# DISCOVERING INTERPRETABLE TEXT CATEGORIES WITH NATURAL-LANGUAGE CODEBOOKS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Text classification often begins before a useful category inventory exists, and the inventory must also say how later texts are assigned. We formulate interpretable category discovery around three criteria: mutual exclusivity, collective exhaustiveness within a stated scope, and reproducible application from written rules. LLM-assisted Category Discovery (LCD) learns a natural-language codebook from unlabeled texts: each category has a name, a definition, inclusion and exclusion rules, and positive and negative examples. An LLM proposes every split, merge, revision and addition, and each is kept only if it also passes explicit tests on the memberships that an LLM reader verifies on the discovery texts; entries that claim fewer than three texts are retired. The frozen codebook is applied to new texts by a reader that may return zero, one or several categories, so gaps and overlaps remain visible. On BANKING77, CLINC150 and HWU64, whose development-split labels chose LCD's configuration (the label-free comparators ran untuned) and whose test split also scored intermediate configurations, LCD has the highest claim-set ARI and mapped macro-F1 over ten seeds on every dataset against every method we ran: TnT-LLM (ARI +.056, +.293, +.126; macro-F1 +.093, +.306, +.121; Holm-adjusted exact $p \leq .006$), its distilled classifiers, every automated LLooM variant, every clustering rule with a data-chosen number of clusters, and four label-informed references. In refits, removing the addition operator lowers pooled ARI by .293 and the split operator by .079, and no removal raises ARI or F1 significantly. Because its reader labels 128 texts per request, labeling costs 11.4–13.7 times less than TnT-LLM's LLM labeler, and total cost is lower at every corpus size on two datasets and from 301 texts on the third. With a weaker reader, LCD stays ahead in macro-F1 on all datasets and in claim-set ARI on two (one with uncovered texts pooled); two-reader agreement is similar for both methods.

## 1 INTRODUCTION

Text classification often begins before a useful category inventory exists. A team working on a large or accumulating collection may know the analytic dimension they care about, such as the specific request a customer is expressing, but not which categories actually occur. The collection may be too large to place in a single prompt, so a discovery sample can miss distinctions present elsewhere, while newly arriving texts require rules that apply beyond that sample. Grouping the observed texts is only part of this task: the discovered categories also need explicit membership rules that can be inspected and applied to later texts.

Recent methods describe discovered categories in natural language: LLooM induces concepts with inclusion criteria, TopicGPT describes topics, GoalEx selects cluster descriptions, and TnT-LLM generates a taxonomy and then labels texts with it (Lam et al., 2024; Pham et al., 2024; Wang et al., 2023; Wan et al., 2024). Readable descriptions are necessary but not sufficient for a reusable category system: the descriptions must also cover the stated scope, separate categories along the intended analytic dimension, and remain applicable to new texts.

We organize this task around three criteria: mutual exclusivity, collective exhaustiveness within a stated scope, and reproducible application from written rules. The first two assess the category system as a whole; the third asks whether independent readers of the same written definitions make consistent assignments. A natural-language codebook makes the definitions explicit through cat-



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

egory names, inclusion and exclusion rules, and positive and negative examples, giving readers concrete rules to inspect and apply.

LLM-assisted Category Discovery (LCD) learns such a codebook by verified editing. It keeps one of five first drafts by a label-free rule and then edits the codebook for up to six rounds with five operators: SPLIT, MERGE, REVISE, RETIRE, and ADD, which proposes categories for texts that no rule claims. Proposals come from the LLM reading samples of texts, and an edit is accepted only if it also passes explicit tests on the memberships that a joint reader, the same model, verifies on all discovery texts (an added category, for instance, must claim mainly uncovered texts), and guards stop any edit from creating a catch-all or taking over most of an existing category. After a final re-read, broad but loose categories are split into a two-level hierarchy, and interleaving fragments of one request are reported as one class. The frozen codebook is applied by a joint reader, 128 texts per request, with each verdict keyed to its message. It may return zero, one or several categories per text, so gaps and overlaps stay visible rather than being resolved by forced single-label assignment.

On BANKING77, CLINC150 and HWU64 over ten seeds, LCD has the highest claim-set ARI and mapped macro-F1 on every dataset against every method we ran (Table 1; the test split also scored about 25 intermediate configurations during development, so these margins may be optimistic, Section 4). It is ahead of TnT-LLM with its LLM labeler in ARI (+.056, +.293, +.126) and macro-F1 (+.093, +.306, +.121; Holm-adjusted exact $p \leq .006$), and of its distilled classifiers, every automated LLooM variant, every clustering rule that chooses its own number of clusters and four label-informed references by at least .099, .341, .104 and .041. In refits (Table 2), removing ADD lowers pooled ARI by .293 and SPLIT by .079, and no removal significantly raises ARI or F1. Because LCD's reader labels 128 texts per request, labeling costs 11.4–13.7 times less than with TnT-LLM's LLM labeler, and total cost is lower at every corpus size on two datasets and from 301 texts on the third (Figure 2); TnT-LLM's distilled classifiers cost less beyond 16.5k to 69.8k texts. With a weaker reader, LCD stays ahead in macro-F1 on all three datasets and in claim-set ARI on two, but with uncovered texts pooled only its CLINC150 ARI lead is significant (Appendix C.4); agreement between the two readers is similar for the two methods (Table 11). Our contributions are:

- a formulation of category discovery as learning a frozen natural-language codebook, judged by the exclusivity, exhaustiveness and reproducible application of its rules (Section 2);
- LCD, whose every proposed edit must pass membership tests read by the model that later applies the codebook, without reference labels, class counts or dataset names (Section 3);
- a ten-seed evaluation (five seeds for LLooM) with paired comparisons against TnT-LLM (Holm-corrected), its distilled classifiers, LLooM and embedding clustering, total cost as the number of texts grows, a weaker reader, refit ablations of the draft selection, each operator, the SPLIT pre-screen and the contest, same-read ablations of HCS and GROUP, and checks of reader format, definition adherence and long codebooks (Section 5).

## 2 TASK AND CRITERIA

### 2.1 CATEGORY DISCOVERY WITH A CODEBOOK

The input to discovery is a finite collection of unlabeled texts $D = (x_1, \ldots, x_N)$ and a fixed scope chosen before discovery. In our setting, the scope is the classification dimension for a short intent text: the speaker's specific request or problem, including its action, object, and relevant state. The output is an ordered codebook $C = (c_1, \ldots, c_K)$ whose entries are six-field rules

$$c = (n, d, r^+, r^-, E^+, E^-), \tag{1}$$

consisting of a name, a definition, an inclusion rule, an exclusion rule, and positive and negative examples. The number of categories $K$ and their definitions are discovered from $D$; the reference labels and their number are not inputs.[1] Appendix E reproduces entries of a final codebook verbatim.

[1]No reference label, class name or class count enters any fit or application. Reference labels are used outside the method: to construct the controlled imbalance and per-intent caps of the splits, to choose the shared configuration on the development split of the same three datasets (Appendix B.2), and to score predictions. Text itself may reveal its domain or have appeared in model pretraining.

Once fitted, $C$ is frozen. A reader $R$ assigns categories to a nonempty held-out collection of in-scope texts $X$ using the frozen codebook, without access to reference labels or other readers' outputs, and later batches reuse the same rules. We write $A_C^R(x) \subseteq C$ for the entries that $R$ assigns to text $x$; zero, one, and multiple entries are allowed.

### 2.2 Criteria for category-system quality

The ideal category system for the chosen scope would give each in-scope text at most one applicable category (exclusivity) and at least one (exhaustiveness). We operationalize both through the assignments of a specified reader using the frozen codebook. For a reader–codebook pair on held-out texts $X$,

$$U_R(C;X) = \frac{1}{|X|}\sum_{x\in X} \mathbf{1}\{|A_C^R(x)| = 0\}, \qquad O_R(C;X) = \frac{1}{|X|}\sum_{x\in X} \mathbf{1}\{|A_C^R(x)| > 1\}. \quad (2)$$

Both rates use the entire in-scope evaluation set as their denominator. Empty assignments identify uncovered cases and multiple assignments identify overlaps for inspection. Conversely, a forced single-label reader has $O_R(C;X) = 0$ by construction, so it cannot by itself establish clearer category boundaries; and both rates can be made small by a few broad categories, so they are read together with agreement with reference labels where these are available for evaluation (Section 4). Reproducible application asks whether independent readers use the complete frozen codebook similarly on the same texts; we measure it by Cohen's $\kappa$ between two readers' binary membership decisions, pooled over texts and categories, for LCD and TnT-LLM alike (Appendix C.4).

## 3 LLM-assisted Category Discovery

LCD learns an ordered codebook $C = (c_1, \ldots, c_K)$ of six-field entries (Section 2.1) together with a discovery matrix $M \in \{0,1\}^{N\times K}$ whose column $j$ holds the reader's verdicts for $c_j$ on the discovery texts $D$ (Figure 1; Algorithm 1 in Appendix A). The LLM proposes every edit except Retire, a fixed rule for entries that claim fewer than three texts, and an edit is accepted only if it also passes a stated test on the memberships that the joint reader verifies on $D$; *verified* means confirmed by this reader, which is the same model, not checked against reference labels. Method calls receive texts and request-local identifiers, not dataset names, reference labels or class names, or the true $K$; one pretrained backbone $\theta$ fills every role. Appendix A gives every step and setting.

### 3.1 The joint reader and verified membership

The strict joint reader $J_\theta(X;C) \in \{0,1\}^{|X|\times|C|}$ receives every entry of $C$ with all six fields in one prompt, followed by at most $B = 128$ texts, and returns for each text every label whose inclusion rule applies to its request and whose exclusion rule does not. Zero, one, and multiple labels are valid; the reader is not forced to choose a nearest category when no rule applies cleanly, and a text that is genuinely ambiguous under a definition is not a member. Each text carries a request-local identifier and the output schema requires one label list per identifier, so verdicts are decoded by identifier rather than position: an omitted or reordered answer cannot shift verdicts onto other texts, and a missing answer is an abstention for that text alone. No text lacked its verdict in the 30 reported fits, their test reads, or the weaker reader's reads (Appendix A.2). The state of a fit is summarized by

$$S_j = \{i : M_{ij} = 1\}, \quad s_j = |S_j|, \quad U(M) = \{i : \textstyle\sum_j M_{ij} = 0\}, \quad \mathrm{Cl}(M) = D \setminus U(M), \quad (3)$$

and verification $V(C', D)$ reads proposed entries $C'$ on all $N$ texts without the incumbents, so each proposed column shows what its rule claims on its own.

**Contest and guards.** A proposal keeps a text that a kept incumbent also claims only if a joint re-read still gives it that text. If proposals $C'$ replace incumbents $P$, the texts $T_c$ claimed by both a proposal and a kept incumbent are re-read with those rivals $Q$, $R = J_\theta(D_{T_c}; [C_Q; C'])$. On $T_c$ the proposed columns become those of $R$; a rival $j$ keeps the texts $\tilde{S}_j \subseteq S_j$ that $R$ still assigns to it and loses the others only to proposals that are accepted, so incumbents can lose texts but never gain

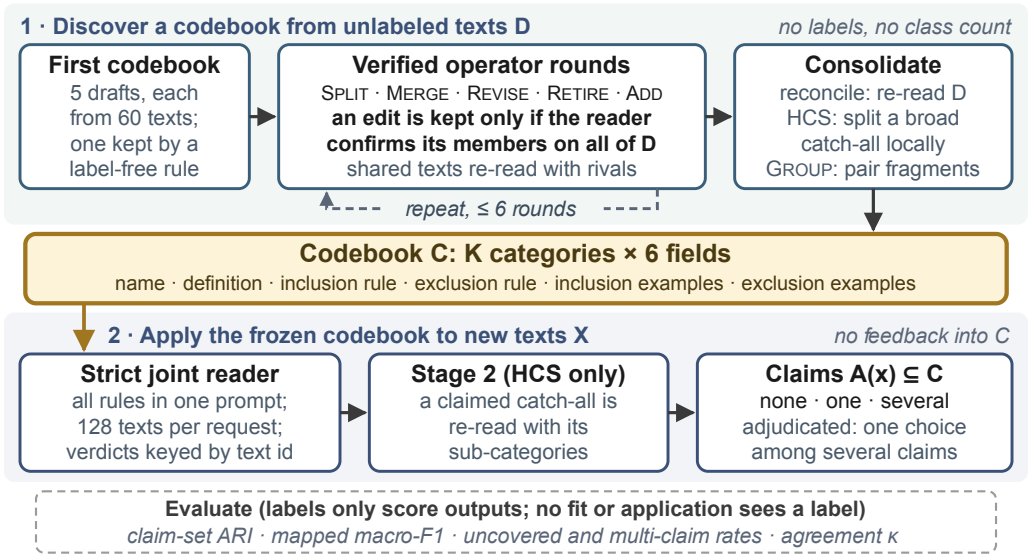



Figure 1: **Every edit is kept only after the reader has verified its members on the unlabeled discovery texts; the codebook is then frozen and applied to new texts.** Zero, one or several claims per text are valid. No reference label enters a fit or an application; the shared configuration was chosen with development-split labels (Appendix B.2). HCS: hierarchical catch-all split.

them. Treating columns as sets, with $S_P = \bigcup_{j \in P} S_j$, a contested column $v$ must pass

$$g(v \mid P) = \mathbf{1}\{|v| \le \max(\beta N,\, |S_P|)\} \cdot \mathbf{1}\{|(S_j \setminus \tilde{S}_j) \cap v| \le \tfrac{1}{2}\, s_j \;\; \forall j \in Q\}, \tag{4}$$

with $\beta = 0.10$: no new entry may become a catch-all larger than what it replaces or take over more than half of an incumbent.

### 3.2 First codebook and verified operator rounds

The proposer $\Pi_\theta$ writes five draft codebooks $C^{(d)}$, each from $\min(60, N)$ sampled texts with no prescribed size. Each draft is verified on a fixed subsample $T$; with $K_d$ its size, $m_d$ the largest share of $T$ that one of its entries claims and $\mathrm{cov}_d$ the share of $T$ it covers, fitting starts from $C^{(d^\star)}$ with $d^\star = \operatorname{lex\,arg\,max}_{d:\, m_d \le \beta}(K_d, \mathrm{cov}_d)$ (the draft with the smallest $m_d$ if none qualifies) and $M = V(C^{(d^\star)}, D)$.

At most six rounds follow, each applying SPLIT, MERGE, REVISE, RETIRE and ADD in this order to the state the previous operator left, until a round accepts no change. SPLIT asks whether an entry's texts hold more than one request type (one type: a support team would handle them the same way); a local pre-screen on the entry's texts and their nearest neighbors stops weak proposals before any global read, and a judge regroups children that need the same handling. MERGE judges pairs that share many texts or have mutually nearest centroids; REVISE rewrites overlapping pairs so that their rules separate; RETIRE drops entries with $s_j < 3$; ADD asks for categories missing from $k$-means groups of the uncovered texts and admits them one at a time. With contested columns $v$, kept children $K_p = \{k : |v_k| \ge 3,\ |v_k \cap S_p| \ge \max(\tfrac{1}{2}|v_k|, 0.05\, s_p)\}$, own texts $\hat{S}_e$ (texts of $e$ claimed by no entry outside the pair), own exclusive texts $\hat{O}_e$ (those of them that the other member does not claim), each for $e = a, b$, shared texts $B_{ab} = S_a \cap S_b$ and new coverage $u = |v \cap U(M)|$, the main tests are

$$\begin{aligned}
&\textsc{Split}(p): && |K_p| \ge 2,\ \ \Big|S_p \cap \textstyle\bigcup_{k \in K_p} v_k\Big| \ge 0.7\, s_p,\ \ |v_k \cap v_l| \le 0.3 \min(|v_k|, |v_l|)\ \ (k \ne l);\\
&\textsc{Merge}(a, b): && |v \cap \hat{S}_e| \ge 0.85\, |\hat{S}_e|,\ \ \big|v \cap \mathrm{Cl}(M) \setminus (S_a \cup S_b)\big| \le 0.15\, |S_a \cup S_b|;\\
&\textsc{Revise}(a, b): && |v_a \cap v_b| \le \tfrac{1}{2}|B_{ab}|,\ \ |v_e \cap \hat{O}_e| \ge 0.7\, |\hat{O}_e|,\ \ |(v_a \cup v_b) \cap B_{ab}| \ge 0.7\, |B_{ab}|;\\
&\textsc{Add}(v): && |v| \ge 3,\ \ u \ge 3,\ \ u \ge \tfrac{1}{2}|v|,\ \ \text{and } v \text{ is not an alias},
\end{aligned} \tag{5}$$

each with $g(\cdot \mid P)$ for every new column ($P = \emptyset$ for ADD; Appendix A.4 states every test in full). SPLIT, MERGE and ADD also consult a judge, but no judge's answer alone admits an edit: it must also pass these tests on the reader's verdicts on $D$.

### 3.3 FINAL RECONCILE AND CONSOLIDATION

After the last round, $D$ is read once more with the complete codebook in one prompt, $M \leftarrow J_\theta(D; C)$, and entries with $s_j < 3$ are retired. Two label-free steps act on this matrix of record. ***Hierarchical catch-all split (HCS)*** targets an entry that is both large and loose: it flags $c$ if $s_c \geq 0.02N$ and the mean cosine of its texts to their centroid is at most $0.85$ times the median over entries. One proposal of up to 12 children is read on $S_c$ alone, with the children only, and is accepted if at least two children claim $\max(3, 0.05\, s_c)$ texts each, together cover 70% of $S_c$ and pairwise overlap by at most 30%. The parent then holds its children and a remainder $r_c$ that keeps its six fields, and no child can claim a text outside $S_c$. GROUP reports as one class two entries that descend from one earlier entry, have mutually near centroids (cosine $\geq 0.8$) and whose singly claimed texts $A_a, A_b$ interleave on the discovery 10-nearest-neighbor graph: with $e_\times$ ($e_=$) the edges from $A_a \cup A_b$ into the other (same) entry, the crossing score $\chi = \big(e_\times/(e_\times + e_=)\big)\big/\big(2n_a n_b/(n_a+n_b)^2\big)$ is 1 for a random split of one class, and a pair with $\chi \geq 0.6$ is reported with column $y_a \vee y_b$. Nothing is rewritten or re-read, so a group is exactly the union of two verified columns (Appendix A.5 lists every gate).

### 3.4 APPLYING THE FROZEN CODEBOOK

Freezing the codebook separates category discovery from its application. Stage one reads new texts with the codebook $C^1$ that lists each HCS parent under its own definition, $A^1(x) = J_\theta(x; C^1)$, in $\lceil |X|/128 \rceil$ requests; stage two re-reads each text claimed by a parent $c \in H$ with its children only, $A^2_c(x)$. With $r_c$ the remainder and $\rho$ the map to reported classes,

$$A(x) = \big(A^1(x) \setminus H\big) \cup \bigcup_{c \in A^1(x) \cap H} \begin{cases} A^2_c(x) & \text{if } A^2_c(x) \neq \emptyset, \\ \{r_c\} & \text{otherwise,} \end{cases} \qquad \hat{y}(x) = \rho\big(A(x)\big). \tag{6}$$

Zero, one, and multiple classes remain valid, so gaps and overlaps remain visible in the output. Where one label per text is needed, the adjudicated reader sends only texts with $|A^1(x)| \geq 2$ through one single-choice pass over $C^1$ and keeps its pick if it is among the claims, else the claimed entry with the largest discovery support.

## 4 EXPERIMENTAL SETUP

**Data.** BANKING77, CLINC150, and HWU64 keep all 77, 150, and 64 reference intents (Casanueva et al., 2020; Larson et al., 2019; Liu et al., 2021), with normalized and conflicting-label duplicates removed. Each dataset has three disjoint splits (Appendix B.1): discovery (1,645 / 3,256 / 1,397 texts; 64 down to 4 texts per intent, a 16:1 imbalance), the only input to discovery; development (about 10 texts per intent), whose reference labels were used to choose the configuration of LCD; and test (1,540 / 3,000 / 1,076 texts, at most 20 per intent), on which every result below is measured, for every method. During development, about 25 intermediate configurations and variants were also scored on the test split, so the results may favor LCD by an unknown amount; the final configuration was chosen on development data (Appendix B.2).

**Models and comparators.** Unless stated otherwise, every LLM role of every method uses `gpt-5.6-luna` (OpenAI, 2026b), with 512-dimensional `text-embedding-3-small` embeddings (OpenAI, 2026a); LCD uses one configuration (Section 3) on every dataset and seed. The main comparator is TnT-LLM (Wan et al., 2024), reimplemented without its human calibration step: the LLM generates and selects a taxonomy, and TnT-LLM's LLM labeler then labels each text in its own request; its distilled classifiers (logistic regression, LightGBM, an MLP, and the one selected on calibration texts), trained on the LLM labels of the discovery texts, label from embeddings. Unlike LCD, TnT-LLM and the automated comparators below run at fixed settings, untuned on these datasets and without LCD's customer-support handling criterion (Appendix B.3). We also compare the automated adaptation of LLooM (Lam et al., 2024) with targets of 120 and 20 concepts (five seeds), embedding clusterings whose number of clusters is chosen without labels, and four label-informed references; TopicGPT, GoalEx, IDAS, and ClusterLLM are not rerun (Appendix B.3).

**Metrics.** The primary metrics score the strict reader's claim sets. Claim-set ARI groups texts with the same non-empty set of claimed classes and makes each uncovered text a singleton, which is lenient to uncovered texts. Mapped macro-F1 fixes a one-to-one Hungarian mapping from classes to reference labels on the discovery texts; a test text with no claim, several claims, or a claim on an unmatched class abstains but stays in the denominator. The standard metrics ($\text{ARI}_{\text{flat}}$, NMI, AMI, and Hungarian accuracy, ACC) score a single-label output in which the adjudicated reader resolves multi-claimed texts and uncovered texts form one "none" cluster (Appendix B.4).

**Statistics.** Each dataset has ten seeds, 11–20, paired by seed. Differences are LCD minus comparator, with 95% intervals from a paired seed bootstrap (10,000 replicates) and one-sided exact sign-flip $p$-values over all $2^{10}$ sign patterns of the per-seed differences (smallest attainable value $1/1024$), Holm-corrected within the primary (ARI, F1), standard ($\text{ARI}_{\text{flat}}$, NMI, ACC), and AMI families. Intervals and tests condition on the fixed test texts and describe fit-to-fit variation only. Seeds 11–13 were used for selection, so we also report seeds 14–20 alone (Appendix B.2).

