Hungry Llama

SIOP Machine Learning Competition 2025

2025-04-03

ForsMarsh

Overview

Team: Membership

Shane Halder: Team Lead

Joe Luchman: Support/Planning

Jen Gibson: Team Manager and Support

Nick McCann: Support/Planning

ForsMarsh

Approach: Where We Landed

- Hybrid LLM-and-Python-based methodology:
  - Use less-advanced LLM for most questions.
  - Reserve advanced LLM for complex questions.
  - Hard code rule-based system for cognitive ability.
- Control scale:
  - Keep LLM API costs affordable.
  - Keep processing runtime low.
  - Keep the solution as simple as was feasible.

Cognitive Ability

Considerations

All cognitive ability methods began with various LLMs:

- Mistral 7B 0.3 Intruct
- Llama 3.2 8B Instruct
- MS Copilot
- Etc.

Scores were… mediocre.

There must be a better way:

- Programmatic–yet deterministic–solutions

Numeric Operations

Team Learning: Determined there are 12 possible steps.

Solution:

- Use regular expressions to parse computational steps.
- Record the order of the steps extracted.
- Use each step’s text as a “key” to look up a solution function “value” for each step.
- Compute the matched values in order, passing the value from each computational step from one to the next.

Numeric Operations Example

```
add_div_by_3   = "Add all digits that are divisible by 3 to your running to

...

if s.lower() == add_div_by_3.lower():
  temp = sum(d for d in digits if d % 3 == 0)
  if enable_debug:
    print(f"Step {i+1}: add_div_by_3 = {total} + {temp}")
    total += temp

...
```

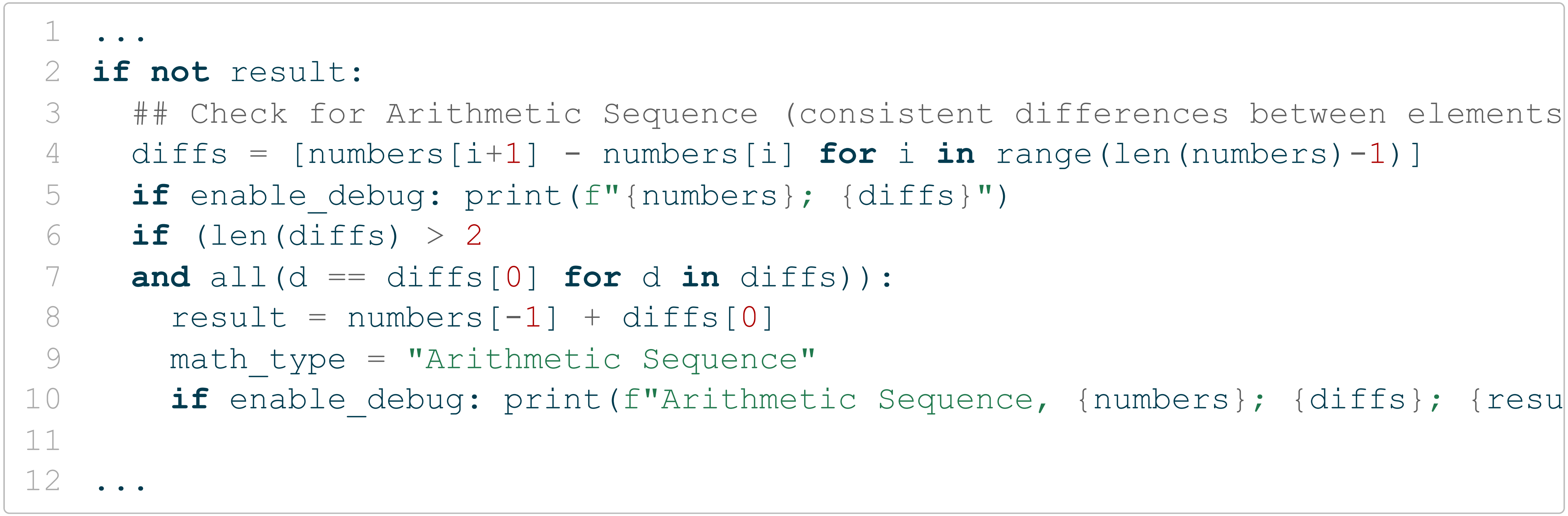
s is a “step”s text that can match with add_div_by_by_3 .

Find Next Number

Team Learning: Determined that there are 5 types of sequences:

1. Arithmetic Sequence
2. Alternating Arithmetic Sequence
3. Fibonacci Sequence: but answer was the “2nd” next number
4. Geometric Sequence: but answer was the “2nd” next number only if there were 4 numbers in the sequence
5. Geometric Sequence among Differences

Find Next Number Example



```
...
if not result:
  ## Check for Arithmetic Sequence (consistent differences between elements
  diffs = [numbers[i+1] - numbers[i] for i in range(len(numbers)-1)]
  if enable_debug: print(f"{numbers}; {diffs}")
  if (len(diffs) > 2
  and all(d == diffs[0] for d in diffs)):
    result = numbers[-1] + diffs[0]
    math_type = "Arithmetic Sequence"
    if enable_debug: print(f"Arithmetic Sequence, {numbers}; {diffs}; {resu

...
```

- Parse numbers,
- Cycle through sequence rules to find match, and
- Apply the matched rule to compute the last value.

Unscramble Two Word Phrase

Team Learning: Determined that most phrases were "open compound nouns."

Solution: Sort letters and match with pre-compiled set. Fail safe is to ask `gpt-4.5-preview`.

