



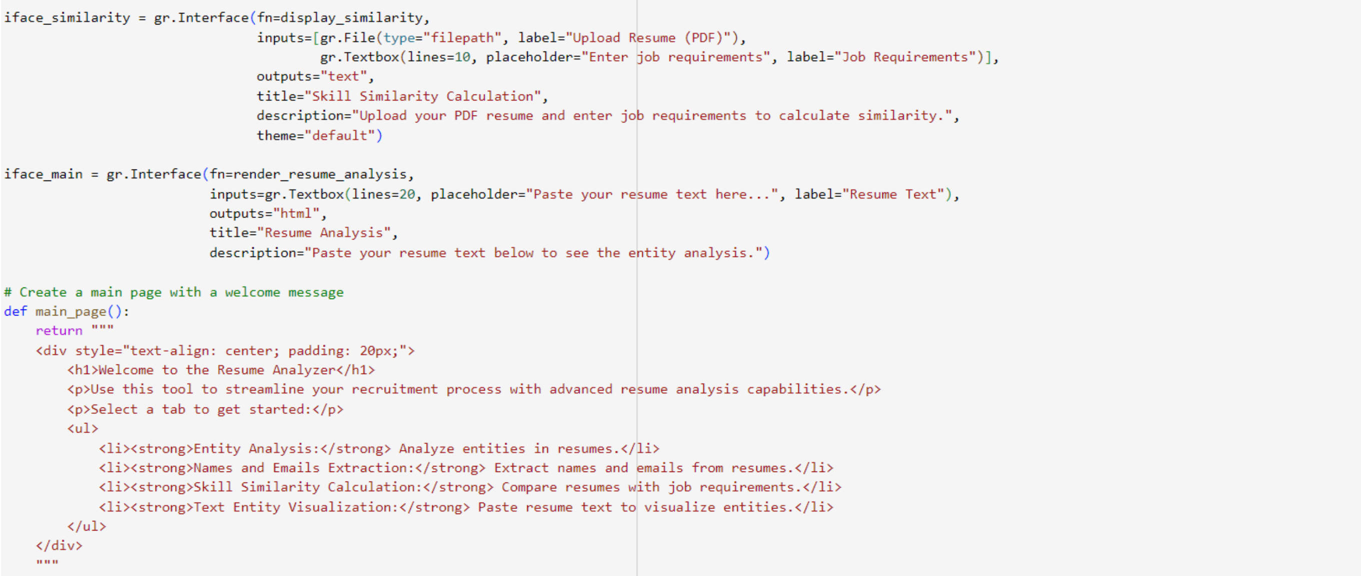
```python
import spacy
import gradio as gr
from spacy import displacy
import pdfplumber
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics.pairwise import cosine_similarity

# Load spaCy model
nlp = spacy.load('en_core_web_sm')

# Define display options for displacy
options = {"ents": ["PERSON", "ORG", "GPE", "EMAIL"], "colors": {"PERSON": "linear-gradient(90deg, #aa9cfc, #fc9ce7)"}}

def extract_text_from_pdf(pdf_path):
    try:
        text = ""
        with pdfplumber.open(pdf_path) as pdf:
            for page in pdf.pages:
                page_text = page.extract_text()
                if page_text:
                    text += page_text + "\n"  # Adding newline to separate pages
        return text
    except Exception as e:
        return f"Error extracting text from PDF: {str(e)}"

def analyze_resume(text):
    doc = nlp(text)
    entities = [(ent.text, ent.label_) for ent in doc.ents]
    return entities

def extract_names_emails(text):
    doc = nlp(text)
    names = [ent.text for ent in doc.ents if ent.label_ == 'PERSON']
    emails = [token.text for token in doc if token.like_email]
    return names, emails
```



```python
def calculate_similarity(resume_text, job_req):
    # Combine job requirements with the resume
    corpus = [job_req] + [resume_text]

    # Vectorize the text
    vectorizer = CountVectorizer().fit_transform(corpus)
    vectors = vectorizer.toarray()

    # Calculate cosine similarity
    cosine_sim = cosine_similarity(vectors)

    # Similarity of job requirements with the uploaded resume
    uploaded_resume_similarity = cosine_sim[0][1] * 100  # Convert to percentage
    output = f"Uploaded resume similarity with job requirements: {uploaded_resume_similarity:.2f}%"
    return output

def display_analysis(resume_file):
    resume_text = extract_text_from_pdf(resume_file)
    if not resume_text.strip():
        return "Error: No text could be extracted from the PDF."

    # Extract and display entities
    entities = analyze_resume(resume_text)
    output_entities = ""
    for entity in entities:
        output_entities += f'{entity[0]} ({entity[1]})\n'

    return output_entities

def display_names_emails(resume_file):
    resume_text = extract_text_from_pdf(resume_file)
    if not resume_text.strip():
        return "Error: No text could be extracted from the PDF."
```



```python
    # Extract names and emails
    names, emails = extract_names_emails(resume_text)
    output = "Names:\n" + "\n".join(names) + "\n\nEmails:\n" + "\n".join(emails)

    return output

def display_similarity(resume_file, job_req):
    resume_text = extract_text_from_pdf(resume_file)
    if not resume_text.strip():
        return "Error: No text could be extracted from the PDF."
    return calculate_similarity(resume_text, job_req)

def render_resume_analysis(resume_text):
    doc = nlp(resume_text)
    html = displacy.render(doc, style="ent", jupyter=False, options=options)
    return html

# Define Gradio interfaces
iface_analysis = gr.Interface(fn=display_analysis,
                              inputs=gr.File(type="filepath", label="Upload Resume (PDF)"),
                              outputs="text",
                              title="Resume Analysis",
                              description="Upload your PDF resume file to analyze entities.",
                              theme="default")

iface_names_emails = gr.Interface(fn=display_names_emails,
                                  inputs=gr.File(type="filepath", label="Upload Resume (PDF)"),
                                  outputs="text",
                                  title="Extract Names and Emails",
                                  description="Upload your PDF resume file to extract names and emails.",
                                  theme="default")
```

