AI Engineer Homework

Instructions

- Provide reproducible code with detailed setup instructions
- Programming language: Python
- Focus on model evaluation, edge case discovery, and iterative improvement
- API deployment is **optional** (bonus points)

Expected Deliverables

1. Git repo with Jupyter notebook containing all experiments and evaluations
2. Experiments should be reproducible with clear model version tracking
3. **Runnable evaluation framework** that works across all model iterations
4. Technical report with findings and improvement analysis. Just a brief write-up is enough. Be ready to discuss things more in depth during the interview.
5. **You can submit partial tasks with an explanation of why you were unable to complete it in full.**

Task Overview

Build and iteratively improve a fine-tuned LLM for domain name suggestions, with emphasis on systematic evaluation, edge case discovery, and model improvement cycles. Make sure the model refuses to generate inappropriate/harmful content domain names. Deploy selected model as API endpoint (Optional).

Tip: in case of time constrains focus on model evaluation and improvement framework instead of fine-tuning/deployment

Required Components

1. Synthetic Dataset Creation

- Create initial synthetic dataset
- Include diverse business types and complexity levels

- Document dataset creation methodology

2. Model Development & Iteration

- **Baseline Model**: Fine-tune initial open-source LLM. You can use common recipes for that.
- **Improved Model(s)**: Address discovered issues through, i.e.:
    - Dataset augmentation
    - Different fine-tuning approaches (LoRA, full fine-tuning, etc.)
    - Hyperparameter optimization
- Save and version all model checkpoints

3. LLM-as-a-Judge Evaluation Framework

Implementation Requirements:

- Design and implement automated evaluation using LLM-as-a-judge
- You may use any third-party API models (GPT-4, Claude, etc.) or fine-tune your own open-source evaluation model
- Create systematic scoring methodology for domain name quality

4. Edge Case Discovery & Analysis

- Systematically discover model failure modes and edge cases
- Categorize and analyze different types of failures
- Demonstrate measurable improvement in handling edge cases
- Document root causes and improvement strategies

5. Safety Guardrails

- Content filtering for inappropriate requests
- Document approach and test with examples

Model Requirements

- **Domain Name Generator**: Must use open source LLM (Llama, Mistral, etc.)
- **LLM-as-a-Judge**: May use third-party API models or fine-tuned open-source models
- All code must be reproducible with clear setup instructions

Technical Report Guidelines

1. Methodology & Initial Results

- Dataset creation approach and baseline model selection
- Initial model performance and evaluation metrics

2. Edge Case Analysis

- **Discovery process**: How you found edge cases
- **Failure taxonomy**: Categories of failures with examples
- **Frequency analysis**: How common each failure type is

3. Iterative Improvement

- **Improvement strategies**: What you tried and why
- **Quantified results**: Before/after metrics for each iteration
- **LLM judge validation**: How you ensured evaluation quality

4. Model Comparison & Recommendations

- **Performance comparison**: Statistical significance of improvements
- **Production readiness**: Which version you'd deploy and why
- **Future improvements**: Next steps for continued improvement

API Development (Optional)

- Simple API endpoint

API Specifications

- Input: JSON with business description field
- Output: JSON with list of domain suggestions, confidence scores, and status

Example Request:

```
{   "business_description": "organic coffee shop in downtown area"  }
```

Example Response:

{   "suggestions": [{"domain": "organicbeanscafe.com", "confidence": 0.92},   {"domain": "downtowncoffee.org", "confidence": 0.87},   {"domain": "freshbreworganic.net", "confidence": 0.83}   ],   "status": "success" }

Safety Example Request:

{   "business_description": "adult content website with explicit nude content" }

Safety Example Response:

{   "suggestions": [],   "status": "blocked",   "message": "Request contains inappropriate content" }