LEARNING FROM USER EDITS IN DICTATION SYSTEMS: ATTRIBUTION AND ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Users of deployed multi-stage systems edit what the system produced, and it is tempting to learn from every edit as if it were a correction. We test this assumption on a production dictation pipeline (speech recognition followed by LLM post-processing) with 900 edits from 45 users. In a non-random human-labeled subset of 359 edits, 47.9% added or removed content rather than correcting a mishearing. The cause of an edit can be estimated: an LLM that sees the logs and an independent transcript of the recording assigns the same category as a human annotator for 81.0% of held-out edits, against 50.2% for always predicting the most common category, although most labels were assigned with a transcript-based suggestion visible. In an adaptation study scored against proxy references, vocabulary learned from edits reduced errors on corrected utterances by 2.2–3.0 points, whereas negative examples learned from every edit increased errors, an increase a small hand-transcribed subset did not confirm. Routing updates by the estimated cause never beat learning vocabulary from every edit. Updates learned from edits need their own validation.

1 INTRODUCTION

Deployed systems built from several model stages—speech pipelines, retrieval-augmented generation, tool-using agents—produce outputs that users then accept, edit, or discard. These reactions are free, plentiful, and arrive without any annotation effort, which makes them an attractive training signal (Joachims et al., 2017; Gao et al., 2024). The simplest way to use them is to treat every edit as a correction: the user changed the output, so the output must have been wrong, so the system should learn to produce what the user wrote.

This reasoning leaves two questions open. First, *did any stage actually make an error?* A user may edit an output because they changed their mind, remembered something to add, or simply prefer a different phrasing, and none of these says anything about the system's accuracy. Second, *if a stage did err, which one?* In a multi-stage system an edit only reveals the final discrepancy, not where it was introduced. We call the task of answering these questions *provenance inference*: recovering the likely cause of an edit from what the system already observes (Figure 1). A third question follows: even when the cause is known, *does the update it suggests actually improve the system?* We treat that as an empirical question for each kind of update.

We study these questions on a production dictation pipeline, where audio is transcribed by a speech recognizer and the transcript is cleaned up by an LLM post-processing stage before the user sees it. The recording is available as independent evidence of what the user actually said, and the realistic way to adapt such a system is non-parametric (adding vocabulary, adding examples or rules to a prompt), because the stages are third-party models that cannot be fine-tuned. Our main contributions, on 900 real edits from 45 users:

- **A measurement of edit types in deployment.** In a human-labeled subset of 359 edits, 47.9% were labeled as adding content the user did not say or removing content they did say, and 32.6% as correcting a misheard word or term. The subset is not a random sample (the share ranges from 40% to 64% across its parts; Section 5), and the label does not establish that the rest of the output was error-free.

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.

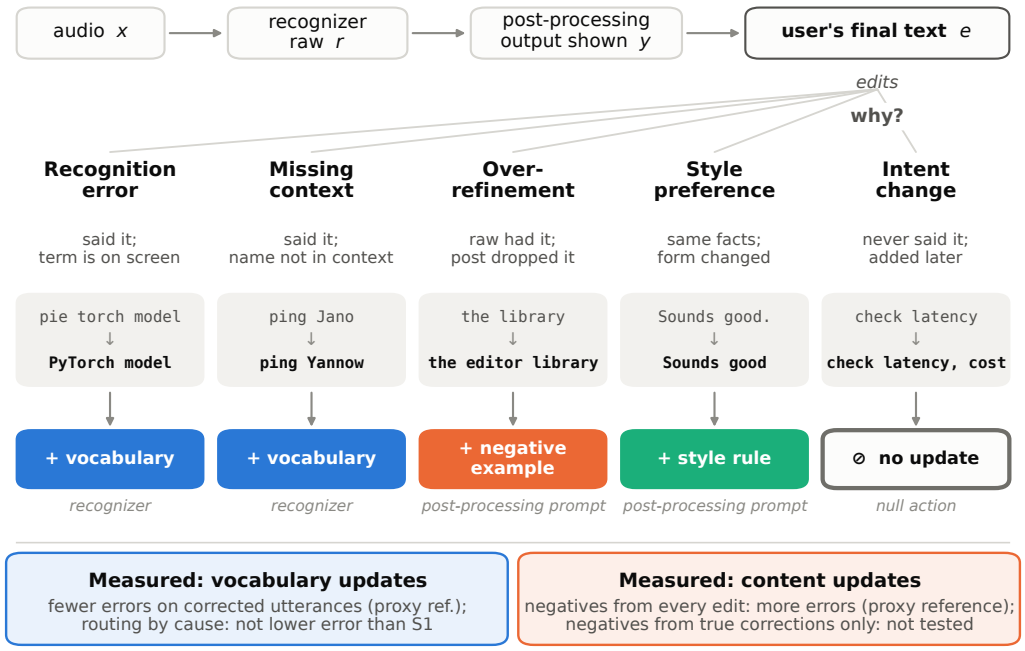

Figure 1: Why did the user edit? Five candidate causes of an edit to a dictation, each paired with one candidate update (examples are constructed). Evidence that separates the causes includes the recording itself (did the user say it?). OVER-REFINEMENT is shown for completeness: no human-labeled edit received it (Section 6). Bottom: what our adaptation study measured, against proxy references (Section 7). Only vocabulary updates showed a benefit, and learning negative examples from genuine corrections alone was never exercised.

- **Evidence for attribution** (Section 6). A prompted LLM agrees with the human labels on 81.0% of held-out edits (majority class 50.2%); adding an independent transcript of the recording raises agreement by 5.4 points, mostly on edits where the annotator had overridden a rule's suggestion. This is agreement with suggestion-assisted labels, not recovery of the true cause.
- **A validation of candidate updates** (Section 7). In an adaptation study run twice, vocabulary learned from edits reduced errors on corrected utterances, negative examples learned from every edit increased errors against proxy references, and no routing by cause had lower error than learning from every edit; routing with the LLM classifier had higher error.
- **An audit of what the evaluation can show** (appendices). Against hand transcripts the negative-example increase is not confirmed, and the difference between references is itself nonzero; an update-action audit shows that negative examples from genuine corrections were never learned, so that update remains untested.

An edit is an observation with several possible causes, not a label; how well its cause can be estimated, and whether an update learned from it helps, are separate empirical questions.

2 PROBLEM FORMULATION

Setting. A deployed system maps an input x through stages $s_1, \ldots, s_n$ to an output shown to the user, and any stage may also read context c. In our pipeline $n = 2$. A recognizer produces a transcript $r = s_1(x, c)$ from audio x; in production its prompt already carries the context (application, text around the cursor, user vocabulary), so r is a context-conditioned, lightly cleaned transcript rather than a raw acoustic hypothesis. A post-processing stage then produces $y = s_2(r)$ by applying the user's formatting preferences. The user may edit y into $e \neq y$. The stages are fixed third-party models, so every update must be non-parametric.

Table 1: Provenance categories and the candidate update each is paired with.

