GEN_AI PROJECT

NAME: L HARSHA VARDHAN
SRN: PES2UG23CS303
SEC: E

PROBLEM STATEMENT: **Smart Resume Parser**

Abstract:

This project implements an automated recruitment tool using Hugging Fac**e** transformers to convert unstructured resumes into structured data. By integrating a BERT-based Named Entity Recognition (NER) pipeline for extracting specific entities like names and organizations with a BART-powered Zero-Shot Classification model for identifying document sections (e.g., "Work Experience" vs. "Education"), the system eliminates the need for manual data entry. The architecture leverages deep contextual embeddings to maintain high accuracy across diverse formatting styles, providing a scalable solution for streamlining HR workflows and candidate screening.

Project:

The goal was to build a specialized NLP agent capable of transforming unstructured resume text into a structured, machine-readable format. I understood that manual data entry from CVs is inefficient, so I built a pipeline that "reads" and "categorizes" information like a human recruiter would.

I designed a dual-engine pipeline using the Hugging Face Transformers library. This approach avoids the need for massive custom datasets by leveraging pre-trained models.

- **Extraction Engine (NER):** I utilized `bert-base-NER` to perform token-level analysis. This allows the system to pinpoint specific entities—names, companies, and universities—within a sentence.
- **Classification Engine (Zero-Shot):** I integrated `bart-large-mnli` to handle section identification. Unlike traditional keyword matching, this model uses Natural Language Inference (NLI) to understand that a header like "Past Roles" is semantically equivalent to "Work Experience."

Pipeline:

Input: Raw text extracted from a resume.

Processing: The text is passed through the BERT model to extract the "Who" and "Where."

Categorization: The BART model labels the intent of the text block.

Output: A structured dictionary containing the detected section and a list of identified entities.

Output:

```
test_resume = """
  Senior Data Scientist at Microsoft from 2021 to Present.
  Previously worked at Tesla as a Junior Engineer.
  Master of Science in Artificial Intelligence from Stanford University.
  """
```

✓ 0.0s

```
--- PARSING RESULTS ---
Predicted Section: Work Experience (0.682)

Entities Extracted:
- Microsoft (ORG)
- Tesla (ORG)
- Artificial Intelligence (MISC)
- Stanford University (ORG)
```

```
test_resume = """
RELEVANT EXPERIENCE
Lead NLP Engineer | TechFlow Solutions, San Francisco | Jan 2022 - Present
- Orchestrated the deployment of a Retrieval-Augmented Generation (RAG) pipeline using LangChain and Pinecone.
- Scaled transformer-based microservices on AWS SageMaker, reducing latency by 40%.
- Mentored a team of 5 junior developers in Agile sprints.

ACADEMIC BACKGROUND
Ph.D. in Computational Linguistics
Massachusetts Institute of Technology (MIT), Cambridge
GPA: 3.9/4.0 | 2017 - 2021

TECHNICAL EXPERTISE
Languages: Python, Rust, C++, SQL.
Frameworks: PyTorch, Hugging Face Transformers, Docker, Kubernetes.
"""
```

✓ 0.0s

```
--- PARSING RESULTS ---
Predicted Section: Technical Skills (0.699)

Entities Extracted:
- TechFlow Solutions (ORG)
- San Francisco (LOC)
- Re (MISC)
- Generation (MISC)
- LangChain (ORG)
- Pinecone (ORG)
- A (MISC)
- ##M (ORG)
- ACADEMIC (ORG)
- Computational (MISC)
- Linguistics (ORG)
- Massachusetts Institute of Technology (ORG)
- MIT (ORG)
- Cambridge (LOC)
- Python (MISC)
- R (MISC)
- C (MISC)
- SQL (MISC)
- P (ORG)
- ##Tor (ORG)
- Face (ORG)
- Dock (ORG)
- Ku (ORG)
- ##net (ORG)
```

```
    ner_tagger = pipeline("ner", model="dslim/bert-base-NER", aggregation_strategy="simple")

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

✓ 3m 47.0s Python

```
are sending unauthenticated requests to the HF Hub. Please set a HF_TOKEN to enable higher rate limits and faster downloads.
ts: 100%|██████████| 199/199 [00:01<00:00, 172.64it/s, Materializing param=classifier.weight]
lassification LOAD REPORT from: dslim/bert-base-NER
           | Status     |  |
-----------+------------+--+-
ense.weight | UNEXPECTED |  |
ense.bias   | UNEXPECTED |  |


   :can be ignored when loading from different task/architecture; not ok if you expect identical arch.
ts: 100%|██████████| 515/515 [00:01<00:00, 484.72it/s, Materializing param=model.shared.weight]
```

```python
def parse_resume_snippet(text):
    candidate_labels = ["Education", "Work Experience", "Technical Skills", "Contact Information"]

    category_result = classifier(text, candidate_labels=candidate_labels)
    top_category = category_result['labels'][0]
    confidence = category_result['scores'][0]

    entities = ner_tagger(text)

    structured_entities = [
        {"word": ent['word'], "label": ent['entity_group'], "score": round(ent['score'], 3)}
        for ent in entities
    ]

    return {
        "detected_section": top_category,
        "confidence_score": round(confidence, 3),
        "entities_found": structured_entities
    }
```