Generative AI Use Reflection

As we have said in our syllabus, the habits you adopt in using generative models are yours to choose, and you should strive to work with these technologies responsibly as a study aid, not a fast-track to an answer. Generative AI technologies possess powerful capabilities that will facilitate not only your coursework now but activities in your professional work later on. There is a nascent but active community of researchers studying how AI tools can be used to help student education, and this reflection is based on key insights from that community.

The purpose of this reflection is to promote beneficial usage of generative AI and to encourage you to leverage those tools to aid in your learning process. Generative AI is ubiquitous, and we encourage you to use these tools to improve your learning process. The intended outcome of working through these questions is a focus on your learning process, rather than your learning output.

If you used any generative AI tool (e.g., ChatGPT, Copilot, Gemini) at any point in an assignment, you must complete the following questions for that assignment. As you complete this reflection, discuss in your own words, and do not use generative AI.

Question 1. Structured self-reflection as a part of work involving generative AI usage improves transparency for instructors and supports students in that they tend to rely less on generated content.[1]

(a) What generative AI tool(s) did you use? Provide the name(s) and model version(s).

  (i) ChatGPT – basic

  (ii) Claude (Free)

(b) Include the dialogues with those tools as screenshots attached to this reflection submission.

(c) Describe in a few sentences the places in the assignment where you used generative AI and how it was used. Why was it used in some places and not others?

  (i) I used it to help expand a function to return a double naming convention as well as a single naming convention for place names. I already had a function that I gave it to improve.

  (ii) I used Claude to help me figure out why I was having toml issues with the CI actions.

  (iii) I use Claude to help edit the css file by adding in '!important' to the end of every line.

Question 2. Over-reliance on generative AI models has been found to diminish learning outcomes. Thus, evaluating and verifying AI outputs, such as by consulting trusted sources like textbooks and journal articles, leads to users gaining better domain knowledge and avoiding that over-reliance.[2]

(a) How have you verified AI generations in the work you submitted?

  (i) Yes, I used it to collect the data.

  (ii) Yes, I was able to get the CI actions to succeed.

  (iii) Yes, it was a minor, tedious edit that it did.

(b) What modifications did you make to the context generated by the tools you used, and why did you make them?

  (i) Since it was just modifying what I gave it, I didn't make any additional modifications.

  (ii) Claude was slightly modifying what I already had so it there were many changes to make.

  (iii) Claude was slightly modifying what I already had so it there were many changes to make.

Question 3. Reflective practice allows us to focus on process—not output—and these practices prioritize meaningful learning over the substance of the final outputs of interacting with a generative model.[3]

---

[1] Combrinck, C., Loubser, N. Student self-reflection as a tool for managing GenAI use in large class assessment. Discov Educ 4, 72 (2025). https://doi.org/10.1007/s44217-025-00461-2

[2] Zhai, C., Wibowo, S. & Li, L.D. The effects of over-reliance on AI dialogue systems on students' cognitive abilities: a systematic review. Smart Learn. Environ. 11, 28 (2024). https://doi.org/10.1186/s40561-024-00316-7

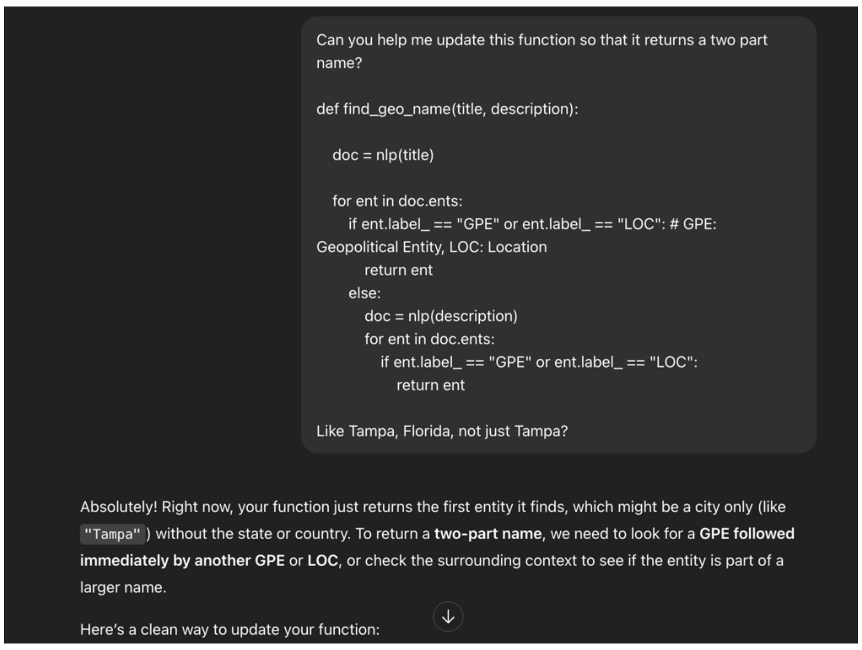
[3] Estaphan, S., Kramer, D., and Witchel, H.J. Navigating the frontier of AI-assisted student assignments: challenges, skills, and solutions. Advances in Physiology Education 49.3 (2025): 633-639. https://doi.org/10.1152/advan.00253.2024

