A Bounded Coordination-Support Capability for Multi-Party Settings: Task-State Monitoring in Firefighter Incident Command

David Grünert and Barbara Morgenstern and Florian Peczinka and Dario Holenstein and Roland Brunner and Alexandre de Spindler
Zurich University of Applied Sciences, Winterthur, Switzerland
{grud,desa}@zhaw.ch,{morgebar,pecziflo,holendar}@students.zhaw.ch
Freiwillige Feuerwehr Stadt Zug, Zug, Switzerland
roland.brunner@stadtzug.ch

Abstract

Many collaboration settings require digital support systems for several humans who coordinate through ongoing communication. We study one such application in firefighter incident command: a dashboard that monitors, from radio transcripts, the state of predefined tasks derived from standard operating procedures (SOPs) and related procedures. Building such a dashboard raises a practical design question: how much transcript structure is actually needed for LLM-based task-state monitoring? More specifically, we examine whether additional transcript structure materially improves monitoring performance, even though it is difficult to obtain reliably from radio communication and increases complexity and latency. We evaluate this question on source-grounded synthetic firefighter scenarios under transcript conditions that vary speaker identity and utterance boundaries, with incremental inference as the deployment-facing condition and full-transcript inference as an offline reference. Across repeated runs, incremental monitoring remains strong across all transcript conditions. Differences between transcript structures are small, continuous transcripts remain competitive, and the main weaknesses are unit-related assignment timing and capturing the completion result, which remain broadly similar across conditions. These results suggest that for this bounded dashboard-support capability, neither speaker identities nor semantically precise utterance boundaries are a primary requirement in the controlled setting studied here.

1 Introduction

Large language models are now widely used in bilateral settings such as chat assistants and individual copilots (Maedche et al., 2019; Nah et al., 2023). Yet many real work settings require digital support for several humans who coordinate through ongoing communication and maintain a shared operational picture rather than seek isolated answers. Recent work on human-AI collaboration therefore argues for moving beyond assistant-centric interaction toward AI systems that support team processes and shared coordination (Seeber et al., 2020; Anthony et al., 2023; Bankins et al., 2024). In mission-critical domains, one concrete support need is to maintain an explicit view of what has been assigned, what remains pending, and what has been completed.

Firefighter incident command provides a concrete instance of this broader problem. Commanders coordinate multiple units through short, interleaved radio transmissions while tracking operational tasks. We study this setting through a command-support dashboard whose task list is already derived from standard operating procedures (SOPs), related procedures, and incident context. The model does not discover new tasks or make operational decisions. Instead, it supports blind-spot checking by proposing conservative state updates for predefined tasks under human oversight.

Once such a dashboard is the target system, a more specific technical question becomes central: how much transcript structure is actually needed for LLM-based task monitoring from multi-party radio traffic? Rich transcript structure may be costly or difficult to obtain robustly in this setting. Speaker identity may require diarization or prior speaker models, both of which remain difficult under noisy multi-speaker conditions and heterogeneous channels (Mehrish et al., 2023). Utterance boundaries may be supplied by voice activity detection, but pause-based segmentation can split incomplete thoughts, while more refined speaker-change or semantic segmentation adds models, engineering effort, and delay (Mehrish et al., 2023). For a command-support system, these are not cosmetic transcript properties but potential deployment bottlenecks because real-time speech systems must trade off model complexity against latency and computational load (Michelsanti et al., 2021).

This motivates the research question of the paper: *how much transcript structure is actually needed to track predefined task state reliably from ongoing multi-party communication?* Specifically, we ask whether task-state monitoring depends on speaker identities and explicit utterance boundaries, or whether a simpler transcript representation suffices for this monitoring task.

We address this question through a controlled evaluation of closed-world task-state monitoring across transcript conditions that vary speaker identity and utterance boundaries while keeping the underlying scenario content fixed. The paper contributes in three ways. First, we formulate a bounded coordination-support problem centered on monitoring the state of predefined SOP- and procedure-related tasks in firefighter incident command. Second, we introduce a source-grounded synthetic dataset construction and validation pipeline that enables controlled comparison of transcript-structure effects while holding operational content constant. Third, we report results for incremental monitoring and a secondary offline full-transcript reference in a reproducible evaluation pipeline with auditable artifacts and local tests for scenario handling, prefix-gold derivation, parsing, and metric computation[1].

The remainder of the paper is structured as follows. Section 2 situates the work in the literature on multi-human AI support and facilitation. Sects. 3–5 then cover the application framing, dataset, and experimental design, before Sect. 6 reports the results and Sect. 7 concludes.