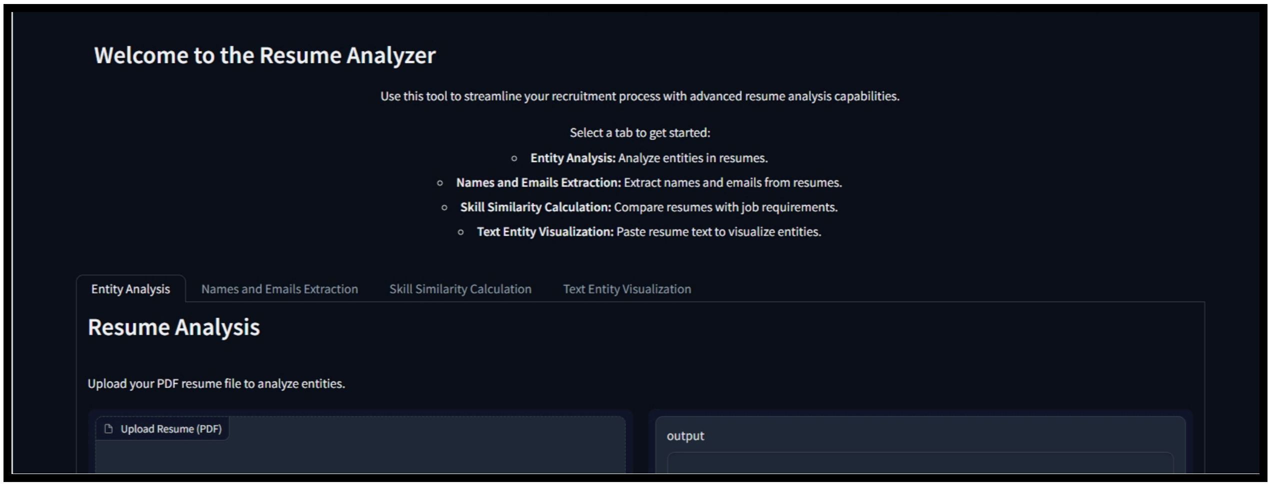


```python
iface_similarity = gr.Interface(fn=display_similarity,
                                inputs=[gr.File(type="filepath", label="Upload Resume (PDF)"),
                                        gr.Textbox(lines=10, placeholder="Enter job requirements", label="Job Requirements")],
                                outputs="text",
                                title="Skill Similarity Calculation",
                                description="Upload your PDF resume and enter job requirements to calculate similarity.",
                                theme="default")

iface_main = gr.Interface(fn=render_resume_analysis,
                          inputs=gr.Textbox(lines=20, placeholder="Paste your resume text here...", label="Resume Text"),
                          outputs="html",
                          title="Resume Analysis",
                          description="Paste your resume text below to see the entity analysis.")

# Create a main page with a welcome message
def main_page():
    return """
    <div style="text-align: center; padding: 20px;">
        <h1>Welcome to the Resume Analyzer</h1>
        <p>Use this tool to streamline your recruitment process with advanced resume analysis capabilities.</p>
        <p>Select a tab to get started:</p>
        <ul>
            <li><strong>Entity Analysis:</strong> Analyze entities in resumes.</li>
            <li><strong>Names and Emails Extraction:</strong> Extract names and emails from resumes.</li>
            <li><strong>Skill Similarity Calculation:</strong> Compare resumes with job requirements.</li>
            <li><strong>Text Entity Visualization:</strong> Paste resume text to visualize entities.</li>
        </ul>
    </div>
    """
```

```
# Combine the main page with interfaces
def create_main_interface():
    with gr.Blocks() as main_interface:
        gr.Markdown(main_page())
        with gr.Tab("Entity Analysis"):
            iface_analysis.render()
        with gr.Tab("Names and Emails Extraction"):
            iface_names_emails.render()
        with gr.Tab("Skill Similarity Calculation"):
            iface_similarity.render()
        with gr.Tab("Text Entity Visualization"):
            iface_main.render()
    return main_interface

# Create and launch the main interface
main_interface = create_main_interface()
main_interface.launch()
```

