Disentangling Annotator Skill from Verifier Strictness in Cross-Verified Dialogue Annotation

Zihao Tao[1] **and John A. Prado**[1,2] **and Ignazio Steven LaManna**[1]
Ryan Puterbaugh[1] and Mim Datta[1] and Julia Hirschberg[1]
[1]Columbia University, USA [2]University of Alberta, Canada
julia@cs.columbia.edu

Abstract

Some dialogue corpus projects use a verify-after-annotation workflow: a second team member reviews a submitted file and records corrections. The resulting correction count mixes two signals, annotator accuracy and verifier strictness. We separate these signals for RASwDA, an audio-anchored re-alignment of 1,045 Switchboard file sides (105,005 corrections across 977 change logs) produced by five team members in 2024–2025. Verification is crossed: each of three identified annotators was checked by four different verifiers, with overlap in both directions. A cross-classified mixed-effects model on per-file corrections-per-interval assigns 10.8% of the variance to annotator identity, while the verifier random effect collapses to zero (singular fit, stable across seven leave-one-out and response-choice refits). Thus, for this boundary-realignment task, we find no detectable verifier identity effect once annotator identity, batch, and file length are controlled. Boundary placement, not label selection, accounts for 58.5% of corrections corpus-wide. This helps explain why verifier-specific strictness has little room to appear: timestamp adjustments are anchored in the audio, while DA-label changes account for only 3.0% of corrections. We release the action-typed change logs so other projects can run the same annotator-verifier decomposition on their own verification data.

1 Introduction

Many corpus-construction projects use a verify-after-annotation workflow: a second team member reviews a submitted file and records corrections before release. When the project has access to an external signal such as audio, this verification step can catch timestamp errors, omitted segments, and label changes in a structured log. The per-file correction count is useful, but incomplete: it mixes two sources, the annotator's accuracy and the verifier's strictness threshold. Without separating them, the log alone cannot tell whether a high-correction file reflects a less accurate annotation pass, a stricter verifier, or simply a harder file.

We address this confound using the RASwDA corpus, a manual re-alignment of the 1,155-conversation Switchboard Dialog Act corpus (Godfrey et al., 1992; Jurafsky et al., 1997). The re-alignment task aligns SwDA dialogue-act intervals to the original audio while retaining the inherited DA label unless the re-alignment makes that label untenable. We analyze a *file side*, one speaker channel of a Switchboard conversation, as the unit of annotation and verification. Between 2024 and 2025, five team members produced 977 per-file change logs containing 105,005 corrections in five action categories. The verification schedule is what makes the analysis possible: each of the three identified annotators was checked by four different verifiers, with overlap in both directions (Table 2). This crossed design lets us fit a mixed-effects model with separate annotator and verifier random effects.

On per-file normalised corrections (corrections per non-silence interval), the variance partition assigns 10.8% to annotator identity, 80.6% to file-level residual, and zero to verifier identity, stably across seven sensitivity refits. We interpret this as a result about this corpus, not a universal claim about verification: for this audio-anchored realignment task, who annotated a file changes the expected correction count, while who verified it has no detectable effect once annotator identity, batch, and file length are controlled.

We make three contributions.

- We release the RASwDA verification log dataset: 977 machine-readable change CSVs, one per file side, totalling 105,005 corrections with per-row action type, time interval, and DA labels before and after correction (§3).
- We present a cross-classified mixed-effects analysis that partitions per-file correction-rate

Proceedings of the 27th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 414–422
August 2–5, 2026.

variance into annotator, verifier, batch, and file-difficulty components. The verifier share collapses to zero (§5), a corpus-level quality check that a single before/after label-overlap number cannot support.

- We show that boundary placement, not DA-label selection, accounts for 58.5% of corrections corpus-wide and remains dominant across all three annotators. This pattern helps explain the verifier-interchangeability finding: most corrections are objective timestamp adjustments, not inter-subjective label judgments (§6).