Provenance z	Candidate update $g(z)$	Adaptation surface
RECOGNITION, MISSING-CONTEXT	add term to vocabulary	recognizer (native vocabulary parameter)
OVER-REFINEMENT	add negative example	post-processing prompt
STYLE	add style rule	post-processing prompt
INTENT-CHANGE	$a_\varnothing$ (do nothing)	—

Provenance. The provenance z of an edit is its latent cause. We use five categories $\mathcal{Z}$:

- RECOGNITION: the pipeline misheard something the user said (an ordinary word, or a term that appears in the available context but was still missed);
- MISSING-CONTEXT: the user said a name or term that appears nowhere in the available context;
- OVER-REFINEMENT: post-processing changed correct content into incorrect content;
- STYLE: the facts are right, and the user changed punctuation, fillers, repetitions, function words, or formatting;
- INTENT-CHANGE: the user added content they did not say, or removed content they did say (historically defined as "nothing was wrong"; see below).

These are the definitions our annotators followed (Section A). Three operational details matter for interpreting the results. First, "available context" means the on-screen text shown to annotators and classifiers; the user vocabulary is logged for 643 of the 900 edits but was not consulted when labeling or classifying. Second, because the recognizer already uses context, a first-stage cleanup error is attributed to RECOGNITION, not to post-processing. Third, each edit receives one label, although a single edit can combine a misheard word with added content (Section 9). An INTENT-CHANGE label therefore means that the edit's labeled change was not dictated; it does not mean that the output contained no system error, even though our guideline, prompt and E1 reference treat it that way. The LLM classifier's prompt was written with slightly different boundaries for OVER-REFINEMENT and STYLE (Section 3).

Provenance is not observed; it is estimated from what the system has, $f : (c, r, y, e, x) \mapsto \hat{z}$, including the recording x.

Candidate updates. A routing function $g : \mathcal{Z} \to \mathcal{A}$ maps each cause to a candidate update. Because the stages are fixed, the updates are non-parametric: add a term to the recognizer's vocabulary, add a negative example or a style rule to the post-processing prompt, or the null action $a_\varnothing$ (do not update). Table 1 pairs each category with one candidate update. Whether that update actually helps is a separate, empirical question, and our adaptation study (Section 7) answers it only for the updates that were actually exercised. A naive learner applies a fixed update to every edit without estimating f.

Scope. The formulation is stage-agnostic; the categories and actions are specific to speech, and we do not claim they transfer to other systems.

3 INFERRING PROVENANCE

Given an edit, a provenance classifier f decides which of the five causes produced it. We compare three kinds of classifier, each built on a different source of evidence: which stage the error survived, what the recording contains, and a language model's reading of the text. None of them is trained.

Stage-diff rule: which stage did the error survive? This was our initial heuristic, and it is the cheapest possible signal. If post-processing left the transcript unchanged ($r = y$) and the user still edited it, the rule blames the recognizer. If post-processing changed the text and moved it further from what the user wanted, it blames post-processing. The rule implicitly assumes that every edit is a correction. Because it only needs the logged stage outputs, we evaluate it on a spot-check rather than on the held-out split (Section 6).

Audio-grounded rule: what does the recording say? The recording is an independent witness to what the user actually said. We transcribe it a second time with a recognizer that sees no context (Whisper large-v3; Radford et al., 2023). Whisper detects the language poorly on short clips, so we infer the language from the displayed text and pass it to Whisper explicitly. The rule then compares the displayed text y with the edit e and asks three questions about each changed span, in order:

1. *Did only the form change?* If the change touches only punctuation or casing, or only deletes filler words or repetitions, the span is STYLE. This is narrower than the annotation guideline: function-word and number-format changes fall through to the next questions.
2. *Did the user say the new words?* If the edited words appear in the independent transcript and the original words do not, the user said them and the pipeline misheard. Likewise if the original and edited words sound alike (compared as toneless pinyin for Chinese and as spelling otherwise).
3. *Otherwise,* the recording does not support the change: INTENT-CHANGE.

A mishearing is MISSING-CONTEXT when the corrected words look like a name or term, the independent recognizer also failed to produce them, and they appear nowhere in the on-screen context; otherwise it is RECOGNITION. An edit counts as INTENT-CHANGE if any of its spans does. The rule's thresholds were set on the development split and then frozen.

Prompted LLM: what does a careful reader of the text conclude? A general-purpose LLM receives the transcript r, the displayed text y, the on-screen context, the edit e, whether y differs from r, and two text-similarity scores. It returns one of the five categories with a short justification (the prompt is in Section B). We test three versions of the prompt, each adding one thing to the previous:

- **v1:** the definitions of the five categories;
- **v2:** v1 plus our annotation guidelines for the boundary between INTENT-CHANGE and STYLE;
- **v2+audio:** v2 plus the independent Whisper transcript, with instructions on how to use it as evidence.

The prompt's category definitions predate the final annotation guidelines and differ from them in two places: its OVER-REFINEMENT also covers a post-processor that replaced an already wrong transcript with a different wrong one, and its STYLE includes some wording changes (its Example C). We report the prompts exactly as they were run.

4 ADAPTING FROM EDITS

E1 asks which updates learned from edits improve the system, and whether knowing *why* an edit happened helps choose them. We simulate a system that keeps learning from its users' edits and compare strategies that differ in one thing only: the rule that decides which update to make.

How the simulated system learns. Each user has their own record of what has been learned, holding three lists: vocabulary terms, negative examples, and style rules. The record is per user because in production all three are user-scoped. We replay a user's edits in the order they happened; each edit may add zero or more entries to the record. To evaluate, we freeze the record and re-run the pipeline on that user's later recordings: the vocabulary goes to the recognizer through its native keyword parameter, and the most recent 20 negative examples and 20 style rules are appended to the post-processing prompt.

Strategies. All strategies draw on the same set of updates; they differ only in the decision rule. Section G lists which updates each strategy actually stored.

- **S0, no learning:** ignores every edit.
- **S1, learn from every edit:** never skips an edit. If the edit swaps short spans (at most four tokens on each side), the new spans become vocabulary terms; any other edit is stored as a negative example ("the system wrote y, the user wanted e"). S1 never writes style rules.

Table 2: Labels of the 359 human-labeled edits. This subset was selected partly by earlier heuristics; see Section D for its composition. Post-processing changed the text of 115 of these edits, but none was labeled OVER-REFINEMENT.

Provenance	Edits	Share
INTENT-CHANGE	172	47.9%
RECOGNITION	84	23.4%
STYLE	70	19.5%
MISSING-CONTEXT	33	9.2%
OVER-REFINEMENT	0	0.0%

- **S2, surface filter:** S1, but first discards edits whose edited text is short (below the 25th percentile of its length in tokens) or whose changed fraction is large (above the 75th percentile). The original design also discarded edits with low recognizer confidence, which the logs do not record.
- **S3, route by estimated cause:** classifies each edit and applies the candidate update from Table 1; INTENT-CHANGE triggers no update. We run it with two classifiers: the audio-grounded rule, which has the fewest intent-change edits routed to an update in E0, and the LLM with the transcript, which agrees most with the human labels.
- **S4, route by adaptation-set labels:** S3 with the reference labels of the adaptation edits in place of a classifier. At the time of the runs, 81.3% of these labels (460 of 566) were the audio rule's own output and 18.7% were human labels, so S4 is neither an oracle nor an upper bound on routing.

To see where S1's effects come from, we also split it into its two halves: **S1-vocabulary** and **S1-negatives**. **S3+** and **S4+**, defined before we ran them, learn vocabulary from every edit as S1 does and route only the other updates by cause (Section H).

5 DATA AND SETUP

Production pipeline. The system is a dictation application. For each utterance a multimodal recognizer produces a cleaned transcript from audio and a prompt carrying the application, the text around the cursor, and a user vocabulary (gpt-4o-transcribe for 467 of our edits, gpt-transcribe for 375, other or unrecorded models for the rest). A second LLM stage (gemini-3.5-flash) applies the user's formatting preferences. The logs contain the output of each stage and the user's final edit.

