Generative AI and its Applications

Name : Amrutha P J	SRN : PES2UG23CS059	Section : A

Project Date : 20-01-2026

Project Report: Smart Resume Parser

1. Project Overview & Objective

For the applied component of Unit 1, I chose to implement a Smart Resume Parser. The objective was to build a tool that can take unstructured text (a raw resume) and automatically extract structured information—specifically the candidate's name, the organizations they have worked for, and their location/education.

This project gave me the opportunity to apply Named Entity Recognition (NER), a core NLP concept we studied, to a real-world business automation problem.

2. Model Selection & Architecture

I selected the dbmdz/bert-large-cased-finetuned-conll03-english model from the Hugging Face Hub.

- **Why this model?** This is a version of BERT that has been specifically fine-tuned on the CoNLL-03 dataset, which is the gold standard for NER tasks. It is trained to recognize four types of entities: Persons (PER), Organizations (ORG), Locations (LOC), and Miscellaneous (MISC).
- **The Aggregation Strategy:** I learned that BERT uses a WordPiece tokenizer, which often splits words into fragments (e.g., Stanford might become Stan and ##ford). To handle this, I used the aggregation_strategy=simple parameter in the pipeline. This was a crucial step; without it, the model returned fragmented tokens that were hard to read. With it, the pipeline automatically reconstructed the full words for me.

3. Implementation Details

The core logic of my project involved post-processing the raw model output to create a clean, usable profile.

- **Logic Flow:** I wrote a function parse_resume() that accepts raw text and runs the NER pipeline.
- **Filtering:** I implemented a confidence threshold check (score > 0.85) to ensure that we only extracted high-quality entities and reduced false positives.
- **De-duplication:** A major challenge I observed was that a candidate might mention their company (Google) multiple times. To solve this, I used Python set() data structures to store the entities, which automatically removed duplicates and kept the output clean.

4. Observations & Results

I tested the parser with a synthetic resume for a Lead Data Scientist named Sophia R. Turner.

- **Performance:** The model successfully identified Sophia R. Turner as a PER (Person), and correctly classified Microsoft, Tesla, and Stanford University as ORG (Organizations).
- **Nuance:** Interestingly, it correctly identified San Francisco and New York as locations.
- **Limitation:** I noticed that the model sometimes confuses universities and companies because they are both tagged as Organizations in the standard dataset. This taught me that while pre-trained models are powerful, domain-specific fine-tuning would be needed to distinguish between a School and a Workplace perfectly.

5. Conclusion

This project demonstrated the power of Transfer Learning. With just a few lines of code, I was able to leverage a model that had already read millions of documents to solve a complex extraction task. It shifted my perspective from just running a model to building a system around a model, where the pre- and post-processing logic is just as important as the neural network itself.

*****