2 Background and Related Work

The Switchboard Dialog Act corpus. Switchboard (Godfrey et al., 1992) consists of ∼2,400 two-way telephone conversations recorded in 1990–1992. A 1,155-conversation subset was annotated with the SWBD-DAMSL tag set (Jurafsky et al., 1997; Allen and Core, 1997), producing the Switchboard Dialog Act (SwDA) corpus with ∼205,000 DA-tagged utterances. The original forced alignment, produced by a GMM-HMM recognizer (Shriberg et al., 1998), contains errors during quiet utterances, background noise, and speaker overlap (Stolcke et al., 2000). The RASwDA project addresses these through manual re-alignment in Praat (Boersma and Weenink, 2024) followed by verification of every re-aligned file. Because verifiers inspect and correct the submitted TextGrid, post-verification labels are not independent annotations of the post-alignment labels. We therefore treat before/after label overlap as a descriptive stability signal, not as inter-annotator agreement.

Annotator modelling. Aggregation methods for noisy multi-annotator labels, from Dawid and Skene (1979)'s EM-based estimator to recent Bayesian (Passonneau and Carpenter, 2014; Paun et al., 2018) and disagreement-aware (Weerasooriya et al., 2023; van der Meer et al., 2024) approaches, estimate annotator reliability from label distributions, but treat reliability as a single static parameter and do not use verifier corrections as evidence. Bayerl and Paul (2011) catalogue factors influencing inter-coder agreement but do not model individual annotator change over time.

Annotation error detection. Recent work detects label errors post hoc (Nahum et al., 2025).

Action type	Definition
BOUNDARY_SHIFT	Segment boundary moved; DA label retained
INSERT	New interval added that annotator omitted
DELETE	Annotator-added interval removed
DA_CHANGE	DA label changed without moving a boundary
MODIFY	Both boundary and label changed

Table 1: Five verifier action types recorded in each `*_changes.csv`. Each row also stores the original and corrected time interval, the utterance text, and the DA label before and after the correction.

Our approach is preventive and longitudinal: structured verification logs record every correction at the time it is applied, which supports trajectory analysis and downstream classification (Zhang and Wang, 2025).

Cross-classified random effects in annotation. Mixed-effects models with crossed random factors are standard in psycholinguistics for separating subject and item variance (Baayen et al., 2008). Psychometric work on rater-mediated assessment treats analogous behaviour as rater severity or rater effects, often with many-facet Rasch models (Eckes, 2019). In corpus annotation, however, verification logs are rarely modeled as a crossed annotator-by-verifier design. Most annotation-quality reports collapse the verifier-vs-annotator question into a single label-overlap number, or report per-annotator means without controlling for who verified each file. In our design, the same person can appear as annotator on some files and verifier on others, and every annotator is checked by multiple verifiers. That crossing is what identifies the two variance components separately.

Our contribution. We measure annotator quality through verifier corrections rather than through label overlap, and we model verifier strictness as a separate random effect rather than absorbing it into the annotator estimate.

3 Verification Infrastructure

For each annotator-submitted TextGrid, a trained verifier reviewed the file against the original audio and recorded every edit in a machine-readable `*_changes.csv` (one per conversation side). These logs are the primary data source for the analyses below.

Each row of a change log records a single

correction in one of five action categories (Table 1), together with the original and corrected time interval, the utterance text, and the DA label before and after. A representative row from `sw2152_A_changes.csv` reads:

```
action=BOUNDARY_SHIFT,
t_orig=(124.31, 126.07),
t_new=(124.31, 125.42),
text="yeah I think so",
da_orig=aa, da_new=aa
```

which records that the verifier shortened a backchannel-agreement (`aa`) interval by 0.65 seconds while keeping the DA label. This level of detail enables aggregate analysis (correction counts) and fine-grained inspection (e.g. which DA transitions verifiers correct most often, or which boundaries shift earlier vs. later).

Per-file metrics. We use two quantities. The raw count corrections/file is the total number of rows in a file's change log. To control for file complexity, we also report corrections/interval, corrections divided by the number of non-silence intervals in the verified TextGrid (excluding intervals marked `SIL` or ⟨`laughter`⟩). The normalisation partly removes the conversation-length confound: a 300-interval file with 90 corrections ($corrections/interval = 0.30$) reflects better work than a 100-interval file with 60 corrections ($corrections/interval = 0.60$), even though the raw count is higher. Other sources of file difficulty (speaker overlap, disfluency density) remain unmodelled and are picked up by the file-level random effect in §5.