Edits. We collected 900 edited utterances from 45 users between 2026-08-27 and 2026-09-23, all on macOS. We kept edits of moderate size; very large rewrites were dominated by non-dictation artifacts such as pasted text. Speech is English and Chinese, often code-switched within a sentence: 499 of the 900 transcripts contain Chinese. Recordings are available for 875. Users are unevenly active: the 13 most active users contribute 762 edits.

Annotation. One annotator labeled edits while listening to the recording, following written guidelines (Section A); unintelligible edits were excluded. The resulting 359 labels (Table 2) come from four sheets that were drawn and presented differently (Section D). Only 20 were drawn uniformly at random; 200 were labeled by accepting or changing the audio rule's suggestion. The share of INTENT-CHANGE is 40%, 64%, 50% and 50% in these four parts, so Table 2 describes this labeled subset, not the 900 edits.

A second annotator labeled 50 of the labeled edits blind, from the written guidelines alone; for 28 of these, the first annotator's label had been given with the rule's suggestion visible. Agreement was 70% on the five categories (Cohen's $\kappa = 0.50$, 95% bootstrap interval [0.30, 0.68]) and 82% on whether an edit is a change of intent ($\kappa = 0.63$, [0.40, 0.84]). The most frequent disagreements were INTENT-CHANGE versus STYLE (5 of 15) and INTENT-CHANGE versus a mishearing (4); the second annotator never used MISSING-CONTEXT.

Table 3: Agreement of provenance classifiers with the human labels on the held-out split (205 edits, none used to design any classifier; 200 of the labels were assigned with the audio rule's suggestion visible). LLM: Gemini 3.8 Flash, one run per prompt. Brackets: 95% bootstrap intervals over edits. †On the reviewed edits the rule's agreement equals the share of its suggestions the annotator accepted.

Classifier	5-way agreement	Action agreement	INTENT-CHANGE → update	Correction → none
Majority class (INTENT-CHANGE)	50.2	50.2	—	—
LLM, text only (v1)	72.2 [65.9, 78.0]	77.6	29%	11%
LLM + guidelines (v2)	75.6 [69.8, 81.5]	81.5	10%	25%
LLM + guidelines + transcript (v2+audio)	81.0 [75.6, 86.3]	84.9	13%	16%
Audio-grounded rule†	75.6 [69.8, 81.5]	80.5	2%	36%

E0 protocol. A random split would place edits we had already tuned on into the test set, so we split by use: every edit that informed a design choice (the audio rule's thresholds or the prompt variants) forms the development split (154 edits); every other human-labeled edit forms the held-out split (205 edits), on which each classifier was scored once. 200 held-out labels were assigned with the audio rule's suggestion visible, so E0 measures agreement with these labels, not recovery of the true cause (Section 6). The LLM is Gemini 3.8 Flash with thinking at its default setting.

E1 protocol. We simulate adaptation with a pipeline that mirrors production's structure: a recognizer with a native vocabulary parameter (gemini-3.5-transcribe, verbatim mode) followed by gemini-3.5-flash with the production cleanup prompt adapted to text input. Unlike production, the context goes only to post-processing, and the users' existing production vocabularies are not loaded. We keep the stages separate because the recognizer's cleanup mode accepts no prompt and dropped clauses when combined with a vocabulary (Section E). For the 22 users with at least five edits, each user's edits are ordered in time; the first 70% (566) are used for adaptation and the last 30% (245) for testing. Utterances without intelligible speech or with non-dictation artifacts are excluded. All test labels are human: 41 given with no suggestion shown, 6 with the stage-diff rule's label shown, and 198 by accepting or changing the audio rule's suggestion.

Metrics. We score each test utterance against a proxy for what was said: the user's edit for RECOGNITION, MISSING-CONTEXT and STYLE edits, and the production output for INTENT-CHANGE edits. The latter treats the production output as correct, so any recognition error remaining in an INTENT-CHANGE utterance is part of the reference. On a random 59 of the test utterances that we transcribed by hand, this proxy is 0.070–0.091 MER from the hand transcript, and scoring INTENT-CHANGE utterances against the user's edit instead would put the reference 0.290 away (Section F). We report the mixed error rate (MER: Chinese characters and words, punctuation ignored), paired per utterance against S0, with 95% intervals from a bootstrap that resamples users. Because punctuation is ignored, MER barely registers STYLE corrections: on the 42 STYLE test utterances the production output is 0.005 MER from the user's edit (character error rate with punctuation: 0.062). The corrected subset therefore mostly measures content corrections.

6 E0: WHAT EVIDENCE DISTINGUISHES EDIT TYPES?

The stage-diff rule mostly picks up changes of intent. On a spot-check of edits the stage-diff rule attributed to the recognizer, the annotator agreed on 5 of 21; 13 were changes of intent. In this spot-check most such edits were new content.

Classifier comparison. Table 3 compares the classifiers. Because RECOGNITION and MISSING-CONTEXT lead to the same candidate update, we also report agreement on the chosen action, and two routing rates: the share of INTENT-CHANGE-labeled edits that a classifier would route to an update, and the share of edits labeled as a correction that it would route to no update. These count routing decisions, not effects of learning.

Classifiers agree with the labels well above the majority class. Every classifier exceeds the majority class by 22–31 points. For scale, two human annotators agreed on 70% of a separate

Table 4: Change in MER relative to S0 on 245 held-out utterances from 22 users (negative is better; proxy references), averaged over two independent runs of every strategy; 95% intervals from a user-level bootstrap. Run-to-run variation is described in Section 9. Section H plots these and the pre-registered variants.

Strategy	Corrected (124)	INTENT-CHANGE (121)	All (245)
S1: learn from every edit (vocabulary + negative examples)	−0.030 [−0.054, −0.006]	−0.003 [−0.026, +0.022]	−0.017 [−0.035, +0.006]
S2: surface filter, then S1	−0.026 [−0.052, −0.000]	−0.001 [−0.018, +0.020]	−0.014 [−0.031, +0.007]
S3: route by audio-rule label	−0.022 [−0.047, −0.001]	−0.001 [−0.026, +0.027]	−0.011 [−0.030, +0.009]
S3: route by LLM (v2+audio) label	−0.015 [−0.028, +0.005]	+0.025 [−0.010, +0.066]	+0.005 [−0.016, +0.036]
S4: route by adaptation-set labels‡	−0.025 [−0.046, −0.006]	+0.003 [−0.022, +0.028]	−0.011 [−0.029, +0.009]
Decomposing S1			
S1, vocabulary updates only	−0.029 [−0.049, −0.006]	−0.001 [−0.025, +0.025]	−0.015 [−0.033, +0.007]
S1, negative examples only	+0.008 [+0.003, +0.014]	+0.004 [−0.000, +0.008]	+0.006 [+0.002, +0.010]

‡81.3% of the adaptation labels are the audio rule's output; S4 is not an oracle (Section 4).

sample (Section 5). These agreement rates show that the logs carry evidence about the edit type, not that the classifiers recover the true cause.

