DOI 10.34185/1562–9945–5–166–2026–08
UDC 004.934:004.89

A.M. Ivanko

JOINT LLM RECONCILIATION OF MULTIPLE ASR TRANSCRIPTS FOR SCRIPT–COMPLIANCE CUE DETECTION IN CONTACT CENTERS

Annotation. Contact–center quality assurance audits script compliance – whether the agent greeted the caller, identified the company, asked for the caller's name, and closed politely – using automatic speech recognition (ASR) transcripts. The cues that carry these checks are short formulaic phrases, and they are exactly what an imperfect recognizer drops or garbles, so the operative metric is recall of the cue rather than word error rate. Classic ASR system combination votes at the token layer and can delete a word only if one engine heard it, which is precisely the word a compliance check needs. This paper therefore asks where the ensemble should live for a detection task and compares four fusion points under a single LLM compliance judge: single–ASR, token–level, decision–level, and joint–hypothesis–level reconciliation. On a disagreement–enriched stratum of a 3057–call contact–center corpus, joint reconciliation is the best arm (macro–F1 0.913), significantly above single–ASR, decision–level OR and ROVER for both judges, and the ordering survives re–scoring on a stratum containing only human–set labels. Three controls show that the gain requires a nonidentical, call–matched second transcript, and attribute it partly to text visible only in that transcript.
Keywords: automatic speech recognition, ASR system combination, contact–center quality assurance, script compliance detection, large language models, LLM–as–judge, hypothesis–level fusion, word error rate, evaluation protocol.

Introduction. Contact centers increasingly audit script compliance at scale: ASR transcribes a call and a downstream classifier checks whether the agent performed required steps – a greeting, a company or "customer support" self–identification, asking for the caller's name, and a courteous closing. The cues that carry these checks are short, formulaic phrases, and they are exactly what an imperfect recognizer tends to drop or garble. A missing word is a missed cue, so what governs this task is detection, not transcription quality. Recall motivates it – an erased cue cannot be recovered downstream – but an auditor equally cannot afford to flag steps never performed, so we report positive–class F1 and Cohen's κ per parameter rather than recall alone. The two cues carrying most of the gain improve by opposite routes: agent–name by recall, company–id by precision.

Statement of the problem. This mismatch matters because a common ASR–combination baseline for using more than one recognizer – system combinations such as ROVER [1] and MOVER [8] – fuses at the token layer to minimize word error rate (WER)

© Ivanko A.M., 2026
Copyright for this paper by its authors. Use permitted under License CC BY 4.0.

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

and token voting can delete a word only one engine heard, which is precisely the word a compliance check needs.

We therefore ask a placement question rather than proposing a new architecture: where should the ensemble live when the metric is cue recall? Holding two de–correlated ASR engines fixed, we compare four fusion points under one LLM compliance judge: 1) single – one transcript, one judge; 2) token–level ROVER/TOV – merge the transcripts, then judge; 3) decision–level OR – judge each transcript and take the union of verdicts; and 4) joint–hypothesis–level JOINT – one judge reads both transcripts and reconciles them.

Contributions. A natural objection is that JOINT "just concatenates two transcripts in a prompt." Three results say otherwise. On a disagreement–enriched subset of a 3057–call corpus, JOINT significantly improves over single–ASR, decision–level OR and the reference ROVER implementation for both judges (macro–F1 0.913), while against our own stronger time–overlap voter the contrast does not resolve. The concatenation is not what helps: handing the judge the same transcript twice scores below handing it once and a transcript of a different call is worse still, while evidence visible only in the second transcript – 6.9% of positive verdicts – supplies 35.7% of the verdicts where JOINT correctly overturns the single arm. An agreement–stratified protocol reported on two populations makes explicit which estimates are affected by label–construction circularity, including that the silver label–building judge is one of those evaluated.

Related Work

ASR system combination. Combining recognizers at the token level is a long line of work: ROVER aligns hypotheses into a confusion network and votes [1, 16], with extensions that add language–model information [17] or a trained classifier over the network [3]; MOVER revisits this for modern meeting systems [8]. A separate thread fuses multiple ASR outputs for a downstream task rather than for the transcript itself – attention over multiple hypotheses [9] and multi–ASR fusion for pseudo–labeling [13], both of which still emit a single merged transcript.

Closest to our work is post–ASR fusion under an LLM: Stepachev et al. [20] prompt an LLM over multiple ASR system outputs for downstream emotion recognition and report that transcript ranking and "system output fusion" help, and crucially that the metric used to select the transcript is decisive. We share the premise that the LLM, not a confusion network, is where multiple recognizers should meet, but differ on three axes. 1) They treat fusion as a single design choice (rank–then–select or concatenate) for a holistic label; we make the fusion point itself the independent variable and contrast four placements under one fixed judge. 2) Our target is cue recall – short, localized phrases a single ASR error can erase outright, unlike emotion, which is diffuse over the utterance and rarely hinges on one token; hence reconciliation rather than selection. 3) We add an agreement–stratified evaluation protocol, which does not appear in prior post–ASR fusion work.

LLM–as–judge and juries. Using an LLM to score outputs is now standard [21], and jury methods aggregate several judges. Recent analyses show correlated judge errors cap the value of larger panels [10] and that confidence gating can give reliability guarantees [7].

These vary the judge over a single input; we hold one judge and vary the ASR input, letting the judge reconcile the two renderings.

Contact–center compliance. Prior compliance detection is largely text–based: on–hold–script detection with a fine–tuned classifier cross–checked against system hold timestamps [2], transcription pipelines [12], and silence/hold monitoring from audio [4]. We treat compliance as a multi–ASR, training–free judging problem scored with an agreement–stratified protocol.

Research objective. This work determines where an ASR + LLM compliance pipeline should place its ensemble of recognizers when what governs the task is detection of a short script cue – a cue an ASR error can erase outright – rather than word error rate. Four fusion points are compared for a fixed pair of recognizers under a single compliance judge, and the mechanism behind the observed ordering is constrained by input interventions and a per–verdict evidence analysis.

Method and Evaluation Protocol. Each call is scored on seven binary compliance parameters: the agent states their own name (agent–name), identifies the company or "customer support" (company–id), asks the caller's name (caller–name), and, at closing, is courteous, thanks the caller, gives a well–wish, and offers further help. A parameter is judged over the relevant window of the agent channel.