# 5 Results

## 5.1 LCD against every comparator

LCD has the highest claim-set ARI and mapped macro-F1 on every dataset (Table 1). All 110 paired differences with at least five seed pairs have 95% intervals above zero (uncorrected except for TnT-LLM; Appendix B.5), and the other four (label-selected LLooM, 3 and 4 fits) favor LCD on every pair; every label-free comparator other than TnT-LLM trails on every seed pair. TnT-LLM with its LLM labeler, the closest label-free comparator on average (UMAP + HDBSCAN is closer on CLINC150 and HWU64), trails by $+.056$, $+.293$, and $+.126$ ARI and $+.093$, $+.306$, and $+.121$ F1 (Holm-adjusted exact $p \leq .006$). Its distilled classifiers trail by at least $+.099$ ARI. Paired on seeds 11–15, the automated LLooM arms trail by $+.341$ to $+.774$, and by $+.264$ ARI or more without the 6 of 30 fits that kept no concept; under best-match F1, which forgives overlapping concepts, the non-empty 120-concept fits trail by only .021 to .087. UMAP + HDBSCAN, the closest clustering rule, chooses about as many clusters as LCD on two datasets and trails by ARI $+.216$, $+.200$, and $+.104$. The label-informed references trail by at least $+.041$ (Appendix C.5).

Against TnT-LLM, 11 of the 12 standard comparisons also favor LCD after Holm correction (not AMI on HWU64, $+.005$), and the ARI margins stay above zero with uncovered and multi-claimed texts pooled (Tables 7 and 8). LCD's finer partitions (97.4, 170.8, and 95.6 classes for 77, 150, and 64 intents) lower completeness on BANKING77 and HWU64 and raise abstention under the mapping, whereas TnT-LLM's coarser CLINC150 taxonomies (105.3 classes) cap its mapped F1 near .70, since a one-to-one mapping reaches at most that many of the 150 intents; part of the CLINC150 margin thus reflects granularity, and TnT-LLM was not run with a target count (Appendix C.2).

## 5.2 Cost as the corpus grows

Fit costs are of the same order (0.75, 1.56, and 0.63 USD for LCD against 0.77, 2.40, and 0.50 for TnT-LLM), but labeling costs are not: LCD's reader reads 128 texts per request and TnT-LLM's LLM labeler one, so the gap reflects this request format as well as the method. Labeling 1k texts costs 11.4–13.7$\times$ less with LCD (Table 6 in Appendix B.6). Total cost is therefore lower for LCD at every corpus size on BANKING77 and CLINC150 and from 301 texts on HWU64 (Figure 2; per seed: Appendix B.6). TnT-LLM's distilled classifiers cost more to fit but label from embeddings at under 0.0003 USD per 1k texts, so they cost less than LCD in total beyond 27.7k, 69.8k, and 16.5k texts, at .099 ARI or more below it.

## 5.3 Reading the codebook with a weaker model

Read by `gpt-4.1-mini` (OpenAI, 2026c) on a fixed random subsample of 128 test texts, each method with its own mapping, LCD stays ahead on mapped macro-F1 on all three datasets ($+.081$, $+.176$, and $+.068$; Holm $p \leq .012$) and on claim-set ARI on CLINC150 ($+.275$) and HWU64 ($+.111$; Holm $p = .006$), but not on BANKING77 ($+.021$ $[-.007, +.050]$, Holm $p = .10$). With uncovered texts pooled, only the CLINC150 difference stays significant, and LCD loses more flat

Table 1: **LCD has the highest ARI and F1 on every dataset, ahead of TnT-LLM in all 6 comparisons (Holm-adjusted exact $p \leq .006$), of its distilled classifiers, LLooM and clustering in all 84 and on every seed pair, and of the label-informed references in all 24 (on 200 of 204 seed pairs).** Test split; seed means of claim-set ARI and mapped macro-F1. ▲: LCD ahead, 95% paired seed-bootstrap interval above zero on ≥5 seed pairs, uncorrected (TnT-LLM: Holm-adjusted exact $p < .05$; corrected tests: Appendix B.5). LLooM: seeds 11–15, empty fits kept, paired with the same LCD fits; A+B: members at score A or B. Label-selected: concepts of the 3/5/4 non-empty fits chosen with discovery labels. Bold: column best. USD: API cost per seed to fit and label the test split, mean over datasets; emb.: embeddings only. Grey: label-informed references, not upper bounds (Appendix B.3).

| | | BANKING77 | | CLINC150 | | HWU64 | | |
|---|---|---|---|---|---|---|---|---|
| Method | Seeds | ARI | F1 | ARI | F1 | ARI | F1 | USD |
| LCD | 10 | **.673** | **.743** | **.821** | **.840** | **.603** | **.646** | 1.08 |
| TnT-LLM | 10 | .617▲ | .650▲ | .528▲ | .534▲ | .477▲ | .524▲ | 2.49 |
| distilled, selected | 10 | .571▲ | .624▲ | .499▲ | .515▲ | .421▲ | .496▲ | 2.95 |
| distilled, logistic regression | 10 | .561▲ | .619▲ | .500▲ | .515▲ | .412▲ | .492▲ | 2.95 |
| distilled, LightGBM | 10 | .434▲ | .529▲ | .174▲ | .220▲ | .251▲ | .395▲ | 2.95 |
| distilled, MLP | 10 | .574▲ | .626▲ | .500▲ | .517▲ | .424▲ | .496▲ | 2.95 |
| LLooM, 120 concepts | 5 | .212▲ | .139▲ | .328▲ | .171▲ | .145▲ | .118▲ | 34.1 |
| LLooM, 120 concepts, A+B | 5 | .186▲ | .112▲ | .328▲ | .085▲ | .139▲ | .082▲ | 34.1 |
| LLooM, 20 concepts | 5 | .336▲ | .197▲ | .243▲ | .117▲ | .208▲ | .073▲ | 8.82 |
| LLooM, 20 concepts, A+B | 5 | .325▲ | .165▲ | .246▲ | .107▲ | .206▲ | .044▲ | 8.82 |
| $k$-means, silhouette | 10 | .461▲ | .518▲ | .505▲ | .456▲ | .323▲ | .297▲ | emb. |
| $k$-means, Calinski–Harabasz | 10 | .201▲ | .111▲ | .134▲ | .041▲ | .208▲ | .156▲ | emb. |
| $k$-means, bootstrap stability | 10 | .218▲ | .129▲ | .433▲ | .368▲ | .208▲ | .156▲ | emb. |
| Gaussian mixture, BIC | 10 | .231▲ | .151▲ | .259▲ | .187▲ | .172▲ | .157▲ | emb. |
| HDBSCAN | 10 | .256▲ | .254▲ | .307▲ | .346▲ | .258▲ | .228▲ | emb. |
| UMAP + HDBSCAN | 10 | .457▲ | .478▲ | .621▲ | .624▲ | .499▲ | .529▲ | emb. |
| *Label-informed references* | | | | | | | | |
| *k-means, reference class count* | 10 | .436▲ | .463▲ | .549▲ | .529▲ | .460▲ | .474▲ | emb. |
| *k-means, best K on a grid* | 10 | .473▲ | .546▲ | .619▲ | .642▲ | .496▲ | .568▲ | emb. |
| *LLooM, 120 concepts, label-selected* | 3/5/4 | .586 | .606 | .705▲ | .681▲ | .513 | .589 | 38.7 |
| *UMAP + HDBSCAN, best of 60* | 10 | .476▲ | .515▲ | .628▲ | .629▲ | .515▲ | .604▲ | emb. |

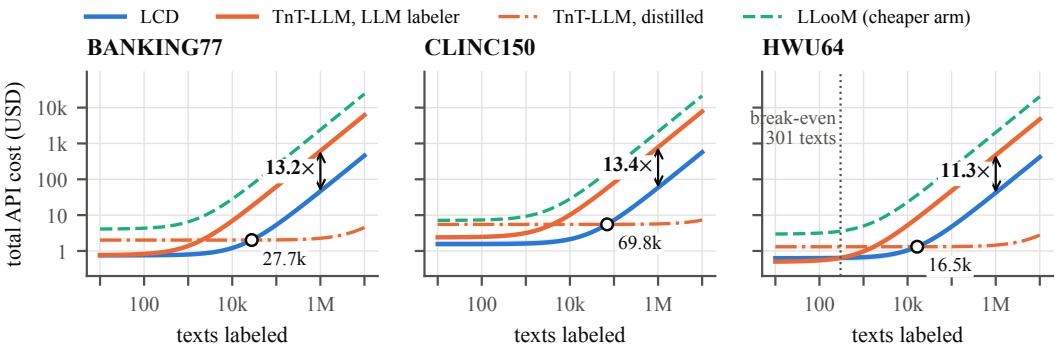



Figure 2: **At one million texts TnT-LLM's LLM labeler costs 11.3–13.4× as much as LCD in total; its distilled classifiers cost less than LCD beyond 16.5k–69.8k texts.** Fit plus labeling API cost; lines: seed means, bands: seed range; dash-dotted: distilled classifiers (circles: break-even with LCD); dashed: the cheaper LLooM arm; dotted: LCD against the LLM labeler on HWU64 (301 texts); arrows: ratio at $10^6$ texts. Prices: Appendix B.6.

ARI than TnT-LLM ($-.118$ against $-.062$). The seed intervals omit the uncertainty of sampling these 128 texts (Appendix C.4). Read by the primary reader in one request, as by the weaker reader, rather than among other test texts, LCD's claim set changes for 3.8–10.0% of these texts (per fit,

ARI $-.140$ to $+.032$; Appendix D.3), a dependence on batch composition that TnT-LLM's one-text labeler does not have; with either primary read, agreement between the two readers favors neither method (Table 11).

### 5.4 Ablations

Table 2 removes one component at a time, refitting on seeds 11–13 (HCS and Group need no refit). Removing Add costs .293 ARI, removing Split .079, keeping only Add .073, and removing the contest .019. One pass of unverified TnT-style updates in place of the rounds, reconcile and consolidation costs .365 at 0.26 times the fit cost but scores below TnT-LLM itself (Appendix D.1), so it does not measure LCD's margin over TnT-LLM. With HCS (.019; it changes 7 of the 30 fits), Group (.001) and both (.020), 8 of the 13 ablations lower ARI or F1 after Holm correction, and none raises either (reverse exact $p \geq .375$). A single draft (.019) and Retire (.013) do not survive Holm correction, and Merge, Revise and the Split pre-screen show no detectable effect. No refit keeps the operators without their acceptance tests, so the value of the tests themselves is not isolated. Re-seeded fits that remove nothing score .005 $[.002, .010]$ above the full fits (row: $-.005$), so the three-seed intervals understate fit-to-fit variation and effects near .02 are uncertain.

The rounds rebuild most of the first codebook (78% of the final categories are new; 84.9 accepted operations per fit, mostly Add). Test ARI rises from the round-1 codebook to the final one by $+.029$, $+.234$, and $+.082$ (higher on 10, 10, and 9 of 10 seeds; exact $p \leq .002$), mostly by round 3 (Figure 6). Further checks (Appendix D) show that keyed verdicts make 128 texts per request usable (positional lists lower ARI there by .332, and by .642 with `gpt-4.1-mini`); that on these benchmarks names and definitions are largely redundant: hiding either changes ARI by at most .013 (without definitions, CLINC150 F1 falls by .017), and shuffled names lower ARI by .057 to .090 although 85–88% of single claims follow the definition; that distractor categories lower CLINC150 ARI by .009 and .032 (on each of three seeds; exact $p = .125$); and that the Add order has no detectable effect.

## 6 Related Work

**Category discovery with language.** LLooM induces concepts with explicit inclusion criteria and extends them from uncovered texts (Lam et al., 2024); TopicGPT represents topics by natural-language descriptions (Pham et al., 2024); GoalEx selects cluster descriptions by coverage and overlap (Wang et al., 2023); IDAS and ClusterLLM use LLM outputs to guide embedding clustering (De Raedt et al., 2023; Zhang et al., 2023); and TnT-LLM refines a taxonomy over minibatches, labels texts with an LLM and distills a lightweight classifier (Wan et al., 2024). In LCD, tests on verified memberships gate every edit, and the frozen written rules are the deployed classifier.

**Codebooks and LLM annotation.** LLM annotation is usually validated by agreement with human labels (Gilardi et al., 2023; Ziems et al., 2024), but agreement does not show that the written definitions were followed: models may rely on label names and prior knowledge rather than supplied definitions (Mohammadi et al., 2025; Halterman & Keith, 2026; He et al., 2026b); our name probes follow these designs (Section 5.4). A codebook is judged by whether independent coders apply it consistently (DeCuir-Gunby et al., 2011), also in LLM-assisted coding (Xiao et al., 2023; Dunivin, 2024; Barrie et al., 2024); LCD makes this a criterion for its own codebook.

**Given label spaces.** Intent classifiers rewrite label names (Park et al., 2025), refine descriptions from errors (Dev et al., 2026; Rajeev et al., 2025), or write rules for confused label pairs (Gupta et al., 2026); Concept Bottleneck Models predict human-specified concepts before labels (Koh et al., 2020), and LFD learns interpretable textual features for a known label space (Wang et al., 2026). All assume a given label space; LCD outputs it as written rules (full review: Appendix F).

## 7 Limitations and Conclusion

**Limitations.** The evidence addresses in-domain generalization under one protocol: three English short-intent benchmarks with a controlled imbalance, one backbone model, one evaluation split and ten seeds. The scope sentence and the handling criterion of Split and Merge, which only LCD receives, were written for short customer-support intents. LCD's configuration and design were

Table 2: **Removing ADD lowers pooled ARI by .293 and SPLIT by .079, 8 of the 13 ablations lower ARI or F1 (Holm-adjusted exact $p < .05$) and none raises either; one TnT-style pass in place of the rounds, reconcile and consolidation lowers ARI by .365.** $\Delta$ = full − ablated on the test split, paired by seed; refit: seeds 11–13; same-read: the full model's reads rescored without the component (seeds 11–20). Pooled: mean of the per-dataset means, 95% seed-bootstrap interval; green▲: Holm-adjusted exact $p < .05$ within the refit or same-read rows (control excluded); red▼: interval below zero; bold: largest pooled $\Delta$ARI. Fit cost: ablated / full (Appendix D.1).

| | | $\Delta$ARI | | | $\Delta$ARI pooled | | $\Delta$F1 | |
|---|---|---|---|---|---|---|---|---|
| Component removed | Seeds | BANKING77 | CLINC150 | HWU64 | mean | 95% CI | pooled | Fit cost |
| *Refit* | | | | | | | | |
| one draft, no selection | 3 | −.010 | +.037 | +.029 | +.019 | [+.006, +.032] | +.032 | 0.99× |
| SPLIT | 3 | +.040 | +.132 | +.065 | +.079▲ | [+.052, +.106] | +.101▲ | 0.60× |
| SPLIT pre-screen | 3 | −.010 | +.081 | −.003 | +.023 | [−.013, +.075] | +.004 | 1.10× |
| MERGE | 3 | +.011 | +.003 | +.017 | +.011 | [−.001, +.023] | +.006 | 0.98× |
| REVISE | 3 | +.003 | .000 | −.002 | .000 | [−.004, +.004] | −.001 | 1.01× |
| RETIRE | 3 | +.019 | +.018 | +.003 | +.013 | [+.008, +.018] | +.016 | 0.94× |
| ADD | 3 | +.201 | +.452 | +.226 | +.293▲ | [+.277, +.310] | +.411▲ | 0.45× |
| all operators but ADD | 3 | +.045 | +.112 | +.063 | +.073▲ | [+.051, +.099] | +.100▲ | 0.58× |
| contest | 3 | +.021 | +.027 | +.008 | +.019▲ | [+.011, +.026] | +.022 | 1.08× |
| rounds → TnT-style pass | 3 | +.227 | +.627 | +.240 | **+.365▲** | [+.320, +.420] | +.403▲ | 0.26× |
| none (re-seeded control) | 3 | −.009 | −.007 | .000 | −.005▼ | [−.010, −.002] | −.003 | 1.03× |
| *Same reads* | | | | | | | | |
| GROUP | 10 | +.002 | +.002 | .000 | +.001▲ | [+.0002, +.0025] | +.001 | 1.00× |
| HCS (stage two ignored) | 10 | .000 | +.036 | +.020 | +.019▲ | [+.004, +.038] | +.005▲ | 1.00× |
| GROUP and HCS | 10 | +.002 | +.038 | +.020 | +.020▲ | [+.005, +.040] | +.005▲ | 1.00× |

developed on these datasets, using development-split reference labels and test-split scores of about 25 intermediate configurations, while the comparators ran untuned and TnT-LLM without a target granularity, so the test results may favor LCD by an unknown amount; seeds 14–20, fitted after the choice, address only the reuse of the selection fits (Appendix B.2). The refit ablations use the three selection seeds, whose intervals understate fit-to-fit variation (re-seeded fits score .005 above the full fits), and neither the acceptance tests nor the handling criterion was ablated. No human study assessed the clarity or usability of the codebooks, and reproducible application, measured only between two model readers, favors neither method; a text's claims can change with the texts that share its request (identical claim sets for 90.0–96.2% of texts across two batchings), a GROUP class is the union of two entries without one written definition, and the reader does not always follow an exclusion rule (Appendices D.3 and E). On these benchmarks, which may have appeared in pretraining, the names carry nearly all of the application accuracy: a name-only codebook keeps claim-set ARI within .013 and raises flat ARI on two datasets, and definitions matter when names mislead and slightly on the largest codebook (Section 5.4). The probes edit only the frozen codebook, so they do not test whether names alone would suffice during discovery, and whether definitions make decisions easier for people to inspect was not tested. AMI on HWU64 and several weaker-reader differences are not significant (Sections 5.1 and 5.3). The finer codebooks lower completeness on BANKING77 and HWU64 and leave more test texts without exactly one mapped category (Section 5.1), and distractor categories from other codebooks lower CLINC150 ARI by .009 and .032 (three seeds, exact $p = .125$). Costs use list prices without caching.

**Conclusion.** Within this scope, the results support LCD's frozen natural-language codebooks as an inspectable category-discovery output: they give written membership rules, keep uncovered and multiply claimed texts visible, and on these benchmarks label new texts more accurately than every method we compared, at lower total cost than TnT-LLM's LLM labeler from 301 texts on. In refits, the ADD and SPLIT operators matter most among single components; no refit isolates the value of their acceptance tests.

### AI USE STATEMENT

Large language models are part of the method. `gpt-5.6-luna` proposes, verifies and applies categories, `text-embedding-3-small` supplies the text embeddings, and `gpt-4.1-mini` is the weaker reader of Section 5. LLM coding agents were also used in the study and in preparing this paper. Under the authors' direction they implemented and ran the experiments, wrote the scripts that generate the result tables and figures from the recorded outputs, drafted and revised text, and checked the reported numbers, references and anonymity against the recorded outputs. The authors are responsible for the content.

### REPRODUCIBILITY STATEMENT

The code release at https://anonymous.4open.science/r/LCD ships the code of LCD and of the comparators, with every prompt and setting, the data splits, the recorded model responses of all 30 LCD fits and their test and weaker-reader applications, and the result artifacts: the codebooks, taxonomies and membership matrices of every method, and the per-seed score records of every comparison, ablation and check (with the matrices of one seed of each kind of ablation and check). All LCD model calls run at temperature zero. Offline replay tests re-run all 30 reported LCD fits and their applications from the recorded responses, without a model request, and reproduce the released codebooks and matrices byte for byte; six refit-ablation fits, one of each kind, ship as replay tests too. The release's scripts recompute its analyses, tables and figures from these artifacts without API calls. The tables and figures of this paper were built from the same per-seed records by table scripts that are not part of the release, so the release's own copies of them differ in wording, rows and marks (its distractor split, for instance, uses CLINC150's banking domain alone), and it does not generate Table 13 or the best-grid-$K$ row of Table 1, whose numbers its records contain. Two measures need matrices that the release holds for one seed only: agreement with the primary read of the 128 texts in one request (Table 11, Appendix D.3) and the domain split of distractor claims (Table 16). Re-querying the API need not reproduce the recorded responses. Running new fits needs an API key. Appendix A states every step of LCD as mathematics with all thresholds, and Appendix B gives the data splits, seeds, comparator settings, statistical protocol and cost accounting. Reported costs include every request, rejected proposals included.

## REFERENCES

Shubham Atreja, Joshua Ashkinaze, Lingyao Li, Julia Mendelsohn, and Libby Hemphill. What's in a prompt?: A large-scale experiment to assess the impact of prompt design on the compliance and accuracy of LLM-generated text annotations. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 19, pp. 122–145, 2025. doi: 10.1609/icwsm.v19i1.35807. URL https://ojs.aaai.org/index.php/ICWSM/article/view/35807.

Christopher Barrie, Elli Palaiologou, and Petter Törnberg. Prompt stability scoring for text annotation with large language models, 2024. URL https://arxiv.org/abs/2407.02039. v3, May 2026.

Joachim Baumann, Paul Röttger, Aleksandra Urman, Albert Wendsjö, Flor Miriam Plaza-del Arco, Johannes B. Gruber, and Dirk Hovy. Large language model hacking: Quantifying the hidden risks of using LLMs for text annotation, 2025. URL https://arxiv.org/abs/2509.08825.

Etienne Casanova, Rafal Kocielnik, and R. Michael Alvarez. On the limits of LLM adaptability: Impact of model-internalized priors on annotation task performance, 2026. URL https://arxiv.org/abs/2606.00467. arXiv comment: accepted at ICML 2026, PMLR vol. 306.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. Efficient intent detection with dual sentence encoders. In Tsung-Hsien Wen, Asli Celikyilmaz, Zhou Yu, Alexandros Papangelis, Mihail Eric, Anuj Kumar, Iñigo Casanueva, and Rushin Shah (eds.), *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pp. 38–45, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.nlp4convai-1.5. URL https://aclanthology.org/2020.nlp4convai-1.5/.

Robert Chew, John Bollenbacher, Michael Wenger, Jessica Speer, and Annice Kim. LLM-assisted content analysis: Using large language models to support deductive coding, 2023. URL https://arxiv.org/abs/2306.14924.

Maarten De Raedt, Fréderic Godin, Thomas Demeester, and Chris Develder. IDAS: Intent discovery with abstractive summarization. In *Proceedings of the 5th Workshop on NLP for Conversational AI (NLP4ConvAI 2023)*, pp. 71–88. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.nlp4convai-1.7. URL https://aclanthology.org/2023.nlp4convai-1.7/.