Guidelines fix definitions; the transcript recovers mishearings. Writing the annotation guidelines into the prompt mostly removes confusions about definitions and lowers the share of INTENT-CHANGE-labeled edits routed to an update from 29% to 10%. Adding the independent transcript mostly helps the other way. In text, a word the recognizer misheard looks like a word the user added afterwards, and the transcript separates them: the share of corrections routed to no update drops from 25% to 16%, and five-way agreement rises by 5.4 points (paired 95% interval [+0.5, +10.2]), the only pairwise difference whose interval excludes zero. On development data the same ordering held with a different model family (Section C).

What the held-out labels can and cannot show. The audio rule's row is not an independent measurement: on the 200 reviewed edits its agreement is, by construction, the annotator's acceptance rate (151 of 200). A related concern is that the v2+audio prompt reads the same Whisper transcript that produced those suggestions. Its gain over v2 is, however, concentrated on the 49 edits where the annotator rejected the rule's suggestion (59.2% against 38.8%), not on the accepted ones (88.1% against 86.8%; Section D). This argues against the gain being mere agreement with the rule, but with only 5 fully blind held-out labels we cannot exclude an influence of the shared evidence.

The two kinds of evidence err in opposite directions. The LLM leans toward "this was an error"; the audio rule leans toward "new content" (2% of INTENT-CHANGE edits routed to an update, 36% of corrections routed to none). On a blind sample of 20 edits drawn before the review, the rule agreed with the annotator on 70%.

Over-refinement was not observed. Post-processing changed the text of 115 labeled edits, and none was labeled OVER-REFINEMENT. The guideline has no decision step for this category, and the only written definition an annotator saw was the narrow one (correct content made incorrect; Section A). This is an absence under our protocol, not evidence that post-processing never errs.

7 E1: WHICH CANDIDATE UPDATES HELP?

Table 4 reports the change in error rate relative to S0, on the utterances whose edit was labeled a correction and on the INTENT-CHANGE utterances, against the proxy references of Section 5; Section F repeats key comparisons against hand transcripts.

Learning vocabulary from every edit reduced errors, without knowing the cause. Vocabulary is the one update with a measured benefit. S1-vocabulary, which adds the short replaced spans of every edit to the recognizer's vocabulary, lowers MER on the corrected utterances by −0.029 [−0.049, −0.006]. All of S1's gain comes from this half: adding the negative examples changes nothing measurable (S1 against S1-vocabulary: −0.001 [−0.008, +0.006]). Every other strategy that learns vocabulary moves in the same direction, by 2.2–3.0 points for S1, S2, S3 with the audio rule and S4; for S3 with the LLM classifier the interval includes zero. Against the hand transcripts of 59 test

utterances the direction holds (S1-vocabulary -0.0198 [-0.0374, $+0.0001$]; S1 -0.0218 [-0.0386, -0.0011]). The gain did not require knowing why the user edited: 96 of S1's 193 vocabulary updates come from edits labeled INTENT-CHANGE (Section G), yet no strategy measurably changes the INTENT-CHANGE utterances (S1-vocabulary: -0.001 [-0.025, $+0.025$]). Because about half of the test utterances are INTENT-CHANGE, the intervals over all utterances include zero. We claim only that no harm was measurable here, not that vocabulary learned from every category is harmless in general.

Negative examples learned from every edit increased errors against proxy references. S1-negatives increases MER overall ($+0.006$ [$+0.002$, $+0.010$]) and on corrected utterances ($+0.008$ [$+0.003$, $+0.014$]), in both runs ($+0.007$ and $+0.005$). S1 stored 373 negative examples; 303 (81%) come from edits labeled INTENT-CHANGE, 264 of them labeled by the audio rule. The hand-transcribed subset does not confirm the increase. On those 59 utterances, the same outputs give $+0.0053$ against the proxy and $+0.0004$ against the hand transcript, and the difference between the two references is itself $+0.0049$ [$+0.0005$, $+0.0095$] (Section F). The measured increase therefore depends partly on the reference. 59 utterances cannot show that it is absent.

Routing by cause did not improve on learning vocabulary from every edit. In paired comparisons between strategies (Section H; post hoc, not corrected for multiple comparisons), no routed strategy had lower MER than S1 or S1-vocabulary. S3 with the LLM classifier had higher MER than S1 ($+0.022$ [$+0.008$, $+0.041$]) and than S1-vocabulary ($+0.020$ [$+0.008$, $+0.037$]), and S3+ had higher MER than S1-vocabulary ($+0.006$ [$+0.001$, $+0.013$]). For S2, S3 with the rule, S4 and S4+ the intervals against S1 include zero. The pre-registered S3+ criterion was not met. Two features of the runs limit what this comparison tests. First, routing never stored a negative example: S3 with the rule, S4, S3+ and S4+ stored none, and the two stored by S3 with the LLM came from edits labeled INTENT-CHANGE (Section G). The benefit of learning negative examples only from genuine corrections was therefore never tested. Second, the routed strategies differ from S1 and from each other in both vocabulary and style rules, and MER barely registers style, so E1 cannot isolate the effect of style rules. We make no claim that they help or hurt.

Routing with the LLM classifier. S3 with the LLM classifier had the largest average increase on the INTENT-CHANGE utterances ($+0.025$ [-0.010, $+0.066$]; $+0.019$ and $+0.031$ in the two runs), most of it from one user (86% and 74%). It differs from S3 with the rule in both vocabulary (190 against 97 terms) and style rules (72 against 53), so we attribute the difference to neither.

Takeaway. The simplest update worked: learning vocabulary from every edit reduced errors on corrected utterances, and knowing the cause of the edit was not needed for it. Negative examples learned indiscriminately increased proxy-referenced errors, and routing by cause did not improve on vocabulary alone. Whether knowing the cause helps the other content updates remains open, because the configuration that would test negative examples from genuine corrections was never exercised. Each kind of candidate update needs its own validation.

8 Related work

Learning from user edits assumes the edit is the signal. The closest prior work learns from users' edits to an LLM agent's output: PRELUDE infers a description of the user's latent preference from the edit history and conditions future generations on it, avoiding fine-tuning (Gao et al., 2024). Its learned preferences correspond roughly to our style rules, and its motivation—adapt without touching model weights—is ours. The difference is that we also ask why an edit happened. PRELUDE treats every edit as expressing a preference (evaluated with a simulated user). In our labeled subset of real edits, nearly half were labeled as new content rather than a correction or a preference, and negative examples learned from every edit increased errors against proxy references. Such learners could benefit from estimating why an edit happened and from validating the update it suggests; our study does not show that either step is required.

Implicit feedback is abundant but biased. Work on learning to rank from clicks established that implicit feedback is cheap and plentiful but systematically biased, and that the remedy is to model

how the feedback was generated—for clicks, the probability that a result was examined—before learning from it (Joachims et al., 2017). Edits are implicit feedback with a differently structured bias: corrections are mixed with changes of intent. Provenance inference is analogous to the propensity model: it models how an edit came about.

Attributing failures to components is hard. In multi-agent LLM systems, identifying which agent and which step caused a failure is difficult even for strong reasoning models; the best method in a recent benchmark attributes the responsible agent correctly about half the time (Zhang et al., 2025). Our setting adds a question they do not face: whether any component failed at all.

Non-parametric adaptation of speech systems. Speech pipelines offer adaptation surfaces that do not require training the recognizer: contextual biasing toward a list of phrases (Pundak et al., 2018), and LLM-based correction of recognizer output (Chen et al., 2023), the role our post-processing stage plays. These works supply the surfaces; we study what feedback should be fed into them.

