📄 Home Assignment — Conversational Agent KPI

Context

We are building a conversational agent to replace outbound operations at health centers. Its task is to convince patients to schedule their annual visits. Since these calls represent the company, quality matters.

This assignment focuses on analyzing **“repair cues”** — the words/phrases an agent uses to fix a breakdown in understanding (asking the patient to repeat, confirm, slow down, etc.).

Examples of repair cues:

- **Ask to repeat:** “Can you repeat?”, “Say that again”, “Pardon?”, “¿Puede repetir?”, “No le entendí”.
- **Rephrase:** “Let me rephrase”, “In other words…”, “I’ll restate that.”
- **Hearing/ASR issues:** “Bad connection”, “You’re breaking up”, “One more time, please.”
- **Confirmation of understanding:** “Just to confirm…”, “To make sure I understood…”

Your Task

1. **Build a multi-pattern matcher** to detect repair cues.
    - The matcher must have a per-turn runtime independent of the number of cue patterns *k*.
    - Target complexity: **O(L+M)**, where L = length of the turn and M = number of matches.
    - Acceptable approaches: Aho–Corasick, DFA/trie with failure links.
    - **Not acceptable:** running k independent regexes.

2. **Handle text normalization**:
    - Unicode NFC normalization
    - Case-folding (case-insensitive)
    - Accent-insensitive matching (e.g., “entendí” ≈ “entendi”)
3. **Benchmark your matcher** against a naive k-regex baseline:
    - Compare runtime at k = {10, 50, 200}.
    - Use the provided transcripts.
    - Report performance (tokens/sec or ms/turn).
4. **Usage of KPI (Short Written Task)**
   Please answer briefly (bullet points acceptable):
    - **Robustness/Recovery KPI:** How would you use repair cue frequency as a proxy to measure the agent’s ability to recover from misunderstandings?
    - **Comprehension Quality KPI:** How would you use repair cue usage as a proxy to measure the agent’s baseline comprehension quality (e.g., ASR/NLU accuracy)?
    - **Friction KPI:** How would you use repair cue patterns as a proxy to measure patient frustration or conversational friction?

What to Submit

- **Method note (5–8 lines):** Data structure used, time/memory bounds, and why it achieves O(L+M).
- **Cue list file (`repair_cues.txt`):** ≥40 cues, grouped by category (English and Spanish allowed).
- **Micro-benchmark results:** Runtime comparison against regex baseline.

- **Integration rule:** Implementation of the “fire only after patient turn” constraint.
- **Code:** Clearly structured, runnable (Python preferred, but any language is fine).
- **KPI usage answers** (short write-up, ½–1 page).

Time Budget

Please scope yourself to ≤4 hours. We value clarity and rigor over extras.

📂 Provided Files

- `transcripts.txt` — 10 example conversations (expand)
- `repair_cues.txt` — starter template (expand to ≥40)