4.0 Result & Discussion

Main Page



Welcome to the Resume Analyzer

Use this tool to streamline your recruitment process with advanced resume analysis capabilities.

Select a tab to get started:

- **Entity Analysis:** Analyze entities in resumes.
- **Names and Emails Extraction:** Extract names and emails from resumes.
- **Skill Similarity Calculation:** Compare resumes with job requirements.
- **Text Entity Visualization:** Paste resume text to visualize entities.

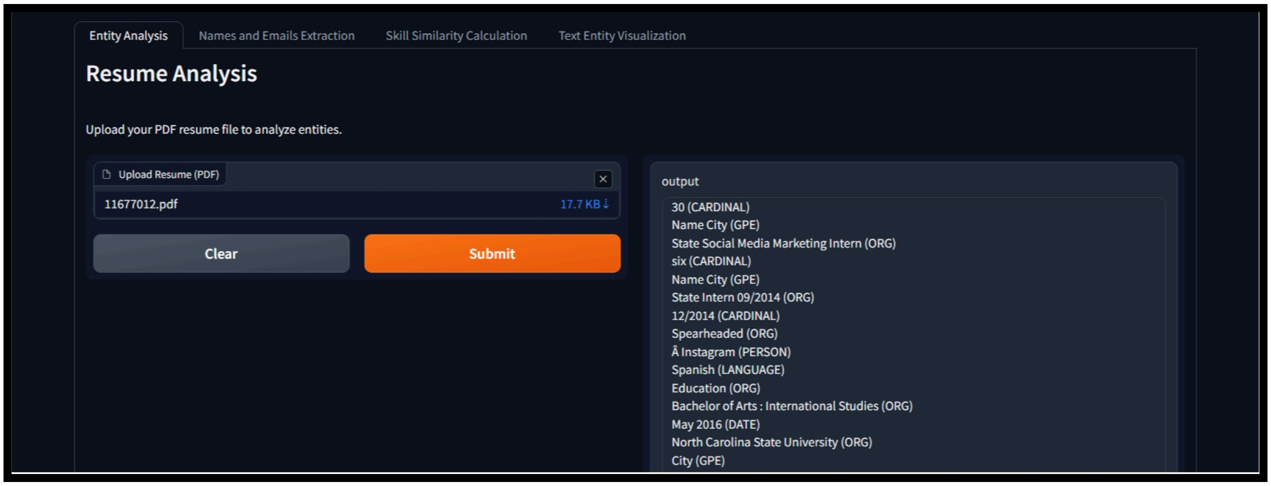
Resume Analysis

Upload your PDF resume file to analyze entities.