9 Limitations and discussion

Labels and definitions. One annotator labeled all edits; a second agreed moderately ($\kappa = 0.50$; $\kappa = 0.63$ on intent versus not). Most held-out and test labels were given by accepting or changing the audio rule's suggestion, so E0 measures agreement with these labels; independent blind labels would be needed to separate classifier accuracy from that anchoring. Nothing was relabeled under the operational caveats of Sections 2 and 3.

One label per edit. An edit can combine a misheard word with added content. The audio rule flags 75 edits in which it judges one changed span misheard and another span new; its row rule labels all of them INTENT-CHANGE. Of the 26 flagged edits that have human labels, 11 were not labeled INTENT-CHANGE. The rule's span judgments have not been checked by hand, so we cannot say how often edits mix causes; span-level labels would be needed.

Proxy references and simulation. The negative-example effect changes with the reference, and the hand-transcribed subset is small (Section F). The simulated pipeline is weaker than production (MER 0.181 against 0.045 on corrected utterances, both against the user's edit, which starts from production's wording).

Scale and scope. 22 users with a median of 18.5 adaptation edits each (mean 25.7). Each strategy was run twice, with one sample per run. Between the two S0 runs the mean MER moved by -0.001, but 42 of 245 utterances changed (13 by more than 0.05), so a near-zero mean difference does not show that sampling variation is small. One application, one platform, English and Chinese.

Closing the loop on implicit feedback. In our labeled subset nearly half of the edits were labeled as changes of intent, and one update learned from every edit increased errors against proxy references. Updates learned from implicit feedback should be validated before they are deployed; whether estimating each edit's cause improves them remains open.

Ethics statement

The recordings and edits are real user data from a production dictation application, collected under the service's terms of use and privacy policy. The research data were de-identified at the account level (pseudonymized): users appear only as opaque account identifiers, with no names or contact details. We do not describe the data as anonymous, since a voice or the content of a dictation can identify a person even without an account name; access was therefore limited to the authors and one additional annotator from the same team, and recordings, transcripts and screen context are not released. The paper reports only aggregate statistics and constructed or paraphrased examples, and the production prompts are not disclosed. Classifiers and adaptation ran through commercial model APIs: Google Gemini, which the production system already uses, and, for prompt development only, Anthropic Claude; Whisper ran locally.

REPRODUCIBILITY STATEMENT

The provenance categories, annotation guidelines (Section A) and classifier prompt (Section B) are given in full; the audio-grounded rule is specified in Section 3, and the adaptation executor, strategies and metrics in Sections 4 and 5. Model versions: Gemini 3.8 Flash (E0), gemini-3.5-transcribe and gemini-3.5-flash (E1), Whisper large-v3. The E0 split was frozen before any held-out scoring and each classifier was scored once; E1 results are averaged over two independent runs. Every number, table and figure in this paper is generated by one script from the stored predictions and model outputs, without new model calls; the E1 update records are replayed from cached model calls and checked against the stored outputs. The data cannot be released, which limits exact replication.

AI USE STATEMENT

We used generative AI tools to edit and condense the text of this paper, including the abstract, and to help write analysis code. We did not use generative AI tools for research ideation, data annotation, or generating results.

Separately, AI models are part of what this paper studies and of the methods it uses: Gemini models serve as provenance classifiers (Section 3) and as stages of the simulated pipeline (Section 4); Whisper provides the independent transcript; and Anthropic Claude (Claude Sonnet 5) was used during prompt development (Appendix **??**; see also the Ethics Statement).

We reviewed all AI-assisted work. Every number in the text is generated by a script from stored results and was checked against the result files, and all AI-edited text was reread against the results. We take full responsibility for the content of this paper, including text, claims and code produced with AI assistance.

REFERENCES

Chen Chen, Yuchen Hu, Chao-Han Huck Yang, Sabato Marco Siniscalchi, Pin-Yu Chen, and Eng Siong Chng. HyPoradise: An open baseline for generative speech recognition with large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 31665–31688, 2023. arXiv:2309.15701.

Ge Gao, Alexey Taymanov, Eduardo Salinas, Paul Mineiro, and Dipendra Misra. Aligning LLM agents by learning latent preference from user edits. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. arXiv:2404.15269.

Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. Unbiased learning-to-rank with biased feedback. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (WSDM)*, 2017. arXiv:1608.04468.

Golan Pundak, Tara N. Sainath, Rohit Prabhavalkar, Anjuli Kannan, and Ding Zhao. Deep context: End-to-end contextual speech recognition. In *IEEE Spoken Language Technology Workshop (SLT)*, 2018. arXiv:1808.02480.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pp. 28492–28518, 2023. arXiv:2212.04356.

Shaokun Zhang, Ming Yin, Jieyu Zhang, Jiale Liu, Zhiguang Han, Jingyang Zhang, Beibin Li, Chi Wang, Huazheng Wang, Yiran Chen, and Qingyun Wu. Which agent causes task failures and when? on automated failure attribution of LLM multi-agent systems. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, 2025. arXiv:2505.00212.

A ANNOTATION GUIDELINES

Listen to the recording first, then decide.

1. Did the user *say* the edited words? If the edit restores words that are audible, the pipeline misheard: RECOGNITION, or MISSING-CONTEXT if the corrected span is a name or term that appears nowhere in the on-screen context.
2. If the entity appears in the context and was still missed, it is RECOGNITION, not MISSING-CONTEXT.
3. If the user added words they did not say, or deleted meaningful words they did say, it is INTENT-CHANGE. This includes plural changes ("link"→"links") and dropping a content word that changes the subject.
4. If only punctuation (including a sentence-final period), fillers, repetitions, function words, or formatting changed, it is STYLE.
5. If the recording is unintelligible, mark it unclear; it is excluded.

How the guidelines were applied. The guidelines were finalized while the first sheets were being labeled; 36 labels were resolved afterwards from the annotator's notes and 11 were changed to follow the final wording. OVER-REFINEMENT has no decision step above; the second annotator's page defined it as "raw had it right and post-processing broke it". The labeling interface showed the screen text, or the text before the cursor when no screen text was logged; the user vocabulary was not shown. Each edit received one label.