2 Background

This section situates the paper at the intersection of multi-human AI support and team information-processing support. It motivates why a dashboard-centered command-support system can be understood as one instance of a broader class of digital support systems for shared coordination, and why task-state monitoring is a meaningful bounded capability within that class.

A large share of current generative AI applications remains focused on the individual user: question answering, drafting, summarisation, or recommendation for a single human operator (Maedche et al., 2019; Nah et al., 2023). By contrast, many work settings require models to interpret contributions distributed across several people, maintain state over time, and support shared rather than individual understanding. Reviews of human-AI teaming show that much of the literature still focuses on bilateral assistants, dyads, or otherwise narrowly scoped team configurations (Lyons et al., 2021; O'Neill et al., 2022; Anthony et al., 2023; Bankins et al., 2024). This leaves open how AI systems should support shared state and coordination across ongoing multi-party interaction through a shared artefact such as a dashboard.

Research on Group Support Systems has long shown that collaboration quality depends not only on who contributes, but also on how information is processed and coordinated within the group (Nunamaker et al., 1991; Dennis and Valacich, 1993; Dennis et al., 2001; Briggs et al., 2003). Facilitation has been central in this tradition because groups systematically suffer from process losses such as production blocking, dominance effects, incomplete information pooling, and premature convergence (Stasser and Titus, 1985; Dennis and Valacich, 1993; Briggs et al., 2003). Recent work on AI in teams suggests that digital artefacts may take over selected support capabilities, especially those involving consistency, timing, monitoring, or large-scale information processing, while humans retain judgement, legitimacy, and contextual authority (Dellermann et al., 2019; Seeber et al., 2020; Dennis et al., 2023).

Our paper focuses on one such capability: maintaining an explicit representation of task state from ongoing multi-party communication. In facilitation terms, this is primarily an information-processing support function. The command-support system does not replace operational decision making. Rather, it externalises on the dashboard what the team has already established: which tasks have been handed out, which unit appears responsible, which tasks have been completed, and what completion result has been reported.

Technically, this formulation is related to dialogue state tracking and information extraction. Dialogue state tracking maintains structured state representations from dialogue history in task-oriented systems (Balaraman et al., 2021). Information and event extraction similarly recover structured records, events, or arguments from text, often against predefined schemas (Chambers and Jurafsky, 2011; Xiang and Wang, 2019). Our setting differs in that the state variables are predefined SOP- and procedure-related operational tasks, the evidence is distributed across multi-party radio com-

[1] https://github.com/zhaw-iwi/swisstext26_pub

munication, and the output is used for incremental dashboard support rather than autonomous dialogue management or open-ended event discovery.

We study this capability in a deliberately bounded operational setting rather than in the most open-ended form of multi-human coordination. If a digital support system cannot reliably maintain assignment and completion state for predefined tasks, then more advanced functions such as summarisation, blind-spot detection, or process coaching are unlikely to be trustworthy. Conversely, if task-state monitoring proves robust under reduced transcript structure, this suggests that at least some multi-human information-processing functions may not require expensive preprocessing pipelines.

Taken together, these strands motivate AI support for shared coordination, but they do not yet answer the more specific deployment question studied in this paper: how much transcript structure is required for closed-world task-state monitoring from ongoing multi-party communication? Firefighter incident command is a useful case because the dashboard-centered support function is operationally meaningful, the communication is genuinely multi-party, and transcript enrichment can be costly. For a bounded function such as digital task-state tracking, richer transcript preprocessing may help, but it is not obvious that speaker labels and clean utterance boundaries are necessary.

3 Application Context and Requirements

Firefighter incident command provides a concrete example of the broader class of multi-human digital support settings outlined in the previous section. In this setting, the command-support system is not meant to converse with a single user in isolation. Instead, it supports shared coordination by helping incident command maintain an explicit, inspectable view of operational task state as evidence arrives through multi-party radio traffic. We use this case not as the whole claim of the paper, but as a demanding setting from which to derive requirements for one bounded coordination-support capability.

3.1 Application Setting

The user-facing surface is a dashboard that presents the task list together with current task status and recent communication context, as shown in Figure 1. Its role is to reduce memory burden and preserve an explicit overview of open and completed tasks. The dashboard thus serves as the interface of a bounded information-processing support system for the command team. It helps keep tasks from getting lost in noisy, interleaved, time-critical radio communication.

Figure 2 situates this dashboard within our implemented command-support system. Predefined tasks originate from SOPs, related procedures, and incident context, for example from baseline checklists associated with the incident type, from object-specific information, and from a building register. This information is available and selected at dispatch time. Radio communication then provides the dynamic evidence for whether those tasks have been assigned to firefighter units, acknowledged, and completed. This paper therefore treats task-state monitoring as a closed-world problem: the task set is supplied in advance, and transcript evidence is used to update task state.