4 Crossed Annotator × Verifier Data

The RASwDA team consists of five members who took on annotator and verifier roles in different combinations across the project. For compact tables, we use stable team pseudonyms T1–T5. Three team members (T1–T3) worked as both annotator and verifier; T4 and T5 appear only in the verifier role. The 503 file sides with no identified annotator are not the work of a sixth team member but files whose annotator attribution we could not recover from the dispatch records; we treat them as missing data and report a missingness diagnostic in the Limitations section.

Coverage. The corpus contains 1,045 file sides (speaker A or B for each of the verified conversations). We cross-reference four sources: verifier dispatch logs maintained by individual team members, post-verification TextGrid folders, `*_changes.csv` filenames, and annotator self-records. This attributes 542 file sides to one of T1–T3 as annotator and 1,025 file sides to one of T1–T5 as verifier. The residual 503 annotator-blank and 20 verifier-blank rows are treated as missing.

	Verifier					
Annot.	**T1**	**T2**	**T3**	**T4**	**T5**	**Unv.**
T1	—	164	63	13	43	5
T2	49	—	44	16	13	0
T3	38	38	—	46	10	0

Table 2: Annotator × verifier crosstab (file sides). Diagonals for T1–T3 are empty because the team avoided self-verification; T4 and T5 have no diagonal because they worked only as verifiers. "Unver." = annotator-identified file sides with no verifier recorded. Each annotator was verified by all four other team members, giving the crossed structure required by the mixed-effects model in §5.

Crossed design. Table 2 reports the annotator × verifier crosstab over the 537 file sides where both roles are identified. The off-diagonal mass is substantial: each annotator was checked by four different verifiers, with 164 T1 files verified by T2, 63 by T3, 13 by T4, and 43 by T5 (and analogously for T2 and T3). This crossing makes the mixed-effects analysis in §5 possible; without it, the annotator and verifier random effects would be inseparable.

Temporal ordering. For each file side we have the verifier's submission date (from the dispatch logs); for 542 of the attributed file sides we also have the annotator's submission date. T1 submitted across 20 distinct dates spanning 2024-06-07 to 2025-02-10; T2 across 10 dates (2024-09-22 to 2025-02-10); T3 across 11 dates (same window). Verifiers submitted across 6–24 distinct dates spanning 2025-02-16 to 2025-10-29. Each submission date corresponds to a real batch (mean 12–14 files per annotation date, 13–21 per verification date), reflecting the workflow in which feedback returns to the annotator only at batch boundaries.

For this reason, we use ordinal batch indices rather than calendar dates as the experience axis: for each person, batch 1 is their first submission, batch 2 their second, and so on. This normalises across team members who joined the project at different calendar times and gives the model a per-person experience axis.

Attribution methodology. The annotator and verifier columns in our metadata table were constructed by reconciling four independent sources: per-team-member dispatch logs (which list files the team member was assigned to verify, with dates), the post-verification TextGrid folders (organised by verifier), the `*_changes.csv` log filenames (which encode neither annotator nor verifier directly), and annotator self-records returned during a follow-up sweep. We treat a row as *confirmed* when at least two sources agree and as *single-source* when only one applies; a sensitivity refit on the confirmed-only subset is reported in the Limitations section. Files marked as zero-correction (verified with no edits, hence no `*_changes.csv`) are recovered from the dispatch logs and enter the analysis with a correction count of zero rather than as missing rows; omitting them would inflate per-batch means and dampen any improvement signal.

5 Mixed-Effects Analysis

We model per-file normalised correction count ($y_i = corrections/interval$ for file i) as a linear function of annotator and verifier experience, controlling for file length, with random intercepts for annotator, verifier, and batch:

$$\begin{aligned} y_i = \beta_0 &+ \beta_1 \,\text{ann\_batch}_i + \beta_2 \,\text{ver\_batch}_i \\ &+ \beta_3 \,\log n_i^{\text{int}} + u^A_{a(i)} + u^V_{v(i)} \\ &+ u^B_{a(i),\,\text{ann\_batch}_i} + \varepsilon_i, \end{aligned} \tag{1}$$

where $a(i)$ and $v(i)$ index annotator and verifier, n_i^{int} is the file's non-silence interval count, and the random terms u^A, u^V, u^B capture per-annotator, per-verifier, and per-(annotator,batch) intercepts. The model is fit by REML with the `lmer` function from the `lme4` R package (Bates et al., 2015) on the 537-file complete-case subset (both roles identified).