B CLASSIFIER PROMPT

The v2 prompt file is reproduced below as it was sent to the model, including its header lines (em dashes are typeset as `--`); all worked examples in it are constructed. The v2+audio variant appends the second block and adds an `AUDIO_TRANSCRIPT` line to each record. Each record is appended after the prompt as `RAW`, `POST`, `CONTEXT`, `EDIT`, `POST_CHANGED_RAW` and the two similarity scores.

```
# Provenance Classifier -- v2

v2 = v1 + the annotation guidelines the human labels follow (intent_change /
style_preference boundary). Tuned on the dev split only.

Note on constructed examples: the few-shot examples below are invented to
illustrate each category -- they are not real user transcripts. Never paste
real production data into this file (it may end up in a public repo).

## Task

You are given one dictation-correction event from a speech-to-text pipeline.
The pipeline produced a raw ASR transcription, then an LLM post-processing
step refined it before showing it to the user. The user then edited the
result. Decide **why** the user's final text differs from what the pipeline
produced -- classify it into exactly one of five categories, then justify
briefly.

You are not just diffing text. You are inferring intent: would fixing the
system (adding a vocabulary entry, adding a negative example to the
post-processing prompt, adding a style rule) have prevented this edit, or
was there nothing for the system to fix?

## Input you will receive

- `RAW`: the raw ASR transcription (before post-processing)
- `POST`: the post-processed output (what the user actually saw)
- `CONTEXT`: on-screen context available to the system at the time (may be empty) --
  `screen_text` / `selected_text` / `text_before_cursor` / `text_after_cursor`
- `EDIT`: the user's final edited text

Auxiliary computed signals (reference only, do not treat as the final answer):
- `POST_CHANGED_RAW`: whether POST differs from RAW at all
- `SIM_RAW_EDIT`, `SIM_POST_EDIT`: character-level similarity between RAW/EDIT and POST/EDIT

## Categories

1. **recognition_error** -- the discrepancy already existed in RAW (or POST
   made no difference to it). The error is attributable to the ASR stage.
   Fix: if the wrong content is an entity/proper noun, add it to the
   vocabulary/bias list.
```

```
2. **over_aggressive_refinement** -- POST substantively changed the content
   relative to RAW, and that change did *not* converge toward what the user
   actually wanted. Note: this does **not** require RAW to have been
   correct -- it also covers cases where RAW was already wrong and POST
   replaced it with a *different*, more fluent-sounding but still-wrong
   guess (this is the more common real-world pattern; a clean "RAW was
   perfect, POST broke it" case is rare). Fix: add a negative example to the
   post-processing prompt.

3. **missing_context** -- RAW and POST agree (the pipeline did what it was
   designed to do), but got an entity/term wrong because it genuinely was
   not available anywhere in CONTEXT -- not because of an acoustic mishearing,
   but because there was no way to know it without world knowledge the
   system doesn't have. Fix: add the entity to the vocabulary/bias list.

4. **style_preference** -- RAW, POST, and EDIT agree on factual content; the
   edit changes only phrasing, punctuation, filler words, or formatting
   preference. Nothing was factually wrong. Fix: add a style rule to the
   post-processing prompt.

5. **intent_change** -- the edit changes the actual meaning or adds new
   content unrelated to correcting anything the pipeline produced. The user
   changed their mind, or is continuing/adding a new thought -- including
   adding words they never said, or deleting meaningful words they did say.
   Fix: none -- this is the null action. Do not update anything.

## Decision guidance (common confusions)

- If `POST_CHANGED_RAW` is false, categories 1 and 2 collapse to just
  category 1 (recognition_error) -- post-processing cannot be blamed for a
  discrepancy it had no hand in.
- Don't default to recognition_error just because RAW/POST differ from EDIT.
  Check whether the difference is a *factual* correction (categories 1-3) or
  a *preference/intent* difference (categories 4-5) -- a single-token change
  can be either, depending on whether the original was factually wrong or
  just not the user's preferred phrasing.
- `missing_context` vs `recognition_error`: both can produce a wrong entity.
  Use CONTEXT -- if the correct entity/term appears nowhere in `screen_text` /
  `selected_text` / `text_before_cursor` / `text_after_cursor`, the context
  was not enough to get it right: nothing available to the system could have
  led it to the correct spelling (missing_context). If the correct entity
  *does* appear in CONTEXT and the system still got it wrong, or the wrong
  word is an ordinary word rather than an entity/term, treat it as a
  mishearing (recognition_error).
- `intent_change` vs `over_aggressive_refinement`: if EDIT is largely
  unrelated in topic to RAW/POST (not just differently worded), lean
  intent_change even if similarity scores are low for both RAW and POST --
  low similarity to both does not by itself mean POST made things worse.
- The most costly mistake is calling an intent_change an error: the system
  would then learn to insert or delete content nobody dictated. Only choose
  recognition_error / missing_context when the POST version plausibly
  *sounds like* what the user said (a mishearing, a near-homophone, a
  misspelled name). If the EDIT introduces content with no acoustic
  resemblance to anything in POST, it is new content: intent_change.
- `intent_change` vs `style_preference` -- judge by whether the facts change:
  - adding content (a clause, a qualifier, a link, a plural the speaker did
    not say, e.g. "link" -> "links") -> intent_change
  - deleting meaningful words, even a single one that changes the subject or
    scope (e.g. dropping "the boss" from "the boss of the company said...")
    -> intent_change
  - changing only punctuation (including dropping a sentence-final period),
    fillers ("um", "like", ""), repetitions, or function words that
    don't change the facts -> style_preference
  - reformatting the same content (numbers as digits, casing, spacing)
    -> style_preference

## Output format

Respond with a single JSON object, nothing else:

'''json
{
  "reasoning": "1-3 sentences",
  "category": "recognition_error | over_aggressive_refinement | missing_context |
    style_preference | intent_change",
  "confidence": "high | medium | low"
}
'''

## Worked examples (constructed, not real data)
```

````
**Example A**
- RAW: "meeting is at 3pm with the finance team"
- POST: "meeting is at 3 PM with the finance team"
- CONTEXT: (empty)
- EDIT: "meeting is at 3pm with the finance team, room 4B"
- POST_CHANGED_RAW: true (case/formatting of "3pm" only)

```json
{
  "reasoning": "POST only normalized formatting. The edit adds new information (room 4B)
    unrelated to anything the pipeline got wrong.",
  "category": "intent_change",
  "confidence": "high"
}
```

**Example B**
- RAW: "can you ping Jano about the release"
- POST: "can you ping Jano about the release"
- CONTEXT: text_before_cursor: "...let's sync on the release notes before Friday..."
- EDIT: "can you ping Yannow about the release"
- POST_CHANGED_RAW: false

```json
{
  "reasoning": "RAW and POST agree, so post-processing isn't at fault. The correct name
    'Yannow' appears nowhere in CONTEXT, so nothing available to the system could have
    produced this spelling.",
  "category": "missing_context",
  "confidence": "medium"
}
```

**Example C**
- RAW: "the uh throughput number was like really really bad this week honestly"
- POST: "The throughput number was quite poor this week."
- CONTEXT: (empty)
- EDIT: "The throughput number was really bad this week."
- POST_CHANGED_RAW: true

```json
{
  "reasoning": "POST correctly cleaned disfluencies but substituted 'quite poor' for the
    user's own emphasis 'really bad' -- a wording choice, not a factual error. Nothing here
    was wrong.",
  "category": "style_preference",
  "confidence": "medium"
}
```

## Additional input: independent audio transcript

`AUDIO_TRANSCRIPT` is what a second, independent speech recognizer
(Whisper large-v3, given no context) heard in the same recording. Use it as
evidence about **what the user actually said**, not as the correct answer --
it also mishears, especially names and domain terms.

- If words the user added or changed in EDIT appear in AUDIO_TRANSCRIPT (or
  something that sounds like them does), the user did say them: the pipeline
  got it wrong -> recognition_error / missing_context, not intent_change.
- If the added/changed words are absent from AUDIO_TRANSCRIPT, and it
  otherwise matches POST, the user most likely typed new content afterwards
  -> intent_change.
- If the user deleted words that AUDIO_TRANSCRIPT also contains, they said
  them and removed them later -> intent_change (or style_preference if they
  are only fillers / repetitions).
- If AUDIO_TRANSCRIPT garbles the same span as POST (both wrong, differently),
  the span was hard to hear; that points to a recognition problem, not to
  new content.
````

C E0 ON DEVELOPMENT DATA WITH A SECOND MODEL FAMILY

During prompt development we ran the three prompt variants with Claude Sonnet 5 on the development edits (103, 103 and 102 parsed responses; majority class 44.7%, 44.7% and 44.1%). Five-way agreement was 54.4 (text only), 66.0 (+ guidelines) and 69.6 (+ guidelines + transcript); the rate of INTENT-CHANGE edits routed to an update was 61%, 35% and 29%. The ordering matches the

held-out results with Gemini (Table 3); the absolute numbers are not comparable, since the prompts were tuned on these edits.

