Exercise A — Evidence-Grounded Extraction & Evaluation

Goal

Build a mini evaluation pipeline that scores **non-canonical extraction** using **evidence spans** as the anchor.

Ashwam journals are messy and human. **Symptoms, food, emotion, and mind concepts cannot be canonicalized** into fixed vocabularies. Your evaluation must still be objective and testable.

Input Provided

You'll receive:

1. **8–10 synthetic Ashwam journal entries** (mixed symptoms, food, emotion, mind)
2. A small **golden reference file** containing, for each journal:

- `domain` (symptom | food | emotion | mind)
- `evidence_span` (exact quote from the journal)
- `polarity` (present | absent | uncertain)
- `intensity_bucket` (low | medium | high | unknown) **or** `arousal_bucket` for emotion
- `time_bucket` (today | last_night | past_week | unknown)

No canonical labels will be provided.

Tasks

1) Design the extraction schema

Define the structure of an extracted **"semantic object"**:

- Clearly state which fields are **constrained** vs **free-text**
- Explain why this schema supports:
  - safety (no hallucinations)
  - evaluation (objective scoring)
  - extensibility (future attributes)

2) Propose an extraction approach

Describe how you would extract semantic objects from journal text:

- Prompt-only or pipeline (LLM-only, LLM + rules, etc.)
- How you enforce **evidence grounding**
- How you handle **uncertainty / abstention** safely

> You do not need to run a real model unless you choose to. You may also load provided predictions and focus on the evaluation harness.

3) Design the evaluation method

Explain exactly:

- How predicted objects are **matched to gold objects**
- What counts as **TP / FP / FN** *without canonical labels*
- How you score:
  - polarity
  - intensity/arousal buckets
  - time buckets

4) Run a mock evaluation

Show example outputs for **2–3 journals** and demonstrate how your scoring works:

- show matched objects
- show how TP/FP/FN were derived
- show example metric computation

5) Failure analysis

Identify at least **3 realistic failure modes**, and for each:

- why it happens
- how your system detects and/or mitigates it

Output Requirements

A) Extraction output schema (JSON)

Your pipeline must output semantic objects in JSON.

B) Scorer

Your scorer must compare **predicted objects vs gold** without relying on canonical labels, and compute at least:

- Object-level **Precision / Recall / F1**

  (matching based on evidence overlap + domain)
- **Polarity accuracy**
- **Bucket accuracy** (intensity/arousal/time) with a clearly defined rule
- **Evidence coverage rate**

  (% predicted items whose evidence spans are valid substrings of the journal text)

C) CLI entrypoint

Provide a CLI that:

- runs parse/extract (or loads provided predictions)
- runs scoring
- writes outputs

Example:

```
python -m ashwam_eval run --data ./data --out ./out
```

D) Output files

Your CLI must produce:

- `out/score_summary.json`
- `out/per_journal_scores.jsonl`

Constraints (strict)

- You may **not** map symptoms/food/emotion/mind into a fixed enum list.
- **Every extracted item must include an `evidence_span` copied from the journal text.**
- Avoid hallucinations: if evidence is missing, abstain or mark uncertain.
- Your solution must be deterministic: **same input → same output**.

What We're Evaluating

- Ability to reason beyond label accuracy
- Evidence-grounded extraction design
- Evaluation design maturity (objective + reproducible)
- Restraint (avoid over-parsing; abstain when unsure)
- Engineering hygiene (clean CLI + tests + readable code)

Data Package - What's inside

- `data/journals.jsonl` — **10** synthetic Ashwam journal entries (messy, mixed language, multi-sentence, negation, uncertainty)

- `data/gold.jsonl` — gold references using **evidence spans + polarity + buckets + time**, **no canonical labels**
- `data/sample_predictions.jsonl` *(optional)* — example predicted outputs for 2 journals to sanity-check a scorer
- `README.txt` — quick spec of fields and rules