



Reading the Rap Sheet

Teaching a small model to read charge text — and knowing when it shouldn't decide

2026-10-21 09:00:00 -0500 · unpublished draft · review PDF

DRAFT · TODO banner image: assets/images/rapsheet-banner.png

A small model reads the charge. A versioned lookup reads the outcome. A threshold decides who gets to decide.

Finding Not Not Guilty.

— a real disposition string, 2,437 times



Introduction

Steven B Miers · October 21, 2026 · v1.00

Three strings in one public court table all mean the same thing: `FNG`, `Verdict-Not Guilty`, and `— 2,437 times — Finding Not Not Guilty`. The first is an abbreviation, the second is a sentence, the third is a data-entry artifact that a human reads correctly in half a second and a regex never will. Above all three sits `Nolle Prosecution : 714,681 rows` — two-thirds of the whole table — the prosecutor declining to proceed. Thirty-six distinct outcome strings in total; eleven of them appear fewer than fifty times; three appear exactly once.

That table is the whole argument of this piece. It holds **two different problems wearing the same record**, and the mistake most AI-for-records products make is solving both with the same tool.



Part 1: Concepts



Two vocabularies, one record

Every row carries a **charge title** — `AGG UUW/VEH/PREV CONVICTION`, `RET THEFT/DISP MERCH/<$300/PRECONV` — and a **disposition** — what the court did with it.

The disposition is a closed vocabulary. Thirty-six strings. A person can read all of them in ten minutes, decide what each one *means* (convicted, acquitted, not prosecuted, vacated, not final), write that decision down, give it a version number, and sign it. That artifact is `disposition_map.yaml`. It is not a model. When a disputant asks why their record said *convicted*, the answer is a line in a file with a version on it, not a logit.

The charge title is an open vocabulary — mostly. There are 1,365 distinct titles, abbreviated by clerks over decades. That's few enough that you *could* hand-write the lookup, and it would still be stale by spring: new statutes add titles every year, and the county's legacy case system spelled the old ones differently. A lookup memorizes. A model has a chance at the title it has never seen, and if it's calibrated, the decency to say when it isn't sure. That is a job for a model — a *small* one, fine-tuned on this county's text, that maps a title to one of the prosecutor's own offense categories. There are 78 values in that label column, and one of them — **PROMIS Conversion**, nearly ten thousand rows — is not an offense at all: it marks records migrated from a legacy case system. The label column has the same disease as the disposition column.

Rules are not weights. Put the closed vocabulary in a lookup a human owns; put the open vocabulary in a model a human supervises. The architecture below exists to keep those two things from blurring into each other.

What calibration buys you

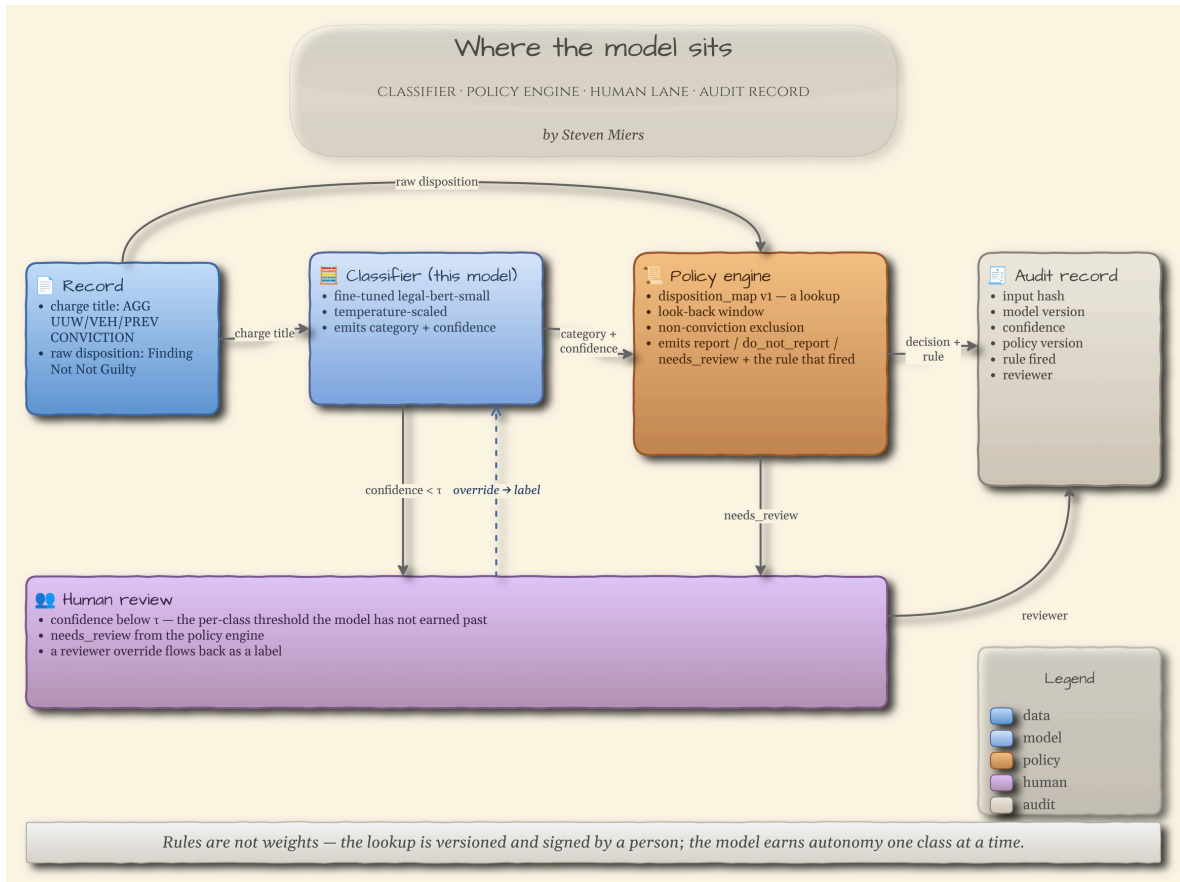
A classifier's softmax output is a score, not a probability, until you make it one. A fine-tuned transformer that says "92%" is, out of the box, usually overconfident — the 92% bucket might be right 80% of the time. **Temperature scaling** (Guo et al., 2017) fixes that with one scalar T fitted on held-out data: divide the logits by T before the softmax and the reported confidence starts to mean what it says. It changes no decisions — the argmax is identical — it only makes the confidence honest. That honesty is the precondition for everything that follows, because the next step treats confidence as a *budget*.

