

CSE 476 Final Project Report

Kenisha Kaushal

GitHub: https://github.com/kenishakaushal/CSE476_FinalProject

1. Overview

This project implements an inference-time answering agent for the CSE 476 Final Project. Instead of training a model, the agent builds a robust inference pipeline around the provided LLM (`bens_model`).

The system uses structured prompting, answer extraction, retry logic, and concurrent processing to reliably generate predictions for all test questions.

2. How the Agent Works

Structured Prompting

Every question is wrapped in a short chain-of-thought system prompt instructing the model to end with:

`FINAL ANSWER: <value>`

This ensures consistent and easily parsed outputs.

Answer Extraction

`extract_answer():`

- isolates text after `FINAL ANSWER:`
- removes whitespace, trailing punctuation, leading signs
- normalizes numeric formatting

Retry Logic

`run_agent()` attempts up to three calls using exponential backoff (1s, 2s, 3s) to handle empty or malformed responses.

Batch Processing + Autosave

`generate_all_answers():`

- processes questions in parallel batches of 10
- autosaves progress after each question to `cse_476_final_project_answers_partial.json`
- preserves input order and logs progress

3. Key Implementation Details

The core implementation resides in `my_agent.py`:

- `call_model_chat_completions()`: model API wrapper
- `ask_solver()`: builds prompts and queries the LLM
- `extract_answer()`: parses/normalizes outputs
- `run_agent()`: full per-question pipeline with retries
- `generate_all_answers()`: batching, concurrency, autosave

Integration with the submission script occurs in:

```
cse476_final_project_submission/generate_answer_template.py
```

where `build_answers()` calls `generate_all_answers()`.

4. Reproducibility

Requirements:

- ASU VPN or network access
- Python 3 + `requests`

Install:

```
pip install requests
# macOS fallback:
pip install --break-system-packages requests
```

Regenerate predictions:

```
cd cse476_final_project_submission
python generate_answer_template.py
```

Single-question test:

```
from my_agent import run_agent
print(run_agent("What is 25 * 4?"))
```