This application framing is also consistent with preliminary feedback from a prototype demonstration of our command-support system with active firefighters. Beyond transcript capture itself, they emphasized the value of temporal tracking features, such as seeing how long a task has remained open or how long a crew has been silent. We treat this feedback as informal practitioner grounding rather than as a formal user-study result.

3.2 Monitoring Problem

Emergency radio communication differs substantially from the dialogue structure often assumed in conversational NLP. Messages are short, domain-specific, and often elliptical. Operational meaning is distributed across acknowledgements, updates, and follow-up reports rather than contained in a single self-sufficient utterance. For the command-support system, useful support therefore depends on reconstructing task state from distributed conversational evidence rather than isolated turns.

This creates the concrete design pressure behind our research question. If the dashboard requires speaker identities, the implementation may depend on diarization or prior speaker models despite radio-channel noise and changing crews (Mehrish et al., 2023). If it requires semantically well-formed utterance boundaries, pause-based segmentation may be insufficient and additional speaker-change or semantic segmentation steps may be needed before task monitoring can run (Mehrish et al., 2023). These components increase engineering effort and add latency, which is a relevant concern in real-time speech systems that must trade off model complex-

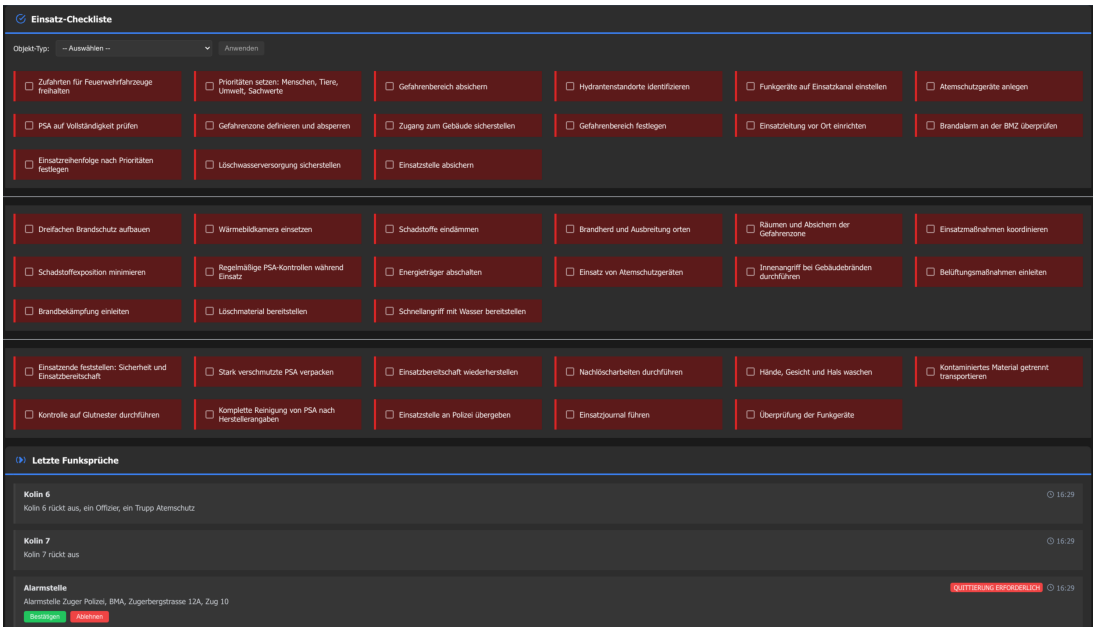

Figure 1: Dashboard interface of our command-support system.

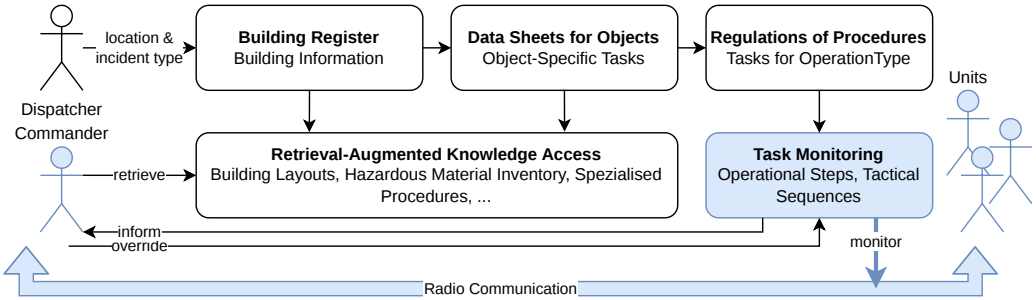

Figure 2: Operational flow of our command-support system, with dashboard components shown in blue.

ity against computational load (Michelsanti et al., 2021). In our setting, such components may fail precisely in the noisy conditions where operational support is most valuable. We therefore investigate how much structure the transcript representation actually needs for this monitoring task. More broadly, the firefighter case can thus be understood as one example of a digital support system that performs a bounded, process-relevant information-processing function inside a human team.