Entity Analysis
Names and Emails Extraction
Skill Similarity Calculation
Text Entity Visualization
Upload Resume (PDF)
output

Figure 1.0 Main Page

This main page contains 4 tab tools which are Entity Analysis, Names and Emails Extraction, Skills Similarity Calculation and Text Entity Visualization. On this page, users can click on any tab to do resume analysis. Refer to user needs.



Entity Analysis
Names and Emails Extraction
Skill Similarity Calculation
Text Entity Visualization
Upload Resume (PDF)
Clear
Submit
output

Resume Analysis

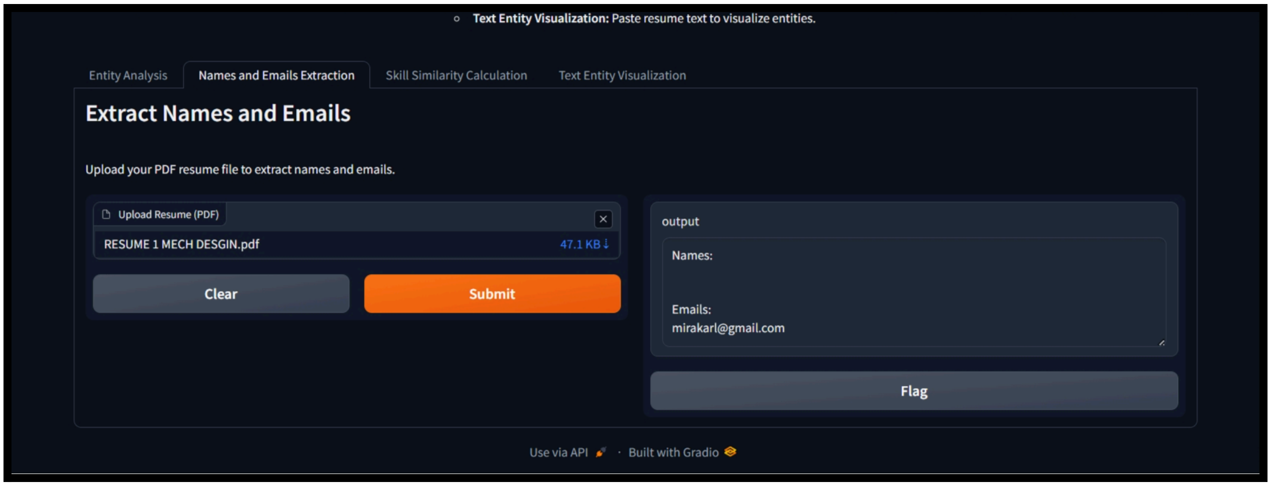
Upload your PDF resume file to analyze entities.

11677012.pdf 17.7 KB

```
30 (CARDINAL)
Name City (GPE)
State Social Media Marketing Intern (ORG)
six (CARDINAL)
Name City (GPE)
State Intern 09/2014 (ORG)
12/2014 (CARDINAL)
Spearheaded (ORG)
Â Instagram (PERSON)
Spanish (LANGUAGE)
Education (ORG)
Bachelor of Arts : International Studies (ORG)
May 2016 (DATE)
North Carolina State University (ORG)
City (GPE)
```

Figure 2.0 Entity Analysis Tab

On this tab, user can upload their resume pdf to identify the identity into a few option such as Organization (ORG), Geopolitical Entity like state (GPE), Date (DATE).



○ **Text Entity Visualization:** Paste resume text to visualize entities.