In plain terms, the model asks whether some annotators receive more corrections after we control for who verified the file, and whether some verifiers make more corrections after we control for who annotated it. The crossed schedule in Table 2 is what makes those two questions separable.

Variance partition. Table 3 reports the decomposition. The verifier random effect collapses to zero (`lme4` reports *boundary (singular) fit*), the residual file-level term dominates at 80.6% of total variance, and the annotator random effect contributes a further 10.8%.

Source	Var. comp.	% of total
Annotator (random)	0.0108	10.8%
Verifier (random)	0.0000	0.0%
Batch within annotator	0.0087	8.7%
Residual (file-level)	0.0807	80.6%
Fixed effects (R^2_m)	—	1.4%

Table 3: Variance partition of per-file normalised corrections from Eq. 1, fit on 537 file sides with both annotator and verifier identified. Variance components are REML estimates; the bottom row reports the marginal R^2 for the three fixed effects (Nakagawa and Schielzeth, 2013). The verifier component is estimated at the boundary (singular fit); see text and sensitivity checks.

Why verifier variance is zero: mechanism. A singular fit at `(1 | verifier)` means the REML optimum lies at $\sigma^2_V = 0$: the likelihood is maximised by assigning no between-verifier variance. Three features of the data make this plausible. First, the file-level residual is an order of magnitude larger than any verifier-level variance the data could plausibly accommodate, so small systematic verifier differences would be indistinguishable from the file-difficulty noise floor. Second, the verifier batch slope β_2 (Table 4) absorbs any common temporal drift in verifier behaviour before the random effect is estimated. Third, with only five verifiers, between-group variance is estimated from a sample of size five and REML is conservative in this regime. In this setting, the data gives no evidence of systematic strictness differences between the five verifiers once annotator identity, batch, and file length are controlled.

Fixed-effect estimates. Table 4 reports the experience slopes (β_1, β_2) and the length covariate (β_3). Neither learning slope reaches conventional significance: β_2 for verifier experience is marginal (p=0.06) and negative, while β_1 for annotator experience is indistinguishable from zero. We treat both slopes as exploratory: annotators might improve with practice, while verifiers could either apply fewer corrections as the team converges or more corrections as their strictness increases.

Trajectories. Figure 1 shows the per-person experience trajectories. Panel (a) plots each of T1, T2, and T3 as annotator against their personal annotation-batch index; panel (b) does the analogous plot for T1–T5 as verifier against the verification-batch index. LOWESS fits on per-file points (solid curves) are flat or noisy in both panels,

Effect	Est. (95% CI)	p
β_1 ann. batch index	-0.002 (-0.011, 0.007)	0.60
β_2 ver. batch index	-0.004 (-0.009, 0.000)	0.06
$\beta_3 \log n^{\text{int}}$	-0.008 (-0.090, 0.075)	0.86

Table 4: Fixed-effect estimates from Eq. 1; CIs are profile-likelihood. Neither learning slope is significant at α=0.05; the verifier-batch slope is marginal and negative (fewer corrections per interval in later verifier batches). File length has no effect once normalisation is in place.

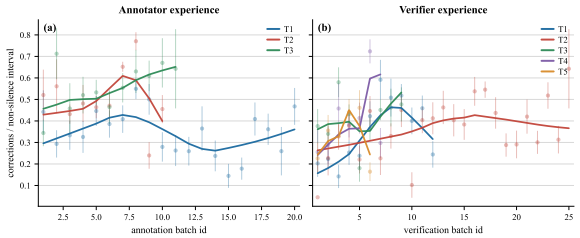

Figure 1: Per-batch mean normalised corrections (corrections/interval) for each team member as annotator (panel a) and as verifier (panel b). Solid curves are LOWESS fits on per-file points; faint markers show per-batch means with ± 1 SE error bars. The x-axis is each person's personal batch index. Read together with Table 4: the flat-to-noisy trajectories are consistent with the null batch slopes. The corrections-per-interval rate varies much more across files than across batches.

and the faint per-batch means show no coherent downward drift, which visually confirms the null fixed-effect slopes in Table 4. The team did not visibly improve over calendar time; the corrections-per-interval rate is instead dominated by file-to-file difficulty.

