



RESUME SCREENING ASSISTANT

Name: SWATHI UMADI

Registration no.: 22BCE8841

College name: VIT -AP UNIVERSITY

Branch: CSE (Core)

Date of Submission: 8th July, 2025

● **Table of Contents**

1.Introduction

2.Objective

3.Tools & Technologies Used

4.Methodology / Working

5.Code Snippets with Explanation

6.Screenshots / Output Results with proper explanation

7.Links for the project

8.Challenges Faced & Solutions

9. Conclusion

10. References

● INTRODUCTION

This project focuses on automating the classification of resumes using Generative AI. In recruitment processes, screening thousands of resumes manually is time-consuming and inefficient. By leveraging a zero-shot classification approach using a pre-trained Large Language Model (LLM) from Hugging Face, we classify resumes into relevant job roles without the need for traditional machine learning training.

● OBJECTIVE

To build a GenAI-powered system that automatically classifies resumes into categories such as Data Science, HR, Software Engineering, etc., using zero-shot inference. The aim is to eliminate the need for model training and achieve intelligent classification through natural language understanding.

● TOOLS & TECHNOLOGIES USED

Platform: Google Colab

Programming Language: Python

Libraries:

- `transformers` (Hugging Face)
- `pandas`
- `re` (Regular expressions)
- `tqdm`

Model Used: facebook/bart-large-mnli (Zero-shot classification)

Data Source: Resume dataset in CSV format

● METHODOLOGY /WORKING

The project follows these steps:

1. **Data Upload:** Upload a CSV file containing resumes.
2. **Data Cleaning:** Clean the resume text by removing punctuation, lowercasing, and trimming whitespace.
3. **Zero-shot Model Setup:** Load the pre-trained BART LLM from Hugging Face's model hub.
4. **Category Definition:** Provide the model with predefined job role labels.
5. **Classification:** For each resume, ask the model to predict the most relevant job role using zero-shot classification.
6. **Output:** Store and download the predicted results in a new CSV file.

● CODE SNIPPETS WITH EXPLANATION

Step 1: Upload the Resume File

```
from google.colab import files  
  
uploaded = files.upload()
```

This code opens a file chooser in Colab so we can upload the csv file directly from the device.

Step 2: Install Transformers Library

```
!pip install transformers
```

Installs the transformers package from Hugging Face, which provides access to pretrained LLMs like BART.

Step 3: Import Required Libraries

```
import pandas as pd

import re

from transformers import pipeline

from tqdm import tqdm
```

- Pandas is used for data handling
- re for text cleaning using regex
- Pipeline from transformers loads the GenAI model
- tqdm shows progress bar while looping through resumes

Step 4: Load the Resume Dataset

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

df.columns = df.columns.str.strip()
```

Loads the uploaded resume file into a DataFrame and removes unwanted whitespace in column names

Step 5: Clean the Resume Text

```
def clean_text(text):

    if pd.isnull(text):

        return ""

    text = str(text).lower()

    text = re.sub(r'\s+', ' ', text)
```

```
text = re.sub(r'^\w\s|', "", text)

return text.strip()

df['CleanResume'] = df['Resume_str'].apply(clean_text)
```

Cleans the resume content by:

- Converting to lowercase
 - Removing extra spaces
 - Removing punctuation
- Creates a new column called CleanResume with cleaned text.

Step 6: Load the Zero-Shot Classification Model

```
classifier = pipeline("zero-shot-classification", model="facebook/bart-large-mnli")
```

Initializes a **zero-shot classifier** using Hugging Face's [facebook/bart-large-mnli](#) model. This model can classify text into categories **without training**.

Step 7: Define Job Role Categories

```
candidate_labels = ["Data Science", "HR", "DevOps", "Testing", "Design",  
"Software Engineer", "Business Analyst"]
```

These are the job categories you want the model to choose from when classifying resumes. You can customize this list based on your project.

Step 8: Predict the Resume Category using GenAI

```
predicted_labels = []

for resume in tqdm(df['CleanResume'].head(100)): # Use full data if system is fast
```

```

    if not resume.strip():

predicted_labels.append("Unknown")

continue

result = classifier(resume, candidate_labels)

predicted_labels.append(result['labels'][0])

```

For each cleaned resume:

- It asks the GenAI model to choose the best-fitting label from **candidate_labels**.
- The top predicted label is saved to **predicted_labels** list.
Uses tqdm to show progress in Colab.

Step 9: Save Predictions to a New File

```

df['PredictedCategory_GenAI'] = predicted_labels

df.to_csv("GenAI_Classified_Resumes.csv", index=False)

from google.colab import files

files.download("GenAI_Classified_Resumes.csv")

```

- Adds a new column to the DataFrame with the predicted category
- Saves the updated DataFrame to a new CSV
- Triggers a download so you can save the results to your system

• SCREENSHOTS/OUTPUT

