

Pibit MLE-2 Round 1 Assignment

Assignment: Prompt Directory Validator Module

Submission Required

1. Zip File containing Python Module
2. Timeline - 2 days

Submission Expectations

1. We will evaluate on the prompts added in this module.
2. We will evaluate on the code quality, repo structure and completeness of code shared.
3. In interviews, we will go in-depth about different modules/frameworks used.

Objective

Build a Python module to update and validate a directory of prompts stored in `.txt` files. The module should detect:

1. **Redundant instructions** (repeated guidance that adds no new value)
2. **Conflicting instructions** (contradictory requirements)
3. **Prompt completeness** according to a given strategy

The module should **suggest fixes** and **update the prompt files** if approved by the user.

Prompt Strategy Compliance

Every prompt must follow these rules:

- **Task** – Clear description of what to do.
 - **Success Criteria** – Measurable, verifiable conditions for completion.
 - **Examples with Edge Cases** – Including at least one edge case; must **not** contain PII.
 - **CoT/TOT Steps if Required** – Explicitly include “Chain of Thought” or “Tree of Thought” guidance where reasoning is complex.
 - **No Secrets / No PII** – Must not contain personal information, credentials, or confidential data.
-

Functional Requirements

1. Input:

- Directory path containing `.txt` files (each file contains one prompt).
- Optional CLI or API arguments (e.g., `--fix`, `--report`).

2. Validation Tasks:

- Detect redundant phrases within a prompt.
- Detect contradictory guidance (e.g., “Be concise” and “Write 5000 words” in the same prompt).
- Check for missing required sections (Task, Success Criteria, Examples, CoT/TOT).
- Check for prohibited content (PII, secrets).

3. Reporting:

- Generate a report (JSON + CLI table) of issues found.
- Report should include file name, type of issue, and suggested fix.

4. Update Mechanism:

- If `--fix` is passed, module will auto-update the prompt files with corrections.

5. Extensibility:

- Code structured to allow adding new validation rules easily.

6. Testing:

- Include a `/tests` folder with **placeholders** for unit tests.
 - Coverage reports should be generated (e.g., via `pytest --cov`).
-

Technical Requirements

- Python 3.10+
 - Module must be **pip-installable** (`setup.py` or `pyproject.toml`).
 - Must be LLM-compatible — you may use OpenAI API, Hugging Face models, or any approved LLM to assist in:
 - Semantic duplicate detection.
 - Redundancy/conflict identification.
 - Must follow PEP8 + docstring conventions.
-

Deliverables

1. **Source Code** in a pip-installable format.
2. **README.md** with:
 - Installation instructions.

- Example usage (CLI + Python API).
3. **Sample Report** (from validating provided sample prompts).
 4. **Unit Test Placeholders**.
 5. **Coverage Report** (even if coverage is low at this stage).
-

Evaluation Criteria

- **Correctness** – Accurately detects all types of issues.
 - **Code Quality** – Readability, maintainability, modularity.
 - **Extensibility** – Ease of adding new rules.
 - **Testing Framework Readiness** – Even if full tests aren't implemented, structure must be in place.
 - **CLI & API Usability** – Should be easy to run.
 - **Documentation** – Clear, concise, complete.
-

Sample Prompts for Validation

Prompt 1 (Redundant + Missing Sections)

```
Write a detailed guide about planting tomatoes.  
Make sure it is very detailed and descriptive.  
Ensure the guide is extremely detailed with many details.
```

Prompt 2 (Conflicting Instructions)

```
Explain how to bake sourdough bread.  
The explanation must be no longer than 100 words.  
Include a comprehensive 10,000-word historical background on sourdough.
```

Prompt 3 (Missing Edge Cases)

```
Design a test plan for an e-commerce checkout system.  
List the functional requirements.  
Provide success criteria.
```

Prompt 4 (PII Violation)

Write an email to John Smith at john.smith@example.com explaining the meeting schedule.
Include his phone number for confirmation.