Sensitivity checks. We ran three robustness refits. (i) *Leave-one-verifier-out:* re-estimating the partition with each of the five verifiers removed in turn yields $\sigma^2_V = 0$ in all five refits (sample sizes 335–471). No single verifier is propping up or suppressing the verifier share. (ii) *Raw-count response:* refitting with corrections/file as the response (no length normalisation) likewise returns $\sigma^2_V = 0$; the finding is not an artefact of the corrections/interval transform. (iii) *Complete-case only:* the 537-row fit above is already the complete-case fit; attributed rows with missing verifier batch indices would have been silently dropped by `lmer`, so no further restriction applies. All seven refits (main + five LOO + raw-count) agree that verifier identity contributes no measurable variance once the other terms are in the model.

What the partition answers. The cross-classified design answers a stronger question than a single inter-annotator agreement number: it separates which layer of the annotation stack carries the disagreement. Ten percent sits at the annotator layer (the three annotators in our team differ from each other), roughly nine percent at batch-within-annotator (real submission-to-submission variation), zero at the verifier layer, and eighty percent at the file-level residual (files differ in difficulty in ways our covariates do not capture). For a corpus release, the verifier-zero result is the relevant quality-control result: at the precision this study can measure, the expected correction count does not depend on which team member verified the file once annotator, batch, and file length are in the model.

6 What Verifiers Correct

The mixed-effects analysis treats corrections as a scalar quantity, but the change logs also record *which* of the five action types (Table 1) each correction belongs to. The distribution is highly non-uniform and is dominated by timestamp adjustments, not label disputes.

Boundary placement dominates corpus-wide. Across all 977 verified file sides, BOUNDARY_SHIFT accounts for 58.5% of the 105,005 corrections, more than the other four categories combined. INSERT is a distant second at 26.1%, with DELETE at 9.4%, DA_CHANGE at 3.0%, and MODIFY at 2.9%. The dominance of timestamp adjustments over label changes is consistent with the structure of the underlying task: a Praat re-alignment pass touches every interval's time marks, while the DA label is inherited from the SwDA gold standard and only changes when the re-alignment makes the original label untenable.

Why this lets verifiers agree. A timestamp decision is anchored in the audio and the forced-aligner output: once two verifiers agree on where a speech–silence transition falls, their boundary corrections converge. Label edits, by contrast, require inter-subjective judgment about whether an utterance is an `sd` versus a `sv` and leave room for genuine verifier disagreement. The 58.5%/3.0% split between BOUNDARY_SHIFT and DA_CHANGE therefore explains the verifier-interchangeability result of §5: most corrections fall in the category where verifier-specific strictness has the least room to show up. A

Annot.	bs.	da.	ins.	del.	mod.
T1	64.9	0.5	27.8	5.9	0.9
T2	61.1	1.3	23.1	12.9	1.5
T3	41.4	12.8	23.8	14.4	7.5
Corpus	*58.5*	*3.0*	*26.1*	*9.4*	*2.9*

Table 5: Per-annotator share (%) of corrections in each of the five action categories (bs. = BOUNDARY_SHIFT, da. = DA_CHANGE, ins./del./mod. = INSERT/DELETE/MODIFY); corpus row is all 977 change logs including the 503 unattributed file sides. BOUNDARY_SHIFT is modal for every identified annotator, but the dominance varies from 41% (T3) to 65% (T1). Cross-annotator divergence is largest in the semantic edit categories (DA_CHANGE, MODIFY), consistent with the σ^2_A estimate in Table 3.

project in which label edits dominate should expect a larger σ^2_V than we report.