Four fusion points (Figure 1). Two recognizers stay fixed: Qwen3–ASR [19] and Canary–1B–v2 preceded by a telephony–tuned MarbleNet VAD [18, 5], MarbleNet performing voice–activity detection rather than recognition. Because the compliance windows are cut from word timings, each recognizer keeps its own word–level aligner – Qwen3–ForcedAligner–0.6B [19] and the NeMo Forced Aligner [15] – both lossless, every word survives into the aligned list, so windowing adds no deletions of its own. We vary only where the evidence is combined: single (one transcript, one judge); ROVER/TOV (merge by token–level voting, then judge), the two token–level arms being ROVER, the NIST SCTK rover binary, and TOV, our own time–overlap word voter; OR (judge each transcript, take the logical OR – a recall–union); and JOINT (one judge call receives both transcripts and reconciles them). ROVER is handed Qwen first: with uniform confidences every disagreement is a one–one tie and rover resolves it to the first hypothesis, so ties fall to Qwen, the stronger recognizer by WER (Table 2). Neither recognizer exposes word–level confidences, so ROVER votes on occurrence alone – the reference configuration, and the one its published description assumes [1]. All judges are local instruction–tuned LLMs at temperature 0 with the same prompt, so the only variable is the fusion point.

MOVER's released system is not public, so we could not run it; TOV is written in its spirit [8] but is not it. The distinction is not pedantry. With only two systems and no word–level confidences, the tie–breaking and gap–fill rules are our design decisions, and they are measurable rather than incidental: TOV emits 22.2% more agent–channel words than the stronger recognizer, which is what makes it the worst arm on WER and the best token–fusion arm on cue detection (Tables 1 and 3). Printing a published method's name over numbers produced by our own implementation would attribute both that design and that result to work

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

that did not make them. ROVER is therefore the reference implementation, and it is the arm that carries every token–fusion conclusion in this paper; TOV bounds what a straightforward time–constrained voter achieves on two confidence–free systems. The tie order is ours to choose, so we test it: rebuilding ROVER with Canary first – ties to the weaker recognizer – changes 31.3% of agent–channel tokens and all 531 fused transcripts, yet moves macro–F1 by +0.008 [–0.005, +0.021] (gemma) and –0.011 [–0.027, +0.004] (qwen). The signs disagree and both intervals cover zero: the tie order does not limit the arm.

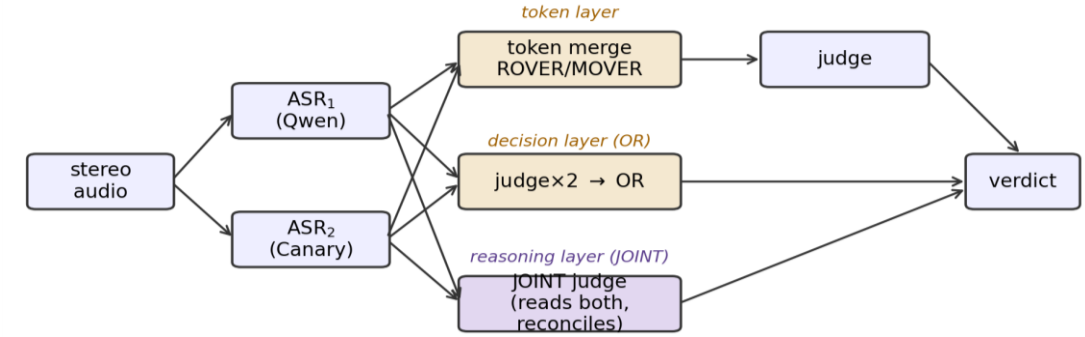


Figure 1 – Four fusion points for the same two ASR engines

Token–level merge (ROVER/TOV), decision–level OR, and joint–hypothesis–level JOINT. Only the combination point changes.

We use 3057 real contact–center calls. The unit of evaluation is a cell – one (call, parameter) pair – so $N_{\text{calls}} = 3057$ and $N_{\text{cells}} = 3057 \times 7 = 21399$; percentages below are over cells and call counts over calls, and the two are not interchangeable.

Labels are of two kinds. Silver, 19692 cells (92.0%): set by detector agreement – a regex detector agreeing with an LLM judge, or, where those disagree, two LLM judges agreeing with each other. Agreement between detectors does not exclude common–mode error, so silver labels are provisional, and no human saw them. Human, 1707 cells (8.0%): a human adjudicated the cell from the call audio, with the transcript window shown alongside it and blind to the regex and judge outputs. Adjudicating from audio makes the label independent of any recognizer – a label read off a transcript cannot distinguish a cue never spoken from one the recognizer deleted, which is the distinction this paper measures – and the blinding keeps a human label from inheriting the bias of a prediction it was shown. The LLM judge that builds the silver tier is Qwen, one of the judges we evaluate, so on a silver cell Qwen is partly scored against its own earlier output. The bias runs in Qwen's favour, making the gemma–over–qwen contrast conservative rather than inflated; it is nonetheless a circularity, and removing it is what the restricted stratum below is for.

The judges were run over the 531 calls recruited by the two adjudication campaigns: 278 with opening–parameter labels and 323 with closing–flag labels, of which 70 carry both ($278 + 323 - 70 = 531$). The stratum is defined by campaign, not by the presence of a human label: 14 further calls carry an isolated human closing label (15 cells) set outside either campaign, and two opening cells were routed to a human but returned no parseable label. All 17 are counted in the 1707 above and none is scored anywhere in this paper. Scoring is per

call – once a call entered the stratum, every parameter of it carrying a usable label was scored – yielding 2124 cells: 832 opening cells (277 agent–name, 277 company–id, 278 caller–name; two of the 834 possible cells lack a parseable label) and 1292 closing–flag cells. Those cells are not uniformly human–adjudicated: 1690 (79.6%) carry a human label – all 1292 closing–flag cells and 398 of the 832 opening cells – and the remaining 434 are silver, i.e. the Qwen–self–scored cells identified above. Every macro–F1 and κ below is therefore computed twice, over all 2124 cells and over the restricted stratum of 1690 whose label a blinded human set. The restriction removes regex and Qwen from the label's assignment but not from its selection, since a cell reaches the human tier by detector disagreement; estimates on it describe a disagreement–enriched population and are not weighted estimates for the corpus. The four closing flags are identical across the two populations, being 100% human–adjudicated. An independently labeled 150–cell sample from the silver tier agrees with the silver labels on 95.3% of cells (143/150; Wilson 95% CI 90.7–97.7%).