3.3 System Requirements

The command-support system must support commanders in checking that SOP-relevant tasks have been addressed. In our setting, these predefined SOP- and procedure-related tasks are already available from SOPs, related procedures, and scenario context before transcript analysis. The scope is therefore closed-world monitoring of predefined tasks rather than open-ended task discovery.

The command-support system must operate under human oversight. Proposed task-state updates must remain transparent to the command team and be subject to human confirmation or override.

The command-support system must support near-real-time dashboard updates during ongoing incidents. Task-state information must be revised as new communication evidence becomes available so that the dashboard remains a current overview.

3.4 Task Formulation

These application requirements lead to a closed-world task formulation. For each predefined task τ and each discrete prefix index k, the model outputs a structured task-state prediction

$$y(\tau, k) = (a, u, c, o).$$

Here, k denotes the current transcript prefix, that is, the communication evidence available up to that point. The tuple components correspond

to `assigned`, `assigned_unit`, `completed`, and `completion_outcome`, respectively. `assigned` (boolean) indicates whether the transcript provides evidence that responsibility for the task has been assigned to a unit and acknowledged. `assigned_unit` (nullable string) indicates which unit is identified as responsible for the task. `completed` (boolean) indicates whether the transcript provides evidence that the task has been completed. `completion_outcome` (nullable string) captures the reported completion result or completion evidence in short textual form.

4 Dataset

We evaluate the monitoring formulation on a controlled synthetic dataset. The synthetic design is deliberate: the research question requires the same underlying operational scenario to be rendered under different transcript-structure conditions while keeping task content fixed. A source-grounded synthetic dataset makes this possible, allowing differences between transcript conditions to be attributed to representation rather than scenario content.

4.1 Source-Grounded Construction

The dataset was constructed with a prompt-based generation and validation pipeline grounded in three source types: Swiss radio-procedure material[2], a firefighter communication transcript used for surface realism, and canonical procedural regulations[3]. These materials were distilled into source notes used for scenario generation and validation.

Each scenario was generated as a fixed-schema JSON object containing an ordered message sequence, a closed-world list of predefined tasks, and gold task states. In addition to the final task state, each gold entry stores the message id at which assignment first becomes valid and, when applicable, the message id at which completion first becomes valid. Prefix-level gold states are then derived deterministically from these transition annotations. Cases in which transition timing would remain ambiguous are resolved through explicit metadata rather than heuristic inference.

Validation was performed in two independent rubric-guided passes: a structural pass for radio style and protocol coherence, and a content pass for operational plausibility, role-task alignment, task sequencing, and traceability of gold states to explicit message evidence. We implemented both dataset generation and validation with `gpt-5.4` as a bounded LLM judgment process under source-grounded constraints (Zheng et al., 2023; Röttger et al., 2024). This source-grounded generation-and-judge setup was used to ground the synthetic data in the intended application setting. The generation–validation–revision loop ran for 12 rounds until the dataset passed both validation streams.

As an additional validation layer before evaluation, all scenario files pass deterministic local checks for schema conformance, sequential message ids, exact alignment between predefined tasks and gold task states, and consistency constraints on assignment/completion labels.

As an additional plausibility check, the extracted structure and resulting scenarios were reviewed within the team. They were judged to align with observed communication structures and contents from the intended application setting. While not a formal annotation study, this review provides additional support for the dataset's plausibility as a set of realistically structured synthetic scenarios.

4.2 Dataset Contents

The final dataset contains five German-language firefighter scenarios with 102 ordered radio messages and 15 predefined tasks in total. Every scenario contains exactly three monitored tasks, yielding 15 task-state traces overall. All 15 tasks are explicitly assigned in the transcript evidence, 12 are completed within the scenario, and 3 remain assigned but incomplete at scenario end.

The five scenarios cover a kitchen fire in an apartment building, an underground-garage fire, a school basement fire, a workshop fire involving gas cylinders, and an attic fire in a row house. Across these scenarios, the monitored task families include water supply, search, ventilation, fire suppression, access control, gas-cylinder cooling, CO measurement, and final control. The assigned units include command-relevant firefighter roles such as `Angriffstrupp`, `Wassertrupp`, `Messgruppe`, `Kontrolltrupp`, and `Verkehrstrupp`.