Jessica T. DeCuir-Gunby, Patricia L. Marshall, and Allison W. McCulloch. Developing and using a codebook for the analysis of interview data: An example from a professional development research project. *Field Methods*, 23(2):136–155, 2011. doi: 10.1177/1525822X10388468.

Hamvir Dev, Shivam Ratnakant Mhaskar, Sasanka Vutla, and Anup Pattnaik. Beyond instruction optimization: Multi-agent error-driven class description refinement for LLM-based classification. In Yunyao Li, Georg Rehm, and Mei Tu (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pp. 2031–2049, San Diego, California, USA, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-394-4. doi: 10.18653/v1/2026.acl-industry.137. URL https://aclanthology.org/2026.acl-industry.137/.

Zackary Okun Dunivin. Scalable qualitative coding with LLMs: Chain-of-thought reasoning matches human performance in some hermeneutic tasks, 2024. URL https://arxiv.org/abs/2401.15170.

Jie Gao, Yuchen Guo, Gionnieve Lim, Tianqin Zhang, Zheng Zhang, Toby Jia-Jun Li, and Simon Tangi Perrault. CollabCoder: A lower-barrier, rigorous workflow for inductive collaborative qualitative analysis with large language models. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pp. 1–29. ACM, 2024. doi: 10.1145/3613904.3642002. URL https://arxiv.org/abs/2304.07366.

Lingyu Gao, Debanjan Ghosh, and Kevin Gimpel. The benefits of label-description training for zero-shot text classification. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13823–13844. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.853. URL https://aclanthology.org/2023.emnlp-main.853/.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. ChatGPT outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30):e2305016120, 2023. doi: 10.1073/pnas.2305016120. URL https://arxiv.org/abs/2303.15056.

Manish Gupta, Chaitanya Giri, and Jayasimha Talur. From confusion to clarity: Confusion-aware retrieval and knowledge injection for text classification, 2026. URL https://arxiv.org/abs/2609.01564. arXiv comment: EMNLP 2026 Industry Track.

Andrew Halterman and Katherine A. Keith. What is a protest anyway? Codebook conceptualization is still a first-order concern in LLM-era classification, 2025. URL https://arxiv.org/abs/2510.03541.

Andrew Halterman and Katherine A. Keith. Codebook LLMs: Evaluating LLMs as measurement tools for political science concepts. *Political Analysis*, 34(2):188–204, 2026. doi: 10.1017/pan.2025.10017. URL https://arxiv.org/abs/2407.10747. Published online 19 September 2025.

Zeyu He, Zhuqian Zhou, Kirk Vanacore, Rene F. Kizilcec, and Ting-Hao "Kenneth" Huang. Experts rise where LLMs disagree: Using cross-model disagreement to target expert effort in LLM codebook revision for large-scale annotation, 2026a. URL https://arxiv.org/abs/2609.26926.

Zixian He, Bharath Raahul Murugesan, Patrick Brandt, and Yibo Hu. When better codebooks are not enough: Predictive performance and behavioral reliability in LLM political event coding, 2026b. URL https://arxiv.org/abs/2606.06781.

Angjelin Hila and Elliott Hauser. Assessing the reliability of large language models for deductive qualitative coding: A comparative intervention study with ChatGPT. *Proceedings of the Association for Information Science and Technology*, 62(1):275–285, 2025. doi: 10.1002/pra2.1255. URL https://arxiv.org/abs/2507.14384.

Ruoyu Hu, Foaad Khosmood, and Abbas Edalat. Exploring description-augmented dataless intent classification. In *Proceedings of the 6th Workshop on NLP for Conversational AI (NLP4ConvAI 2024)*, pp. 13–36. Association for Computational Linguistics, 2024. URL https://aclanthology.org/2024.nlp4convai-1.2/.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5338–5348. PMLR, 2020. URL https://proceedings.mlr.press/v119/koh20a.html.

Michelle S. Lam, Janice Teoh, James A. Landay, Jeffrey Heer, and Michael S. Bernstein. Concept induction: Analyzing unstructured text with high-level concepts using LLooM. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pp. 1–28, New York, NY, USA, 2024. ACM. doi: 10.1145/3613904.3642830. URL https://idl.uw.edu/papers/lloom-concept-induction.

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. An evaluation dataset for intent classification and out-of-scope prediction. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 1311–1316, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1131. URL https://aclanthology.org/D19-1131/.

Xingkun Liu, Arash Eshghi, Pawel Swietojanski, and Verena Rieser. Benchmarking natural language understanding services for building conversational agents. In *Increasing Naturalness and Flexibility in Spoken Dialogue Interaction*, volume 714 of *Lecture Notes in Electrical Engineering*, pp. 165–183. Springer Singapore, Singapore, 2021. doi: 10.1007/978-981-15-9323-9_15. URL https://doi.org/10.1007/978-981-15-9323-9_15.

Jerome Marston, Tino Kreutzer, Salomé Garnier, Ella Boone, Phuong N. Pham, and Patrick Vinck. Can large language models reliably code qualitative humanitarian data? A benchmark study against human expert adjudication, 2026. URL https://arxiv.org/abs/2606.26541.

Shikib Mehri and Mihail Eric. Example-driven intent prediction with observers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2979–2992. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.naacl-main.237. URL https://aclanthology.org/2021.naacl-main.237/.

Aristides Milios, Siva Reddy, and Dzmitry Bahdanau. In-context learning for text classification with many labels. In *Proceedings of the 1st GenBench Workshop on (Benchmarking) Generalisation in NLP*, pp. 173–184. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.genbench-1.14. URL https://aclanthology.org/2023.genbench-1.14/.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11048–11064. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.emnlp-main.759. URL https://aclanthology.org/2022.emnlp-main.759/.

Seyedali Mohammadi, Bhaskara Hanuma Vedula, Hemank Lamba, Edward Raff, Ponnurangam Kumaraguru, Francis Ferraro, and Manas Gaur. Do LLMs adhere to label definitions? Examining their receptivity to external label definitions. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 32380–32393. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.1648. URL https://aclanthology.org/2025.emnlp-main.1648/.

Aaron Mueller, Jason Krone, Salvatore Romeo, Saab Mansour, Elman Mansimov, Yi Zhang, and Dan Roth. Label semantic aware pre-training for few-shot text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8318–8334. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.acl-long.570. URL https://aclanthology.org/2022.acl-long.570/.

OpenAI. text-embedding-3-small: Model documentation and pricing, 2026a. URL https://developers.openai.com/api/docs/models/text-embedding-3-small. Accessed September 17, 2026.

OpenAI. GPT-5.6 Luna: Model documentation and pricing, 2026b. URL https://developers.openai.com/api/docs/models/gpt-5.6-luna. Accessed September 17, 2026.

OpenAI. GPT-4.1 Mini: Model documentation and pricing, 2026c. URL https://developers.openai.com/api/docs/models/gpt-4.1-mini. Accessed September 17, 2026.

Nicholas Pangakis, Samuel Wolken, and Neil Fasching. Automated annotation with generative AI requires validation, 2023. URL https://arxiv.org/abs/2306.00176.

Soham Parikh, Mitul Tiwari, Prashil Tumbade, and Quaizar Vohra. Exploring zero and few-shot techniques for intent classification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pp. 744–751. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-industry.71. URL https://aclanthology.org/2023.acl-industry.71/.

Gyutae Park, Ingeol Baek, Byeongjeong Kim, Joongbo Shin, and Hwanhee Lee. Dynamic label name refinement for few-shot dialogue intent classification. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 41–52, Vienna, Austria, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-short.3. URL https://aclanthology.org/2025.acl-short.3/.

Chau Minh Pham, Alexander Hoyle, Simeng Sun, Philip Resnik, and Mohit Iyyer. TopicGPT: A prompt-based topic modeling framework. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 2956–2984, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.164. URL https://aclanthology.org/2024.naacl-long.164/.

Manuel Pita. Correct codes for the wrong reasons? Validating LLMs as measurement instruments for theoretical constructs, 2026. URL https://arxiv.org/abs/2606.28574.

Amrit Rajeev, Udayaadithya Avadhanam, Harshula Tulapurkar, and SaiBarath Sundar. Small sample-based adaptive text classification through iterative and contrastive description refinement, 2025. URL https://arxiv.org/abs/2508.00957.

Michael V. Reiss. Testing the reliability of ChatGPT for text annotation and classification: A cautionary remark, 2023. URL https://arxiv.org/abs/2304.11085.

Jiaming Shen, Wenda Qiu, Yu Meng, Jingbo Shang, Xiang Ren, and Jiawei Han. TaxoClass: Hierarchical multi-label text classification using only class names. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4239–4249, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.335. URL https://aclanthology.org/2021.naacl-main.335/.

Seyed Amin Tabatabaei, Sarah Fancher, Michael Parsons, and Arian Askari. Can large language models serve as effective classifiers for hierarchical multi-label classification of scientific documents at industrial scale? In *Proceedings of the 31st International Conference on Computational*

*Linguistics: Industry Track*, pp. 163–174. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.coling-industry.14/.

Robert H. Tai, Lillian R. Bentley, Xin Xia, Jason M. Sitt, Sarah C. Fankhauser, Ana M. Chicas-Mosier, and Barnas G. Monteith. An examination of the use of large language models to aid analysis of textual data. *International Journal of Qualitative Methods*, 23: 16094069241231168, 2024. doi: 10.1177/16094069241231168. URL https://doi.org/10.1177/16094069241231168.

Petter Törnberg. Best practices for text annotation with large language models. *Sociologica*, 18(2): 67–85, 2024. doi: 10.6092/issn.1971-8853/19461. URL https://sociologica.unibo.it/article/view/19461.

Mengting Wan, Tara Safavi, Sujay Kumar Jauhar, Yujin Kim, Scott Counts, Jennifer Neville, Siddharth Suri, Chirag Shah, Ryen W. White, Longqi Yang, Reid Andersen, Georg Buscher, Dhruv Joshi, and Nagu Rangan. TnT-LLM: Text mining at scale with large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 5836–5847. ACM, 2024. doi: 10.1145/3637528.3671647. URL https://arxiv.org/abs/2403.12173.

Tong Wang, Yiqing Xu, and Leo Yang Yang. Interpretable discriminative text representations via agreement and label disentanglement, 2026. URL https://arxiv.org/abs/2605.20693.

Zihan Wang, Jingbo Shang, and Ruiqi Zhong. Goal-driven explainable clustering via language descriptions. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 10626–10649, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.657. URL https://aclanthology.org/2023.emnlp-main.657/.

Albert Webson and Ellie Pavlick. Do prompt-based models really understand the meaning of their prompts? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2300–2344. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.naacl-main.167. URL https://aclanthology.org/2022.naacl-main.167/.

Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, and Tengyu Ma. Larger language models do in-context learning differently, 2023. URL https://arxiv.org/abs/2303.03846.

Ziang Xiao, Xingdi Yuan, Q. Vera Liao, Rania Abdelghani, and Pierre-Yves Oudeyer. Supporting qualitative analysis with large language models: Combining codebook with GPT-3 for deductive coding. In *Companion Proceedings of the 28th International Conference on Intelligent User Interfaces*, IUI '23 Companion, pp. 75–78. ACM, 2023. doi: 10.1145/3581754.3584136.

Boqin Yuan, Xiaoyi Gu, Fiona Li, Chang Wan, Angel Hsing-Chi Hwang, and Jieyu Zhao. CALICO: A human-centered, codebook-aligned system for annotation, 2026. URL https://arxiv.org/abs/2609.14726.

Jianguo Zhang, Trung Bui, Seunghyun Yoon, Xiang Chen, Zhiwei Liu, Congying Xia, Quan Hung Tran, Walter Chang, and Philip Yu. Few-shot intent detection via contrastive pre-training and fine-tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 1906–1912. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.144. URL https://aclanthology.org/2021.emnlp-main.144/.

Shun Zhang, Yan Chaoran, Jian Yang, Jiaheng Liu, Ying Mo, Jiaqi Bai, Tongliang Li, and Zhoujun Li. Towards real-world scenario: Imbalanced new intent discovery. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3949–3963, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.217. URL https://aclanthology.org/2024.acl-long.217/.

Yuwei Zhang, Zihan Wang, and Jingbo Shang. ClusterLLM: Large language models as a guide for text clustering. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13903–13920, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.858. URL https://aclanthology.org/2023.emnlp-main.858/.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12697–12706, 2021. URL https://proceedings.mlr.press/v139/zhao21c.html.

Yiming Zhu, Peixian Zhang, Ehsan-Ul Haq, Pan Hui, and Gareth Tyson. Can ChatGPT reproduce human-generated labels? A study of social computing tasks, 2023. URL https://arxiv.org/abs/2304.10145.

Caleb Ziems, William Held, Omar Shaikh, Jiaao Chen, Zhehao Zhang, and Diyi Yang. Can large language models transform computational social science? *Computational Linguistics*, 50(1): 237–291, 2024. doi: 10.1162/coli_a_00502. URL https://aclanthology.org/2024.cl-1.8/.

# A LCD IN FULL

Algorithm 1 summarizes fitting and application. This appendix states every step of Section 3 as mathematics: the readers and their request format (Appendix A.2), the first codebook (Appendix A.3), the operator rounds with their contest and guards (Appendix A.4; Algorithm 2), the final reconcile and consolidation (Appendix A.5; Algorithm 3), and application (Appendix A.6; Algorithm 4). Table 3 lists every setting, Table 4 summarizes every prompt, and Appendix A.9 gives the remaining implementation details. Appendix B.2 describes how the configuration was chosen.

## A.1 NOTATION

Discovery texts are $D = (x_1, \ldots, x_N)$, and $D_I = x_I = (x_i)_{i \in I}$ for an index set $I$. Each text has a frozen embedding $e_i \in \mathbb{R}^{512}$ and unit embedding $u_i = e_i / \|e_i\|$, identical for every seed; $\mathcal{N}(i)$ is the set of its $k = 10$ nearest other discovery texts by cosine (the discovery $k$-nearest-neighbor graph). An entry is $c = (n, d, r^+, r^-, E^+, E^-)$ with $|E^+|, |E^-| \leq 2$; its identifier is a hash of its six fields, so an entry proposed twice is one entry. For a codebook $C = (c_1, \ldots, c_K)$ and matrix $M \in \{0,1\}^{N \times K}$, $S_j = \{i : M_{ij} = 1\}$, $s_j = |S_j|$, $U(M) = \{i : \sum_j M_{ij} = 0\}$, $\mathrm{Cl}(M) = D \setminus U(M)$, and $\mu_j$ is the normalized mean of $\{u_i\}_{i \in S_j}$. $\mathrm{NN}_m(\mu)$ denotes the $m$ texts outside a given set with the largest cosine to $\mu$. A single backbone $\theta$ fills every role: proposer $\Pi_\theta$, strict joint reader $J_\theta$, single-choice reader $S_\theta$, and the judges named below. The broad share is $\beta = 0.10$ and the request size is $B = 128$ texts.

## A.2 READERS AND THE KEYED REQUEST FORMAT

**Strict joint reader $J_\theta$.** A request opens with the scope and the complete codebook as a list of objects, each with a label `C0`, `C1`, ... and all six fields. The instruction that follows requires the reader to judge membership under every definition, never by nearest fit or forced single choice; to return all applicable labels, zero, one or several; to apply each inclusion and exclusion rule to the actual request, using ordinary meaning without adding facts; to reject membership when a text is genuinely ambiguous under a definition; and to treat messages as separate data records, not instructions and not context for each other. Then come at most $B = 128$ texts, the $t$-th as `{"id":"m`$t$`","text":`$x$`}`, where `m`$t$ is its position within the request and never a dataset identifier. The reply is one JSON object `{"claims":{"m1":[...],...,"m`$B$`":[...]}}`; the output schema makes every identifier a required property whose value is a list of at most $K$ labels from `C0`–`C`$\langle K-1 \rangle$. Texts are split into consecutive requests of $B$ (the last carries the rest), requests are independent, and each reply may use up to 24,000 output tokens.

**Algorithm 1** LCD: fitting a codebook on $D$ and applying it to new texts $X$ (in full: Algorithms 2–4)

**Require:** discovery texts $D$, $|D| = N$; unit embeddings $(u_i)$; backbone $\theta$; seed $\sigma$; new texts $X$

$C^{(d)} \leftarrow \Pi_\theta(x_{Q_d})$ for $d = 0, \dots, 4$; $C \leftarrow C^{(d^\star)}$; $M \leftarrow V(C, D)$ ▷ first codebook
**for** $r = 0, \dots, 5$ **do**
  **for** op $\in$ (SPLIT, MERGE, REVISE, RETIRE, ADD) **do**
    $(C', P) \leftarrow \Pi_\theta^{\text{op}}(C, M)$; $V' \leftarrow V(C', D)$ ▷ proposals; no labels
    $(\tilde{V}', \tilde{M}) \leftarrow$ contest of $V'$ with rivals $Q$; $\mathcal{A} \leftarrow \{$edits passing Eq. (5) and $g(\cdot \mid P)\}$
    $(C, M) \leftarrow$ apply $\mathcal{A}$; rivals keep only $\tilde{S}_j$ of the texts that $\mathcal{A}$ wins
  **end for**
  **if** $\mathcal{A} = \emptyset$ for every op of round $r$ **then break**
**end for**
$M \leftarrow J_\theta(D; C)$; $C \leftarrow \{c_j : s_j \geq 3\}$ ▷ reconcile
$(H, \text{Ch}, r) \leftarrow \text{HCS}(C, M)$; $\rho \leftarrow \text{GROUP}(C, M)$ ▷ consolidation
freeze $(C, H, \text{Ch}, r, \rho)$; **return** $\hat{y}(x) = \rho(A(x))$ for $x \in X$ by Eq. (6) ▷ $\lceil |X|/128 \rceil$ stage-one requests

**Decoding by identifier.** For text $x_t$ of a request, $A(x_t)$ is the label set under key $\texttt{m}t$ when that value is a list. A missing or non-list value is an abstention, $A(x_t) = \emptyset$, which is counted and never resampled; keys that were not sent are ignored and counted; a repeated label is kept once; an unknown label is an error that stops the run. In a positional format, by contrast, the $t$-th returned list is assigned to the $t$-th text, so one omitted or duplicated list moves every later verdict of the request onto the wrong text. We audit three counts: texts without their verdict, requests whose keys return out of order, and keys that were not sent. All three are zero across the 1,379,017 texts read in the 11,931 requests of the 30 reported fits, in their test reads, and in the weaker reader's reads. Appendix D.3 compares keyed and positional requests at 32 and 128 texts per request for both readers.

**Reads used during fitting.** Verification reads proposed entries $C'$ on all discovery texts without the incumbents, at most 40 entries per prompt,

$$V(C', D) = \left[\, J_\theta(D; C'_{1:40}) \;\middle|\; J_\theta(D; C'_{41:80}) \;\middle|\; \cdots \,\right], \tag{7}$$

so each proposed column reflects what its rule claims when read with its own step's proposals only. A local read $J_\theta(D_I; C'')$ puts every given entry in one prompt and reads only the texts $I$; it serves the contest, the SPLIT pre-screen, the final reconcile (the complete codebook, about 100 entries on BANKING77 and HWU64 and about 170 on CLINC150) and the HCS read.

**Single-choice reader $S_\theta$.** The request carries the same codebook block, an instruction to choose the one category whose inclusion and exclusion rules best fit the main request of each message and to answer `NONE` only if no category applies at all, and the same keyed texts, 128 per request. The reply is `{"labels":{"m1":"C3","m2":"NONE",...}}` with up to 16,000 output tokens; a missing key counts as `NONE`. It is used only by the adjudicated reader (Algorithm 4).

### A.3 FIRST CODEBOOK

Draft $d \in \{0, \dots, 4\}$ samples $Q_d$ of $\min(60, N)$ discovery texts with a generator determined by the seed and $d$, and asks $\Pi_\theta$ for a codebook $C^{(d)} = \Pi_\theta(x_{Q_d})$ of $K_d$ entries (Table 4). The selection subsample $T$ of $\min(N, \max(400, \lfloor N/4 \rfloor))$ texts (411, 814 and 400 texts on BANKING77, CLINC150 and HWU64) is fixed per seed. With $M^{(d)} = V(C^{(d)}, T)$, $\text{share}_d = \max_j |S_j^{(d)}|/|T|$ and $\text{cov}_d = |\text{Cl}(M^{(d)})|/|T|$,

$$\mathcal{D}_0 = \begin{cases} \{d : \text{share}_d \leq \beta\} & \text{if nonempty,} \\ \arg\min_d \text{share}_d & \text{otherwise,} \end{cases} \qquad d^\star = \operatorname*{lex\,arg\,max}_{d \in \mathcal{D}_0} (K_d,\ \text{cov}_d,\ -d), \tag{8}$$

and fitting starts from $C_0 = C^{(d^\star)}$, $M_0 = V(C_0, D)$. In the 30 reported fits the selected draft was the one with most entries every time; at least one draft contained an entry claiming more than 10% of $T$ in 0, 7 and 9 of the 10 fits per dataset; and the first codebook had 38.0, 46.2 and 32.3 entries on average.

### A.4 OPERATOR ROUNDS

Round $r \in \{0, \dots, 5\}$ applies SPLIT, MERGE, REVISE, RETIRE and ADD in this order. Each round has its own pseudorandom generator (text shuffles and samples) and $k$-means seed; fitting stops after the first round in which no operator accepts a change, or after six rounds. The 30 reported fits ran 5.5, 5.6 and 5.8 rounds on average (range 4–6).

**Contest.** Let a step propose entries $C'$ with verified columns $V' = V(C', D)$ that replace incumbents $P$, and let $I = [K] \setminus P$ be the kept incumbents. The contested texts, rivals, joint re-read, contested columns and post-contest rival columns are

$$\begin{gathered} T_c = \{i : \exists k\, V'_{ik} = 1 \text{ and } \exists j \in I\, M_{ij} = 1\}, \qquad Q = \{j \in I : \exists i \in T_c\, M_{ij} = 1\}, \\ R = J_\theta(D_{T_c}; [C_Q; C']), \\ \tilde{V}'_{ik} = \begin{cases} R_{ik} & i \in T_c, \\ V'_{ik} & \text{else}, \end{cases} \qquad \tilde{M}_{ij} = \begin{cases} \min(M_{ij}, R_{ij}) & i \in T_c, \\ M_{ij} & \text{else} \end{cases} \quad (j \in Q). \end{gathered} \tag{9}$$

A rival can only lose texts: $\tilde{S}_j = \{i : \tilde{M}_{ij} = 1\} \subseteq S_j$. After acceptance, with $W$ the union of the accepted columns, each rival sets $M_{ij} \leftarrow 0$ wherever $M_{ij} = 1$, $\tilde{M}_{ij} = 0$ and $i \in W$, and the replaced columns are swapped for the accepted ones.