Macro–F1 and Cohen's κ, each averaged over the seven parameters (Table 1). "Macro" denotes the mean over the seven parameters, not over the two classes: the per–parameter F1 is the positive–class F1, so a parameter's score reflects detection of the cue rather than of its absence. Per–parameter precision and recall are in Figure 2. Significance uses a paired uuid–cluster bootstrap on macro–F1 (5000 resamples) as the primary test, since cells nest within calls, and a cluster sign–flip permutation test on the same statistic (10000 permutations) where a contrast is close. An exact McNemar test was computed as a cross–check but gates no claim and is not tabulated: it estimates paired item correctness rather than a difference in macro–F1, and ignores the clustering of cells within calls.

Experiments. Table 1 reports macro–F1 and macro–κ for the regex baseline and two LLM judges across all fusion points, on the full 2124–cell stratum and the restricted human–only one. JOINT with the stronger judge (gemma) is best at 0.913 and is the top arm for both judges on both populations. The intermediate arms order differently per judge on the full stratum but agree on the restricted one (JOINT $>$ TOV $>$ ROVER $>$ OR $>$ single); the instability belongs to the contaminated cells, not the judges.

One rule applies throughout: a contrast resolves when the cluster bootstrap interval on macro–F1 excludes zero, the primary test named in the Method. Under it JOINT$>$OR (ΔF1 +0.029) and gemma$>$qwen (+0.033) hold on both populations, and on the restricted stratum JOINT also beats single and OR for qwen (Table 3), so the advantage is not confined to the stronger judge. JOINT exceeds ROVER, the reference token–fusion implementation, for both judges and both populations (gemma +0.023 [+0.010, +0.036] and +0.021 [+0.005, +0.037]; qwen +0.036 [+0.021, +0.051] and +0.031 [+0.014, +0.050]). Against TOV the answer is population–dependent – resolving on the full stratum (+0.015 [+0.004, +0.026]) but not the restricted one (+0.012 [–0.001, +0.026]), the cluster permutation test agreeing (p=0.05 and p=0.14) – so we take the conservative reading and claim no separation from TOV. TOV outscores ROVER (0.898 against 0.890 for gemma), so the arm JOINT fails to

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

separate from the stronger token–fusion arm, and it is the one that is not a published method. Reporting the two single–transcript arms separately answers whether the gain merely inherits the better engine. It does not: Qwen–only leads Canary–only only slightly for every judge (gemma 0.872 against 0.860), while gemma–JOINT at 0.913 exceeds both by 0.041 and 0.053 – several times the gap between the recognizers themselves. Hereafter "single" denotes the Qwen–only arm, the stronger of the two and the production recognizer. The regex arms are competitive on F1 (0.864–0.891, behind only gemma–JOINT and gemma–TOV) but are the worst arms in Table 1 on κ (0.549–0.601), below every LLM arm including single–transcript qwen (0.657). The two statistics disagree because the parameters are strongly imbalanced – positive prevalence ranges from 0.134 (company–id) to 0.906 (agent–name) – and F1 rewards a detector that fires readily on high–prevalence cues, whereas κ discounts the agreement prevalence alone would produce. For the LLM arms the two agree on the ordering, and κ makes it cleaner: JOINT is the top arm for both judges on both statistics (gemma 0.709, qwen 0.704) and exceeds the best token–fusion arm on κ for both (gemma TOV 0.695, qwen TOV 0.664), where on F1 the qwen comparison was ambiguous.

Table 1

Macro–F1 / macro–κ by fusion point, on both evaluation populations

Judge	stratum	stat	Qwen	Canary	ROVER	OR	TOV	JOINT
regex	full	F1	—	—	0.887	0.891	0.864	—
regex	full	κ	—	—	0.601	0.573	0.549	—
regex	restr.	F1	—	—	0.879	0.883	0.852	—
regex	restr.	κ	—	—	0.593	0.554	0.536	—
qwen	full	F1	0.839	0.829	0.845	0.851	0.849	0.880
qwen	full	κ	0.657	0.633	0.657	0.688	0.664	0.704
qwen	restr.	F1	0.796	0.790	0.822	0.805	0.823	0.853
qwen	restr.	κ	0.598	0.584	0.620	0.603	0.635	0.649
gemma	full	F1	0.872	0.860	0.890	0.884	0.898	**0.913**
gemma	full	κ	0.660	0.631	0.693	0.667	0.695	**0.709**
gemma	restr.	F1	0.866	0.854	0.889	0.873	0.898	**0.910**
gemma	restr.	κ	0.623	0.613	0.673	0.617	0.674	**0.682**

Full stratum 2124 cells, restricted human–only stratum 1690 cells; best per population in bold. Qwen and Canary are the two single–transcript arms. Both statistics are means over the seven parameters. The regex detector is defined over a merged transcript and has no single–recognizer arm.

The gains are not uniform (Figure 2). Caller–name (0.93–0.98) and further (0.96–0.99) are near–ceiling across every gemma and qwen arm, so the fusion point is interchangeable there; the remaining closing flags are lower than "near–ceiling" implies – courtesy 0.91–0.95, wellwish 0.88–0.95, and thanked lowest at 0.81–0.89 – so the interchangeability argument covers two parameters, not five. The differences concentrate on the two hardest cues: +0.119

