**Comparison: *Given Technique* vs *Updated Technique***

| **Point** | **Given Technique** | **Updated Technique** | **Effect** |
| --- | --- | --- | --- |
| **Prompt Delivery** | Returns **list of role messages** (system, user) and feeds them directly to pipeline | Uses **render\_chat() with tokenizer.apply\_chat\_template** → converts messages into **TinyLlama’s expected chat format** | Updated Technique is closer to training data, → more reliable outputs |
| **System Prompt** | Inserted as one message in the list | Passed separately via render chat (keeps clear role separation) | Updated Technique cleaner + less drift |
| **Few-Shot Handling** | Embedded manually in user prompt | Embedded manually in user prompt | Same |
| **Detection Output Format** | “Label / Rationale / Flags” enforced in text | Same enforced structure | Same |
| **Maintainability** | Cleaner typing hints (List[Dict[str, Any]]), modular functions | Slightly more verbose, less typing hints | Given Technique easier to extend |
| **Output Consistency** | Sometimes drifts (extra text, not following exact structure) | More consistent (sticks to label/rationale/flags) | Updated Technique better |

**Prompt Differences**

The **wording of the prompts is almost identical** in both approaches:

* Both ask for:
  + Label: Human / AI / Hybrid
  + Rationale (2 points)
  + Flags
* Both include few-shot examples.
* The **difference lies in how the system vs user roles are packaged**.

**Bottom Line**

* **Given Technique**:
  + Simpler, modular, good for experimentation.
  + Risk: prompt drift, model not always following structure.
* **Updated Technique**:
  + Same prompts but wrapped with **chat template formatting** → matches TinyLlama training.
  + Results in **more consistent and rubric-aligned outputs**.

The improvement is **not because of different wording in prompts**, but because the **Updated Technique enforces correct chat role formatting**.