Data Science Report

1. Project Overview

Project Name: AI Agent for Structured Job Information Extraction

Problem Statement

students encounter the difficulty in tailoring the resume every now and then, thats why we came up with this AI resume, cover letter generator, just fill the data your data once, edit it later, tailors your resume and content letter as per the description of the job in any speicifc company.

Solution

I developed an AI agent leveraging a fine-tuned large language model to automatically extract structured information from unstructured job descriptions. The system accepts raw job postings and produces outputs in a consistent JSON schema comprising the following key fields:

- Core Responsibilities
- Required Skills
- Educational Requirements
- Experience Level
- Preferred Qualifications
- Compensation and Benefits

This structured output enables downstream applications such as resume matching engines, hiring dashboards, and job recommendation systems to programmatically process and analyze job postings more effectively.

2. Training Data

Original Dataset

- **File:** `./training_data.xlsx`->trained in PC.
- **Number of rows:** Approximately 11,024
- **Columns include:**
    - job_id
    - category
    - job_title
    - job_description
    - job_skill_set

The dataset consisted of raw job postings with associated metadata such as job titles, skill tags, and categories.

Data Transformation for Fine-tuning

To prepare the data for fine-tuning our model, we converted raw job records into instruction–response pairs suitable for a low-rank adaptation (LoRA) fine-tuning approach. The process involved:

- Extracting core fields such as company name, job title, and the full job description

- Generating clean and valid JSON model responses specifying structured job information
- Constructing prompts in a clear instruction format requesting extraction of specific fields in JSON format
- If any field was missing in the job description, the value was marked as "N/A" to ensure consistent JSON structure

Example transformation:

```
### Instruction:
Extract the following information from the job description in a structured JSON
format.
The JSON should have exactly these keys:
"Core Responsibilities", "Required Skills", "Educational Requirements",
"Experience Level", "Preferred Qualifications", "Compensation and Benefits".
If information for a key is not present, use "N/A".

Job Title: Apple Solutions Consultant
Company: Apple
Job Description:
As an ASC you will be highly influential in growing mind and market ...

### Response:
{
  "Core Responsibilities": "...",
  "Required Skills": "...",
  "Educational Requirements": "...",
  "Experience Level": "...",
  "Preferred Qualifications": "...",
  "Compensation and Benefits": "N/A"
}
```

The final prepared dataset for fine-tuning was saved as `./mistral_training_data.txt` containing the instruction–response pairs separated by newlines.

3. Model and Fine-tuning Setup

Base Model

Mistral-7B-Instruct-v0.2

Reasons for Choosing Mistral

- Smaller VRAM footprint (~4GB quantized), enabling efficient fine-tuning and deployment on consumer GPUs
- Lower hallucination tendency compared to larger models (such as Llama 2-7B or Llama 3.1-8B) for structured extraction tasks
- Strong capabilities in following instruction-based prompts critical for precise JSON extraction

Fine-tuning Method

- Employed LoRA (Low-Rank Adaptation) technique using Hugging Face PEFT library

- LoRA enables efficient fine-tuning by training small adapter modules rather than updating all model parameters, significantly reducing resource requirements and training time
- **Quantization:** BitsAndBytes 4-bit (nf4 with double quantization and fp16 computation) to reduce GPU memory load during training and inference

Fine-tuning Hyperparameters

- **Training epochs:** 3
- **Batch size:** 16
- **Learning rate:** 2e-4
- **Iterations performed:** 200 out of approximately 4,134 planned, because of time constraint.

4. Evaluation Methodology and Outcomes

Evaluation Strategy

- No formal quantitative metrics (precision, recall, F1 score) were computed during this project phase due to time and resource constraints.
- Primary evaluation was qualitative, involving manual review of about 100 random samples from the model output

Qualitative Findings

- Majority of extracted fields aligned accurately with the original job descriptions
- The model produced consistent and cleanly formatted JSON outputs
- Hallucination in skill sets and qualifications was significantly reduced compared to baseline models (Llama 2 and Llama 3) used prior to fine-tuning
- Outputs were human-readable and well-structured for downstream use

5. Justification for Fine-tuning Target and Architecture Choices

Task Specialization: The core objective was to create a model specialized in extracting structured information from loosely formatted job descriptions, a highly domain-specific task demanding strict output formatting. Fine-tuning the Mistral base model with LoRA adapters enabled this specialization efficiently.

Improved Reliability: Base large language models tend to hallucinate or generate inconsistent outputs when tasked with structured data extraction. Fine-tuning mitigates this by adapting the model to precisely map free-text job descriptions to a consistent JSON schema.

Adapted Style: The instruction–response style with explicit keys ensures that the model understands the need for highly structured extraction rather than freeform text generation.

Resource Constraints: Due to limited access to large computational resources and absence of massive, well-aligned job-description–resume paired datasets, LoRA fine-tuning on Mistral (a lightweight quantized model) was the most feasible approach.

Modular Architecture: Separating the job information extraction LLM (Mistral-LoRA) from the resume and cover letter generation (handled by Gemini API) ensures flexibility, maintainability, and improved overall system performance.

6. System Architecture Diagram (Summary)

```
+------------------------------+
|     Job Description Input    |
+-------------+----------------+
              |
       [Fine-tuned Mistral LLM]
          (LoRA Adapters + Quant)
              |
   Structured JSON Extraction
              |
+-------------v------------------+
|        Flask API Server        |
|  (Serves /extract endpoint)    |
+-------------+------------------+
              |
        JSON Job Info Output
              |
[Gemini API (Pre-trained LLM)] <--- Resume and Cover Letter generation
              |
      Tailored Resume & Cover Letter
```

This modular pipeline enables efficient and specialized processing of job postings followed by personalized document generation.