on company–id and +0.099 on agent–name, against under +0.05 on every other parameter. We describe these as the two opening cues rather than as the ones ASR most often erases, since we report no per–cue erasure rate; the concentration is measured and the erasure explanation for it is not. They also improve by different routes, set out below. Agent–name is the hardest parameter and remains difficult even under joint–hypothesis–level fusion with the weaker judge (qwen–JOINT 0.681), while gemma–single reaches 0.786 and gemma–JOINT 0.885. On company–id the regex baseline reaches only 0.696 – against 0.951 for agent–name on the same arm, the gap that exposes it as a near–constant predictor rather than a detector – while gemma–JOINT (0.824) is the strongest arm. The premise of token–level voting is that it yields a better transcript; Table 2 tests that premise directly on this data. Because WER is sensitive to text normalization, we report three scorings of the same alignments: NIST SCTK sclite [11] with GLM contraction alternations; the Whisper EnglishTextNormalizer [14] with jiwer [6], the most lenient; and a stop–word–filtered content–word variant, the strictest. On the agent channel – the channel every compliance parameter is scored on – Qwen3 is best under all three, and neither fused transcript improves on it: ROVER degrades it by 3.0 points and TOV by 13.2 (Whisper–normalized). The cause is insertion, not a lost vote: ties go to Qwen and, as shown above, reversing them changes nothing. What both fused arms do instead is fill gaps – a word one recognizer deleted and the other kept enters the fused transcript. On the same 20 calls ROVER emits 11.3% more agent–channel words than Qwen3 and TOV 22.2% more (17519 against 14341). This also explains why TOV posts the worst WER in Table 2 and the best token–fusion macro–F1 in Table 1, which would otherwise look like an implementation fault. WER charges for every spurious insertion; cue detection does not, because an inserted word cannot erase a cue that is present, and some injected words are ones Qwen dropped. The divergence is this paper's thesis in miniature: the arm that is worst on the transcription objective is the best token–level arm on the detection one. As set out in the Method, the TOV row bounds rather than measures, so the conclusion drawn from Table 2 rests on the ROVER row, where ties fall to Qwen and ROVER still does not reach the stronger recognizer alone. Qwen3 is best and TOV worst under all three scorings, so the conclusion is not an artifact of normalization, though the middle of the ranking is not invariant. Combined with Table 1, this sharpens the claim: token–level fusion is worse on its own objective and no better on ours, whereas joint–hypothesis–level reconciliation improves cue detection without producing a transcript at all.

Table 2

Agent–channel WER (%) against a 20–call human reference, under three scorings

System	NIST sclite	Whisper–norm	content–word
Qwen3–ASR	**25.4**	**23.2**	**29.9**
Canary–1B–v2	29.3	28.0	30.8
ROVER (token fusion)	28.7	26.2	32.2
TOV (our time–overlap voter)	38.8	36.4	36.4

Lower is better. This reference set is a 20–call subset, disjoint in purpose from the 531–call stratum used elsewhere.

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

Macro–F1 collapses false positives and false negatives into one number, so two arms with the same F1 can fail in opposite directions. Figure 2 reports precision, recall and F1 side by side for every parameter and arm: the stem spans precision to recall and F1, their harmonic mean, falls on it, so the stem's length shows how lopsided an arm's errors are and its lower endpoint shows which error dominates (y–axis from 0.30). The two cues carrying the gain sit at opposite extremes, which is why we do not attribute both to one mechanism. On agent–name precision is at or near 1.00 for every LLM arm while recall runs from 0.359 (qwen–single) to 0.825 (gemma–JOINT): the entire error budget is missed cues, and moving the fusion point buys recall without costing precision – the signature of evidence recovery rather than of a changed operating threshold. On company–id the picture inverts: recall is 0.84–0.95 across the six gemma and qwen arms while precision ranges 0.47–0.73, so the errors are false alarms. Decision–level OR is the worst arm for every judge, because a union of two verdicts propagates either engine's spurious match (qwen–OR 0.465, gemma–OR 0.550), whereas JOINT improves precision over its own single arm for both judges (gemma 0.608 to 0.729, qwen 0.620 to 0.647). On caller–name and further the stems collapse and the three statistics coincide, which is why the fusion point is interchangeable there; on the remaining closing flags the stems are short but not degenerate.

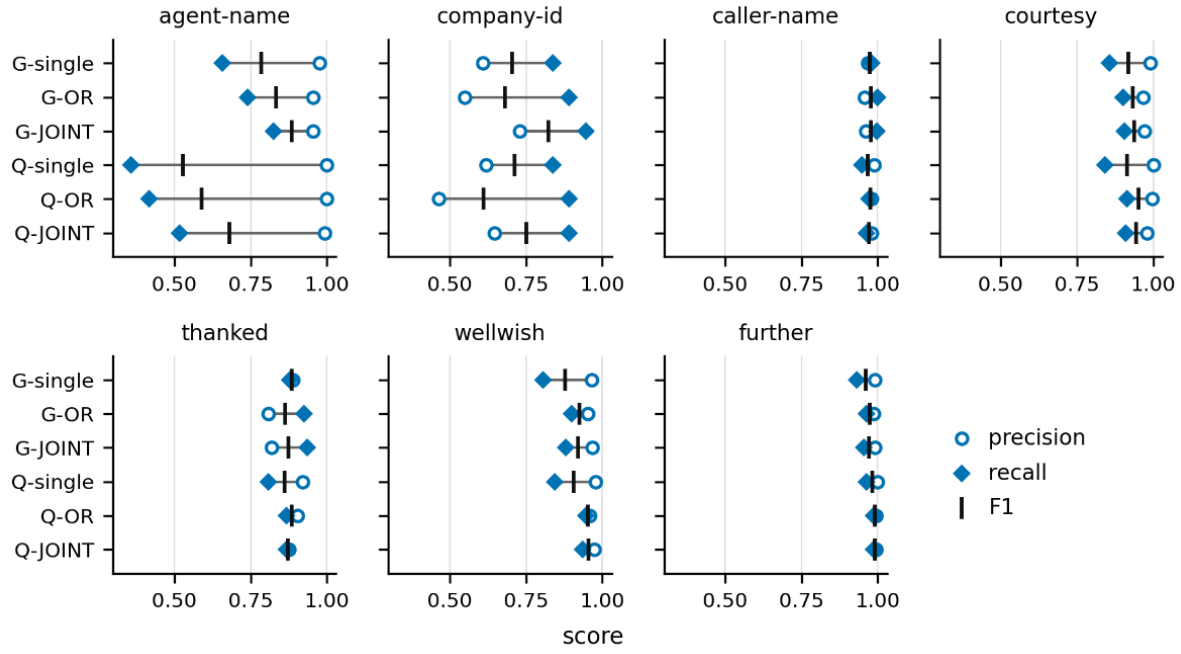


Figure 2 – Precision, recall and F1 by compliance parameter and arm

Table 3

Paired uuid–cluster bootstrap on macro–F1, both populations

contrast	ΔF1	95% CI	ΔF1	95% CI
G–JOINT − G–OR	+0.029	[+0.019, +0.039]	+0.037	[+0.025, +0.049]
G–JOINT − Q–JOINT	+0.033	[+0.017, +0.049]	+0.057	[+0.039, +0.077]
G–JOINT − G–single	+0.041	[+0.029, +0.054]	+0.044	[+0.031, +0.058]
Q–JOINT − Q–single	+0.042	[+0.028, +0.055]	+0.056	[+0.040, +0.074]
Q–JOINT − Q–OR	+0.029	[+0.018, +0.041]	+0.048	[+0.035, +0.063]
G–OR − G–single	+0.012	[+0.004, +0.020]	+0.008	[–0.002, +0.017]
G–JOINT − G–ROVER	+0.023	[+0.010, +0.036]	+0.021	[+0.005, +0.037]
Q–JOINT − Q–ROVER	+0.036	[+0.021, +0.051]	+0.031	[+0.014, +0.050]
G–JOINT − G–TOV	+0.015	[+0.004, +0.026]	+0.012	[–0.001, +0.026]
G–JOINT − G–shuf	+0.196	[+0.174, +0.219]	+0.183	[+0.159, +0.209]
G–single − G–shuf	+0.155	[+0.133, +0.178]	+0.139	[+0.115, +0.163]
G–JOINT − G–dup	+0.117	[+0.098, +0.137]	+0.116	[+0.097, +0.136]
G–dup − G–single	−0.076	[–0.096, –0.057]	−0.072	[–0.093, –0.051]