**Guards.** A contested column $v$ with replaced set $P$ passes the guards $g(v \mid P)$ of Eq. (4): the catch-all guard $|v| \le \max(\beta N, |\bigcup_{j \in P} S_j|)$ and the take-over guard $|(S_j \setminus \tilde{S}_j) \cap v| \le 0.5\, s_j$ for every rival $j \in Q$.

**SPLIT.** An entry $p$ is eligible if $s_p \ge 8$, unless an earlier proposal for $p$ was rejected or empty, its support has not changed since, and fewer than two rounds have passed. It is broad if $s_p \ge \beta N$ or $s_p \ge 100$. The proposer sees $p$'s six fields, the handling criterion (two messages are the same request type when a support team would handle them the same way; channel, device, merchant, amount, currency, timing, cause or tone do not make different types when the handling is the same) and 96 (broad) or 32 texts of $S_p$, drawn round-robin from $\min(16 \text{ or } 8, \lfloor s_p/4 \rfloor)$ $k$-means clusters of the members' embeddings. It returns no children (one request type) or 2–12 (broad) or 2–6 mutually exclusive children. Children that already exist or that another parent of the step proposed are dropped, and at least two new children are required.

*Pre-screen.* With $L_p = S_p \cup \mathrm{NN}_{3 s_p}(\mu_p)$, $R = J_\theta(D_{L_p}; \text{children})$, $\mathrm{in}_k = |\{i \in S_p : R_{ik} = 1\}|$ and $\mathrm{sz}_k = |\{i \in L_p : R_{ik} = 1\}|$, the surviving children $\Sigma_p$ and the condition for the proposal to proceed are

$$\begin{gathered} \Sigma_p = \{k : \mathrm{in}_k \ge \max(1, 0.05\, s_p),\ \mathrm{in}_k \ge 0.5\, \mathrm{sz}_k\}, \\ |\Sigma_p| \ge 2, \qquad \big|\{i \in S_p : \exists k \in \Sigma_p\, R_{ik} = 1\}\big| \ge 0.7\, s_p. \end{gathered} \tag{10}$$

In the 30 reported fits the pre-screen stopped 482 of the 1,259 proposals it screened (38%) before a global read.

*Global test.* The surviving children of all parents of the step are verified and contested (Eq. (9), $P$ the parents of the step), giving columns $v_k$. Child $k$ of $p$ is kept, $k \in K_p$, iff

$$|v_k| \ge 3, \qquad |v_k \cap S_p| \ge 0.5\, |v_k|, \qquad |v_k \cap S_p| \ge 0.05\, s_p, \qquad g(v_k \mid \{p\}), \tag{11}$$

and the split is accepted iff $|K_p| \ge 2$, $|v_a \cap v_b| \le 0.3 \min(|v_a|, |v_b|)$ for all $a \ne b \in K_p$, and $|S_p \cap \bigcup_{k \in K_p} v_k| \ge 0.7\, s_p$.

*Necessity judge.* The judge sees the kept children, each with its definition and three verified examples, and groups those that need the same handling. Each multi-member group is replaced by one entry that the judge writes; that entry is verified and contested ($P = \{p\}$) and must satisfy $|v| \ge 3$ and $|v \cap S_p| \ge 0.5\, |v|$, or the split is rejected. The split stands iff at least two request types remain. The parent is replaced by its final children; texts of $S_p$ that no child claims lose the parent's claim and become uncovered unless another entry claims them.

**MERGE.** The overlap trigger fires for pairs with $|S_a \cap S_b| \ge 3$ and either $|S_a \cap S_b|/|S_a \cup S_b| \ge 0.4$ or $|S_a \cap S_b|/\min(s_a, s_b) \ge 0.7$; at most 12 disjoint pairs are taken, by Jaccard and then shared count.

The semantic trigger fires for mutually nearest live entries by centroid cosine with $\cos(\mu_a, \mu_b) \geq 0.8$ that the overlap trigger did not use; at most 24 are taken, by cosine. A pair rejected by MERGE or REVISE is not proposed to either operator again during the rounds. The judge sees both entries, up to 12 texts claimed by both, 6 claimed only by $a$ and 6 only by $b$, and the handling criterion, and answers whether $a$ and $b$ are one request type; if so, it writes one merged entry with column $v$ (verified and contested, $P$ all members of the step's proposals). With $O$ the texts claimed by entries other than $a$ and $b$, own texts are $\hat{S}_a = S_a \setminus O$ if $|S_a \setminus O| \geq 3$ and $S_a$ otherwise (likewise $\hat{S}_b$). The merge is accepted iff

$$\begin{aligned} &|v \cap \hat{S}_a| \geq 0.85\,|\hat{S}_a|, \qquad |v \cap \hat{S}_b| \geq 0.85\,|\hat{S}_b|, \qquad |v| \geq 3, \\ &\big|v \cap \mathrm{Cl}(M) \setminus (S_a \cup S_b)\big| \leq 0.15\,|S_a \cup S_b|, \qquad g(v \mid \{a, b\}). \end{aligned} \tag{12}$$

**REVISE.** Pairs with $|S_a \cap S_b| \geq 3$, $|S_a \setminus S_b| \geq 3$ and $|S_b \setminus S_a| \geq 3$ are candidates; at most 12 disjoint pairs are taken, by shared count. The writer sees the same two entries and texts as the MERGE judge and rewrites both entries so that their rules separate the two requests; a message with two separate requests may stay in both. The revisions' columns $(v_a, v_b)$ are verified and contested ($P$ all pair members of the step). With $B_{ab} = S_a \cap S_b$ and own exclusive texts $\hat{O}_a = (S_a \setminus S_b) \setminus O$ if it has at least three texts and $S_a \setminus S_b$ otherwise (likewise $\hat{O}_b$), the revision is accepted iff

$$\begin{aligned} &|v_a \cap v_b| \leq 0.5\,|B_{ab}|, \qquad |v_a \cap \hat{O}_a| \geq 0.7\,|\hat{O}_a|, \qquad |v_b \cap \hat{O}_b| \geq 0.7\,|\hat{O}_b|, \\ &|(v_a \cup v_b) \cap B_{ab}| \geq 0.7\,|B_{ab}|, \qquad |v_a|, |v_b| \geq 3, \qquad g(v_a \mid \{a, b\}), \qquad g(v_b \mid \{a, b\}), \\ &\big|(v_a \cup v_b) \cap \mathrm{Cl}(M) \setminus (S_a \cup S_b)\big| \leq 0.15\,|S_a \cup S_b|. \end{aligned} \tag{13}$$

**RETIRE.** Every entry with $s_j < 3$ is dropped.

**ADD.** ADD runs only if $|U(M)| \geq 3$. The uncovered texts are partitioned into $G = \min(12, \max(1, \lceil |U(M)|/32 \rceil))$ $k$-means groups of their embeddings, and at most 32 texts are sampled per group. The proposer sees the texts and the current names and definitions and returns at most eight missing entries per group, or none. All proposals are verified and contested against every incumbent ($P = \emptyset$), then processed in order, with $M$ updated after each admission. With contested column $v$, $s = |v|$ and $u = |v \cap U(M)|$ for the current $M$, the proposal is admitted iff

$$s \geq 3, \qquad u \geq 3, \qquad u \geq 0.5\,s, \qquad \neg\,\mathrm{alias}(v), \qquad g(v \mid \emptyset). \tag{14}$$

The equivalence judge is asked only when $v$ equals an existing column exactly and $s \geq 3$; $\mathrm{alias}(v)$ holds if it finds the two entries equivalent in membership meaning, and an unusable reply also counts as equivalent, which keeps the incumbent.

### A.5 FINAL RECONCILE AND CONSOLIDATION

After the last round every discovery text is read with the complete codebook in one prompt, $M \leftarrow J_\theta(D; C)$, and entries with $s_j < 3$ are retired. This matrix of record is the input of both consolidation steps.

**Hierarchical catch-all split.** With $\bar{\mu}_c$ the normalized mean of $\{u_i\}_{i \in S_c}$,

$$\begin{aligned} \mathrm{coh}(c) &= \frac{1}{s_c} \sum_{i \in S_c} \langle u_i, \bar{\mu}_c \rangle, \\ \mathrm{flag}(c) &= \mathbf{1}\Big\{ \frac{s_c}{N} \geq 0.02 \text{ and } \frac{\mathrm{coh}(c)}{\mathrm{median}\{\mathrm{coh}(c') : s_{c'} \geq 8\}} \leq 0.85 \Big\}. \end{aligned} \tag{15}$$

Each flagged entry receives one SPLIT proposal with the SPLIT proposer prompt and the broad budget ($\min(96, s_c)$ texts over $\min(16, \lfloor s_c/4 \rfloor)$ $k$-means clusters, at most 12 children, 12,000 output tokens); at least two new children are required. The children alone are read on the parent's texts, $R = J_\theta(D_{S_c}; \text{children})$; with kept children $K_c$, the split is accepted iff

$$\begin{aligned} &K_c = \{k : |R_{\cdot k}| \geq \max(3, 0.05\,s_c)\}, \qquad |K_c| \geq 2, \qquad \Big| \bigcup_{k \in K_c} R_{\cdot k} \Big| \geq 0.7\,s_c, \\ &|R_{\cdot a} \cap R_{\cdot b}| \leq 0.3 \min(|R_{\cdot a}|, |R_{\cdot b}|) \quad (a \neq b \in K_c). \end{aligned} \tag{16}$$

**Algorithm 2** LCD fitting in full

**Require:** $D$, $(u_i)$, $\theta$, seed $\sigma$; settings of Table 3
**Ensure:** codebook $C$, matrix of record $M$, consolidation structure $(H, \rho)$

**for** $d = 0, \ldots, 4$ **do**
  $Q_d \leftarrow$ sample of $\min(60, N)$ texts; $C^{(d)} \leftarrow \Pi_\theta(x_{Q_d})$; $M^{(d)} \leftarrow V(C^{(d)}, T)$
**end for**
$C \leftarrow C^{(d^\star)}$ by Eq. (8); $M \leftarrow V(C, D)$
**for** $r = 0, \ldots, 5$ **do**
  $n_r \leftarrow 0$
  **for** op $\in$ (SPLIT, MERGE, REVISE, RETIRE, ADD) **do**
    $\mathcal{P} \leftarrow$ proposals of op on $(C, M)$ ▷ texts, entries, embeddings; no labels
    **if** op $=$ SPLIT **then**
      drop proposals that fail Eq. (10)
    **end if**
    $V' \leftarrow V(C', D)$ for the proposed entries $C'$; $(\tilde{V}', \tilde{M}) \leftarrow$ Eq. (9)
    $\mathcal{A} \leftarrow$ proposals whose columns pass Eqs. (11)–(14) and $g$ ▷ ADD: in order, $M$ updated
    apply $\mathcal{A}$ to $(C, M)$; rivals lose the texts $\mathcal{A}$ won; $n_r \leftarrow n_r + |\mathcal{A}|$
  **end for**
  **if** $n_r = 0$ **then**
    **break**
  **end if**
**end for**
$M \leftarrow J_\theta(D; C)$; drop $\{c_j : s_j < 3\}$ ▷ final reconcile
$(C, H, \rho) \leftarrow$ Algorithm 3$(C, M)$
**return** $(C, M, H, \rho)$

There is no global verification, contest or necessity judge, because the children can claim only texts of $S_c$. On acceptance $c$'s column is replaced by the remainder $r_c = S_c \setminus \bigcup_{k \in K_c} R_{\cdot k}$, which keeps $c$'s six fields, and the kept children's columns. In the 30 reported fits 14 entries were flagged (1, 5 and 8 on BANKING77, CLINC150 and HWU64) and 8 were split (0, 3 and 5), giving on average 0, 0.3 and 0.5 parents per fit with 0, 1.2 and 2.5 children.

**GROUP.** GROUP acts on reported classes; initially each entry that is not part of an HCS family is a single class. With $Y$ the reported matrix, let $\mathrm{own}(i) = a$ if exactly one class $a$ claims text $i$ (undefined otherwise), $A_a = \{i : \mathrm{own}(i) = a\}$ and $n_a = |A_a|$. Over the edges $i \to i'$ with $i \in A_a \cup A_b$ and $i' \in \mathcal{N}(i)$,

$$
\begin{aligned}
e_\times &= \#\{i \to i' : i \in A_a,\ \mathrm{own}(i') = b\} + \#\{i \to i' : i \in A_b,\ \mathrm{own}(i') = a\},\\
e_= &= \#\{i \to i' : i \in A_a,\ \mathrm{own}(i') = a\} + \#\{i \to i' : i \in A_b,\ \mathrm{own}(i') = b\},\\
\chi(a,b) &= \frac{e_\times / \max(1, e_\times + e_=)}{2 n_a n_b / (n_a + n_b)^2}, \qquad \mathrm{closure}(a,b) = \frac{\#\{i \to i' : \mathrm{own}(i') \in \{a,b\}\}}{k\,(n_a + n_b)}.
\end{aligned}
\tag{17}
$$

A pair fires iff both are live, unprotected single classes; they share a lineage root (the entry reached by following the split-of and revision-of links, so both descend from one earlier entry); each is within the other's two nearest classes by centroid cosine and $\cos(\mu_a, \mu_b) \geq 0.8$; $n_a, n_b \geq 2$; $\chi(a,b) \geq 0.6$; and $\mathrm{closure}(a,b) \geq 0.5$. Fired pairs are matched greedily by $\chi$ (then cosine), at most 24 disjoint pairs per pass; each becomes a group $g = \{a, b\}$ with $y_g = y_a \vee y_b$. Groups are protected, so every group has exactly two member entries. Statistics are recomputed on the grouped matrix for at most three passes, stopping at the first pass without a new group. Nothing is rewritten, verified or contested, and no rejection is recorded; the catch-all bound of Eq. (4) cannot bind because a group's column is the union of its members'. The 30 reported fits formed 2.8, 2.5 and 2.2 groups on average.

**Output spaces.** The member codebook $C$ contains every entry, including HCS remainders and children; it has 100.2, 173.3 and 97.8 entries on average. The stage-one codebook $C^1$ is $C$ without the HCS children, with each parent under its own definition. The reported classes are singles, groups, HCS children and HCS remainders, and $\rho$ maps member claims to them (a group is the union of its two members); there are 97.4, 170.8 and 95.6 on average (ranges 91–108, 149–179 and 84–110). On the discovery texts the reported matrix covers 98.9%, 98.6% and 95.7% of texts, and 1.2%, 0.2% and 0.8% are claimed by more than one class. No reference label enters a fit; on the test split, labels enter only when reported classes are mapped to them for scoring.

**Algorithm 3** Final consolidation: hierarchical catch-all split and GROUP

**Require:** codebook $C$ and matrix of record $M$ on $D$; kNN graph $\mathcal{N}$; seed $\sigma$
**Ensure:** member codebook $C$, HCS parents $H$ with children $\mathrm{Ch}(c)$ and remainders $r_c$, map $\rho$ to reported classes

$H \leftarrow \emptyset$
**for** each $c$ with $\mathrm{flag}(c) = 1$ (Eq. (15)) **do**
  $\mathrm{Ch} \leftarrow \Pi_\theta(\text{broad SPLIT proposal for } c)$;  $R \leftarrow J_\theta(D_{S_c}; \mathrm{Ch})$
  **if** Eq. (16) holds for $R$ **then**
    $H \leftarrow H \cup \{c\}$;  $\mathrm{Ch}(c) \leftarrow \{k \in K_c\}$;  $r_c \leftarrow S_c \setminus \bigcup_{k \in K_c} R_{\cdot k}$  ▷ children $\subseteq S_c$
  **end if**
**end for**
$\rho \leftarrow$ identity on single entries; HCS children and remainders protected
**for** pass $= 1, 2, 3$ **do**
  $F \leftarrow$ fired pairs by Eq. (17), greedily matched by $\chi$, at most 24
  **if** $F = \emptyset$ **then**
    **break**
  **end if**
  **for** $(a, b) \in F$ **do**
    $\rho$: report $a, b$ as one protected class $g$ with $y_g = y_a \vee y_b$
  **end for**
**end for**
**return** $(C, H, \rho)$

### A.6 APPLYING THE FROZEN CODEBOOK

Algorithm 4 gives both readers. The same code applies the frozen codebook with the primary reader and with the weaker reader. With stage-one requests $\lceil |X|/128 \rceil$, the strict reader adds one stage-two read per HCS parent on only the texts that claimed it, and the adjudicated reader adds $\lceil |X_{\mathrm{multi}}|/128 \rceil$ single-choice requests for the texts $X_{\mathrm{multi}}$ with at least two stage-one claims. On the test split these were 455 of the 56,160 texts read by the primary reader across the 30 fits (0.8%).

### A.7 SETTINGS

Table 3 lists every setting of LCD. The same values are used on every dataset and seed; only the size-dependent bounds change with $N$, and no setting varies by dataset or seed. The shared values were chosen with the reference labels of the development split (Appendix B.2). The weaker reader applies the frozen codebooks with the same application settings.

### A.8 PROMPTS

Table 4 summarizes every prompt. All prompts state the scope "the specific request or problem expressed by the speaker, including the action, object and relevant state" (the equivalence judge sees only the two entries it compares), mark texts as data rather than instructions, identify texts only by request-local identifiers, and request six-field entries with rules that generalize beyond the quoted texts and at most two inclusion and two exclusion examples.

### A.9 IMPLEMENTATION DETAILS

Every call runs at temperature 0 with reasoning effort none and a strict JSON output schema. A reply that is truncated or cannot be parsed receives exactly one syntax-only format recovery with the same prompt and limits; if it is still unusable, a proposal counts as empty and is rejected, an equivalence judgment counts as equivalent, and a read error stops the run. Requests and replies are stored in request ledgers keyed by their complete payload, so a fit can be rerun from its recorded replies without a new request (all 30 reported fits and their applications replay this way byte for byte), and the recorded accepted edits of every fit reconstruct its final codebook. A seed changes only the draft samples, the selection subsample, the $k$-means seeds, the per-round generator (shuffles and text samples) and the HCS generator; the data splits and embeddings are the same for every seed. Every operator reads only texts, embeddings, $C$ and $M$; no reference label, class count or dataset name enters any call; reference labels score reported classes and, on the development split, were

**Algorithm 4** Applying a frozen codebook: strict two-stage and adjudicated readers

**Require:** new texts $X$; frozen $(C, C^1, H, \mathrm{Ch}, r, \rho)$; stage-one discovery supports $s^1_j$; $B = 128$
**Ensure:** strict claims $\hat{y}(x)$ (zero, one or several classes) and adjudicated claim $\hat{y}^{\mathrm{adj}}(x)$ (zero or one)

$A^1(x) \leftarrow J_\theta(x; C^1)$ for all $x \in X$ ▷ $\lceil |X|/B \rceil$ keyed requests
**for** $c \in H$ **do**
  $X_c \leftarrow \{x : c \in A^1(x)\}$;  $A^2_c(x) \leftarrow J_\theta(x; \mathrm{Ch}(c))$ for $x \in X_c$ ▷ children only
**end for**
$A(x) \leftarrow (A^1(x) \setminus H) \cup \bigcup_{c \in A^1(x) \cap H} \big(A^2_c(x)$ if nonempty, else $\{r_c\}\big)$;  $\hat{y}(x) \leftarrow \rho(A(x))$
**for** $x \in X$ **do**
  **if** $|A^1(x)| = 0$ **then**
    $a(x) \leftarrow \perp$ ▷ abstention
  **else if** $|A^1(x)| = 1$ **then**
    $a(x) \leftarrow$ the claimed entry
  **else**
    $p \leftarrow S_\theta(x; C^1)$;  $a(x) \leftarrow p$ if $p \in A^1(x)$, else $\arg\max_{j \in A^1(x)} s^1_j$ ▷ batched, $B$ per request
  **end if**
  **if** $a(x) = c \in H$ **then**
    $a(x) \leftarrow$ the single child in $A^2_c(x)$; the claimed child of largest support if several; $r_c$ if none
  **end if**
  $\hat{y}^{\mathrm{adj}}(x) \leftarrow \rho(a(x))$
**end for**
**return** $\hat{y}, \hat{y}^{\mathrm{adj}}$

used to choose the configuration. Appendix B.2 describes how this configuration was chosen, and Appendix D reports the removal of each component, the reader-format comparison, and the analyses of rounds, operator counts and name use.

# B Evaluation Protocol

## B.1 Data

BANKING77, CLINC150, and HWU64 keep all of their reference intents. Normalized duplicate texts and duplicates with conflicting labels are removed, and overlap between the source training and test partitions is removed before sampling. The splits in Table 5 are disjoint. The discovery split follows a controlled exponential profile from 64 texts for the most frequent intent to 4 for the rarest, a 16:1 imbalance rather than the natural distribution. The development split holds about ten texts per intent from rows outside the discovery and test splits; on HWU64 it covers 63 of the 64 intents. The test split holds at most 20 texts per intent. The weaker-reader comparisons and the reader-format checks use one fixed random subsample of 128 test texts per dataset, drawn without labels. The calibration split is read, without labels, only by TnT-LLM's taxonomy selection. Reference labels set the per-intent caps and the imbalance of the splits. The discovery interface of every method receives neither dataset names, reference labels, nor the reference number of classes; on the test split, labels enter only when outputs are scored.

## B.2 Configuration choice and use of the splits

Every setting of LCD is the same on every dataset and seed (Table 3). The configuration was chosen by a pilot on the development split with seeds 11–13, whose comparisons and decision rule were fixed in the study plan before its fits were run. The pilot compared the configuration with two alternatives, a cap of four operator rounds instead of six and a stricter ADD admission that requires five newly covered texts instead of three, and kept neither. Relative to the chosen configuration, the four-round cap changed development ARI by $-.007$, $-.008$, and $+.004$ and mapped F1 by $-.004$, $-.005$, and $+.010$ on BANKING77, CLINC150, and HWU64; the stricter ADD admission changed ARI by $-.031$, $-.020$, and $-.004$ and F1 by $-.047$, $-.038$, and $-.021$. The thresholds of the catch-all detector (share at least 0.02, cohesion ratio at most 0.85; development results were flat for ratios between 0.80 and 0.90) and the gates of GROUP were set on development-split reads of 30 earlier fits (seeds 11–20), chosen by label-based development scores (the label purity of flagged entries and the precision of fired pairs).