Per-annotator divergence concentrates in label edits. Table 5 splits the shares by annotator. The cross-annotator gap in BOUNDARY_SHIFT share is large (T3 41.4%, T2 61.1%, T1 64.9%) but explains little in practice: boundary corrections are triggered by alignment drift, which varies with the audio, not with the annotator. The cross-annotator divergence concentrates in the semantic-edit categories: T3's files received 12.8% DA_CHANGE corrections (against 0.5–1.3% for the other two) and 7.5% MODIFY (against 0.9–1.5%). This pattern is consistent with the $\sigma^2_A \approx 11\%$ estimate in Table 3: the between-annotator variance lives primarily in how the original annotator handled borderline label and utterance-boundary decisions, not in how cleanly they re-aligned speech.

DA spikes track file difficulty, not annotator regression. Despite contributing only 3% of corpus-wide corrections, DA_CHANGE events cluster on specific files rather than distributing uniformly. Inspection of the highest-DA_CHANGE files in our corpus shows they are disproportionately conversations with heavy backchannel exchanges (where `b`/`aa`/`ny` distinctions are genuinely ambiguous) or extended joint laughter (where the boundary between `x` and substantive turns is unclear). The variance partition assigns the corresponding share to the file-level residual ($\sigma^2_e \approx 81\%$) rather than to any batch random effect, consistent with this reading: hard files are hard regardless of who annotated them or when.

Boundary-shift provenance. Because boundary shifts can reflect either annotator edits or residual forced-alignment errors, we ran a post-hoc three-state comparison between the original, annotated, and verified TextGrids. Among high-confidence same-text boundary changes from annotated to verified files, 50.3% refined a boundary the annotator had already moved toward the verified position, 32.2% corrected an original alignment error left unchanged by the annotator, and 16.2% reverted or corrected an annotator-introduced boundary change. These counts make the interpretation more cautious: many boundary corrections are verifier refinements of an active realignment process, not simple annotator failures.

Implication for tooling. If boundary placement accounts for nearly 60% of verifier effort across the team, future re-alignment projects should first improve boundary-placement support at annotation time. Examples include visual cues for low-confidence boundaries from the forced aligner, or a second-pass boundary-only review before full DA verification. We do not evaluate such interventions here, but the correction-type distribution points to boundary support before DA-label retraining.

7 Discussion

What a zero verifier share means. A common concern about cross-annotator verification is that per-annotator quality estimates are contaminated by whichever verifier happened to check an annotator's files. The verifier random effect u^V in Eq. 1 addresses this: it absorbs per-verifier leniency before the annotator random effect is estimated. On the RASwDA data the estimate lands at $\sigma^2_V = 0$ (singular fit), stable across seven sensitivity refits. We treat this as evidence of verifier interchangeability, not as a failed analysis. Once annotator, batch, and file length are controlled, the data does not support treating the five verifiers as systematically different. The crossed design is what makes this claim identifiable: without off-diagonal mass in Table 2, u^A and u^V would be confounded and only their sum recoverable.

Why the annotator share survives. $\sigma^2_A \approx 10.8\%$ is small in absolute terms but stable and interpretable: Table 5 localises the between-annotator gap in the semantic-edit categories (DA_CHANGE, MODIFY), where label judgment matters and the audio does not force convergence. The asymmetry

between σ^2_A and σ^2_V is what we take away: who annotated a file changes the expected correction count; who verified it does not.

Why this matters beyond RASwDA. The empirical zero-verifier result is specific to RASwDA's audio-anchored boundary-realignment task. The modeling setup can still be useful elsewhere. For corpus projects that use a verify-after-annotation workflow, the log format we release is modest: one change CSV per verified side, with columns for the action type, original and corrected intervals, and DA labels before and after (§3). With those logs, verifier strictness becomes an estimable source of variation rather than an unmeasured source of noise. A *non-zero* σ^2_V elsewhere would be just as informative: it would prompt verifier recalibration before release, or a per-verifier adjustment of downstream quality estimates.

Implications for future projects. For future projects, we recommend three design choices. (i) Log corrections at the action-category level rather than as scalar counts alone; the boundary-vs-label split in Table 5 is invisible without it and motivates targeted tooling. (ii) If verifier effects matter for the release claim, schedule verification so that every annotator is checked by at least two distinct verifiers, with overlap in both directions; this gives Eq. 1 the off-diagonal mass it needs. (iii) Release the change logs alongside the corpus so downstream users can re-run the variance partition under their own assumptions about what counts as an error.