5000 resamples. G = gemma, Q = qwen, shuf = the shuffled–transcript control arm, dup = the duplicate–transcript control arm. Columns 2–3 are the full stratum, columns 4–5 the restricted stratum. A contrast resolves only when the interval excludes zero. Deltas are computed on unrounded macro–F1, so one may differ by 0.001 from a subtraction of Table 1's printed values.

A fifth of the stratum carries a label no human set, selected by detector agreement, which makes those cells easy by construction and inflates every arm rather than only the label builders. Re–scoring on the 1690 human–set cells (Tables 1 and 4) leaves the headline intact – gemma–JOINT 0.913 to 0.910 – while the contrasts move both ways: every contrast against a non–token arm grows (JOINT over OR +0.029 to +0.037, over single +0.041 to +0.044, over qwen–JOINT +0.033 to +0.057) and every contrast against a token–fusion or control arm shrinks slightly (over ROVER +0.023 to +0.021, over TOV +0.015 to +0.012, over the shuffled arm +0.196 to +0.183). The gain over judging one transcript is larger on the clean cells, the margin over token fusion smaller. One conclusion weakens outright: decision–level OR no longer separates from single–ASR for gemma (+0.008, interval spanning zero). The cost of the restriction is statistical power on the opening parameters – caller–name falls from 278 cells to 76 and company–id from 277 to 114, so their restricted–stratum estimates are correspondingly noisier.

The restriction also settles what the regex baseline is doing. Its agent–name F1 rises on the 208 cells it never labeled, from 0.951 to 0.970, which is the opposite of what label–construction circularity predicts. The explanation is degeneracy: on those cells it answers "yes" 208 times out of 208, and 0.970 is exactly the F1 a constant "yes" earns at the restricted stratum's 0.942 prevalence; its κ there is 0.000. The regex baseline is not inflated by the la-

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

bels it helped build – it is a prevalence–exploiting constant on this parameter, which is precisely the failure κ is designed to expose and F1 is not.

A shuffled–transcript control. Aggregate F1 cannot distinguish reconciliation from a simpler account – that two transcripts merely make the judge readier to answer yes – so we re–ran JOINT with ASR VERSION 2 replaced by a transcript of a different call: a seeded derangement over the same stratum, no call paired with itself, everything else fixed including the instruction that the two transcripts are of the same audio. Macro–F1 falls from 0.913 to 0.717, not back to the single–transcript arm (0.872) but 0.155 below it. Both drops resolve: JOINT over the shuffled arm is +0.196 [+0.174, +0.219] and single over it +0.155 [+0.133, +0.178] (Table 3), holding on the restricted stratum (+0.183 and +0.139). Macro–κ separates them more sharply, 0.709 to 0.192: once the second transcript belongs to another call, almost none of the judge's agreement with gold survives chance correction. The lift does not survive the swap, and the swap costs more than the lift was worth – so a transcript of the same call is necessary for the second input to help; the next control shows it is not sufficient. The collapsed arm is not degenerate, which matters because a judge that simply stopped answering would drop the same way: it returns 1280 positive and 844 negative verdicts against the real arm's 1359 and 765, agrees on only 68.7% of cells, and its disagreements run both ways, 372 lost and 293 gained. A permissiveness account predicts more positives; we observe fewer.

A duplicate–transcript control. That leaves a second alternative alive: that the gain comes from the JOINT prompt, or from reading the same evidence twice, rather than from combining two different renderings. A third arm makes ASR VERSION 2 byte–identical to VERSION 1 – both the Qwen window the single arm reads – everything else fixed including the JOINT paragraph. Reconciliation predicts this arm falls back to the single arm; a prompt or re–reading account predicts it keeps the lift. Neither happens: it lands below the single arm, at 0.796 against 0.872 (Table 4). JOINT exceeds it by 0.117 [+0.098, +0.137] – nearly three times JOINT's margin over single – and the duplicate arm is worse than single by 0.076 [–0.096, –0.057], both $p < 0.001$ under the cluster permutation test. The lift is therefore not a property of the prompt or of re–reading: the benefit requires a second transcript that actually differs. One asymmetry qualifies that reading. Duplication helps on the three opening parameters (+0.044 over single) and hurts sharply on the four closing flags (-0.166), where the judge turns conservative – its positive rate on further falls from 0.635 to 0.427 against a gold rate of 0.675. A plausible reason is positional rather than informational: closing cues sit at the end of their window, and a second copy pushes the first copy's ending into the middle of the context, while an opening cue stays at the top. We did not test this, so we report the aggregate – the prompt account is refuted – and flag the duplicate arm as a cleaner control for start–anchored cues than end–anchored ones.

Table 4

The four judge arms under gemma, on both populations

arm	F1	κ	agree	F1	κ	agree
JOINT (real 2nd ASR)	**0.913**	**0.709**	—	**0.910**	**0.682**	—
single (no 2nd ASR)	0.872	0.660	92.7%	0.866	0.623	91.9%
dup (V2 = V1)	0.796	0.503	83.7%	0.794	0.472	80.3%
shuf (V2 = other call)	0.717	0.192	68.7%	0.727	0.172	63.1%

All four read the same windows and differ only in what occupies ASR VERSION 2. Columns 2–4 are the full stratum, columns 5–7 are the restricted stratum. Agreement is the share of cells on which an arm returns the same verdict as JOINT.