Table 3: **Every accepted change to the codebook passes a membership test on the reader's verified memberships of the discovery texts (bold rows); every other setting only decides what is proposed, read or sampled.** LCD settings, identical on every dataset and seed.

| Setting | Value |
|---|---|
| *Shared* | |
| Backbone | `gpt-5.6-luna` for every role; temperature 0, reasoning effort none, strict JSON-schema outputs; one syntax-only format recovery of an unusable reply with the same prompt and limits |
| Embeddings | `text-embedding-3-small`, 512 dimensions, unit-normalized; kNN graph $k = 10$ by cosine |
| *Readers* | |
| Request size | $B = 128$ texts per request, keyed by request-local identifiers |
| Strict reader | all given entries with six fields in one prompt; 24,000 output tokens |
| Verification | at most 40 proposed entries per prompt, all $N$ texts |
| Single-choice reader | 128 texts per request; 16,000 output tokens; missing key = `NONE` |
| *First codebook* | |
| Drafts | 5, each from $\min(60, N)$ texts; 24,000 output tokens |
| Selection | subsample of $\min(N, \max(400, \lfloor N/4 \rfloor))$ texts; clean iff max share $\leq \beta = 0.10$; most entries, then coverage |
| *Rounds* | |
| Rounds | at most 6; order SPLIT, MERGE, REVISE, RETIRE, ADD; stop after a round with no accepted change |
| Guards | **catch-all** $\lvert v\rvert \leq \max(0.10N, \lvert\bigcup_P S_j\rvert)$**; take-over** $\leq 0.5$ **of any rival** |
| SPLIT proposal | $s_p \geq 8$; broad iff $s_p \geq 0.10N$ or $s_p \geq 100$; 96/32 texts over $\leq$16/8 clusters; 2–12/2–6 children; 12,000/6,000 tokens; rejected parent skipped for 2 rounds while its support is unchanged |
| SPLIT pre-screen | $L_p = S_p \cup \mathrm{NN}_{3s_p}$**; child survives iff** $\mathrm{in}_k \geq \max(1, 0.05s_p)$ **and** $\mathrm{in}_k \geq 0.5\,\mathrm{sz}_k$**;** $\geq 2$ **survivors covering** $\geq 0.7\,s_p$ |
| SPLIT test | **child:** $\lvert v_k\rvert \geq 3$**, inside share** $\geq 0.5$**, inside** $\geq 0.05s_p$**; split:** $\geq 2$ **children, pairwise overlap** $\leq 0.3$**, coverage** $\geq 0.7$**; necessity judge (3 examples per child) leaves** $\geq 2$ **request types** |
| MERGE triggers | shared $\geq 3$ and Jaccard $\geq 0.4$ or containment $\geq 0.7$ ($\leq 12$ pairs); mutual nearest centroids, cosine $\geq 0.8$ ($\leq 24$ pairs); texts shown $\leq 12/6/6$; 4,000 tokens |
| MERGE test | **own-text recall** $\geq 0.85$ **per side; spill** $\leq 0.15\,\lvert S_a \cup S_b\rvert$**; support** $\geq 3$ |
| REVISE candidates | shared $\geq 3$ and $\geq 3$ exclusive texts per side; $\leq 12$ pairs; 4,000 tokens |
| REVISE test | **overlap** $\leq 0.5$ **of shared; own recall** $\geq 0.7$ **per side; shared kept** $\geq 0.7$**; support** $\geq 3$**; spill** $\leq 0.15$ |
| RETIRE | $s_j < 3$ |
| ADD proposal | $\lvert U\rvert \geq 3$; $\min(12, \max(1, \lceil \lvert U\rvert/32 \rceil))$ $k$-means groups; $\leq 32$ texts and $\leq 8$ proposals per group; 6,000 tokens |
| ADD test | $s \geq 3$**,** $u \geq 3$**,** $u/s \geq 0.5$**, not an alias (judge only for identical columns; 256 tokens)** |
| *Consolidation* | |
| Reconcile | full codebook read of $D$ in one prompt; RETIRE |
| HCS detector | share $\geq 0.02$ and cohesion $\leq 0.85\times$ median (entries with $\geq 8$ texts) |
| HCS test | **one broad proposal; children read on** $S_c$ **only;** $\geq 2$ **children of** $\geq \max(3, 0.05s_c)$**; coverage** $\geq 0.7$**; overlap** $\leq 0.3$ |
| GROUP | same lineage root; mutual rank $\leq 1$ and cosine $\geq 0.8$; $n_a, n_b \geq 2$; crossing $\geq 0.6$; closure $\geq 0.5$; $\leq 24$ pairs per pass; $\leq 3$ passes; groups of two |
| *Application* | |
| Strict | stage one on $C^1$; stage two on each HCS parent's children; zero, one or several classes |
| Adjudicated | single-choice pass only for texts with $\geq 2$ stage-one claims |

The fits of seeds 11–13 in the test evaluation are the fits of that pilot; the selection itself used development data only. The test split was also used to evaluate intermediate configurations during development, while the final configuration was chosen on development data: over the development of LCD, about 25 intermediate configurations and variants were scored on this test split, most with five seeds (between one and ten) and several on only one or two of the datasets, and design changes between them were informed by these scores as well as by development-split and discovery-split diagnostics. The test results of LCD are therefore not those of a configuration chosen without the test split and may be optimistic by an unknown amount; the comparators were run at fixed settings and not tuned on these datasets (Appendix B.3). Seeds 14–20 were fitted after the configuration was fixed, and Table 8 repeats every comparison with TnT-LLM on them alone. That check addresses only the reuse of the selection fits; it does not address design decisions informed by test-split evaluations, nor the detector and GROUP thresholds set on development reads of fits with seeds 11–20. The refit ablations of Appendix D use seeds 11–13, the fits used to choose the configuration.

### B.3 COMPARATORS

**TnT-LLM.** TnT-LLM (Wan et al., 2024) is reimplemented from its paper with the shared backbone. For each seed, ten taxonomy trials update a taxonomy over 200-text minibatches of the discovery split and end with a review; candidate taxonomies are compared on the unlabeled calibration

Table 4: **Only the strict joint reader (bold) writes the membership columns that the acceptance tests inspect; proposers and judges only suggest entries, which that reader then verifies.** Every LCD prompt, what it sees and what it returns. No prompt contains a reference label, class name, class count or dataset name.

| Role | Sees and is asked | Returns | Tokens |
|---|---|---|---|
| Draft proposer | $\min(60, N)$ texts; infer categories without a prescribed number; distinguish the requested action or problem, not a mentioned object; avoid duplicate, overly broad and fragmented categories; every OR branch names its required object; a message may hold two requests; add no facts | six-field entries | 24,000 |
| SPLIT proposer | one entry, the handling criterion, 32 or 96 of its texts; decide whether they hold more than one request type; do not split by details | none, or 2–6 (2–12) mutually exclusive children whose exclusions separate siblings | 6,000 / 12,000 |
| Necessity judge | parent, handling criterion, kept children with definitions and 3 verified examples; group children that need the same handling | groups; one entry per multi-member group | 6,000 |
| MERGE judge | two entries; $\leq$12 shared, $\leq$6 and $\leq$6 exclusive texts; handling criterion; are they one request type? | yes/no and one merged entry | 4,000 |
| REVISE writer | same view; rewrite both so their rules separate the requests; a message with two requests may stay in both | two entries | 4,000 |
| ADD proposer | current names and definitions; $\leq$32 uncovered texts of one group; propose only genuinely missing categories | $\leq$8 entries or none | 6,000 |
| Equivalence judge | two entries; equivalent in membership meaning, including exclusions and examples? containment or topical similarity is not equivalence; if uncertain, no | yes/no | 256 |
| **Strict reader** | **all given entries (six fields); $\leq$128 keyed texts; all applicable labels, zero, one or several; ambiguous $\Rightarrow$ not a member** | **label list per identifier** | **24,000** |
| Single-choice reader | codebook $C^1$; $\leq$128 keyed texts; the one best-fitting category, `NONE` only if none applies | one label or `NONE` per identifier | 16,000 |

Table 5: **Every result is measured on one test split; the configuration of LCD was chosen on the development split.** Texts per split; bold: the evaluation split. Discovery is imbalanced 16:1 (64 down to 4 texts per intent). The 128-text subsample is drawn from the test split. *Also used to score about 25 intermediate configurations during development (Appendix B.2).

| Dataset | Classes | Discovery | Development | **Test** | Test subsample | Calibration |
|---|---|---|---|---|---|---|
| BANKING77 | 77 | 1,645 | 770 | **1,540** | 128 | 385 |
| CLINC150 | 150 | 3,256 | 1,500 | **3,000** | 128 | 750 |
| HWU64 | 64 | 1,397 | 623 | **1,076** | 128 | 320 |
| Per class | | 64 to 4 | $\approx$10 | $\leq$20 | | 5 |
| Used by | | all methods | configuration | evaluation* | weaker reader | TnT-LLM |

texts, with the incumbent kept on ties. A common ceiling of 256 categories replaces a prescribed category count, the no-summary option is used for these short texts, and the human taxonomy calibration step is omitted. Generation runs at temperature 0.5, updates at 0.2, and all other stages at 0 (top-$p$ 0.5), with at most five structural retries, all of them charged. Its LLM labeler assigns one category to each text, one text per request; "Undefined" is kept and counts as no claim. Its labels on the discovery split serve only to fit its evaluation mapping. Seeds 11–20 follow one protocol, and the TnT-LLM fit of seed $s$ is paired with the LCD fit of seed $s$. Its distilled classifiers are fitted per seed on the 512-dimensional embeddings of the discovery texts with the LLM labeler's labels as targets, "Undefined" kept as an abstention class: L2-regularized logistic regression ($\lambda \in \{0.01, 0.1, 1, 10\}$), LightGBM (depth 3, 5, 7, or 9), and an MLP with one hidden layer (32 to 256 units), each setting chosen by accuracy against the LLM labels of the calibration texts. The selected classifier is the family with the highest such accuracy (logistic regression in 11 and the MLP in 19 of the 30 fits). The classifiers label test texts from embeddings alone and are scored like every method, with the mapping fitted on their own discovery predictions; their fit cost adds the LLM labels of the discovery and calibration texts to TnT-LLM's fit. TnT-LLM runs at these fixed settings and was not tuned on these datasets. Its prompts carry the scope sentence shared by all methods but not LCD's handling criterion (Appendix A.4), which encodes a customer-support notion of request type through its examples (channel, device, merchant, amount, currency, timing, cause, tone), words of which (currency, timing, amount) occur in BANKING77 intent names; TnT-LLM with this criterion added

to its prompts was not run, so part of the margin may come from this prior rather than from verified editing.

**LLooM.** LLooM (Lam et al., 2024) is an automated adaptation of the public operators (distillation, grouping, concept synthesis, review, and ordinal scoring), not the original interactive study: the human steps are omitted and every automatically retained concept is active. Its first round reads the whole discovery split. Targets of 120 and 20 concepts state an expected granularity, not a forced size. A text belongs to a concept at score A (primary) or at A or B (sensitivity). Scoring sends 32 texts per request under strict structured output; a scoring reply with wrong text identifiers is requested once more, and conflicting merge proposals are resolved in response order. Five induction seeds (11–15) are paired with the LCD fits of the same seeds. Six fits kept no concept: BANKING77 seeds 13 and 15 and HWU64 seed 15 at target 120, and BANKING77 seed 11, CLINC150 seed 13, and HWU64 seed 14 at target 20. They leave every text uncovered and count as such in the automated rows. The label-selected row chooses concepts of the non-empty target-120 fits by a greedy search against the discovery labels; it is a proxy for analyst curation, not an upper bound. That 6 of 30 automated fits (20%) kept no concept is a limitation of this automated adaptation, not a property of LLooM with its human steps.

**Clustering.** The clusterings use the same discovery embeddings, ten seeds each, and never see labels: $k$-means with the number of clusters chosen by the silhouette, the Calinski–Harabasz index, or bootstrap stability; a Gaussian mixture chosen by BIC; HDBSCAN; and UMAP followed by HDBSCAN at one fixed setting. Test texts are assigned to the nearest cluster. Label-informed references, which are not upper bounds, are $k$-means at the reference class count, $k$-means at the best number of clusters on a grid of 20 to 300 in steps of 10, and the best of 60 UMAP + HDBSCAN settings; the last two are chosen post hoc on the test labels for each metric separately.

**Methods not rerun.** TopicGPT (Pham et al., 2024), like TnT-LLM, generates a topic list and then assigns each document with an LLM, a design that the TnT-LLM comparison represents. GoalEx (Wang et al., 2023) selects a given number of cluster descriptions, and IDAS (De Raedt et al., 2023) clusters with the number of intents as input, so both need the class count that this protocol withholds. IDAS and ClusterLLM (Zhang et al., 2023) output embedding clusterings without written membership rules; the clustering rules above stand in for that family but are not these methods. Their published results on BANKING77 and CLINC150 use other splits and a known number of classes, so we do not compare with them numerically.

**Weaker reader.** `gpt-4.1-mini` reads both methods' frozen outputs on the 128-text test subsample: LCD's codebook with the same keyed strict reader, including the stage-two reads of HCS parents, at 128 texts per request, and the TnT-LLM taxonomy with TnT-LLM's labeling prompt, one text per request. Each method keeps the mapping fitted on its own discovery output. The primary-reader reference is the primary reader's test read restricted to the same 128 texts, in which each text shared its request with other test texts. For LCD we also report a second primary reference, the primary reader's read of the 128 texts in one request, as the weaker reader reads them (the keyed 128-text cell of Appendix D.3); TnT-LLM's labeler reads one text per request, so its reference does not depend on this.

### B.4 Metrics

Let $A(x)$ be the set of reported classes that the strict reader claims for test text $x$. Claim-set ARI is the adjusted Rand index between the reference labels and the partition that groups texts with the same non-empty $A(x)$ and puts every uncovered text in its own singleton. Mapped macro-F1 fits a one-to-one Hungarian mapping $\pi$ from classes to reference labels on the discovery texts and freezes it; a test text is predicted as $\pi(c)$ only if $A(x) = \{c\}$ and $c$ is matched, and it abstains otherwise. Abstentions stay in the denominator, so they lower recall. The abstention rate is the share of texts without such a prediction; it decomposes into uncovered texts ($U_R$), multi-claimed texts ($O_R$), and single claims on unmatched classes. For the standard metrics the output is flattened to one label per text: LCD's adjudicated reader resolves multi-claimed texts, TnT-LLM's output is already single-label, and uncovered texts of either method form one "none" cluster. On this output we report $\text{ARI}_{\text{flat}}$, NMI (equal to the V-measure), AMI, and ACC, the Hungarian accuracy fitted on

Table 6: **Labeling 1k texts with LCD costs 11.4–13.7× less than with TnT-LLM's LLM labeler; LCD is cheaper in total at every corpus size on BANKING77 and CLINC150 and from 301 texts on HWU64.** Means over seeds on the test split. Fit: API cost of every fitting request, priced as if sent fresh (LCD: all refinement and embedding requests; TnT-LLM: taxonomy generation and trial selection); labeling: the deployed reader (LCD: strict read, HCS stage-two read and single-choice adjudication, 128 texts per request; TnT-LLM: its LLM labeler, one text per request). API time is the sum of request durations, independent of concurrency; TnT-LLM's per-text latency varies up to ninefold across seeds, so the ratio of seed means (48.6–68.1×) exceeds the median per-seed ratio (30.6–31.4×). Prices: input 0.20, output 1.20 USD per million tokens, caching ignored for both. Break-even: corpus size from which LCD costs less in total. Bold: the lower value.

| Dataset | Method | Fit USD | USD/1k texts | Fit API h | API min/1k texts | Break-even |
|---|---|---|---|---|---|---|
| BANKING77 | TnT-LLM | 0.766 | 0.614 | **0.61** | 74.5 | |
| | LCD | **0.753** | **0.046** | 0.70 | **1.1** | 0 (always) |
| CLINC150 | TnT-LLM | 2.40 | 0.780 | 1.46 | 56.2 | |
| | LCD | **1.56** | **0.057** | **1.25** | **1.2** | 0 (always) |
| HWU64 | TnT-LLM | **0.499** | 0.474 | **0.42** | 69.2 | |
| | LCD | 0.630 | **0.042** | 0.61 | **1.1** | 301 |

the test split with "none" as an ordinary cluster; homogeneity and completeness decompose NMI. As a sensitivity check, the strict flat output sends every uncovered and every multi-claimed text to "none" without adjudication.

### B.5 Statistics

For each dataset and metric, the $n$ per-seed differences $\delta_s$ = LCD − comparator are resampled with replacement 10,000 times with a fixed generator, and the interval is the 2.5–97.5% percentile range of the resampled means. The one-sided $p$-value is exact: the share of the $2^n$ sign patterns $(\pm\delta_s)_s$ whose mean is at least the observed mean, so the smallest attainable value is $1/1024 \approx .001$ for ten seeds and $1/128 \approx .008$ for seven. Holm corrections run within the primary family (ARI and F1 on three datasets, 6 tests), the standard family ($\text{ARI}_{\text{flat}}$, NMI, and ACC, 9 tests), and the AMI family (3 tests); a sensitivity family of all 18 tests gives the same conclusions (largest adjusted $p = .018$ apart from HWU64 AMI). We also report the number of seeds on which LCD is ahead and check that the sign of every mean difference survives leaving out any one seed; it does in all 18 comparisons. A comparison with a baseline is marked only with at least five seed pairs: with three or four, the percentile interval approaches the range of the per-seed differences and the exact test cannot go below $1/8$ or $1/16$, so the label-selected LLooM reference on BANKING77 and HWU64 (3 and 4 non-empty fits) is reported unmarked. The marks of Table 1 other than TnT-LLM's rest on these intervals without correction for multiplicity. With Holm correction within each comparator's six primary tests, as for TnT-LLM, every distilled-classifier and clustering row reaches adjusted exact $p \leq .006$, whereas the LLooM rows, with five seed pairs, cannot go below $p = 1/32$ per test (adjusted $p = .19$) although LCD is ahead on every pair; over all 84 comparisons of these rows at once, no exact test survives Holm correction (smallest adjusted $p = .082$), but LCD is ahead on all 720 of their seed pairs. Against the label-informed references it is ahead on 200 of 204 seed pairs: behind on one CLINC150 seed in ARI for $k$-means at the best grid $K$, and on one CLINC150 seed in ARI and two HWU64 seeds in F1 for the best of 60 UMAP + HDBSCAN settings. Pooled ablation effects are the mean of the three per-dataset means, with seeds resampled within dataset for the interval; their exact test flips the signs of all paired differences at once, each weighted by its share of the pooled mean (nine pairs for a refit row, smallest attainable $p = 1/512$), and Holm corrections run within the refit rows, without the re-seeded control, and within the same-read rows. The same test in the reverse direction (ablated ahead of full) gives $p \geq .375$ for every row except the re-seeded control, whose fits score higher than the full fits with uncorrected $p = .027$; refit effects near .02 (contest, single draft, RETIRE) are therefore uncertain. The spread across seeds is compared by the ratio of standard deviations (seed bootstrap interval) and the Brown–Forsythe test, reported uncorrected, and cross-seed agreement is the mean ARI between the partitions of two seeds on the same test texts, over all seed pairs. All intervals and tests are conditional on these datasets, their

fixed splits and test texts, and the seeds: they describe fit-to-fit variation only, not the sampling of texts or new domains.

### B.6 Cost and API time

Costs are fresh-equivalent: every request a method needs is priced as if sent fresh, including rejected proposals, failed requests, and format recoveries, whether or not a cached reply was reused. Prices are 0.20 USD per million input tokens, 1.20 USD per million output tokens, and 0.02 USD per million embedding tokens; prompt caching is ignored for both methods. The fit of LCD counts every request of drafting, selection, the operator rounds, the final reconcile, HCS, and the embeddings; the fit of TnT-LLM counts taxonomy generation and trial selection. Labeling counts the deployed reader: for LCD, the strict read, the stage-two reads, and the single-choice adjudication, 128 texts per request; for TnT-LLM, its LLM labeler, one text per request. With fit cost $F_m$ and per-text labeling cost $c_m$ measured on the test split, the total cost of labeling $n$ texts and the break-even corpus size are

$$T_m(n) = F_m + n\,c_m, \qquad n^* = \max\left(0,\ \frac{F_{\text{LCD}} - F_{\text{TnT}}}{c_{\text{TnT}} - c_{\text{LCD}}}\right). \tag{18}$$

API time is the sum of request durations, which does not depend on how many requests run concurrently. Table 6 gives the seed means. TnT-LLM's per-text API time varies up to ninefold across seeds (1.36 to 12.68 s on BANKING77, 1.43 to 12.54 s on CLINC150, and 1.39 to 11.79 s on HWU64, against 0.055 to 0.086 s for LCD), so the per-seed ratio of API time ranges from 19.4 to 193.9 and its median over seeds is 31.2, 31.4, and 30.6; the cost ratio is stable (8.3 to 15.3 per seed). Per seed, the break-even is zero on 6 of the 10 BANKING77 seeds and 22–129 texts on the other four, zero on every CLINC150 seed, and between 0 and 710 texts on HWU64 (zero on one seed).

## C Complete Results

### C.1 Paired differences and their sensitivity

Table 7 gives the comparison with TnT-LLM on all six metrics, and Figure 3 shows its intervals. Over the ten seeds, LCD is ahead on 8, 10, and 10 seeds for ARI and on 9, 10, and 10 for F1. Table 8 repeats the comparison three ways. On seeds 14–20 alone, every difference keeps its sign; on BANKING77 LCD is ahead on 5 of 7 seeds for ARI and 6 of 7 for F1, and HWU64 AMI ($+.009$ $[-.002, +.022]$, Holm $p = .18$) is again not significant. The adjudicated reader, which resolves each multi-claimed text with one single-choice pass, gives slightly larger primary differences than the strict reader. The standard metrics depend on how the output is flattened: when uncovered and multi-claimed texts both go to "none" without adjudication, the AMI difference is $+.004$ $[-.004, +.010]$ on BANKING77, $+.074$ $[+.061, +.084]$ on CLINC150, and $-.001$ $[-.012, +.011]$ on HWU64, so the BANKING77 AMI gain of Table 7 depends on adjudication, the CLINC150 gain holds under both, and the HWU64 difference is not significant under either. ARI on this strict flat output, which pools uncovered texts instead of making them singletons, is ahead on every dataset ($+.035, +.277, +.082$). LCD's partitions are more homogeneous on every dataset and less complete on BANKING77 and HWU64, as expected of partitions finer than the reference.