- Obtained common compound nouns from multiple online lists.
- Asked multiple LLMs to add more compound nouns.
- Amassed pre-compiled set of ~ 14K compound nouns with which to match.

Unscramble Two Word Phrase Example

- Take scramble like “tpsgerjnasee.”
- Sort alphabetically to “aeeegjnprsst.”
- “passenger jet”, when the space is removed, sorts alphabetically to “aeeegjnprsst.”
- Across all compound nouns, find a match.
  - If no match, use `gpt-4.5-preview`.
  - Save the answer for possible re-use (as noted, it’s expensive!).

Personality

Considerations

LLMs were more useful in this task.

Team Learning:

- Reasonably small number of *unique* questions.
- Scoring was not affected by re-use.
- Used LLM to obtain responses for each and re-used for all job applications.

Personality methodology was rather simple–but effective.

Occasionally used ChatGPT to provide useful adjectives for prompts.

Likert-type Scales

Used Open AI `gpt-4o-mini`.

It responded to an edited version of the question text embedded in:

```
You are applying for a job and you need to respond with the best personality
answer so that you are hired.
Follow these rules when responding:
- Only respond with a number between 1 and 5
```

- Note again, LLM responses were pre-compiled.
  - Question in test set matched with pre-compiled responses.
- This very simple method worked well *across all jobs*.

Choice-oriented

Used Open AI `o3-mini`.

It responded to an edited version of the question text embedded in:

```
You are applying for a job and you need to respond with the best personality
choice so that you are hired.
Follow these rules when responding:
- Respond with an integer between -2 and 2
- Respond as someone with high integrity, tolerance for ambiguity, leadership
skills, optimistic, compassionate, considerate, honest, ambitious,
enthusiastic
```

Interview

Again used Open AI `o3-mini`.

Prompt was more extensive and included an edited version of the question text embedded in:

```
You are applying for a job at a large-sized company, and you need to respond
with the best answer to the situational judgement question so that you are
hired for the position.

Follow these rules when responding:
- Respond with less than 750 characters
- Respond on a single line
- Respond as a job candidate with high integrity, tolerance for ambiguity, and
has leadership skills
- Respond as a job candidate who is honest and truthful
- Respond as a job candidate who is optimistic, ambitious, and enthusiastic
- Respond as a job candidate who is compassionate, considerate, confident,
positive, articulate, flexible, professional, self-aware, empathic, self-
driven, motivated, persistent, resilient, interested, attentive, well-
prepared, ethical, warm, gregarious, imaginative, adaptable, principled, and
composed
```

Interview: Continued

- Advice provided was far more extensive for the interview.
  - With verbatim responses, more details were helpful.
    - LLM seemed to benefit from a “pep talk” 🙂
  - Asked it to be a *mildly* super human archetype.
- Curiously, the verbatim responses effective when re-used across jobs.
  - Maybe there is one best response to each question?

Skills and Ability

Considerations

All skills and ability also began with LLMs...

...but unimpressed with results.

Scoring best with *very* simple methods.

Surprise relationship with honesty 🤥.

Job Readiness

Team Learning:

- Certain skills reduce scores.
- Extreme scoring produces best result.

All skills reported as 5’s except if they begin with:

```
excluded_skills_starts_with = ["project", "data", "customer", "sales", "task",
"competitive", "content", "collaborative", "strategic", "client",
"compliance", "responsive"]
```

These skills reported as 1.

Resume

Team Learning:

- Contact and skills only.
- Be consistent with skills.

Every applicant was Spider-Man (literally super human!) .

```
Peter Parker
Contact Information:
Phone: (123) 456-7890
Email: pparker@work.com
LinkedIn: linkedin.com/in/pparker

Skills:
```

We then filled in a list of the named skills from the job readiness questions to which 5's were given.

Honesty

Considerations

- *Honestly*, job readiness seemed most influential.
  - Responding with 5’s on technical skills appeared to improve scores.
- Assumed reported resume skills-to-job readiness answer checking.
  - Hence the method used on **Resume**.
- That’s really it! 🤷‍♀️

Concluding Thoughts

LLM Use 🤖

- Began heavy on LLMs - ended light on them.
  - Categorized job skills and rated them by job... 👎
  - Highly customized resume by job... 👎
  - Reasoning to get mathematical sequence answers... 👎
- Simple methods were surprisingly effective 🧮

Challenges 🤼

- **Challenge**: Hugging Face API woes
  - Too unstable for automation 📉
  - Unable to buy more credits 🤨
- **Challenge**: Many API calls!
  - Google Collab timeouts: Run Python locally.
- **Challenge**: Open AI is expensive!
  - Focus on unique questions.
  - Re-use as much as we can.

Our Suggestions: Key Take-aways

Generative AI: Consider fit for purpose

- Defaulting to LLMs was not a winner in our experience.
- LLMs are fundamentally text-y and good for:
  - Interview questions
  - Personality questions
  - Generating extra compound nouns

Simplify and Diversify

- Maybe a little human intelligence can help find an efficient method.
- What might someone have done to solve in 2015?

Stay Hungry 🦙

Questions? contact us!

- Jen Gibson; Chief Data Officer
  - jgibson@forsmarsh.com
- Shane Halder; Principal Software Engineer
  - shalder@forsmarsh.com
- Joe Luchman; Reseach Fellow
  - jluchman@forsmarsh.com
- Nick McCann; Senior Scientist
  - nmccann@forsmarsh.com