The controls above intervene on the input; they show the second transcript matters without showing what the judge takes from it. We re–ran JOINT asking it to quote the span it relied on and matched every quote against the two windows it was given. It is never asked which transcript it used – a model that fabricates a cue can fabricate the attribution as easily – so each quote is located by string match, present when a contiguous run of at least 70% of its normalized tokens occurs in the window. The analysis covers all seven parameters and the 2126 cells the re–run reached, two more than the 2124 scored elsewhere because bucketing a quote needs no gold label. Asking for a quote barely perturbs the verdicts: the re–run agrees with the scored arm on 828 of 834 opening cells (99.3%) and 1220 of 1292 closing cells (94.4%), the latter with near–symmetric disagreements (41 against 31) – noise, not a shift.

Of 1368 positive verdicts, 322 (23.5%) quote evidence found in only one transcript (Table 5). That figure is not by itself the mechanism, and it is worth being precise about why: the single–transcript arm reads the Qwen window, so the 16.6% of verdicts resting on Qwen–only evidence cite text the single arm already had. Only the 6.9% resting on Canary–only evidence cite text the single arm could not see. Taken alone, 6.9% looks too small to carry a +0.041 macro–F1 gain.

It is not taken alone. What matters is not how often second–transcript evidence appears but where, and it is concentrated almost entirely on the decisions the single arm gets wrong. Three quarters of Canary–only verdicts – 71 of 95 – are cells the single arm called negative, against 7.8% of the verdicts resting on evidence present in both transcripts: a nine–fold enrichment. Of the 129 verdicts where JOINT correctly overturns a single–arm negative, 46 (35.7%) rest on Canary–only evidence, from a bucket that is 6.9% of all positives. Evidence visible only in the second transcript is thus rare overall and dominant exactly where the arms disagree, which is what an evidence account predicts and a prompt or permissiveness account does not. Fabrication is rare but not absent: five verdicts (0.37%) quote text in neither window. Recovery and fabrication move together with the match threshold – at 50–60% they are 15.9–19.5% and 0.00%, at exact matching 37.3% and 2.5%.

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

Table 5

Where the quoted evidence occurs, and where it changes a decision

evidence occurs in	n	share	single NO	% of bucket	gold agrees
BOTH	1041	76.1%	81	7.8%	69
QWEN–ONLY	227	16.6%	18	7.9%	13
CANARY–ONLY	**95**	**6.9%**	**71**	**74.7%**	**46**
NEITHER	**5**	**0.37%**	1	—	1
total	**1368**		**171**		**129**

All seven parameters, over the 2126 cells the evidence re–run reached; 1368 positive verdicts. QWEN–ONLY is text the single–transcript arm already sees, CANARY–ONLY text it cannot. Match rule: a contiguous run of 70% or more of the quote's normalized tokens. "single NO" counts cells the single–transcript arm called negative; the last column keeps only those where the gold label agrees with JOINT. NEITHER is the fabrication rate; no verdict returned an empty quote.

Conclusion. For contact–center script compliance – a cue–recall task in which an ASR error can erase the evidence – where the recognizers are combined materially affects performance within the evaluated two–ASR setting. A capable judge reconciling both transcripts in one call (JOINT) is the best arm, significantly above decision–level OR and single–ASR for both judges and on a restricted stratum of human–set labels only. Token–level voting, the common WER–driven baseline, led in no configuration: JOINT significantly exceeds the reference ROVER implementation for both judges and both populations, while against our own stronger time–overlap voter the contrast does not resolve.

The building blocks are established and we concede the field as Related Work sets it out: attention over multiple hypotheses [9], multi–ASR fusion for transcript improvement [13], and post–ASR fusion under an LLM for a holistic label [20]. Using an LLM to combine multiple ASR outputs is not what we claim as new. The contribution is empirical and methodological rather than architectural, and is threefold: 1) making the fusion point the independent variable for a cue–recall objective, and showing placement is an operative lever; 2) a pair of input interventions that falsify the mechanism rather than illustrate it – the second input replaced once by a transcript of a different call and once by a copy of the first, both scoring below the single–input arm – which, with per–verdict evidence bucketing, turns "reconciliation" from a label into a measured quantity; and 3) an agreement–stratified protocol on two populations, exposing both that a strong–looking regex baseline is a prevalence–exploiting constant and that one evaluated judge helped build the silver labels.

The recommended configuration is a drop–in for any production ASR+LLM compliance auditor already running two recognizers: relative to a single–ASR pipeline it costs a second ASR pass plus one judge call per item, with no cross–system hypothesis alignment and no fine–tuning. Per–recognizer forced alignment still cuts the windows exactly as in the single–ASR baseline; what JOINT removes is the alignment of one engine's hypothesis to the other's that a confusion network requires. Against decision–level OR, which also needs both transcripts, JOINT halves the judge calls. The guidance is to invest in a capable reconciling judge rather than in token–level fusion, and the agreement–stratified protocol is reusable by

QA teams wanting honest estimates against provisional silver labels without over–crediting the detectors that produced them. The scope is narrow: a single English contact–center corpus, the recognizer count fixed at two and the controls varying one judge, so placement is not compared against adding engines, multilingual generalization is untested, end–to–end cost is not measured, and because the evidence analysis locates a quoted span by string match it cannot separate a judge that read a cue from one that guessed and happened to quote a matching span.

ЛІТЕРАТУРА

1. Fiscus J. G. A Post–Processing System to Yield Reduced Word Error Rates: Recognizer Output Voting Error Reduction (ROVER). *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. 1997. С. 347–354.
DOI: 10.1109/ASRU.1997.659110.

2. Galimzianov D., Vyshegorodtsev V. Text–Based Detection of On–Hold Scripts in Contact Center Calls. *Machine Learning and Applications: An International Journal*. 2024. Вип. 11. № 2. С. 1–9. DOI: 10.5121/mlaij.2024.11201.

3. Hillard D., Hoffmeister B., Ostendorf M. та ін. iROVER: Improving System Combination with Classification. *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*. 2007. С. 65–68. DOI: 10.3115/1614108.1614125.

4. Ingle D., Kumar A., Gogineni K. та ін. Real–Time Monitoring of Silences in Contact Center Conversations. *Interspeech 2022*. 2022. С. 1951–1952. URL: https://www.isca–archive.org/interspeech_2022/ingle22_interspeech.pdf.

5. Jia F., Majumdar S., Ginsburg B. MarbleNet: Deep 1D Time–Channel Separable Convolutional Neural Network for Voice Activity Detection. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2021. С. 6818–6822.
DOI: 10.1109/icassp39728.2021.9414470.