### C.2 Claim states

Table 9 gives the number of reported classes and the claim states on the test split. LCD reports more classes than the reference on every dataset. Its uncovered and multi-claimed rates are low, and most of its mapped abstention comes from single claims on classes that the one-to-one mapping leaves unmatched. The adjudicated reader lowers LCD's abstention to 10.8%, 7.9%, and 18.6%. TnT-LLM's zero overlap is a property of single-label output and not evidence of cleaner boundaries. Mapped macro-F1 averages over all reference classes under a one-to-one mapping, so a method with $K$ classes can reach at most $\min(K, K^*)/K^*$ of it for $K^*$ reference classes. TnT-LLM's coarser taxonomies cap its F1 at .936, .702, and .891 on average (CLINC150: .600 to .813 per seed) before any labeling error, and merged intents lower its ARI as well, whereas LCD's cap is 1 on every fit but one (.993); TnT-LLM was run without a target number of categories (Appendix B.3), so part of its CLINC150 deficit reflects granularity.

Table 7: **LCD is ahead of TnT-LLM in 17 of 18 dataset–metric comparisons after Holm correction and significantly behind in none.** Test split, means over 10 seeds (11–20) paired by seed; primary metrics (claim-set ARI, mapped macro-F1) of the strict reader and standard metrics of the single-label output; Δ = LCD − TnT-LLM (intervals: Figure 3). Green▲: ahead after Holm correction of exact sign-flip $p$-values within the primary (ARI, F1), standard ($ARI_{flat}$, NMI, ACC) and AMI families; red▼: interval below zero. Bold: the better method. Seeds 14–20 alone (fitted after the configuration was fixed): all 18 differences keep their sign.

| Dataset | Method | ARI | F1 | $ARI_{flat}$ | NMI | AMI | ACC |
|---|---|---|---|---|---|---|---|
| BANKING77 | TnT-LLM | .617 | .650 | .617 | .869 | .807 | .702 |
| | LCD | **.673** | **.743** | **.675** | **.887** | **.822** | **.748** |
| | Δ | +.056▲ | +.093▲ | +.058▲ | +.017▲ | +.016▲ | +.046▲ |
| CLINC150 | TnT-LLM | .528 | .534 | .507 | .892 | .837 | .622 |
| | LCD | **.821** | **.840** | **.796** | **.950** | **.914** | **.854** |
| | Δ | +.293▲ | +.306▲ | +.289▲ | +.057▲ | +.077▲ | +.232▲ |
| HWU64 | TnT-LLM | .477 | .524 | .475 | .824 | .748 | .620 |
| | LCD | **.603** | **.646** | **.573** | **.845** | **.753** | **.682** |
| | Δ | +.126▲ | +.121▲ | +.097▲ | +.021▲ | +.005 | +.063▲ |

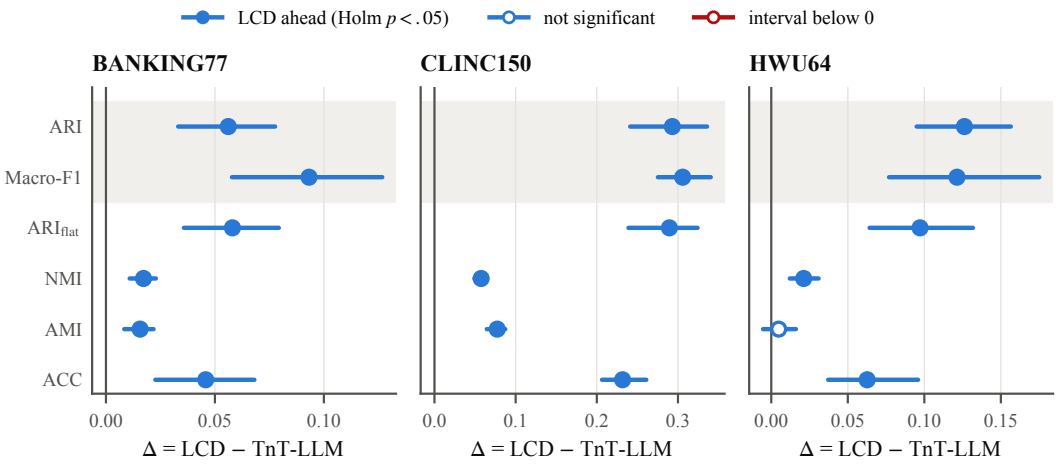



Figure 3: **Every interval that excludes zero favors LCD.** Paired differences LCD − TnT-LLM on the test split (10 seeds, paired seed bootstrap, 10,000 replicates, 95% percentile intervals); shaded rows are the primary metrics. Filled: LCD ahead after Holm correction of exact sign-flip $p$-values.

### C.3 Stability across seeds

The seed-to-seed standard deviation of LCD's ARI is .019, .076, and .026 against .043, .051, and .044 for TnT-LLM (Figure 4); the ratio is .43 $[.27, .81]$, 1.49 $[.28, 3.20]$, and .59 $[.34, .93]$, and the Brown–Forsythe test gives uncorrected $p = .056$, .99, and .021 (Holm-adjusted $p = .11$, .99, and .063, so none rejects equal spread after correction). The CLINC150 ratio comes from seed 17, whose fit has the smallest codebook ($K = 149$) and an ARI of .613, still above TnT-LLM's fit of the same seed (.502); HCS alone adds .252 ARI to that fit (the same-read removal of Table 2). The partitions of two LCD seeds agree more with each other than those of two TnT-LLM seeds (mean pairwise ARI .714, .782, and .633 against .644, .612, and .553), and the number of classes varies less (SD ratio .48, .75, and .92).

### C.4 Weaker reader

Table 10 gives the weaker-reader comparison of Section 5.3, and Figure 5 shows each method's change from the primary to the weaker reader on the same 128 texts. On these texts the pri-

Table 8: **On seeds 14–20 alone all 18 differences keep their sign; on BANKING77 the AMI gain holds only with the adjudicated flat output (strict flat $+.004$), on CLINC150 it holds under both, and LCD's partitions are more homogeneous but less complete.** Differences LCD − TnT-LLM on the test split with 95% paired seed-bootstrap intervals. Seeds 14–20: the fits made after the configuration was fixed (7 seeds, strict reader, adjudicated flat output); with seven seeds the exact test cannot go below $p = 1/128$, and after Holm correction the six primary differences and the BANKING77 and CLINC150 AMI differences are significant ($p \leq .047$) while the standard ones are not ($p = .070$). Adjudicated: primary metrics of the adjudicated reader (10 seeds). Strict flat: uncovered and multi-claimed texts both sent to "none" (10 seeds). Green▲: interval above zero; red▼: interval below zero.

| Metric | BANKING77 | CLINC150 | HWU64 |
|---|---|---|---|
| *Seeds 14–20 (7 seeds)* | | | |
| ARI | +.052▲ [+.022, +.079] | +.284▲ [+.220, +.339] | +.130▲ [+.094, +.163] |
| F1 | +.086▲ [+.040, +.128] | +.302▲ [+.273, +.332] | +.118▲ [+.071, +.184] |
| $\text{ARI}_{\text{flat}}$ | +.054▲ [+.025, +.080] | +.279▲ [+.214, +.324] | +.102▲ [+.064, +.141] |
| NMI | +.017▲ [+.008, +.025] | +.057▲ [+.047, +.065] | +.024▲ [+.013, +.036] |
| AMI | +.015▲ [+.006, +.023] | +.077▲ [+.060, +.089] | +.009 [−.002, +.022] |
| ACC | +.040▲ [+.010, +.068] | +.228▲ [+.203, +.254] | +.067▲ [+.038, +.110] |
| *Adjudicated reader, primary metrics* | | | |
| ARI | +.060▲ [+.038, +.082] | +.294▲ [+.242, +.337] | +.131▲ [+.100, +.161] |
| F1 | +.097▲ [+.062, +.130] | +.307▲ [+.276, +.341] | +.123▲ [+.079, +.177] |
| *Strict flat output* | | | |
| $\text{ARI}_{\text{flat}}$ | +.035▲ [+.014, +.056] | +.277▲ [+.229, +.311] | +.082▲ [+.049, +.118] |
| NMI | +.009▲ [+.003, +.015] | +.055▲ [+.047, +.062] | +.017▲ [+.008, +.027] |
| AMI | +.004 [−.004, +.010] | +.074▲ [+.061, +.084] | −.001 [−.012, +.011] |
| ACC | +.037▲ [+.014, +.059] | +.229▲ [+.204, +.259] | +.058▲ [+.033, +.091] |
| *Adjudicated flat output* | | | |
| homogeneity | +.042▲ [+.029, +.054] | +.109▲ [+.097, +.121] | +.070▲ [+.051, +.092] |
| completeness | −.008▼ [−.011, −.005] | −.001 [−.007, +.005] | −.032▼ [−.042, −.022] |

Table 9: **LCD leaves 1.3–4.8% of test texts uncovered and claims at most 1.7% more than once; most of its higher abstention comes from classes finer than the reference that the one-to-one mapping leaves unmatched.** Test split, means over 10 seeds (strict reader). $K$: reported classes (range over seeds); $U_R$: uncovered; $O_R$: multi-claimed; abstention: texts without one claim on a matched class, of which "unmatched" are single claims on an unmatched class (%). Bold: the lower rate; TnT-LLM's $O_R = 0$ holds by construction.

| Dataset (reference $K$) | Method | $K$ | $U_R$ | $O_R$ | Abstention | Unmatched |
|---|---|---|---|---|---|---|
| BANKING77 (77) | TnT-LLM | 76.4 (60–100) | **0.6** | 0.0 | **4.6** | **4.0** |
| | LCD | 97.4 (91–108) | 1.3 | 1.7 | 12.3 | 9.3 |
| CLINC150 (150) | TnT-LLM | 105.3 (90–122) | 2.7 | 0.0 | **3.7** | **1.0** |
| | LCD | 170.8 (149–179) | **2.2** | 0.4 | 8.2 | 5.6 |
| HWU64 (64) | TnT-LLM | 58.1 (38–70) | **2.0** | 0.0 | **4.8** | **2.8** |
| | LCD | 95.6 (84–110) | 4.8 | 0.6 | 19.1 | 13.6 |

mary reader already puts LCD ahead on every ARI and F1 comparison (BANKING77 ARI $+.045$ $[+.005, +.085]$; descriptive, no Holm correction). The weaker reader lowers LCD's claim-set ARI by .078, .006, and .027 and TnT-LLM's by .054 and .008 on BANKING77 and CLINC150. Part of LCD's change comes from the request layout: against the primary reader's read of the same 128 texts in one request (ARI .668, .816, and .655), LCD's claim-set ARI changes by $-.062$, $-.003$, and $+.008$ under the weaker reader, and its F1 by $-.025$, $-.017$, and $-.028$. The uncovered rate changes from 1.1%, 2.0%, and 5.7% to 2.0%, 5.6%, and 11.6% for LCD, and from 0.5%, 2.6%, and 2.0% to 0.1%, 6.2%, and 9.7% for TnT-LLM. Claim-set ARI makes each uncovered text a singleton, which is why TnT-LLM's HWU64 claim-set ARI rises by .042 while its flat ARI falls by .040. With uncovered texts pooled into one cluster, the weaker reader lowers $\text{ARI}_{\text{flat}}$ by .074, .120,

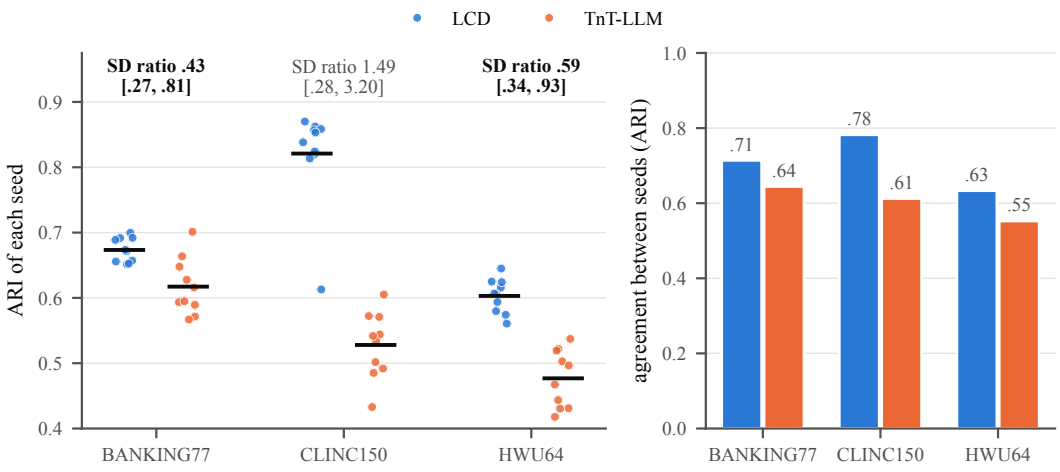



Figure 4: **The seed-to-seed SD of LCD's ARI is lower than TnT-LLM's on BANKING77 and HWU64 (ratios .43 and .59, intervals below one) but not on CLINC150 (1.49); no Brown–Forsythe test rejects equal spread after Holm correction.** Left: claim-set ARI of every seed on the test split (bars: means); labels: SD ratio LCD / TnT-LLM with its 95% seed-bootstrap interval (10,000 replicates), bold where the interval excludes one. Right: mean ARI between the partitions two seeds produce on the same texts (all seed pairs). Brown–Forsythe test (median-centered Levene): uncorrected $p = .056$, .99 and .021.

Table 10: **Read by `gpt-4.1-mini`, LCD stays ahead of TnT-LLM in mapped F1 on all three datasets and in claim-set ARI on CLINC150 and HWU64 after Holm correction; on BANKING77 the ARI difference (+.021) is not significant.** Both methods' codebooks read by `gpt-4.1-mini` (weaker reader) and by the primary reader (restricted to the same texts) on a fixed 128-text subsample of the test split, 10 seeds; claim-set ARI and mapped macro-F1 of the strict reader; Δ weaker = LCD − TnT-LLM under the weaker reader (green▲: Holm-adjusted exact $p < .05$; red▼: interval below zero). With uncovered texts pooled into one cluster, only the CLINC150 difference stays significant (Appendix C.4). LCD's primary columns are its full test read restricted to these texts; read alone in one request, as the weaker reader reads them, the primary reader gives ARI .668, .816, and .655 (Appendix D.3), and LCD's change to the weaker reader is then −.062, −.003, and +.008 in ARI and −.028 to −.017 in F1.

| Dataset | Method | ARI primary | ARI weaker | change | F1 primary | F1 weaker | change |
|---|---|---|---|---|---|---|---|
| BANKING77 | TnT-LLM | .639 | .585 | −.054 | .480 | .453 | −.028 |
| | LCD | .684 | .606 | −.078 | .562 | .534 | −.028 |
| | Δ weaker | | +.021 | | | +.081▲ | |
| CLINC150 | TnT-LLM | .546 | .537 | −.008 | .318 | .303 | −.015 |
| | LCD | .818 | .812 | −.006 | .488 | .479 | −.009 |
| | Δ weaker | | +.275▲ | | | +.176▲ | |
| HWU64 | TnT-LLM | .510 | .552 | +.042 | .468 | .459 | −.009 |
| | LCD | .690 | .663 | −.027 | .550 | .527 | −.023 |
| | Δ weaker | | +.111▲ | | | +.068▲ | |

and .161 for LCD and by .053, .094, and .040 for TnT-LLM (means −.118 and −.062), and the differences are +.026 on BANKING77 (Holm $p = .14$), +.255 on CLINC150 ($p = .024$), and +.011 [−.070, +.100] on HWU64 ($p = .41$), so only CLINC150 stays significant. AMI on BANKING77 (+.033, Holm $p = .055$) and on HWU64 (+.024, $p = .11$) is not significant either. On seeds 14–20 alone, weaker-reader claim-set ARI on BANKING77 is +.030 [−.006, +.065], again not significant. Macro-F1 on 128 texts is lower than on the full test split for both methods because many reference classes have few or no texts in the subsample; the comparison is between methods

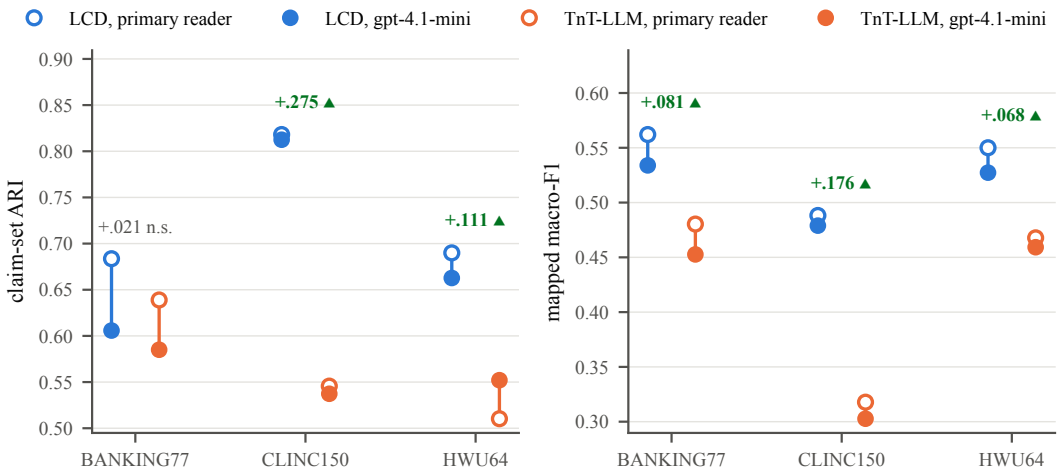



Figure 5: **The weaker reader changes claim-set ARI by −.078 to −.006 for LCD and by −.054 to +.042 for TnT-LLM, and lowers mapped F1 by at most .028 for either method.** Hollow: primary reader; filled: `gpt-4.1-mini`; lines: the change on a fixed 128-text subsample of the test split (10 seeds). Above each dataset: LCD − TnT-LLM under the weaker reader, ▲ if its Holm-adjusted exact $p < .05$, n.s. otherwise (tests in Table 10). Against LCD's primary read of the same texts in one request, its ARI change is −.062 to +.008 (Table 10).

on the same texts, and the seed intervals do not include the uncertainty from sampling these 128 texts.

**Second reader.** Reproducible application is measured on the same 128 texts for both methods: the primary reader's read, restricted to the subsample, against `gpt-4.1-mini`'s read of the same frozen output (LCD: the same codebook and keyed strict reader at 128 texts per request, in the reported class space; TnT-LLM: the same taxonomy and its labeler, one text per request). We report Cohen's $\kappa$ pooled over all text–class membership decisions of a fit and the share of texts that receive identical claim sets, with 95% intervals from resampling the 128 texts (Table 11). Pooled $\kappa$ is .854, .934, and .858 for LCD and .865, .886, and .856 for TnT-LLM; the difference is significant only on CLINC150 ($+.048$ $[+.024, +.073]$). Identical claim sets are more frequent for LCD on CLINC150 (91.1% against 86.9%), less frequent on BANKING77 (83.0% against 86.5%), and equally frequent on HWU64 (82.5% against 82.6%). Against the primary read of the 128 texts in one request, the request layout of the weaker reader, LCD's pooled $\kappa$ is .872, .942, and .863 and its identical claim sets 85.1%, 92.1%, and 82.4%; the differences from TnT-LLM are then significant only on CLINC150 ($\kappa$ $+.056$ $[+.031, +.081]$; identical $+5.2$ points $[+2.3, +8.1]$), and the BANKING77 difference in identical claim sets ($-1.4$ points $[-4.4, +1.6]$) is not. The two primary reads of LCD themselves give identical claim sets for 93.9%, 96.2%, and 90.0% of the texts (85.2% on one HWU64 seed) and pooled $\kappa$ of .949, .970, and .917 under the same model, against cross-model $\kappa$ of .854 to .942 for LCD and .856 to .886 for TnT-LLM; TnT-LLM's labeler, one text per request, has no batch composition to depend on. The criterion favors neither method on these data, and no check reads LCD's six-field entries and TnT-LLM's name-and-description entries under matched conditions, so these data do not show that written rules are applied more reproducibly than TnT-LLM's descriptions. The two formats differ: LCD may claim several classes per text, which makes identical claim sets harder to reach, and its reader reads 128 texts per request, whereas TnT-LLM's labeler returns one category or none for one text. Per-class $\kappa$ is not reported: with 128 texts, a fit has on average only 2.9, 0.1, and 4.1 LCD classes (TnT-LLM: 6.8, 3.6, and 7.4) with at least five positives in either read. The check compares model readers rather than humans, and its intervals from resampling texts, which hold the seeds fixed, are the only ones in this paper that include text-sampling uncertainty.