```
[ ] from google.colab import files
```

```
# Upload Resume.csv from your computer
uploaded = files.upload()
```



Choose Files Resume.csv

- Resume.csv(text/csv) - 56273235 bytes, last modified: 7/8/2025 - 100% done
- Saving Resume.csv to Resume.csv

```
[ ] !pip install transformers
```



```
Requirement already satisfied: transformers in /usr/local/lib/python3.11/dist-packages (4.53.0)
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-packages (from transformers) (3.18.0)
Requirement already satisfied: huggingface-hub<1.0,>=0.30.0 in /usr/local/lib/python3.11/dist-packages (from transformers) (0.33.1)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.11/dist-packages (from transformers) (2.0.2)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-packages (from transformers) (24.2)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.11/dist-packages (from transformers) (6.0.2)
Requirement already satisfied: regex<=2019.12.17 in /usr/local/lib/python3.11/dist-packages (from transformers) (2024.11.6)
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from transformers) (2.32.3)
Requirement already satisfied: tokenizers<0.22,>=0.21 in /usr/local/lib/python3.11/dist-packages (from transformers) (0.21.2)
Requirement already satisfied: safetensors<=0.4.3 in /usr/local/lib/python3.11/dist-packages (from transformers) (0.5.3)
Requirement already satisfied: tqdm<=4.27 in /usr/local/lib/python3.11/dist-packages (from transformers) (4.67.1)
Requirement already satisfied: fsspec<=2023.5.0 in /usr/local/lib/python3.11/dist-packages (from huggingface-hub<1.0,>=0.30.0->transformers) (2023.3.2)
Requirement already satisfied: typing-extensions<=3.7.4.3 in /usr/local/lib/python3.11/dist-packages (from huggingface-hub<1.0,>=0.30.0->transformers) (4.14.0)
Requirement already satisfied: hf-xet<2.0.0,>=1.1.2 in /usr/local/lib/python3.11/dist-packages (from huggingface-hub<1.0,>=0.30.0->transformers) (1.1.5)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests->transformers) (3.4.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests->transformers) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests->transformers) (2.4.0)
Requirement already satisfied: certifi<2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests->transformers) (2025.6.15)
```



```
classifier = pipeline("zero-shot-classification", model="facebook/bart-large-mnli")
```



```
/usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), s
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(

config.json: 1.15k/? [00:00<00:00, 42.4kB/s]

model.safetensors: 100% [00:41<00:00, 1.63G/1.63G [00:41<00:00, 165MB/s]

tokenizer_config.json: 100% [00:00<00:00, 26.0/26.0 [00:00<00:00, 2.12kB/s]

vocab.json: 899k/? [00:00<00:00, 10.4MB/s]

merges.txt: 456k/? [00:00<00:00, 15.7MB/s]

tokenizer.json: 1.36M/? [00:00<00:00, 26.3MB/s]

Device set to use cpu
```



```
[8] df = df.head(100)

predicted_labels = []

for resume in tqdm(df['CleanResume']):
    if not resume.strip():
        predicted_labels.append("Unknown")
        continue
    result = classifier(resume, candidate_labels)
    predicted_labels.append(result['labels'][0]) # top predicted label
```

100%|██████████| 100/100 [2:23:32<00:00, 86.13s/it]

```
df['PredictedCategory_GenAI'] = predicted_labels

df.to_csv("GenAI_Classified_Resumes.csv", index=False)