6. Jitsi JiWER: Evaluate your speech–to–text system with similarity measures.

7. Jung J., Brahman F., Choi Y. Trust or Escalate: LLM Judges with Provable Guarantees for Human Agreement. *International Conference on Learning Representations (ICLR)*. 2025. DOI: 10.48550/arXiv.2407.18370. URL: https://arxiv.org/abs/2407.18370.

8. Kamo N., Ochiai T., Delcroix M. та ін. MOVER: Combining Multiple Meeting Recognition Systems. *Interspeech 2025*. 2025. С. 3424–3428. DOI: 10.21437/interspeech.2025–1614.

9. Kano T., Ogawa A., Delcroix M. та ін. Attention–Based Multi–Hypothesis Fusion for Speech Summarization. *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. 2021. С. 487–494. DOI: 10.1109/asru51503.2021.9687977.

10. Kohli G. S. Nine Judges, Two Effective Votes: Correlated Errors Undermine LLM Evaluation Panels. *arXiv.org*. 2026. DOI: 10.48550/arXiv.2605.29800.
URL: https://arxiv.org/abs/2605.29800.

11. National Institute of Standards and Technology SCTK: NIST Scoring Toolkit.

12. Płaza M., Pawlik Łukasz, Deniziak S. Call Transcription Methodology for Contact Center Systems. *IEEE Access*. 2021. Вип. 9. С. 110975–110988.

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

DOI: 10.1109/access.2021.3102502.
13. Prakash J., Kumar B., Hacioglu K. та ін. Better Pseudo–labeling with Multi–ASR Fusion and Error Correction by SpeechLLM. *Interspeech 2025*. 2025. С. 579–583.
DOI: 10.21437/interspeech.2025–1707.
14. Radford A., Kim J. W., Xu T. та ін. Robust Speech Recognition via Large–Scale Weak Supervision. *Proceedings of the 40th International Conference on Machine Learning (ICML)*. 2023. Вип. 202. С. 28492–28518. DOI: 10.48550/arXiv.2212.04356.
URL: https://proceedings.mlr.press/v202/radford23a.html.
15. Rastorgueva E., Lavrukhin V., Ginsburg B. NeMo Forced Aligner and its application to word alignment for subtitle generation. *Interspeech 2023*. 2023. С. 5257–5258.
URL: https://www.isca–archive.org/interspeech_2023/rastorgueva23_interspeech.pdf.
16. Schwenk H., Gauvain J.–L. Combining Multiple Speech Recognizers using Voting and Language Model Information. *6th International Conference on Spoken Language Processing (ICSLP 2000)*. 2000. Вип. 2. С. 915–918. DOI: 10.21437/icslp.2000–419.
17. Schwenk H., Gauvain J.–L. Improved ROVER using Language Model Information. *ISCA ITRW ASR2000: Automatic Speech Recognition — Challenges for the New Millennium*. 2000. С. 47–52. URL: https://www.isca–archive.org/asr_2000/schwenk00_asr.pdf.
18. Sekoyan M., Koluguri N. R., Tadevosyan N. та ін. Canary–1B–v2 & Parakeet–TDT–0.6B–v3: Efficient and High–Performance Models for Multilingual ASR and AST. *arXiv.org*. 2025. DOI: 10.48550/arXiv.2509.14128. URL: https://arxiv.org/abs/2509.14128.
19. Shi X., Wang X., Guo Z. та ін. Qwen3–ASR Technical Report. *arXiv.org*. 2026.
DOI: 10.48550/arXiv.2601.21337. URL: https://arxiv.org/abs/2601.21337.
20. Stepachev P., Chen P., Haddow B. Context and System Fusion in Post–ASR Emotion Recognition with Large Language Models. *IEEE Spoken Language Technology Workshop (SLT), GenSEC Challenge*. 2024. DOI: 10.48550/arXiv.2410.03312.
URL: https://arxiv.org/abs/2410.03312.
21. Zheng L., Chiang W.–L., Sheng Y. та ін. Judging LLM–as–a–judge with MT–Bench and Chatbot Arena. *Neural Information Processing Systems*. 2023. С. 46595–46623.
DOI: 10.52202/075280–2020.

REFERENCES

1. Fiscus, J. G. (1997). A post–processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER). *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 347–354.
https://doi.org/10.1109/ASRU.1997.659110
2. Galimzianov, D., & Vyshegorodtsev, V. (2024). Text–based detection of on–hold scripts in contact center calls. *Machine Learning and Applications: An International Journal*, *11*(2), 1–9. https://doi.org/10.5121/mlaij.2024.11201
3. Hillard, D., Hoffmeister, B., Ostendorf, M., Schlüter, R., & Ney, H. (2007). iROVER: Improving system combination with classification. *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, 65–68. https://doi.org/10.3115/1614108.1614125