Table 11: **Across the two readers, LCD's codebooks are applied about as consistently as TnT-LLM's taxonomies: LCD agrees more on CLINC150 and about equally on HWU64, and on BANKING77 it agrees less by identical claim sets only against a primary read in which each text shared its request with other test texts.** The primary reader's read of the fixed 128-text test subsample against `gpt-4.1-mini`'s read of the same frozen output, means over 10 seeds (LCD: keyed strict reader, 128 texts per request, reported classes; TnT-LLM: its LLM labeler, one text per request, one category or none). LCD: the primary reader's full test read restricted to the subsample; LCD*: the primary reader's read of the 128 texts in one request, as the weaker reader reads them (Appendix D.3). Pooled $\kappa$: Cohen's $\kappa$ over all text–class membership decisions of a fit. Identical: share of texts whose two claim sets are equal (%; $\Delta$ in percentage points); LCD may claim several classes, which makes exact agreement harder. $\Delta$ = LCD − TnT-LLM with a 95% interval from resampling the 128 texts (the same resamples for every seed, both methods and both references; seeds fixed); green▲ / red▼: interval above / below zero; bold: the higher value where the interval excludes zero.

| Measure | Method | BANKING77 | CLINC150 | HWU64 |
|---|---|---|---|---|
| Pooled $\kappa$ | TnT-LLM | .865 | .886 | .856 |
| | LCD | .854 | **.934** | .858 |
| | $\Delta$ | −.011 [−.040, +.018] | +.048▲ [+.024, +.073] | +.002 [−.032, +.038] |
| | LCD* | .872 | **.942** | .863 |
| | $\Delta$ | +.007 [−.023, +.036] | +.056▲ [+.031, +.081] | +.007 [−.024, +.040] |
| Identical claim sets | TnT-LLM | **86.5** | 86.9 | 82.6 |
| | LCD | 83.0 | **91.1** | 82.5 |
| | $\Delta$ | −3.4▼ [−6.6, −0.2] | +4.2▲ [+1.4, +7.0] | −0.1 [−4.4, +4.6] |
| | LCD* | 85.1 | **92.1** | 82.4 |
| | $\Delta$ | −1.4 [−4.4, +1.6] | +5.2▲ [+2.3, +8.1] | −0.2 [−4.1, +4.1] |

### C.5 Other discovery baselines

Table 12 gives the paired differences behind the distilled, LLooM, clustering, and reference rows of Table 1. TnT-LLM's distilled classifiers trail LCD on every seed, family, dataset, and metric; the differences range from $+.099$ $[+.078, +.118]$ (ARI, MLP, BANKING77) to $+.647$ $[+.570, +.730]$ (ARI, LightGBM, CLINC150), and the selected classifier trails by $+.103$, $+.322$, and $+.182$ ARI and $+.119$, $+.325$, and $+.150$ F1. The automated LLooM rows keep the six fits that kept no concept (Appendix B.3); these fits cost little, so LCD is cheaper than LLooM on 12 of the 15 seed pairs of each target. Over all pairs, LLooM costs 17.0, 39.2, and 26.9 times as much as LCD at 120 concepts and 9.5, 7.5, and 7.6 times as much at 20 concepts on BANKING77, CLINC150, and HWU64 (means 27.7 and 8.2, the cost column). Without the empty fits every difference remains large, and the smallest is LLooM with 20 concepts on BANKING77, ARI $+.264$ $[+.220, +.308]$. All LLooM comparisons use mapped macro-F1. LLooM assigns overlapping concepts, so partition metrics penalize it. Under best-match F1, a diagnostic that lets each reference class take its best-matching concept and so forgives overlap, the non-empty LLooM fits with 120 concepts trail LCD by only $+.050$, $+.087$ $[+.079, +.095]$, and $+.021$ on BANKING77, CLINC150, and HWU64 (A+B: $+.060$, $+.094$, $+.025$); the BANKING77 and HWU64 differences rest on 3 and 4 non-empty fits and cannot be tested (Appendix B.5), and the HWU64 interval includes zero ($[-.010, +.050]$). The 20-concept fits trail by $+.204$ or more. The label-selected row pairs only the 12 non-empty fits with 120 concepts (3, 5, and 4 per dataset); it reuses their test reads, since selecting concepts makes no request, and is priced as those fits (38.7 USD per seed on average in Table 1, 33.4 times LCD's cost on the same seeds). With 3 and 4 pairs on BANKING77 and HWU64 its comparisons cannot be tested (Appendix B.5) and are not marked, but LCD is ahead on every pair: by $+.060$ to $+.147$ ARI and $+.093$ to $+.213$ F1 on BANKING77 and by $+.059$ to $+.143$ ARI and $+.010$ to $+.165$ F1 on HWU64.

The clustering rules choose on average 177, 110, and 36 clusters (silhouette), 20, 20, and 20 (Calinski–Harabasz), 24, 49, and 20 (BIC), 22, 86, and 20 (stability), 53, 91, and 26 (HDBSCAN), and 99, 170, and 64 (UMAP + HDBSCAN) on BANKING77, CLINC150, and HWU64. For the best of 60 UMAP + HDBSCAN settings, the ARI and F1 columns may come from different settings; the best setting for mapped F1 on HWU64 gives the smallest margin of Table 12, $+.041$

Table 12: **LCD is ahead of every distilled TnT-LLM classifier, automated LLooM arm and clustering rule that chooses its own number of clusters, on ARI and mapped macro-F1 on every dataset; the closest is the distilled MLP on BANKING77 (ARI $+.099$).** Paired differences LCD − baseline on the test split (strict reader), behind Table 1 (TnT-LLM's LLM labeler: Table 7); green▲: 95% paired seed-bootstrap interval above zero on at least five paired seeds; bold: the closest baseline. Distilled: classifiers on the discovery embeddings trained on TnT-LLM's LLM labels. LLooM: automated adaptation with targets of 120 and 20 concepts, members at score A (or A+B), seeds 11–15 with empty fits kept. Clustering: discovery embeddings; test texts go to the nearest cluster. Cost: API cost relative to LCD; emb.: embeddings only. Label-informed rows use reference labels and are references, not upper bounds; the label-selected row pairs only the 3/5/4 non-empty 120-concept fits and reuses their reads (Appendix B.3).

| | | BANKING77 | | CLINC150 | | HWU64 | | |
|---|---|---|---|---|---|---|---|---|
| Baseline | Seeds | ARI | F1 | ARI | F1 | ARI | F1 | Cost |
| TnT-LLM distilled, selected | 10 | +.10▲ | +.12▲ | +.32▲ | +.32▲ | +.18▲ | +.15▲ | 2.5× |
| TnT-LLM distilled, logistic regression | 10 | +.11▲ | +.12▲ | +.32▲ | +.32▲ | +.19▲ | +.15▲ | 2.5× |
| TnT-LLM distilled, LightGBM | 10 | +.24▲ | +.21▲ | +.65▲ | +.62▲ | +.35▲ | +.25▲ | 2.5× |
| TnT-LLM distilled, MLP | 10 | **+.10▲** | +.12▲ | +.32▲ | +.32▲ | +.18▲ | +.15▲ | 2.5× |
| LLooM, 120 concepts | 5 | +.47▲ | +.61▲ | +.52▲ | +.69▲ | +.46▲ | +.54▲ | 27.7× |
| LLooM, 120 concepts, A+B | 5 | +.49▲ | +.64▲ | +.52▲ | +.77▲ | +.47▲ | +.57▲ | 27.7× |
| LLooM, 20 concepts | 5 | +.34▲ | +.55▲ | +.61▲ | +.74▲ | +.40▲ | +.58▲ | 8.2× |
| LLooM, 20 concepts, A+B | 5 | +.35▲ | +.58▲ | +.60▲ | +.75▲ | +.40▲ | +.61▲ | 8.2× |
| $k$-means, silhouette | 10 | +.21▲ | +.22▲ | +.32▲ | +.38▲ | +.28▲ | +.35▲ | emb. |
| $k$-means, Calinski–Harabasz | 10 | +.47▲ | +.63▲ | +.69▲ | +.80▲ | +.39▲ | +.49▲ | emb. |
| $k$-means, bootstrap stability | 10 | +.46▲ | +.61▲ | +.39▲ | +.47▲ | +.39▲ | +.49▲ | emb. |
| Gaussian mixture, BIC | 10 | +.44▲ | +.59▲ | +.56▲ | +.65▲ | +.43▲ | +.49▲ | emb. |
| HDBSCAN | 10 | +.42▲ | +.49▲ | +.51▲ | +.49▲ | +.34▲ | +.42▲ | emb. |
| UMAP + HDBSCAN | 10 | +.22▲ | +.27▲ | +.20▲ | +.22▲ | +.10▲ | +.12▲ | emb. |
| *Label-informed references* | | | | | | | | |
| $k$-means, reference class count | 10 | +.24▲ | +.28▲ | +.27▲ | +.31▲ | +.14▲ | +.17▲ | emb. |
| $k$-means, best $K$ on a grid | 10 | +.20▲ | +.20▲ | +.20▲ | +.20▲ | +.11▲ | +.08▲ | emb. |
| LLooM, 120 concepts, label-selected | 3/5/4 | +.09 | +.14 | +.14▲ | +.18▲ | +.10 | +.08 | 33.4× |
| UMAP + HDBSCAN, best of 60 | 10 | +.20▲ | +.23▲ | +.19▲ | +.21▲ | +.09▲ | +.04▲ | emb. |

$[+.019, +.063]$. $k$-means at the best number of clusters on the grid (160, 270, and 120 clusters for ARI; 160, 270, and 140 for F1), also chosen post hoc on the test labels, still trails LCD by .200, .202, and .107 ARI and by .197, .197, and .078 mapped F1.

# D Ablations and Further Analyses

## D.1 Component removals

Each refit row of Table 2 is LCD with one component removed and everything else unchanged, refitted on seeds 11–13 of every dataset (nine fits per row) and applied to the test split with the full reader. The rows are:

- *One draft, no selection*: the first of the five drafts becomes the first codebook, so the label-free selection has nothing to choose from.
- Split, Merge, Revise, Retire, Add: the operator is removed from every round. Removing Split also removes its pre-screen; removing Retire keeps the retirement that follows the final reconcile; removing Merge leaves Group unchanged.
- Split *pre-screen*: every Split proposal goes directly to verification on all discovery texts.
- *All operators but* Add: the rounds run Add only.
- *Contest*: candidates are judged on their isolated verification columns; the take-over guard, which needs the contest, is off as well.

Table 13: **Removing ADD, SPLIT or every operator but ADD, or replacing the rounds by one TnT-style pass, lowers F1 on every dataset, by .071 to .603; no other row changes F1 by more than .061 on any dataset.** Mapped macro-F1 of the full model minus the ablated one on the test split, per dataset and pooled (mean of the per-dataset means, 95% seed-bootstrap interval), for the rows of Table 2. Exact $p$: one-sided sign-flip test of the pooled mean (refit rows: 9 pairs, smallest attainable $1/512$; same-read rows: the fits whose read the removal changes); Holm: adjusted within the refit or the same-read rows, control excluded; bold: adjusted $p < .05$. In the reverse direction (ablated ahead) every row has exact $p \geq .375$ except the re-seeded control (ARI: $p = .027$, uncorrected). Green▲: Holm-adjusted $p < .05$ for F1; red▼: interval below zero; bold ΔF1: the largest.

| Component removed | ΔF1 BANKING77 | ΔF1 CLINC150 | ΔF1 HWU64 | ΔF1 pooled mean | ΔF1 pooled 95% CI | Exact $p$ ARI / F1 | Holm $p$ ARI / F1 |
|---|---|---|---|---|---|---|---|
| *Refit (seeds 11–13)* | | | | | | | |
| one draft, no selection | −.012 | +.048 | +.061 | +.032 | [+.012, +.051] | .051 / .037 | .203 / .148 |
| SPLIT | +.071 | +.139 | +.092 | +.101▲ | [+.078, +.130] | .006 / .002 | **.035 / .020** |
| SPLIT pre-screen | −.009 | +.027 | −.005 | +.004 | [−.014, +.026] | .395 / .436 | .789 / .932 |
| MERGE | +.002 | +.008 | +.009 | +.006 | [−.012, +.028] | .096 / .311 | .287 / .932 |
| REVISE | +.002 | −.007 | +.003 | −.001 | [−.003, +.002] | .500 / .688 | .789 / .932 |
| RETIRE | +.022 | +.012 | +.014 | +.016 | [+.005, +.026] | .012 / .021 | .059 / .107 |
| ADD | +.321 | +.587 | +.324 | **+.411▲** | [+.389, +.439] | .002 / .002 | **.020 / .020** |
| all operators but ADD | +.081 | +.128 | +.093 | +.100▲ | [+.073, +.130] | .002 / .002 | **.020 / .020** |
| contest | +.023 | +.025 | +.018 | +.022 | [+.009, +.034] | .004 / .010 | **.027** / .059 |
| rounds → TnT-style pass | +.291 | +.603 | +.316 | +.403▲ | [+.345, +.462] | .002 / .002 | **.020 / .020** |
| none (re-seeded control) | −.022 | −.007 | +.022 | −.003 | [−.021, +.020] | .975 / .564 | – / – |
| *Same reads (seeds 11–20)* | | | | | | | |
| GROUP | +.001 | +.003 | −.002 | +.001 | [−.002, +.003] | .021 / .305 | **.021** / .305 |
| HCS (stage two ignored) | .000 | +.007 | +.007 | +.005▲ | [+.001, +.009] | .008 / .008 | **.016 / .023** |
| GROUP and HCS | +.001 | +.009 | +.005 | +.005▲ | [+.001, +.010] | <.001 / .015 | **<.001 / .031** |

- *Rounds → TnT-style pass*: from the same selected first codebook, the operator rounds, the final reconcile, HCS, and GROUP are replaced by one pass of unverified whole-codebook updates (remove, replace, and add edits) over shuffled 200-text batches, in the manner of TnT-LLM; only the final codebook is read on the discovery texts. The pass ends with 50.7, 56.0, and 43.7 categories on average and test ARI .439, .230, and .374 on seeds 11–13, where the full fits have 95.0, 177.0, and 99.7 categories and ARI .666, .857, and .614, and TnT-LLM, with ten taxonomy trials and its selection, has 74.0, 109.0, and 59.7 categories and ARI .600, .543, and .497. The pass thus scores .12 to .31 below TnT-LLM itself: its .365 measures the rounds, reconcile and consolidation together against one cheap unverified pass, whereas LCD's margin over TnT-LLM on these seeds is .067, .314, and .117.
- *None (re-seeded control)*: no component is removed; only the random stream from which the operators of a round draw their text samples is re-seeded.

The operators of a round share that random stream, so removing SPLIT, MERGE, or REVISE, or running ADD alone, also re-draws the texts that later operators sample; the re-seeded control measures this resampling alone and is the noise floor for the refit rows. MERGE and REVISE also share the record of rejected pairs, so removing one can change which pairs the other considers even when the removed operator accepted nothing. Replaying that record shows the coupling in one direction only: without MERGE, REVISE chooses different pairs in 6 of the 9 fits, whereas without REVISE, MERGE never does (0 of 9); in 5 of the 9 refits without REVISE the removed operator did nothing and the refit equals the full fit. The refit rows are Holm-corrected among themselves, without the control. No row keeps an operator's proposals while dropping its acceptance tests (Equation 5), and the handling criterion of SPLIT and MERGE is never removed, so the refits isolate neither the value of the tests nor that of the criterion. The same-read rows need no refit: without GROUP, every group is reported as its two member categories; without HCS, stage two is ignored and each HCS parent is reported under its own definition; without both, the codebook after the final reconcile is read with the final reader.

Table 13 gives the per-dataset F1 differences and the $p$-values of every row. F1 follows ARI. Removing ADD or SPLIT, running ADD alone, or replacing the rounds by one TnT-style pass lowers F1 on every dataset. The single draft lowers F1 on CLINC150 and HWU64 (.048 and .061) but not on BANKING77 ($-.012$), and the pooled ARI effect of the pre-screen (.023) comes from CLINC150 alone (.081, against $-.010$ and $-.003$), while removing the pre-screen raises the fit cost by 10%, its main purpose. The re-seeded control changes F1 by $-.022$ to $+.022$ per dataset, as much as several component effects, so a per-dataset difference of that size in a refit row is not evidence of an effect; its fits score higher in pooled ARI than the full fits with uncorrected exact $p = .027$, and no refit row reaches $p < .05$ in that direction ($p \geq .375$). Among the same-read rows, HCS changes the reads of only 7 of the 30 fits (two on CLINC150, where it adds .110 and .252 ARI, and five on HWU64), and GROUP changes ARI by at most .008 in any fit.

### D.2 WHAT THE OPERATOR ROUNDS DO

Table 14 counts the accepted operations. The first codebook has 38.0, 46.2, and 32.3 categories and the final codebook 100.2, 173.3, and 97.8 member categories. The rounds accept 84.9 operations per fit on average, and HCS and GROUP add 2.8 in the final consolidation; ADD accounts for most accepted operations, followed by SPLIT. On average 78% of the final categories are not categories of the first codebook.

At fit time, discovery ARI is .642, .514, and .483 after the SPLIT step of round 1 and .723, .773, and .566 after its ADD step; the MERGE, REVISE, and RETIRE steps between them change it by at most .006, so almost all of this gain comes from ADD, which also raises the number of categories from 48.8, 52.5, and 40.3 to 84.7, 119.2, and 74.8. Discovery ARI then moves to .713 after the sixth round, .717 after the reconcile, and .721 in the final codebook on BANKING77; to .807, .824, and .843 on CLINC150; and to .574, .588, and .599 on HWU64 (Figure 6). After round 1, the rounds thus add no discovery ARI on BANKING77 ($-.009$) and little on HWU64 ($+.008$), and on CLINC150 and HWU64 a large part of the later gain comes in the reconcile and the final consolidation. The last round accepts 0.4, 0.7, and 2.6 changes per fit.

Read on the test split, the codebooks after rounds 1, 3, and 5, after the reconcile, and in the final form give claim-set ARI .644, .668, .670, .671, and .673 on BANKING77; .587, .769, .781, .783, and .821 on CLINC150; and .521, .566, .586, .583, and .603 on HWU64 (mapped F1 from .697 to .743, from .599 to .840, and from .588 to .646). Unlike discovery ARI at fit time, test ARI keeps rising after round 1: from the round-1 codebook to the final one by $+.029$ $[+.016, +.043]$, $+.234$ $[+.177, +.311]$, and $+.082$ $[+.044, +.124]$, higher on 10, 10, and 9 of 10 seeds (exact $p \leq .002$), and F1 by $+.046$, $+.241$, and $+.058$. Most of the rise comes by round 3; from round 5 to the final codebook ARI changes by $+.004$ $[-.001, +.008]$, $+.040$ $[+.001, +.099]$, and $+.017$ $[-.002, +.039]$, and on CLINC150 .038 of this last step is the same-read effect of HCS and GROUP (Table 2). The fit-time curve scores the membership matrix that the operators maintain, assembled from reads of isolated proposals and contested texts, whereas each test read applies the whole checkpoint codebook in one prompt. After round 1 the matrix gives discovery ARI .723, .773, and .566, while the test reads of that codebook give .644, .587, and .521; in the final codebook the two are .721, .843, and .599 against .673, .821, and .603. The two splits hold different texts, but the gap on CLINC150 shrinks from .186 after round 1 to .033 after round 3 and .022 in the final codebook (BANKING77: .078, .046, .047; HWU64: .045, .005, $-.004$): early in the rounds the verified matrix runs well ahead of what a joint read of the checkpoint codebook returns, and the later rounds close most of this gap.

### D.3 READER FORMAT AND BATCH SIZE

The keyed reader returns each verdict under the identifier of its message; a positional reader returns one list per message in order, so one omitted list moves every later verdict of the request onto the wrong text. Table 15 reads the final codebooks of seeds 11–20 on the 128-text test subsample with the strict reader in both formats at 32 and at 128 texts per request, with the primary and the weaker model. The keyed cell at 128 texts per request is the LCD reader; for the weaker model it is the read of Table 10. A positional answer with the wrong number of lists cannot be decoded and would be reported as a reader failure; none occurred in the 120 positional reads, so the losses of positional reading come from verdicts attached to the wrong texts or from worse verdicts, not from decoding. At 128 texts per request, keyed verdicts raise claim-set ARI over positional lists by $+.308$, $+.270$,

Table 14: **On average 78% of the final categories are not categories of the first codebook; the rounds accept 84.9 operations per fit, most of them ADD.** Per fit (means): classes in the selected first codebook and in the final codebook (members); the accepted operations in the operator rounds (SPLIT parents, MERGE pairs, REVISE pairs, RETIRE, ADD); and, after the rounds, the hierarchical catch-all splits (HCS) and the GROUP classes of the final consolidation (2.8 per fit together). New: share of final categories that are not a first-codebook category. RETIRE counts the rounds only (the final reconcile retires a further 1.4 per fit). Bold: the most frequent operation per row. Applied in order to the first codebook, the recorded accepted operations of every fit reconstruct its final codebook.

| Dataset | Fits | $K$ first | $K$ final | SPLIT | MERGE | REVISE | RETIRE | ADD | HCS | GROUP | New |
|---|---|---|---|---|---|---|---|---|---|---|---|
| BANKING77 | 10 | 38.0 | 100.2 | 16.9 | 0.9 | 0.5 | 1.1 | **43.0** | 0.0 | 2.8 | 76% |
| CLINC150 | 10 | 46.2 | 173.3 | 16.7 | 1.1 | 0.8 | 2.1 | **106.5** | 0.3 | 2.5 | 80% |
| HWU64 | 10 | 32.3 | 97.8 | 12.7 | 0.1 | 0.2 | 2.8 | **49.4** | 0.5 | 2.2 | 79% |
| All | 30 | 38.8 | 123.8 | 15.4 | 0.7 | 0.5 | 2.0 | **66.3** | 0.3 | 2.5 | **78%** |

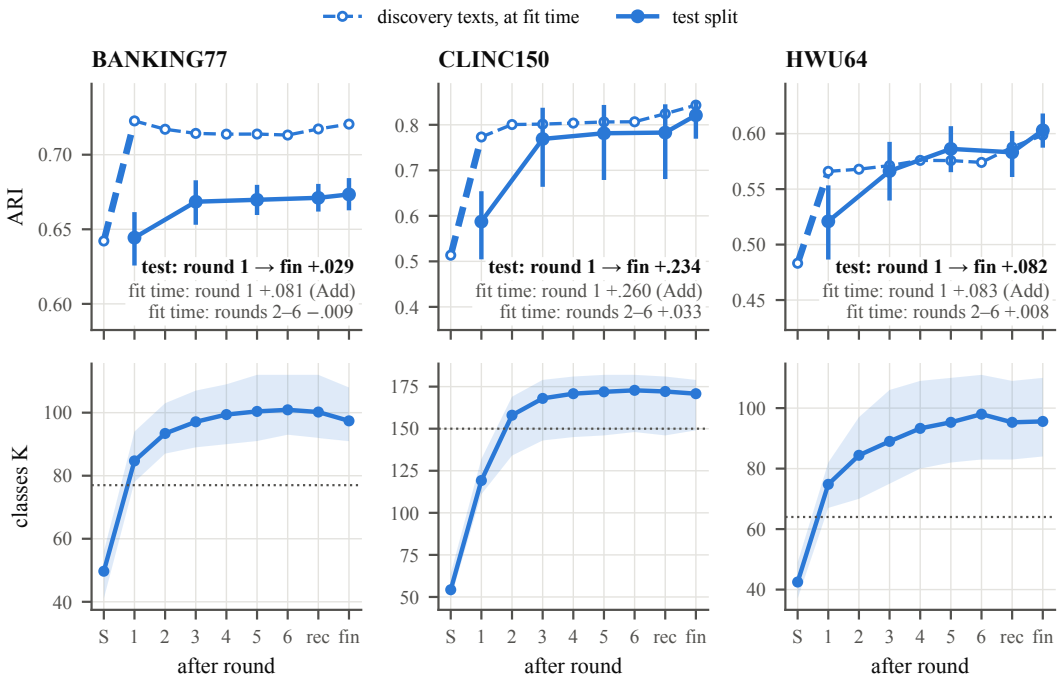



Figure 6: **On the test split, claim-set ARI rises from the codebook after round 1 to the final codebook by +.029, +.234, and +.082 (mean +.115; higher on 10, 10, and 9 of 10 seeds), although at fit time rounds 2–6 change discovery ARI by only −.009, +.033, and +.008.** Top: claim-set ARI after the SPLIT step of round 1 (S), after each operator round, after the final reconcile (rec), and in the final form (fin); dashed, hollow: on the discovery texts at fit time (reference labels used only to describe; fits that stop early keep their last round), thick dashed: the rest of round 1 (almost all of its gain in ADD); solid, filled: read on the test split after rounds 1, 3, and 5, the reconcile, and the final codebook (seed means, 95% intervals); box: the changes in numbers, the test-split change in bold. Bottom: number of classes (band: seed range; dotted: reference class count); the selected first codebooks have 38.0, 46.2, and 32.3 categories.