from google.colab import files
files.download("GenAI_Classified_Resumes.csv")
```



	A	B	C	D	E	F	G	H	
1	ID	Resume_s	Resume_h	Category	CleanResu	PredictedCategory_GenAI			
2	16852973	HR	<div	HR	hr adminis	HR			
3	22323967	HR	<div	HR	hr specialis	HR			
4	33176873	HR	<div	HR	hr director	HR			
5	27018550	HR	<div	HR	hr specialis	HR			
6	17812897	HR	<div	HR	hr manage	Testing			
7	11592605	HR	<div	HR	hr generali	HR			
8	25824789	HR	<div	HR	hr manage	HR			
9	15375009	HR	<div	HR	hr manage	HR			
10	11847784	HR SP	<div class=	HR	hr specialis	Design			
11	32896934	HR	<div	HR	hr clerk su	HR			
12	29149998	HR AS	<div class=	HR	hr assistan	HR			
13	11480899	HR M	<div class=	HR	hr manage	HR			
14	23155093	HR	<div	HR	hr manage	HR			
15	11763983	HR GE	<div class=	HR	hr generali	HR			
16	27490876	HR CC	<div class=	HR	hr coordin	HR			
17	32977530	HR CL	<div class=	HR	hr clerk su	Testing			
18	93002334	HR AN	<div class=	HR	hr analyst	HR			
19	24184357	HR	<div	HR	hr director	DevOps			
20	73077810	HR GE	<div	HR	hr generali	HR			
21	13879043	HR	<div	HR	hr consulti	Testing			
22	30163002	HR GE	<div	HR	hr generali	HR			
23	18827609	HR	<div	HR	hr associat	HR			
24	25676643	HR	<div	HR	hr specialis	HR			
25	12786012	HR CC	<div class=	HR	hr coordin	HR			
26	87968870	HR	<div	HR	hr generali	HR			
27	46258701	HR CC	<div class=	HR	hr coordin	HR			
28	14225422	HR M	<div	HR	hr manage	HR			
29	29297973	HR	<div	HR	hr represe	HR			
30	18334783	HR DI	<div class=	HR	hr director	HR			

	A	B	C	D	E	F	G	H
61	27523575	HR	<div	HR	hr associat	HR		
62	39081840	HR	<div	HR	hr coordin	Testing		
63	20925036		<div	HR	summary c	Testing		
64	15576950	HR SP	<div class=	HR	hr specialis	HR		
65	29564653	HR AL	<div	HR	hr adminis	HR		
66	14886205	HR	<div	HR	hr manage	HR		
67	17412079	HR	<div	HR	hr consulti	HR		
68	16861758	HR	<div	HR	hr assistan	HR		
69	93112113	HR M	<div	HR	hr manage	Design		
70	15041689	HR	<div	HR	hr director	HR		
71	80162314	HR PF	<div	HR	hr professi	HR		
72	20993320	HR	<div	HR	hr coordin	HR		
73	14640322	HR GE	<div class=	HR	hr generali	HR		
74	26289308		<div	HR	senior hr n	HR		
75	23408537	HR AS	<div class=	HR	hr assistan	HR		
76	28808263	HR PA	<div class=	HR	hr payroll a	HR		
77	18084150	SR. H	<div class=	HR	sr hr gener	HR		
78	26671167		<div	HR	directv hr	HR		
79	44476983	HR BU	<div class=	HR	hr business	HR		
80	16877897	ASSIS	<div class=	HR	assistant n	HR		
81	29091445	HR AS	<div class=	HR	hr associat	HR		
82	25724495		<div	HR	regional hr	HR		
83	24508725		<div	HR	field hr ass	HR		
84	11698189	HR	<div	HR	hr employe	HR		
85	27165830		<div	HR	territory h	HR		
86	47470864	GENE	<div class=	HR	general hr	HR		
87	34740556	SENI	<div class=	HR	senior hr b	HR		
88	18557164	HR BU	<div class=	HR	hr business	HR		
89	19179079		<div	HR	recruiting a	HR		
90	13376919		<div	HR	regional hr	HR		

< >

GenAI_Classified_Resumes

+

● LINKS FOR THE PROJECT

Google colab link:

 resume_screen_genai.ipynb

● CHALLENGES FACED & SOLUTIONS

1.Challenge: Low accuracy when using small data samples or ML models

Solution: Switched to full dataset + later replaced with GenAI model

2.Challenge: OpenAI GPT-3.5 API required payment/quota

Solution: Used Hugging Face's free BART-based zero-shot classifier

3.Challenge: Long resume text took time to process

Solution: Limited classification to 100 rows during testing; can scale up later

● CONCLUSION

The project successfully demonstrates how **Generative AI**, specifically a pre-trained **zero-shot classification model (facebook/bart-large-mnli)** from Hugging Face, can be used to automate the classification of resumes into predefined job categories. Unlike traditional machine learning approaches, this model required **no manual training** and was capable of understanding and assigning labels purely based on semantic content in the resume.

The implementation:

Eliminated the need for labeled training data
Handled textual variations intelligently
Achieved classification with good accuracy and interpretability
This approach proves highly scalable, customizable, and well-suited for
real-world HR and resume screening applications in the **era of Generative AI**.

● REFERENCES

- 1.Hugging Face Transformers Library
<https://huggingface.co/transformers>
- 2.BART Model for Zero-Shot Classification:
<https://huggingface.co/facebook/bart-large-mnli>
- 3.Google Colab (Cloud IDE used):
<https://colab.research.google.com>
- 4.Resume Dataset (Kaggle-based):
Resume dataset from Kaggle
- 5.Pandas Library Documentation:
<https://pandas.pydata.org/docs/>
- 6.Scikit-learn for vectorization (ML version):
<https://scikit-learn.org/stable/>