(a) Let’s reflect more broadly. What did you learn in this assignment?
   (i) I learned that there are different ways to write toml files. I thought they were all related to poetry, but they are also hatch. We also didn’t end of going this route and ended using poetry.

(b) Thinking about the process of your learning rather than the output, how was your learning process impacted by the use of generative AI?
   (i) I learned a lot from this project and genAI helped me with understanding topics prior to the project or conceptual things. I believe I actually understand how all the different pieces of the app work together now.

(c) If you hadn’t used generative AI, what would have gone differently?
   (i) I would have just spent longer trying to find answers and completing tedious parts of tasks by hand.

(i)



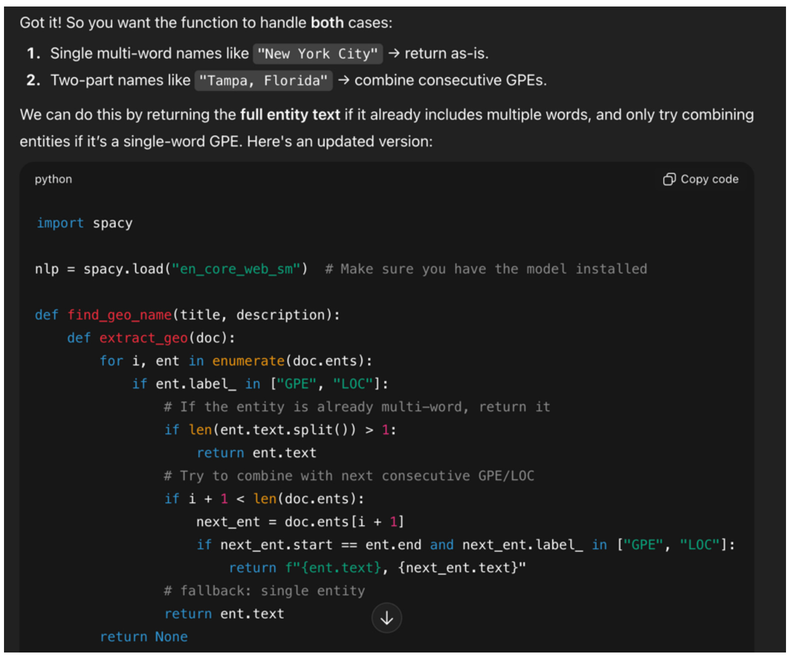
Can you help me update this function so that it returns a two part name?

```
def find_geo_name(title, description):

   doc = nlp(title)

   for ent in doc.ents:
      if ent.label_ == "GPE" or ent.label_ == "LOC": # GPE:
Geopolitical Entity, LOC: Location
         return ent
      else:
         doc = nlp(description)
         for ent in doc.ents:
            if ent.label_ == "GPE" or ent.label_ == "LOC":
               return ent
```

Like Tampa, Florida, not just Tampa?

Absolutely! Right now, your function just returns the first entity it finds, which might be a city only (like `"Tampa"`) without the state or country. To return a **two-part name**, we need to look for a **GPE followed immediately by another GPE** or **LOC**, or check the surrounding context to see if the entity is part of a larger name.