8 Conclusion

We released 977 per-file verification logs from the RASwDA re-alignment of Switchboard, covering 105,005 corrections in five action categories with per-row time intervals and DA labels before and after. On this data we fit a cross-classified mixed-effects model on per-file corrections per interval. The variance partition assigns 10.8% to annotator identity, 8.7% to batch-within-annotator, 80.6% to file-level residual, and zero (singular fit, stable across seven refits) to verifier identity. For the released corpus this is the relevant quality-control result: once annotator, batch, and file length are controlled, we find no detectable verifier identity effect on correction rates. Boundary placement dominates corrections at 58.5% across the team, and most verifier decisions are anchored in the audio rather than in label judgment, which is why the verifier random effect has little to estimate in this task. Projects with action-typed per-file logs can fit the same cross-classified random effects model to their own verification data.

Limitations

Power limit on the verifier-interchangeability claim. The $\sigma^2_V = 0$ result is a null finding: at $N = 537$ complete-case file sides and five verifiers, we cannot distinguish "verifiers are truly indistinguishable" from "there is a real but small verifier effect the data lack power to resolve." The leave-one-verifier-out refits all reproduce the zero, which rules out one-verifier-driven instability but does not establish a tight upper bound on the effect. The appropriate framing is "no detectable verifier identity effect once annotator, batch, and file length are controlled," not "verifiers are identical." A dataset with tens of verifiers would be required to tighten this further.

Task-specific generalization. The empirical result is tied to the structure of RASwDA: the project mainly re-aligns existing dialogue-act intervals to audio, and the modal correction type is an audio-anchored boundary shift. Annotation projects dominated by subjective dialogue-act labels, discourse relations, or pragmatic functions may show nonzero verifier severity effects even under the same crossed design. What should transfer is the method: crossed verification logs make that effect measurable when it exists.

Unattributed annotator rows and informative missingness. 503 of the 1,045 file sides have no recoverable annotator attribution; they enter the corpus row of Table 5 and the verifier-side counts in §4 but are excluded from the 537-file complete-case mixed-effects fit. The missingness is informative: Mann–Whitney U tests reject equality of both n^{int} and total corrections between the attributed and unattributed pools ($p < 10^{-8}$ in both cases), and a χ^2 test on verifier identity likewise rejects ($p < 10^{-20}$). The unattributed rows are on median longer (282 vs. 231 non-silence intervals) but receive fewer corrections (51 vs. 91) than attributed rows. The variance partition should therefore be read as applying to the complete-case subset; extrapolation to the unattributed 503 rows requires assuming the annotator-blank files were produced by a member of the same three-person pool with

similar behavioural parameters, an assumption we cannot directly test.

File-difficulty residual. The corrections/interval normalisation controls for length but not for harder-to-quantify difficulty signals: speaker overlap density, disfluency rate, channel quality, topic familiarity. These are absorbed by the file-level residual variance $\sigma_e^2 = 80.6\%$ of total, which is a large mass that mixes measurement error with unmodelled file difficulty rather than pure noise. Decomposing it further would require covariates we do not have.

Boundary-shift source. The provenance analysis in §6 is post-hoc and relies on same-text interval matching across the original, annotated, and verified TextGrids. We therefore report only high-confidence same-text boundary changes and exclude ambiguous split, merge, and weak-match cases from the percentages. The analysis helps interpret the main pattern in the boundary corrections, but it is not a manual adjudication of every shifted boundary.

Annotator-side calendar precision. Verifier dispatch logs record exact submission dates. Annotator-side dates are recovered from self-reports for 542 of the file sides; the remaining attributed files use file-ID order as an ordinal proxy. Because the model uses batch-rank rather than calendar date, this affects the granularity of the experience axis but not its direction, and since neither learning slope reached significance, the precision of the x-axis is not the main constraint.