Extract Names and Emails

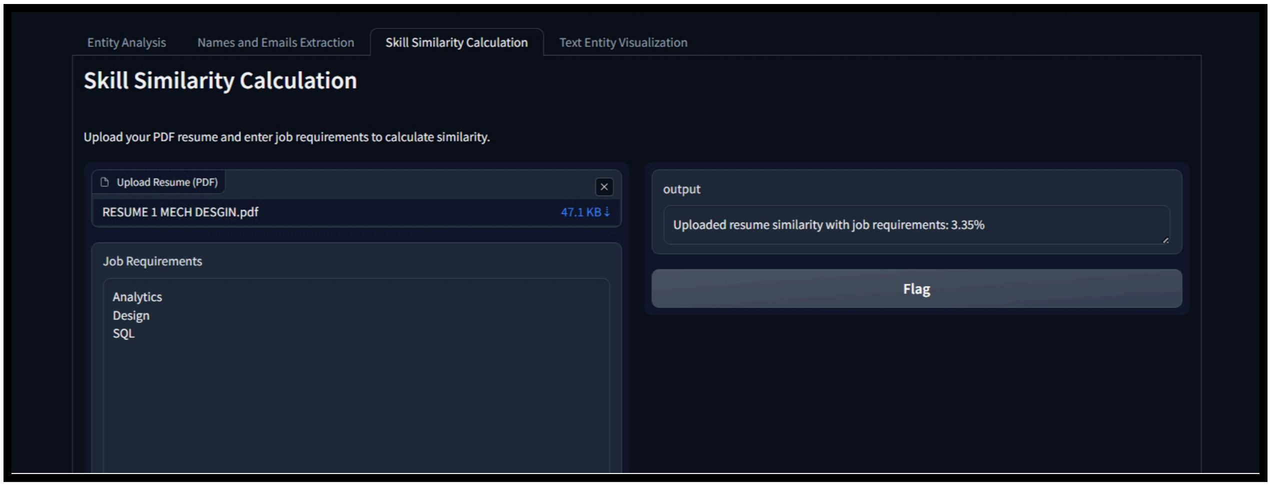
Upload your PDF resume file to extract names and emails.

RESUME 1 MECH DESGIN.pdf 47.1 KB

```
Names:

Emails:
mirakarl@gmail.com
```

Figure 3.0 Extract names and Emails Tab

On this tab, users can upload their resume for the system to extract their name and any email stated. This is because extracting their names and other contact information like emails make the reviewing process easier especially for the recruiter.


Entity Analysis
Names and Emails Extraction
Skill Similarity Calculation
Text Entity Visualization
Flag

Skill Similarity Calculation

Upload your PDF resume and enter job requirements to calculate similarity.

Upload Resume (PDF)

RESUME 1 MECH DESGIN.pdf 47.1 KB

Job Requirements

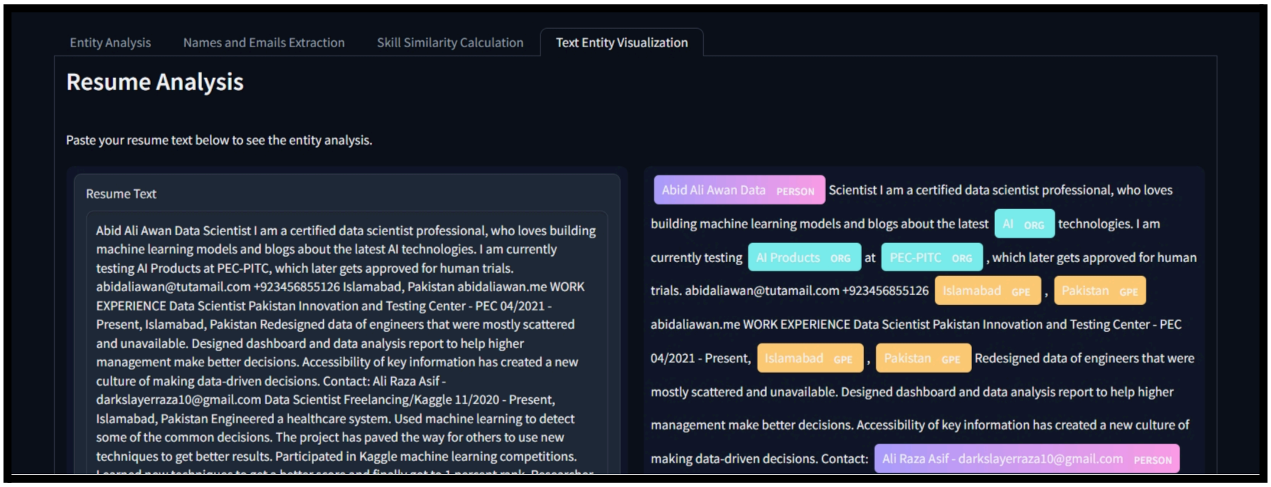
Analytics
Design
SQL

output

Uploaded resume similarity with job requirements: 3.35%

Figure 4.0 Skills Similarity Calculation Tab

On this tab users can upload the Resume and lose the job requirement the company needed. The next column will shows the output where it shown the skill similarity percentage.


