Name: Kartik Palcharla

SRN: PES2UG23CS259

SEC: D

Problem Statement

Category 3: Analysis & Extraction (The "Smart" Agents)

10. Smart Resume Parser

- **Goal**: Upload a resume text and automatically extract specific fields: Name, University, Company.
- **Tech**: pipeline('ner') to find PER (Person) and ORG (Organization) entities.

Abstract

This project implements an automated resume parsing tool using the Hugging Face transformers library. The objective is to convert unstructured resume text into structured JSON data suitable for Applicant Tracking Systems (ATS). By leveraging a pre-trained BERT model (dslim/bert-base-NER) fine-tuned for Named Entity Recognition, the application identifies key entities such as candidate names, organizations, and locations. To address common limitations in generic NLP models such as misclassifying technical terms or failing to distinguish universities from companies the system incorporates a hybrid post-processing layer. This ensures high-accuracy extraction of **Candidate Name**, **Education**, **Work Experience**, and **Technical Skills** from raw text.

Short Documentation

1. What I Understood

The core challenge of this project is Information Extraction (IE). Resumes are semi-structured documents where humans can easily identify a "Company" versus a "Skill," but machines struggle because they lack context.

- **The Model's Role:** A standard Named Entity Recognition (NER) model can identify "Google" as an organization and "New York" as a Location.
- **The Gap:** Standard models fail on domain-specific nuance. For example, they often classify "Stanford University" as a Company (because it is an organization) and misclassify capitalized technical terms like "Docker" or "Kubernetes" as names of people (e.g., "Dock" and "Ku").
- **The Solution:** A pure AI model is insufficient. A robust solution requires a Hybrid Approach: using Deep Learning and rule-based software engineering to refine and correct the output.

2. What I Built

I built a Python class named ReliableResumeParser that processes raw resume text in two stages:

Stage 1: AI Pass (BERT)

- I used the dslim/bert-base-NER model via the Hugging Face pipeline.
- I enabled aggregation_strategy="simple" to automatically stitch fragmented tokens (e.g., "San", "Fran", "##cisco") back into coherent words ("San Francisco").

Stage 2: Logic Pass (Heuristics) To fix the errors observed in testing, I implemented a custom logic layer on top of the AI's output:

- **Confidence Filtering:** The system discards any entity where the model's confidence score is below 85% to prevent hallucinated data (like the "Dock" error).
- **Education Separation:** The model groups all organizations together. I added logic to check for keywords like "University" or "College" to move these entities from the "Companies" list to a dedicated "Universities" list.
- **Skill Extraction:** I created a "Known Skills" lookup. If the text contains words like "Python," "SQL," or "AWS," they are forcibly categorized as "Skills," overriding any incorrect classification by the AI.
- **Name Fallback:** If the AI fails to detect a name the system defaults to capturing the first line of the document so that no resume goes unidentified.

Output: The final output is a clean, deduplicated JSON object containing separated lists for Candidate Name, Companies, Universities, Locations, and Detected Skills.

Outputs:

Resume

```
# --- TEST ---
if __name__ == "__main__":
    real_resume = """
    John Smith
    Seattle | john.smith@email.com

EXPERIENCE

Software Engineer
Microsoft | Redmond, WA
2018 - Present
- Developed new features for Azure cloud services using C# and .NET.
- Collaborated with teams in London and Tokyo to improve server uptime.

Junior Developer
Amazon | Seattle, WA
2016 - 2018
- Maintained the internal inventory system using Java and AWS.
- Fixed bugs in the checkout process.

EDUCATION

Bachelor of Science in Computer Science
PES University | Seattle, WA"""
```

Output

```
Analyzing resume...

--- 📄 Parse Results ---
{
    "Candidate Name": "John Smith",
    "Companies": [
        "Microsoft",
        "Amazon"
    ],
    "Universities": [
        "PES University"
    ],
    "Locations": [
        "WA",
        "Seattle",
        "Tokyo",
        "London",
        "Redmond"
    ],
    "Detected Skills": [
        "Java",
        "Azure",
        "AWS"
    ],
    "Debug_Log": [
...
        "Accepted: Seattle (LOC) - 1.00",
        "Accepted: WA (LOC) - 1.00"
    ]
}
```