Sample size for inferential claims. With three annotators and five verifiers, our random-effects estimates have wide confidence intervals and should be read as evidence about this team's workflow rather than population-level claims about dialogue annotators in general. The methodological contribution, that cross-classified variance partition is a feasible and informative QC diagnostic on verification logs, does not depend on n.

No annotator-profile covariates. We do not collect prior annotation experience, linguistics background, or training-hour covariates for the five team members. Such covariates would let the model explain *why* the random intercepts differ; without them we can only report *that* they differ.

No annotator-by-verifier interaction term. Equation 1 treats annotator and verifier effects as additive. A pair-specific interaction would ask whether, for example, one verifier is unusually strict toward one annotator. With three annotators, five verifiers, empty self-verification cells, and uneven cell counts, that interaction is weakly identified and would be unstable. We leave pair-specific bias modeling to larger verification designs.

References

James Allen and Mark Core. 1997. DAMSL: Dialog act markup in several layers. Draft, University of Rochester.

R. H. Baayen, D. J. Davidson, and D. M. Bates. 2008. Mixed-effects modeling with crossed random effects for subjects and items. *Journal of Memory and Language*, 59(4):390–412.

Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. 2015. Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1):1–48.

Petra Saskia Bayerl and Karsten Ingmar Paul. 2011. What determines inter-coder agreement in manual annotations? A meta-analytic investigation. *Computational Linguistics*, 37(4):699–725.

Paul Boersma and David Weenink. 2024. Praat: Doing phonetics by computer. Computer program. Version 6.4, retrieved 2024.

A. P. Dawid and A. M. Skene. 1979. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):20–28.

Thomas Eckes. 2019. *Many-Facet Rasch Measurement*, pages 153–175. Routledge.

John J. Godfrey, Edward C. Holliman, and Jane McDaniel. 1992. SWITCHBOARD: Telephone speech corpus for research and development. In *Proceedings of the 1992 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 1, pages 517–520, San Francisco, CA. IEEE.

Daniel Jurafsky, Elizabeth Shriberg, and Debra Biasca. 1997. Switchboard SWBD-DAMSL shallow-discourse-function annotation coders manual, draft 13. Technical Report 97-02, University of Colorado, Boulder.

Omer Nahum, Nitay Calderon, Orgad Keller, Idan Szpektor, and Roi Reichart. 2025. Are LLMs better than reported? Detecting label errors and mitigating their effect on model performance. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 26782–26809, Suzhou, China. Association for Computational Linguistics.

Shinichi Nakagawa and Holger Schielzeth. 2013. A general and simple method for obtaining R^2 from generalized linear mixed-effects models. *Methods in Ecology and Evolution*, 4(2):133–142.

Rebecca J. Passonneau and Bob Carpenter. 2014. The benefits of a model of annotation. *Transactions of the Association for Computational Linguistics*, 2:311–326.

Silviu Paun, Bob Carpenter, Jon Chamberlain, Dirk Hovy, Udo Kruschwitz, and Massimo Poesio. 2018. Comparing Bayesian models of annotation. *Transactions of the Association for Computational Linguistics*, 6:571–585.

Elizabeth Shriberg, Rebecca Bates, Andreas Stolcke, Paul Taylor, Daniel Jurafsky, Klaus Ries, Noah Coccaro, Rachel Martin, Marie Meteer, and Carol Van Ess-Dykema. 1998. Can prosody aid the automatic classification of dialog acts in conversational speech? *Language and Speech*, 41(3–4):443–492.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–374.

Michiel van der Meer, Neele Falk, Pradeep K. Murukannaiah, and Enrico Liscio. 2024. Annotator-centric active learning for subjective NLP tasks. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18537–18555, Miami, Florida, USA. Association for Computational Linguistics.

Tharindu Cyril Weerasooriya, Alexander Ororbia, Raj Bhensadadia, Ashiqur KhudaBukhsh, and Christopher Homan. 2023. Disagreement matters: Preserving label diversity by jointly modeling item and annotator label distributions with DisCo. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4679–4695, Toronto, Canada. Association for Computational Linguistics.

Wenbo Zhang and Yuhan Wang. 2025. Act2P: LLM-driven online dialogue act classification for power analysis. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20494–20504, Vienna, Austria. Association for Computational Linguistics.