Lecture 3: Support Triage + API Actions

Grounding, state, and safe external calls

University of Chicago

December 29, 2025

What you will build today

- A ticket triage pipeline (priority, category, route) using structured output
- A grounded draft response that cites a knowledge base (KB)
- A proposed action plan mapped to an allow-listed set of API calls
- A safe execution layer: validation + logging + failure handling

Business problem

Scenario: Customer support receives many tickets with uneven quality.

Goal: For each ticket, produce:

- triage labels (priority/category/route)
- a grounded draft reply (citing KB snippets)
- *optional* actions (create issue, refund request, CRM note) via APIs

Inputs (provided in `lecture_3/data`)

- `tickets.jsonl`: support tickets
- `kb/`: short markdown KB articles
- `api_stub_data.json`: local "API" data used by stub calls

New workflow primitive introduced

- **External actions**: turn model output into calls (safely!)
- **State**: tickets move through stages; we keep an audit log

Pipeline

1. Parse ticket $\rightarrow$ triage JSON
2. Retrieve relevant KB snippets (simple search) $\rightarrow$ context
3. Draft reply with citations $\rightarrow$ JSON
4. Propose actions $\rightarrow$ JSON (allow-listed)
5. Validate + execute stub APIs; write logs

Exercises

- Improve routing accuracy with better prompts
- Reduce hallucinations by enforcing citations
- Add a “safe executor” that rejects invalid actions
- Add retry logic and better logs (what failed, why, what next)

Deliverable

- Notebook: `notebooks/lecture_3_support_triage_api.ipynb`
- Output tables in `data/outputs/`:
  - `triage.csv`, `draft_replies.csv`, `actions_log.csv`

Extensions / Optional challenges

- **Stronger grounding**: require exact quotes; reject replies that reference uncited facts
- **Better retrieval**: upgrade keyword overlap to embeddings or hybrid; measure impact on quality
- **Ticket state machine**: enforce allowed transitions (new $\rightarrow$ triaged $\rightarrow$ awaiting_customer $\rightarrow$ resolved)
- **Safer execution**: per-action schema validation + dry-run mode + retry/backoff + richer logs
- **Adversarial tickets**: include prompt injection in ticket text; harden with instruction hierarchy + allow-lists