D E0 LABELS: SOURCES AND EVALUATION SCOPE

Table 5 lists where the 359 human labels come from. "Suggestion shown" means the annotator saw a rule's label and accepted or changed it; this is human review, not independent blind labeling. The E1 adaptation set additionally uses 460 labels produced by the audio rule alone (Section 4).

Table 5: Sources of the human labels. Counts after excluding unintelligible edits.

Sheet (how drawn; what the annotator saw)	Edits	INTENT-CHANGE	RECOGNITION/MISSING-CONTEXT	STYLE
Sampled from edits an early text rule could not attribute; no suggestion shown	117	47 (40%)	35	35
Sampled from edits the stage-diff rule called RECOGNITION; its label shown	22	14 (64%)	5	3
Uniform random sample; no suggestion shown	20	10 (50%)	8	2
E1 test edits (and 3 others); audio rule's label shown, accept or change	200	101 (50%)	69	30

Held-out split. Of the 205 held-out edits, 5 come from the uniformly random sheet and 200 from the reviewed sheet, where the annotator accepted 151 of the rule's suggestions and changed 49. The labels cover 103 INTENT-CHANGE, 54 RECOGNITION, 32 STYLE and 16 MISSING-CONTEXT edits, and no OVER-REFINEMENT. Agreement on the accepted and on the changed edits was 74.8%/61.2% (v1), 86.8%/38.8% (v2), 88.1%/59.2% (v2+audio) and 100.0%/0.0% (audio rule, by construction). On the 5 blind held-out edits the four classifiers agreed on 5, 5, 4 and 4. Table 6 gives the confusion matrices; the prompt's OVER-REFINEMENT was predicted once by each LLM variant and never occurs in the labels.

Table 6: Confusion matrices on the held-out split (rows: human label with count; columns: predicted).

	RECOGNITION	MISSING-CONTEXT	STYLE	INTENT-CHANGE	OVER-REFINEMENT
LLM, text only (v1)					
RECOGNITION (54)	35	8	2	9	0
MISSING-CONTEXT (16)	3	12	0	1	0
STYLE (32)	3	0	28	1	0
INTENT-CHANGE (103)	11	2	16	73	1
LLM + guidelines (v2)					
RECOGNITION (54)	23	10	1	20	0
MISSING-CONTEXT (16)	2	11	0	3	0
STYLE (32)	1	0	28	3	0
INTENT-CHANGE (103)	1	1	7	93	1
LLM + guidelines + transcript (v2+audio)					
RECOGNITION (54)	36	6	1	11	0
MISSING-CONTEXT (16)	2	12	0	2	0
STYLE (32)	1	0	28	3	0
INTENT-CHANGE (103)	7	1	4	90	1
Audio-grounded rule					
RECOGNITION (54)	16	10	0	28	0
MISSING-CONTEXT (16)	0	11	0	5	0
STYLE (32)	0	1	27	4	0
INTENT-CHANGE (103)	2	0	0	101	0

Second annotator. The 50 edits were labeled blind from the written guidelines; 28 of the first annotator's labels on them had been given with the audio rule's suggestion visible, so the comparison is not fully independent on that side. The second annotator marked 2 edits unclear; these count as disagreements. Table 7 lists the disagreements. The sample was drawn from the labeled edits; the drawing step is not scripted in our repository.

E PROBING A COMMERCIAL RECOGNIZER

We probed gemini-3.5-transcribe (September 2026). It exposes a native vocabulary parameter (`custom_vocabulary`, up to 1,000 terms, no per-term weights) and a "smart" cleanup mode,

Table 7: The 15 disagreements between the two annotators.

Pair of labels	Edits
INTENT-CHANGE vs STYLE	5
INTENT-CHANGE vs RECOGNITION/MISSING-CONTEXT	4
STYLE vs RECOGNITION	2
RECOGNITION vs MISSING-CONTEXT	2
second annotator: unclear	2

but accepts no prompt: text in the input is ignored and a system instruction is rejected. On a Chinese utterance, smart mode combined with a custom vocabulary dropped a whole clause in 3 of 3 runs, while smart mode alone and verbatim mode with the vocabulary did not; smart mode also dropped a sentence-final word on an English utterance. On near-silent audio it returned an empty transcript for one clip and the same invented sentence in 3 of 3 runs for another. These observations are from our design notes; the raw probe outputs were not retained. They motivate running recognition in verbatim mode and cleanup as a separate stage.

F REFERENCE SENSITIVITY OF THE E1 RESULTS

One author transcribed a random 60 of the 245 test utterances (fixed seed) from the audio, starting from an independent Whisper draft (edited in 32 cases) and without seeing the production output or the user's edit; one recording was unintelligible. The remaining 59 utterances cover 20 of the 22 users and 30 INTENT-CHANGE, 14 RECOGNITION, 13 STYLE and 2 MISSING-CONTEXT labels. Table 8 shows how far each candidate reference is from this hand transcript.

Table 8: MER of each candidate reference against the hand transcript (59 utterances).

Utterances	n	E1 reference	User's edit	Production output
User corrected the system	29	0.091	0.091	0.118
Change of intent	30	0.070	0.290	0.070

Table 9 and Figure 2 re-score the same cached outputs (two-run average) against three references: the proxy on all 245 test utterances, the proxy on the 59 hand-transcribed ones, and the hand transcript on the same 59. The last column is the difference between the two references on identical utterances and outputs; all intervals on the 59 use the same user resamples. For S1-negatives the proxy-referenced increase ($+0.0059$) is of similar size on the subset ($+0.0053$), but against the hand transcript it is $+0.0004$ (unrounded +0.00043), and the difference between references is $+0.0049$ [$+0.0005$, $+0.0095$]. The hand-transcribed subset therefore does not confirm the increase. Its interval ([-0.0044, $+0.0037$]) is too wide to show that the increase is absent. In the single runs the subset values were $+0.0120$ and -0.0014 (proxy) and $+0.0031$ and -0.0022 (hand).

Table 9: Change in MER relative to S0 under different references (two-run average; 95% user-level bootstrap intervals).

	Proxy, all 245	Proxy, 59 subset	Hand transcript, same 59	Proxy − hand, same 59
S1, negative examples only	+0.0059 [+0.0021, +0.0098]	+0.0053 [−0.0013, +0.0099]	+0.0004 [−0.0044, +0.0037]	+0.0049 [+0.0005, +0.0095]
S1, vocabulary only	−0.0152 [−0.0329, +0.0073]	−0.0135 [−0.0300, +0.0063]	−0.0198 [−0.0374, +0.0001]	+0.0064 [−0.0061, +0.0203]
S1 (both)	−0.0167 [−0.0349, +0.0057]	−0.0199 [−0.0348, −0.0005]	−0.0218 [−0.0386, −0.0011]	+0.0019 [−0.0133, +0.0181]

G UPDATE-ACTION AUDIT

Table 10 counts what each strategy actually stored for the test users, replayed from the cached model calls; all 5390 replayed pipeline calls reproduce the stored outputs. For each update type it gives the number stored and the number of adaptation edits that triggered it, split by the edit's label ("corr.": any label other than INTENT-CHANGE; "h": of these, human labels, the rest being audio-rule labels). Negative examples beyond the 20 most recent per user are stored but not used; the number used is

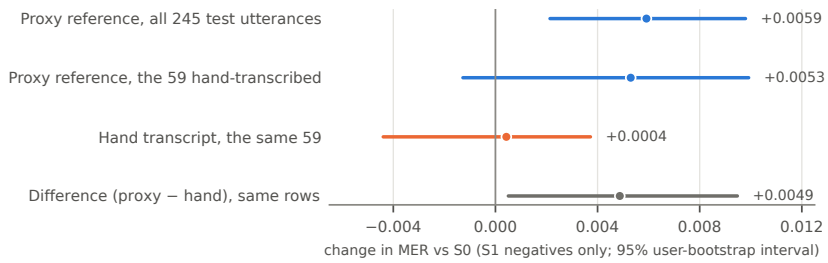

Figure 2: S1-negatives versus S0 under proxy and hand-transcribed references.

in parentheses. Style-rule counts differ between the two runs when shown as two numbers, because the rules are generated by an LLM in each run.