4. Ingle, D., Kumar, A., Gogineni, K., & Vepa, J. (2022). Real–time monitoring of silences in contact center conversations. *Interspeech 2022*, 1951–1952.
https://www.isca–archive.org/interspeech_2022/ingle22_interspeech.pdf
5. Jia, F., Majumdar, S., & Ginsburg, B. (2021). MarbleNet: Deep 1D time–channel separable convolutional neural network for voice activity detection. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6818–6822.
https://doi.org/10.1109/icassp39728.2021.9414470
6. Jitsi. (n.d.). *JiWER: Evaluate your speech–to–text system with similarity measures*. https://github.com/jitsi/jiwer.
7. Jung, J., Brahman, F., & Choi, Y. (2025). Trust or escalate: LLM judges with provable guarantees for human agreement. *International Conference on Learning Representations (ICLR)*. https://doi.org/10.48550/arXiv.2407.18370
8. Kamo, N., Ochiai, T., Delcroix, M., & Nakatani, T. (2025). MOVER: Combining multiple meeting recognition systems. *Interspeech 2025*, 3424–3428.
https://doi.org/10.21437/interspeech.2025–1614
9. Kano, T., Ogawa, A., Delcroix, M., & Watanabe, S. (2021). Attention–based multi–hypothesis fusion for speech summarization. *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 487–494. https://doi.org/10.1109/asru51503.2021.9687977
10. Kohli, G. S. (2026). Nine judges, two effective votes: Correlated errors undermine LLM evaluation panels. *arXiv.org*. https://doi.org/10.48550/arXiv.2605.29800
11. National Institute of Standards and Technology. (n.d.). *SCTK: NIST scoring toolkit*. https://github.com/usnistgov/SCTK.
12. Płaza, M., Pawlik, Łukasz, & Deniziak, S. (2021). Call transcription methodology for contact center systems. *IEEE Access*, *9*, 110975–110988.
https://doi.org/10.1109/access.2021.3102502
13. Prakash, J., Kumar, B., Hacioglu, K., Sharma, B., Gopalan, S., Chetlur, M., Venkatesan, S., & Stolcke, A. (2025). Better pseudo–labeling with multi–ASR fusion and error correction by SpeechLLM. *Interspeech 2025*, 579–583. https://doi.org/10.21437/interspeech.2025–1707
14. Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I. (2023). Robust speech recognition via large–scale weak supervision. *Proceedings of the 40th International Conference on Machine Learning (ICML), Proceedings of Machine Learning Research*, *202*, 28492–28518. https://doi.org/10.48550/arXiv.2212.04356
15. Rastorgueva, E., Lavrukhin, V., & Ginsburg, B. (2023). NeMo forced aligner and its application to word alignment for subtitle generation. *Interspeech 2023*, 5257–5258. https://www.isca–archive.org/interspeech_2023/rastorgueva23_interspeech.pdf
16. Schwenk, H., & Gauvain, J.–L. (2000a). Combining multiple speech recognizers using voting and language model information. *6th International Conference on Spoken Language Processing (ICSLP 2000)*, *2*, 915–918. https://doi.org/10.21437/icslp.2000–419
17. Schwenk, H., & Gauvain, J.–L. (2000b). Improved ROVER using language model information. *ISCA ITRW ASR2000: Automatic Speech Recognition — Challenges for the New Millennium*, 47–52. https://www.isca–archive.org/asr_2000/schwenk00_asr.pdf

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

18. Sekoyan, M., Koluguri, N. R., Tadevosyan, N., Żelasko, P., Bartley, T. M., Karpov, N., Balam, J., & Ginsburg, B. (2025). Canary–1B–v2 & Parakeet–TDT–0.6B–v3: Efficient and high–performance models for multilingual ASR and AST. *arXiv.org*. https://doi.org/10.48550/arXiv.2509.14128
19. Shi, X., Wang, X., Guo, Z., Wang, Y., Zhang, P., Zhang, X., Guo, Z., Hao, H., Xi, Y., Yang, B., Xu, J., Zhou, J., & Lin, J. (2026). Qwen3–ASR technical report. *arXiv.org*. https://doi.org/10.48550/arXiv.2601.21337
20. Stepachev, P., Chen, P., & Haddow, B. (2024). Context and system fusion in post–ASR emotion recognition with large language models. *IEEE Spoken Language Technology Workshop (SLT), GenSEC Challenge*. https://doi.org/10.48550/arXiv.2410.03312
21. Zheng, L., Chiang, W.–L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E. P., Zhang, H., Gonzalez, J. E., & Stoica, I. (2023). Judging LLM–as–a–judge with MT–Bench and chatbot arena. *Neural Information Processing Systems*, 46595–46623. https://doi.org/10.52202/075280–2020

Received 09.06.2026.
Accepted 11.08.2026.
Published 30.10.2026

Спільне узгодження кількох ASR–транскриптів великою мовною моделлю для виявлення ознак дотримання скрипту в контакт–центрах

Класичне комбінування систем розпізнавання мовлення – ROVER та його похідні, зокрема MOVER, – виконує голосування на рівні окремих слів і оптимізує частоту помилок на слові (WER). Натомість у контролі якості контакт–центрів транскрипт є лише проміжним представленням: за ним перевіряють, чи виконав оператор обов'язкові кроки скрипту – привітання, самоідентифікацію компанії, запит імені клієнта та ввічливе завершення розмови. Ознаки, що несуть ці перевірки, є короткими формульними фразами, які недосконале розпізнавання найчастіше спотворює або втрачає, а голосування на рівні токенів здатне вилучити слово, яке почув лише один розпізнавач. Отже, краще транскрибування не є тотожним кращому виявленню ознаки.

Мета дослідження – визначити, на якому рівні конвеєра доцільно розміщувати ансамбль розпізнавачів, якщо цільовою метрикою є повнота виявлення ознаки, а не частота помилок на слові.

За фіксованої пари декорельованих ASR–систем порівняно чотири точки злиття під керуванням однієї моделі–судді: одиночний транскрипт; злиття на рівні токенів (еталонний ROVER та власний голосувальник за часовим перекриттям); злиття на рівні рішень, тобто об'єднання вердиктів за кожним транскриптом окремо; та злиття на рівні гіпотез, коли суддя читає обидва транскрипти й узгоджує їх у межах одного звернення. Оцінювання виконано на корпусі з 3057 дзвінків за сімома параметрами дотримання скрипту, а всі підсумкові порівняння продубльовано на обмеженій підвибірці, що містить лише встановлені людиною мітки.

Злиття на рівні гіпотез є найкращою стратегією (macro–F1 0,913) і статистично значуще переважає одиночний транскрипт, злиття на рівні рішень та ROVER для обох суддів і на обох вибірках. Три контрольні експерименти показують, що при-

ріст потребує неідентичного транскрипта того самого дзвінка: заміна другого транскрипта транскриптом іншого дзвінка знижує macro–F1 до 0,717, а заміна його копією першого – до 0,796, тобто нижче за варіант з одиночним транскриптом, що виключає пояснення через поблажливість судді та через ефект самої підказки; текст, наявний лише у другому транскрипті, становить 6,9 % позитивних вердиктів, але забезпечує 35,7 % випадків, у яких злиття на рівні гіпотез правильно скасовує рішення за одиночним транскриптом. Отже, у межах дослідженої конфігурації з двома розпізнавачами місце розміщення ансамблю істотно впливає на якість виявлення ознак.

Ключові слова: автоматичне розпізнавання мовлення, комбінування систем розпізнавання, контроль якості контакт–центру, виявлення дотримання скрипту, великі мовні моделі, модель–суддя, злиття на рівні гіпотез, частота помилок на слові, протокол оцінювання.

Іванько Артем Миколайович – аспірант кафедри інформаційних технологій та комп'ютерної інженерії, Національний технічний університет «Дніпровська політехніка».
ORCID: https://orcid.org/0009–0002–0491–5374

Ivanko Artem – PhD student, department of information technology and computer engineering, Dnipro University of Technology.
ORCID: https://orcid.org/0009–0002–0491–5374

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)