Entity Analysis
Names and Emails Extraction
Skill Similarity Calculation
Text Entity Visualization

Resume Analysis

Paste your resume text below to see the entity analysis.

Resume Text

Abid Ali Awan Data Scientist I am a certified data scientist professional, who loves building machine learning models and blogs about the latest AI technologies. I am currently testing AI Products at PEC-PITC, which later gets approved for human trials. abidaliawan@tutamail.com +923456855126 Islamabad, Pakistan abidaliawan.me WORK EXPERIENCE Data Scientist Pakistan Innovation and Testing Center - PEC 04/2021 - Present, Islamabad, Pakistan Redesigned data of engineers that were mostly scattered and unavailable. Designed dashboard and data analysis report to help higher management make better decisions. Accessibility of key information has created a new culture of making data-driven decisions. Contact: Ali Raza Asif - darkslayerraza10@gmail.com Data Scientist Freelancing/Kaggle 11/2020 - Present, Islamabad, Pakistan Engineered a healthcare system. Used machine learning to detect some of the common decisions. The project has paved the way for others to use new techniques to get better results. Participated in Kaggle machine learning competitions.

Abid Ali Awan Data PERSON Scientist I am a certified data scientist professional, who loves building machine learning models and blogs about the latest AI ORG technologies. I am currently testing AI Products ORG at PEC-PITC ORG , which later gets approved for human trials. abidaliawan@tutamail.com +923456855126 Islamabad GPE , Pakistan GPE abidaliawan.me WORK EXPERIENCE Data Scientist Pakistan Innovation and Testing Center - PEC 04/2021 - Present, Islamabad GPE , Pakistan GPE Redesigned data of engineers that were mostly scattered and unavailable. Designed dashboard and data analysis report to help higher management make better decisions. Accessibility of key information has created a new culture of making data-driven decisions. Contact: Ali Raza Asif - darkslayerraza10@gmail.com PERSON

Figure 5.0 Skills Similarity Calculation Tab

On this tab, entity analysis can be input by text and the output will identify the entity in colours. Purple shade identify PERSON. Orange shade identify GPE. Blue shade identify ORG.

5.0 Conclusion

The Resume Analyzer project exemplifies the practical application of Natural Language Processing (NLP) in automating the extraction and analysis of crucial information from resumes. This automation brings about substantial advantages in terms of hiring efficiency, as it reduces the time and effort required for initial resume reviews. Consequently, hiring managers and recruiters can allocate their focus towards more strategic tasks. Additionally, this tool promotes consistency and objectivity in resume evaluations, thereby minimising the occurrence of human error and bias. Its scalability renders it suitable for organisations of all sizes, particularly those that handle large volumes of applications.

Potential future enhancements for this project could involve the utilisation of advanced NLP models or machine learning techniques to improve skill matching. Furthermore, there is the possibility of extracting additional relevant information, such as work experience and education history. Moreover, greater user customisation could be implemented to define specific entities or fields based on job requirements or industry standards. Additionally, more accurate entity classification could further enhance the precision of the resume analyzer.

In conclusion, the Resume Analyzer project serves as a testament to how NLP can significantly enhance the resume screening process by providing accurate and efficient extraction of personal details and skill evaluations. This tool offers time savings, consistency, and scalability, making it an invaluable asset for modern hiring practices. As the hiring landscape continues to evolve, tools like the Resume Analyzer will play a crucial role in ensuring effective and fair candidate evaluations, ultimately leading to better hiring decisions and more efficient HR operations.

6.0 References

Awan, A. A. (n.d.). *spaCy Resume Analysis*. Deepnote. Retrieved June 9, 2024, from https://deepnote.com/app/abid/spaCy-Resume-Analysis-81ba1e4b-7fa8-45fe-ac7a-0b7bf3da7826

Lyons, R. (n.d.). *How to List Problem-Solving Skills on a Resume*. TopResume. https://www.topresume.com/career-advice/how-to-list-problem-solving-skills-on-a-resume

Resume Checker: Get Instant Resume Score With Our Grader. (n.d.). Kickresume. Retrieved June 9, 2024, from https://www.kickresume.com/en/resume-checker/

7.0 Appendix

PROJECT_AI.ipynb