Table 10: Updates stored per strategy (summed over test users) and the labels of the edits that triggered them.

	Vocabulary			Negative examples			Style rules		
Strategy	terms	from corr.	from INTENT-CHANGE	stored (used)	from corr.	from INTENT-CHANGE	stored	from corr.	from INTENT-CHANGE
S1	200	97 (33h)	96 (8h)	373 (260)	70 (26h)	303 (39h)	0	0	0
S2	143	69 (29h)	68 (7h)	206 (179)	35 (18h)	171 (32h)	0	0	0
S3 audio rule	97	93 (22h)	3 (3h)	0 (0)	0	0	53/51	58 (21h)	2 (2h)
S3 LLM	190	86 (29h)	87 (4h)	2 (2)	0	2 (1h)	72	61 (23h)	21 (2h)
S4 labels	102	103 (32h)	0	0 (0)	0	0	57/55	64 (27h)	0
S3+	213	104 (33h)	98 (10h)	0 (0)	0	0	53/51	58 (21h)	2 (2h)
S4+	211	105 (34h)	96 (8h)	0 (0)	0	0	57/55	64 (27h)	0
S1 vocab only	200	97 (33h)	96 (8h)	0 (0)	0	0	0	0	0
S1 neg. only	0	0	0	373 (260)	70 (26h)	303 (39h)	0	0	0

Three consequences follow. Routing never stored a negative example from an edit labeled as a correction: the rule-, label- and S3+/S4+-routed strategies stored none, and the 2 stored by S3 with the LLM came from edits labeled INTENT-CHANGE. The candidate update "negative example from a genuine correction" was therefore never tested. Style rules were written only by the routed strategies, which also differ from S1-vocabulary in vocabulary: S3+ and S4+ store 13 and 11 more terms (for 7 and 7 users), so no comparison in E1 isolates style rules. Finally, S1's vocabulary includes terms from 96 edits labeled INTENT-CHANGE.

H ADDITIONAL E1 RESULTS

Two-run averages, change in MER vs S0 on all 245 utterances: pre-registered S3+ (vocabulary from every edit, other updates routed by the audio rule) -0.009 $[-0.028, +0.016]$; S4+ (same, routed by adaptation-set labels) -0.011 $[-0.028, +0.008]$; S1 without the "never copy" instruction in the negative-example block -0.016 $[-0.035, +0.007]$, against -0.017 for S1. The S3+ criterion registered before the run (beat S1 and S2 overall) was not met. Figure 3 shows all strategies by subset, and Table 11 gives paired differences between strategies. These comparisons were not planned before the runs and are not corrected for the 39 tests; they are reported to replace a comparison of overlapping intervals.

Table 11: Paired differences in MER between strategies (first minus second; positive means the first has higher error). Two-run average per utterance, 95% user-level bootstrap intervals, proxy references.

Comparison	Corrected (124)	INTENT-CHANGE (121)	All (245)
S2 − S1	+0.003 [−0.009, +0.015]	+0.003 [−0.005, +0.011]	+0.003 [−0.005, +0.010]
S3 audio rule − S1	+0.008 [−0.010, +0.025]	+0.003 [−0.006, +0.011]	+0.005 [−0.007, +0.016]
S3 LLM − S1	+0.015 [−0.000, +0.037]	+0.029 [+0.011, +0.051]	+0.022 [+0.008, +0.041]
S4 labels − S1	+0.005 [−0.003, +0.011]	+0.006 [−0.004, +0.014]	+0.005 [−0.003, +0.011]
S3+ − S1	+0.008 [−0.004, +0.024]	+0.007 [+0.002, +0.012]	+0.008 [+0.000, +0.016]
S4+ − S1	+0.005 [−0.007, +0.020]	+0.006 [−0.005, +0.014]	+0.006 [−0.005, +0.015]
S2 − S1-vocabulary	+0.002 [−0.012, +0.017]	+0.001 [−0.007, +0.010]	+0.001 [−0.008, +0.010]
S3 audio rule − S1-vocabulary	+0.007 [−0.011, +0.023]	+0.001 [−0.009, +0.009]	+0.004 [−0.008, +0.015]
S3 LLM − S1-vocabulary	+0.014 [+0.002, +0.030]	+0.027 [+0.009, +0.048]	+0.020 [+0.008, +0.037]
S4 labels − S1-vocabulary	+0.004 [−0.002, +0.007]	+0.004 [−0.007, +0.012]	+0.004 [−0.003, +0.009]
S3+ − S1-vocabulary	+0.008 [−0.001, +0.021]	+0.005 [+0.001, +0.008]	+0.006 [+0.001, +0.013]
S4+ − S1-vocabulary	+0.005 [−0.004, +0.016]	+0.004 [−0.007, +0.012]	+0.004 [−0.006, +0.012]
S1 − S1-vocabulary	−0.001 [−0.008, +0.006]	−0.002 [−0.005, +0.001]	−0.002 [−0.005, +0.003]

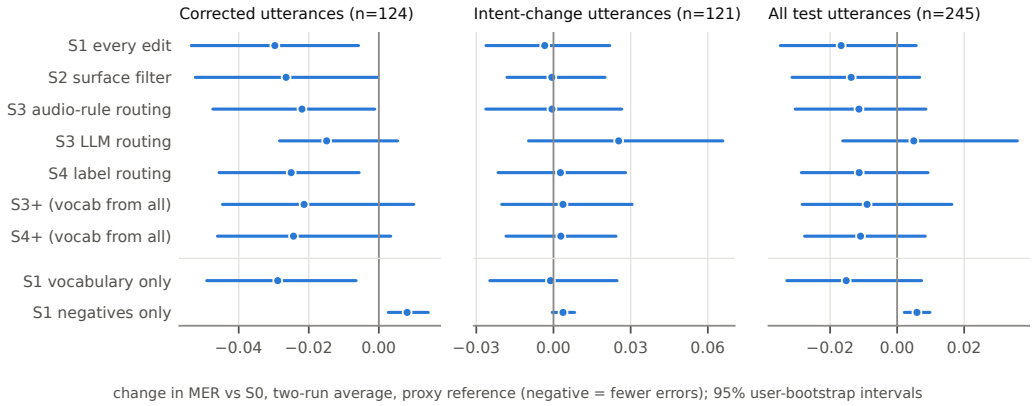

Figure 3: Change in MER relative to S0 by subset (proxy references; two-run average; 95% user-level bootstrap intervals). Routed strategies are kept; none has a lower point estimate than S1.