Here's a clean way to update your function:



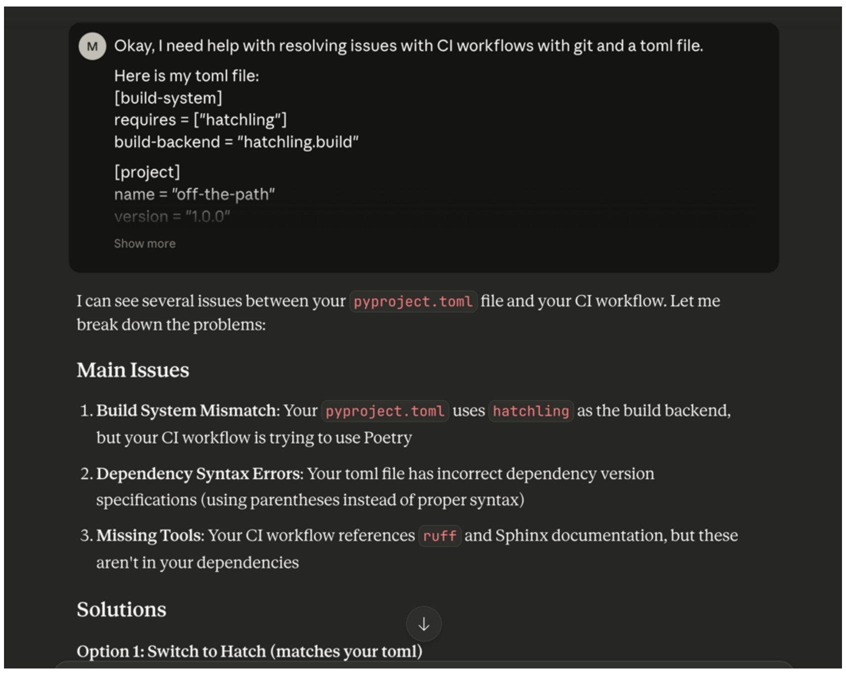
Got it! So you want the function to handle **both** cases:

1. Single multi-word names like `"New York City"` → return as-is.
2. Two-part names like `"Tampa, Florida"` → combine consecutive GPEs.

We can do this by returning the **full entity text** if it already includes multiple words, and only try combining entities if it's a single-word GPE. Here's an updated version:

```python
import spacy

nlp = spacy.load("en_core_web_sm")  # Make sure you have the model installed

def find_geo_name(title, description):
    def extract_geo(doc):
        for i, ent in enumerate(doc.ents):
            if ent.label_ in ["GPE", "LOC"]:
                # If the entity is already multi-word, return it
                if len(ent.text.split()) > 1:
                    return ent.text
                # Try to combine with next consecutive GPE/LOC
                if i + 1 < len(doc.ents):
                    next_ent = doc.ents[i + 1]
                    if next_ent.start == ent.end and next_ent.label_ in ["GPE", "LOC"]:
                        return f"{ent.text}, {next_ent.text}"
                # fallback: single entity
                return ent.text
        return None
```

(iv)



Okay, I need help with resolving issues with CI workflows with git and a toml file.

```
Here is my toml file:
[build-system]
requires = ["hatchling"]
build-backend = "hatchling.build"

[project]
name = "off-the-path"
version = "1.0.0"
```

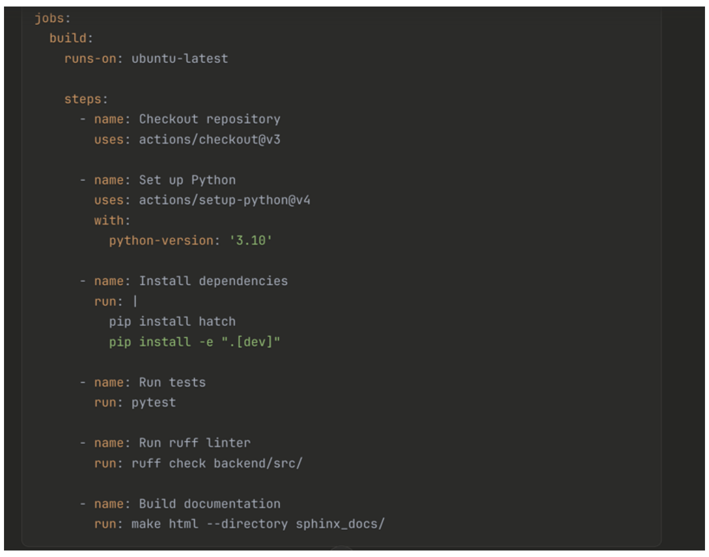
I can see several issues between your `pyproject.toml` file and your CI workflow. Let me break down the problems:

Main Issues

1. **Build System Mismatch**: Your `pyproject.toml` uses `hatchling` as the build backend, but your CI workflow is trying to use Poetry
2. **Dependency Syntax Errors**: Your toml file has incorrect dependency version specifications (using parentheses instead of proper syntax)
3. **Missing Tools**: Your CI workflow references `ruff` and Sphinx documentation, but these aren't in your dependencies

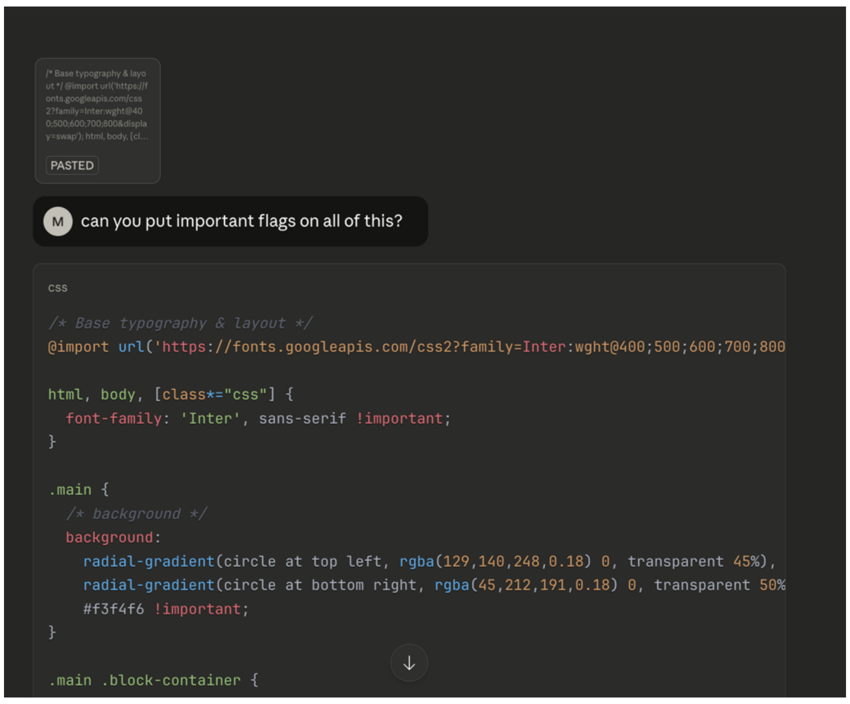
Solutions

Option 1: Switch to Hatch (matches your toml)



```yaml
jobs:
  build:
    runs-on: ubuntu-latest

    steps:
      - name: Checkout repository
        uses: actions/checkout@v3

      - name: Set up Python
        uses: actions/setup-python@v4
        with:
          python-version: '3.10'

      - name: Install dependencies
        run: |
          pip install hatch
          pip install -e ".[dev]"

      - name: Run tests
        run: pytest

      - name: Run ruff linter
        run: ruff check backend/src/

      - name: Build documentation
        run: make html --directory sphinx_docs/
```

(iii)



```
/* Base typography & layout */ @import url('https://fonts.googleapis.com/css2?family=Inter:wght@400;500;600;700;800&display=swap'); html, body, [cl...
```

can you put important flags on all of this?

```css
/* Base typography & layout */
@import url('https://fonts.googleapis.com/css2?family=Inter:wght@400;500;600;700;800

html, body, [class*="css"] {
  font-family: 'Inter', sans-serif !important;
}

.main {
  /* background */
  background:
    radial-gradient(circle at top left, rgba(129,140,248,0.18) 0, transparent 45%),
    radial-gradient(circle at bottom right, rgba(45,212,191,0.18) 0, transparent 50%
    #f3f4f6 !important;
}

.main .block-container {
```