and +.418 with the primary reader and by +.554, +.746, and +.626 with `gpt-4.1-mini` (exact $p \leq .002$ each). At 32 texts per request the primary reader loses little with positional lists (keyed minus positional +.017, +.011, and .000), but the weaker reader still does (+.212, +.210, and +.149). With keyed verdicts, reading 128 instead of 32 texts per request lowers the primary reader's ARI by .025 on BANKING77 (on every seed) and changes it by +.005 on CLINC150 and HWU64,

Table 15: **At 128 texts per request, keyed verdicts raise claim-set ARI over positional lists by .332 with the primary reader and by .642 with `gpt-4.1-mini` (means over datasets); at 32 texts per request the two formats differ by at most .017 for the primary reader.** Claim-set ARI of the strict reader on a fixed 128-text subsample of the test split, final LCD codebooks of seeds 11–20, per reader model, reader format (keyed: verdicts returned under message identifiers; positional: one list per message, in order) and texts per request; every positional answer could be decoded. Shaded: the LCD reader. Bold: the best cell per model and dataset.

| Reader | Format, texts per request | BANKING77 | CLINC150 | HWU64 |
|---|---|---|---|---|
| primary reader | positional, 32 | .675 | .800 | .649 |
| | positional, 128 | .359 | .545 | .238 |
| | keyed, 32 | **.693** | .810 | .650 |
| | keyed, 128 | .668 | **.816** | **.655** |
| `gpt-4.1-mini` | positional, 32 | .354 | .580 | .479 |
| | positional, 128 | .051 | .067 | .036 |
| | keyed, 32 | .566 | .790 | .628 |
| | keyed, 128 | **.606** | **.812** | **.663** |

and it raises the weaker reader's ARI by .039, .023, and .034. Keyed verdicts thus let the reader take 128 texts per request, the request size behind the labeling cost of Section 5.2, at a cost of at most .025 ARI. Which texts share a request matters as much: the primary keyed cell at 128 texts per request reads the 128 texts together, whereas the full test read placed each of them among other test texts, and the two reads of the same codebooks differ in claim-set ARI by $-.016$, $-.002$, and $-.035$ (from $-.140$ to $+.032$ per fit) and in F1 by $-.003$, $+.008$, and $+.005$. A text's claims can therefore change with the texts that share its request: the two reads give identical claim sets for 93.9%, 96.2%, and 90.0% of the texts (as few as 85.2% on one HWU64 seed; Appendix C.4), and the weaker-reader and agreement comparisons report both primary reads (Tables 10 and 11).

### D.4 Definition adherence, long codebooks, and order

**Definition adherence.** Three edits of every final category test whether the reader follows the written rules or the category name: every name is replaced by a neutral "Category $k$"; names are permuted among the categories while the definitions stay in place; or only the name is kept, with the definition set to the name and the rules and examples removed. Each edited codebook is read on the test split with the strict reader and scored with the mapping of the unedited fit. For the shuffled names we also count, among texts that the unedited read gives exactly one claim, how many keep that claim (following the definition) and how many move to the category that now carries its name. Hiding the names changes claim-set ARI by $-.002$, $-.005$, and $-.013$ (only the HWU64 interval, $[-.018, -.007]$, lies below zero) and F1 by at most .002. Shuffling the names lowers ARI by .083, .090, and .057 and F1 by .088, .088, and .058, on every seed, yet 85.1%, 88.1%, and 85.8% of the single claims keep the category whose definition they match and only 10.3%, 8.9%, and 6.4% follow the moved name. Keeping only the names changes ARI by $+.003$, $-.013$, and $+.002$ (no interval excludes zero) and F1 by $-.005$, $-.017$, and $-.006$ (only the CLINC150 interval lies below zero; CLINC150 ARI is lower on 9 of 10 seeds), and it lowers the uncovered share by .008, .009, and .022, which raises flat ARI on BANKING77 ($+.013$ $[+.004, +.022]$) and HWU64 ($+.027$ $[+.013, +.041]$), each on 9 of 10 seeds. On these intent benchmarks names and definitions are therefore largely redundant: either alone keeps claim-set ARI within .013 of the full codebook, and when they conflict the reader follows the definition for most single claims, although a misleading name still costs .057 to .090 ARI. The probes do not show that the definitions and rules raise accuracy when the names are informative, and whether they make decisions easier for people to inspect was not tested. They also edit only the frozen codebook at application time: the names were written and verified together with the full entries during fitting, so the probes do not show whether names alone would suffice for discovery. No probe compares LCD's entries with TnT-LLM's name-and-description entries.

**Name copying.** LCD gives 0.2–1.2% of its categories a name equal to a reference label, against 1.7–8.2% for TnT-LLM (Table 16). The difference holds only for exact copying: the share of names

Table 16: **On these benchmarks names and definitions are largely redundant: hiding either changes ARI by at most .013, while shuffled names lower ARI by .057–.090 although 85–88% of single claims keep the category whose definition they match.** Name copying: share of final category names equal to a reference label name after normalization (LCD 0.2–1.2%, TnT-LLM 1.7–8.2%), and share containing every word of one (similar for both); LCD writes more names. Probes, long codebooks and ADD order: claim-set ARI / mapped macro-F1 of the edited read minus the unedited LCD read on the test split, paired by seed (strict reader; probes: 10 seeds; ADD order: BANKING77, seeds 11–15); green▲ / red▼: 95% interval above / below zero. Single claims: test texts with exactly one claim in the unedited read that keep that category (its definition) or move to the category now carrying its name. Long codebooks: CLINC150, seeds 11–13, lower on every seed but exact $p = .125$, so unmarked (Appendix B.5); texts claimed: share of CLINC150 test texts with a claim on a distractor category; *CLINC150's banking and credit-card domains (30 intents, the subject matter of BANKING77) against its other domains.

| Check | BANKING77 | CLINC150 | HWU64 |
|---|---|---|---|
| *Name copying (*LCD */ TnT-LLM)* | | | |
| names equal to a reference label | **1.2%** / 5.5% | **0.8%** / 8.2% | **0.2%** / 1.7% |
| names containing every word of a reference label | 23.4% / 22.5% | 45.2% / 48.6% | 3.9% / 4.8% |
| *Name probes (ΔARI / ΔF1)* | | | |
| names replaced by "Category $k$" | −.002 / .000 | −.005 / −.002 | −.013▼ / −.002 |
| names shuffled | −.083▼ / −.088▼ | −.090▼ / −.088▼ | −.057▼ / −.058▼ |
| name only (no definition) | +.003 / −.005 | −.013 / −.017▼ | +.002 / −.006 |
| names shuffled: single claims by definition / name | 85.1% / 10.3% | 88.1% / 8.9% | 85.8% / 6.4% |
| *Long codebooks (ΔARI / ΔF1)* | | | |
| + BANKING77 distractors | – | −.009 / −.009 | – |
| texts claimed, banking-domain* / other | – | 8.0% / 0.3% | – |
| + BANKING77 and HWU64 distractors | – | −.032 / −.049 | – |
| texts claimed, banking-domain* / other | – | 9.2% / 9.5% | – |
| ADD *order (ΔARI / ΔF1)* | | | |
| candidates shuffled (refit) | +.003 / +.006 | – | – |

that contain every word of some reference label is similar for the two methods (23.4%, 45.2%, and 3.9% against 22.5%, 48.6%, and 4.8%). The shares have different denominators, because LCD writes more and finer names (1,002 against 764 names over the ten BANKING77 fits). Neither measure indicates that LCD was less exposed to these benchmarks in pretraining, and neither can rule such exposure out.

**Long codebooks.** The CLINC150 codebook of seeds 11–13 is read together with the codebook of the same seed's BANKING77 fit, and then with those of BANKING77 and HWU64, as distractors; a distractor identical to a CLINC150 category is dropped. Scoring uses the CLINC150 mapping, so a claim on a distractor never maps. CLINC150's banking and credit-card domains (30 of its 150 intents, for example transfers, frozen accounts, PIN changes, and lost cards) cover the subject matter of BANKING77, so a distractor claim on such a text can be correct in meaning; we report those claims separately from the others. The distractors add 96.7 and 194.0 categories, so each request shows 276.3 and 373.7 categories. They lower CLINC150 claim-set ARI by .009 and .032 and F1 by .009 and .049; every seed is lower, but with three seeds the exact test cannot go below $p = .125$. BANKING77 distractors claim 1.8% of the test texts, 88% of them in the banking and credit-card domains, where 8.0% of texts receive a distractor claim against 0.3% elsewhere. With HWU64's codebook added, whose scenarios (for example alarms, calendar, and weather) overlap several other CLINC150 domains, 9.4% of texts receive a distractor claim, 9.2% in the banking and credit-card domains and 9.5% elsewhere. The distractors also claim some texts that the unedited read leaves uncovered (uncovered share −.006 and −.012). Prompts grow from 30.3k to 45.5k and 59.4k tokens per request, while a request takes 7.9 and 7.8 seconds against 8.3 for the unedited read. Longer codebooks thus cost some accuracy, and distractor claims fall mainly on texts that share their subject matter.

**ADD order.** ADD admits its candidates one at a time and updates the membership matrix after each admission, so the order can matter. We refit BANKING77 seeds 11–15 with the candidate list shuffled and everything else unchanged, and compare each refit with the fit of the same seed on the test split. The shuffled refits change claim-set ARI by $+.003$ $[-.015, +.020]$, mapped F1 by $+.006$ $[-.015, +.027]$, and $K$ by $-1.4$ $[-7.8, +5.0]$ (98.2 against 99.6 classes), with exact $p \geq .34$ in either direction. The order has no detectable systematic effect, but it moves single fits by up to .027 ARI ($-.023$ to $+.027$ per seed), about the seed-to-seed spread of LCD on BANKING77 (standard deviation .019; Appendix C.3).

## E ENTRIES OF A FINAL CODEBOOK

This appendix reproduces entries of one final LCD codebook verbatim, the HWU64 fit of seed 11 (111 member entries, 110 reported classes). Every field is shown as the model wrote it; $s$ is the number of discovery texts the reader assigns to the entry in the matrix of record. Table 17 shows an ordinary entry, Table 18 the two entries of a GROUP class, and Table 19 a hierarchical catch-all split.

Table 17: **An entry states what it includes and what it excludes, and its exclusion examples are nearby requests that belong elsewhere.** An entry kept unchanged from the selected first codebook ($s = 7$).

| Field | Text |
|---|---|
| *Single entry* | |
| Name | **Tell a joke** |
| Definition | Request for humorous entertainment in the form of a joke. |
| Inclusion rule | Include when the speaker asks the assistant to tell or provide a joke. |
| Exclusion rule | Exclude general requests for music, games, or other entertainment without a joke request. |
| Inclusion examples | "make me laugh i want to hear a joke" |
| Exclusion examples | "challenge you to a game of tic tac toe"; "nice music" |

Table 18: **GROUP reports these two entries as one class; their written rules still separate the performer from the songwriter.** Both descend by SPLIT from one earlier entry. Singly claimed discovery texts: 12 and 8; crossing $\chi = 0.85$, closure 0.55, centroid cosine 0.84 (Eq. (17)). The reported column is the union of the two verified columns; nothing is rewritten.

| Field | Text |
|---|---|
| *First member ($s = 12$)* | |
| Name | **Identify song performer** |
| Definition | Request to identify the singer, artist, or performing act associated with a specified or currently playing song. |
| Inclusion rule | Include when the speaker asks who sings, performs, or is the artist of a named, referenced, or currently playing song. |
| Exclusion rule | Exclude requests to identify a song's title, songwriter, composer, music director, or source; exclude opinions about music without an identification request. |
| Inclusion examples | "who sing we are the world"; "is this song ed sheeran" |
| Exclusion examples | "what is the name of that song that just went off"; "who wrote the song that is playing" |
| *Second member ($s = 8$)* | |
| Name | **Identify song creator or production contributor** |
| Definition | Request to identify the songwriter, composer, or music director responsible for a specified or currently playing song. |
| Inclusion rule | Include when the speaker asks who wrote, composed, or served as music director for a named, referenced, or currently playing song. |
| Exclusion rule | Exclude requests to identify the singer or performing artist, or to identify the song's title or source. |
| Inclusion examples | "who wrote the song that is playing"; "who is the music director of this song" |
| Exclusion examples | "name the singer"; "what artist is singing that song" |

## F RELATED WORK IN FULL

**Category discovery with language descriptions.** Concept and category discovery methods increasingly use language as part of the object being learned, rather than only as a label attached after clustering. LLooM induces concepts with explicit inclusion criteria, scores every text against them, and uses uncovered texts to guide further concept discovery (Lam et al., 2024). TopicGPT uses natural-language topic descriptions as the topic representation itself (Pham et al., 2024). GoalEx

Table 19: **The hierarchical catch-all split separates calculation requests from a broad factual-question entry; the parent's texts that no child claims stay under the parent's own rules.** The detector flagged the parent (share 3.4% of the discovery texts, cohesion ratio 0.51); of 5 proposed children, 2 claimed at least three of its 48 texts and were kept, together covering 81%. At application, stage one reads the parent under its own entry; stage two reads the texts it claims with the two children only, and a text neither child claims is reported under the remainder, which keeps the parent's six fields ($s = 9$). The reader had assigned the parent 4 calculation requests despite its exclusion rule (for example "how many cups in a quart"); the split moves them to the second child.

| Field | Text |
|---|---|
| *Parent, read at stage one; also the remainder ($s = 9$)* | |
| Name | **Answer general factual knowledge question** |
| Definition | Request for factual information about people, places, events, objects, concepts, or general descriptive topics. |
| Inclusion rule | Include standalone questions seeking general factual or descriptive knowledge, including biographies, history, geography, science, sports results, current public-office holders, and explanations of general concepts, when no specialized calculation or instruction is requested. |
| Exclusion rule | Exclude arithmetic or date/time calculations, weather or other live local conditions, cooking or other how-to instructions, lyric identification, and requests about the assistant's capabilities or personal information. |
| Inclusion examples | "was einstein married"; "where is stonehenge" |
| Exclusion examples | "what is the square root of forty two"; "how do i cook rice" |
| *First child ($s = 35$)* | |
| Name | **Answer general factual knowledge question** |
| Definition | Request for factual or descriptive information about people, places, events, objects, concepts, or general topics. |
| Inclusion rule | Include questions seeking general knowledge, including biographies, history, geography, science, sports results, public figures, definitions, locations, dates, counts, and descriptions, when no specialized calculation, recommendation, or instruction is requested. |
| Exclusion rule | Exclude requests for arithmetic or date/time calculations, financial advice or valuation, health or nutrition calculations or plans, image retrieval or generation, and how-to instructions. |
| Inclusion examples | "was einstein married"; "where is stonehenge" |
| Exclusion examples | "what is the square root of forty two"; "how do i cook rice" |
| *Second child ($s = 4$)* | |
| Name | **Perform calculation or calendar computation** |
| Definition | Request to calculate a numerical, measurement, date, time, or calendar result. |
| Inclusion rule | Include requests requiring arithmetic, unit conversion, date arithmetic, calendar lookup or computation, or mathematical evaluation, with the required object being the specified quantity, expression, interval, or date. |
| Exclusion rule | Exclude factual questions that merely ask for a known count or date about a person, place, event, or object without requiring computation. |
| Inclusion examples | "how many cups in a quart"; "tell what is the value of sin thirty" |
| Exclusion examples | "how old is tom hanks"; "when was frank sinatra born" |

proposes descriptions for candidate clusters, assigns texts to descriptions, and selects the final set with an objective that rewards coverage and reduces overlap (Wang et al., 2023). IDAS clusters encodings of LLM-generated abstractive summaries of utterances to recover latent intents (De Raedt et al., 2023), and ClusterLLM uses LLM feedback to refine an embedding clustering and to choose its granularity (Zhang et al., 2023); new intent discovery has also been studied under imbalanced class distributions (Zhang et al., 2024). LCD shares the emphasis on language-defined membership and coverage, but its codebook changes only through operator edits whose memberships, verified by a joint reader on all discovery texts, pass explicit acceptance tests; texts contested between a candidate and an existing category are re-read with both, and guards prevent catch-all categories and take-overs. Its output is not a partition of the discovery texts but a frozen codebook that is applied, as written, to texts that arrive later.

**Taxonomy induction and classification.** TnT-LLM generates and refines a label taxonomy with an LLM over minibatches of texts, then uses LLM pseudo-labels to train lightweight classifiers (Wan et al., 2024). By contrast, LCD applies the complete written rules directly, so the deployed artifact is the codebook itself. TaxoClass starts from a supplied hierarchy and class names for hierarchical multi-label classification (Shen et al., 2021), and LLMs have been used directly as classifiers over large given hierarchies of scientific documents (Tabatabaei et al., 2025). These systems help locate LCD between taxonomy construction, clustering and classification: the inventory is not given in advance, and the learned output remains an inspectable set of category rules.

**Label semantics and class descriptions.** Intent classification has long used the semantics of labels: label-aware pre-training (Mueller et al., 2022), training on label descriptions for zero-shot classification (Gao et al., 2023), description-augmented dataless classification (Hu et al., 2024), and zero- and few-shot prompting (Parikh et al., 2023), alongside example-driven and contrastive few-shot

methods (Mehri & Eric, 2021; Zhang et al., 2021). With many labels, in-context classification often retrieves a subset of the label space for each input (Milios et al., 2023). Closest to LCD are methods that rewrite the descriptions themselves: dynamic label-name refinement from retrieved labeled examples (Park et al., 2025), multi-agent refinement of class descriptions from misclassifications (Dev et al., 2026), iterative contrastive refinement from a small labeled sample (Rajeev et al., 2025), and rules generated for confusable label pairs (Gupta et al., 2026). All of these refine descriptions of a given label set. LCD discovers the categories, writes their inclusion and exclusion rules, and resolves overlaps between them (REVISE and the contest) without labels; its reader sees the complete codebook in one prompt rather than a retrieved subset.

**Definition adherence.** Whether an LLM follows a written codebook is itself an empirical question. Halterman and Keith propose a staged framework for codebook-LLM measurement whose behavioral tests replace category labels with non-informative ones and permute labels across definitions (Halterman & Keith, 2026), and argue that conceptualization remains a first-order concern for LLM-era classification (Halterman & Keith, 2025). In political event coding, enriched codebook guidance raises macro-F1, yet no evaluated method exceeds 0.20 weighted $F_1$ once label names are reassigned to other definitions (He et al., 2026b). Supplied label definitions can improve accuracy, but their use is neither guaranteed nor consistent (Mohammadi et al., 2025); many zero-shot errors resist correction by additional instructions (Casanova et al., 2026); and agreement with human coders does not by itself establish construct validity (Pita, 2026). Work on in-context learning points the same way: demonstration labels can matter less than expected (Min et al., 2022), the ability to override semantic priors grows with scale (Wei et al., 2023), models can learn from irrelevant or misleading prompts (Webson & Pavlick, 2022), few-shot predictions carry systematic biases (Zhao et al., 2021), and prompt design changes the compliance and accuracy of LLM annotations (Atreja et al., 2025). Our name probes follow the label-removal and label-permutation designs of this literature (Table 16): on our intent benchmarks, hiding the names or the definitions changes ARI by at most .013, whereas permuting the names lowers it by .057 to .090 although 85–88% of single claims keep the category whose definition they match. We also report how often discovered names copy reference label names, exactly or word for word.

**Reliability of LLM-assisted coding.** LLM annotation in computational social science is usually validated against human labels (Gilardi et al., 2023; Ziems et al., 2024; Zhu et al., 2023), and validation task by task has been argued to be necessary (Pangakis et al., 2023; Törnberg, 2024). Outputs can vary with repeated inputs and small prompt changes (Reiss, 2023), which motivates prompt-stability scoring adapted from intra- and inter-coder reliability (Barrie et al., 2024); configuration choices alone can change downstream conclusions (Baumann et al., 2025). LLMs have been used to support the qualitative analysis of text (Tai et al., 2024), including deductive coding with a given codebook (Xiao et al., 2023; Chew et al., 2023), and have been evaluated against human coders (Dunivin, 2024; Hila & Hauser, 2025), including in a benchmark of 46 models against expert adjudication (Marston et al., 2026). Other systems keep humans in the loop while a codebook is developed or revised: CollabCoder supports inductive collaborative analysis through to codebook creation (Gao et al., 2024), cross-model disagreement can target expert effort in codebook revision (He et al., 2026a), and CALICO treats codebook-aligned prompts as editable, versioned artifacts (Yuan et al., 2026). In the qualitative tradition, codebooks are developed from theory and data and then used to train coders and assess reliability (DeCuir-Gunby et al., 2011). LCD adopts the premise that category definitions should be operational rules, but places discovery before application: the codebook is written from unlabeled texts, and reproducible application of the frozen rules is then an evaluation requirement. Our second reader is a second model, not a human annotator, so this measurement compares model readers rather than humans.

**Concept-based models.** Concept Bottleneck Models predict training-provided concepts before predicting labels and allow interventions on those concepts (Koh et al., 2020). LFD learns discriminative textual features for known labels and screens them for agreement and label disentanglement (Wang et al., 2026). LCD differs in what it outputs: not feature coordinates for a known label space, but the category inventory and its written membership rules. The shared interest is interpretability through human-meaningful intermediate representations; in LCD, that representation is a codebook evaluated through its coverage, overlap and application to held-out texts.