What abstention buys you

Once confidence is honest, you can set a threshold τ and say: *below τ , the model does not decide; a person does*. That is **selective classification** (Chow, 1970; Geifman & El-Yaniv, 2017), and the honest way to report it is a risk–coverage curve: at each coverage level, what error rate are you accepting? Pick an error budget — this piece uses 0.5% — and the curve tells you how much of the traffic the model may handle alone at that budget.

Do it **per class** and something more useful falls out: a table of which offense categories the model has *earned* the right to decide unaided, which ones route to review, and the one or two it never gets there on. That table is what autonomy looks like when it is earned rather than assumed — the same per-gate question Part 1 and Part 2 of the [autonomous AI series](#) kept asking, now with a number attached to each gate.

Where the model sits



Three slots and a lane. The **classifier** reads the title and emits a category with a calibrated confidence. The **policy engine** reads the raw disposition through `disposition_map v1`, applies the look-back window and the non-conviction exclusion, and emits `report`, `do_not_report`, or `needs_review` — *with the rule that fired*. The **human lane** receives everything below τ and everything the policy flagged, and a reviewer's override flows back as a training label. The **audit record** on the right captures input hash, model version, confidence, policy version, rule, and reviewer, so any decision can be replayed. The statute-lookup and TF-IDF baselines from Part 2 are deliberately absent from the picture: this is the architecture, not the benchmark.

The data, in three sentences

The data is one county's public-domain prosecutions table — about 1.08 million charge-level rows — de-identified by the publisher and pulled by query from a SODA open-data portal. The publisher stopped maintaining it in December 2024, though the platform's own row timestamps were last touched in April 2026 — so rather than trust the word *frozen*, every number in this piece traces to a run manifest that records the query, the row counts, the source's last-updated timestamp, and the seeds. Seven columns are dropped on load and never become features — `race`, `gender`, `age_at_incident`, `judge`, `incident_city`, `law_enforcement_agency`, `unit` — along with `offense_category`, the pre-update label, which would leak the target; the loader refuses a frame that carries any of them.

One more line, because it matters: this is a reference implementation of a *methodology*. It classifies charge text. It never attempts to match a record to a person, and it is not a screening tool.

Part 2: Build

 **Everything here is public.** The [rapsheet package](#), the [narrated notebook](#) that produces every number in this piece, and a [snapshot of the data](#) with the identifiers left out. This section shows only the code that teaches; the notebook has the rest.

DRAFT · §2 — drafted alongside the Tuesday-morning notebook cells: SoQL pull, grouped split, disposition map, Trainer call; one honest compute sentence.

Part 3: Evaluation

DRAFT · §3 — after training: headline table (rules · TF-IDF · legal-bert-small; stratified + temporal), the losses paragraph, F1 reliability, F2 risk-coverage, per-class autonomy table, F3 confusion matrix, the one-county limitation, what's next.

Key Takeaways

DRAFT · 4–5 bullets, after §3

References

1. Cook County State’s Attorney. *Dispositions* (dataset [apwk-dzx8](#), Cook County open-data portal), public domain; no longer maintained as of 30 December 2024. Retrieved 19 October 2026 (1,080,014 rows). Snapshot: [tangledpath/rapsheet-charge-titles](#).
2. Guo, Chuan, et al. “On Calibration of Modern Neural Networks.” *ICML 2017*.
3. Geifman, Yonatan, and Ran El-Yaniv. “Selective Classification for Deep Neural Networks.” *NeurIPS 2017*.
4. Chow, C. K. “On Optimum Recognition Error and Reject Tradeoff.” *IEEE Transactions on Information Theory* 16(1), 1970.
5. Fair Credit Reporting Act, 15 U.S.C. § 1681c(a) (FCRA § 605(a)) — reporting-period limits on adverse information.
6. Miers, Steven. *Semi and Fully Autonomous AI Systems*, [Part 1: HITL](#) and [Part 2: HOTL](#), 2026.
7. Miers, Steven. [rapsheet](#) : package and [notebook](#), 2026. Model card: [\(tangledpath/rapsheet-legal-bert-small, after Tuesday afternoon\)](#).

Steven Miers — [git-steven.github.io](#)