4.3 Limitations

The limitations of the study follow directly from this dataset design and should be read upfront. The dataset is synthetic, small, and intentionally controlled. The scenarios are cleaner and more linear than real firefighter radio traffic, with limited pro-

[2] Bundesamt für Bevölkerungsschutz, Funkmaterial Sprechregeln, February 2004

[3] Feuerwehr Koordination Schweiz FKS, Reglement Einsatzführung, November 2022

tocol noise, dispatch and handover tails, and some bundled tasks that compress multi-step work into a single monitored item. The evaluation also begins from text transcripts rather than raw audio, so it does not include upstream speech-recognition or radio-channel degradation effects. Finally, the task formulation is deliberately narrow: it tests closed-world monitoring of predefined tasks, not open-ended task discovery or broad operational performance. We therefore use the dataset as controlled evidence for transcript-structure sensitivity in dashboard-oriented task monitoring, not as a realism corpus or a direct deployment claim.

5 Experimental Setup

We evaluate the dataset under a 3×2 design that varies transcript structure and processing mode while keeping the underlying scenario content fixed.

5.1 Conditions

We compare three transcript structures: **structured_dialogue**, which includes speaker identities and utterance boundaries, **no_speaker**, which retains utterance boundaries but omits speakers, and **continuous_transcript**, which contains neither speakers nor boundaries. This last condition approximates the minimum structure available when speakers cannot be identified and reliable utterance segmentation is not available.

We evaluate two processing modes: **incremental**, which predicts on each growing message prefix from the first message to the full transcript, and **full_transcript**, which predicts once on the complete transcript.

Incremental processing is the primary application condition. The `full_transcript` condition serves as a secondary offline reference once all evidence is available. For incremental `continuous_transcript` mode, each prefix is rendered as one continuous text string without speaker labels or message boundaries.

5.2 Execution and Reproducibility

All reported values are batch-level means across 12 repeated runs. The notebook runs the evaluation in live mode with `gpt-5.2` at temperature 0.0. Repeated runs are reported because live API inference can still show small residual variation despite deterministic settings. The evaluation prompt instructed the model to behave conservatively: when the transcript did not provide sufficient evidence for assignment or completion, the corresponding fields were to remain `false`/`null`. This reflects the operational preference to avoid overstating task progress under incomplete evidence. The evaluation pipeline persists each request and response artifact using deterministic identifiers over run, scenario, structure condition, processing mode, and prefix index, supporting resume-safe execution and auditability. The repository contains the dataset-generation prompts, validation prompts, source notes, evaluation prompt payloads, and persisted request/response artifacts. The tables report 95% confidence intervals from the batch summaries exported by the notebook and evaluation code.

5.3 Metrics

We report four state-monitoring metrics in both processing modes. In the incremental setting, each metric is computed at every evaluated prefix and then averaged over prefixes. In the `full_transcript` setting, the same metric is computed once on the final scenario transcript.

Assignment accuracy. This metric measures, for each predefined task, whether the predicted `assigned` value matches gold.

Unit assignment accuracy. This metric measures whether the predicted `assigned_unit` matches gold. It is evaluated on tasks for which either gold or prediction marks the task as assigned, so that unit assignment is scored only where a responsible unit is relevant. In the incremental setting, scores are first computed per prefix and then averaged. Prefixes without any unit-assignment support contribute 0.0 by construction.

Completion accuracy. This metric measures, for each predefined task, whether the predicted `completed` value matches gold.

State accuracy. This metric collapses each task state to one of `NOT_ASSIGNED`, `ASSIGNED`, or `COMPLETED`, and then measures whether the predicted three-way state matches gold.

For incremental evaluation, we additionally report assignment and completion detection latency together with assignment and completion miss rate. The gold transition points come from explicit scenario annotations for first assignment and, when applicable, first completion evidence. Assignment is anchored to the first operative command that hands responsibility for the predefined task to the assigned unit. Later refinements to the same unit do not reset the assignment point.

Assignment detection latency. This metric measures the number of prefix steps between the gold assignment point and the first prefix at which the model predicts the task as assigned with the correct unit. It is reported only for successfully detected assignment events.

Completion detection latency. This metric measures the number of prefix steps between the gold completion point and the first prefix at which the model predicts the task as completed while preserving assigned state and the correct unit. It is reported only for successfully detected completion events.

Assignment detection miss rate. This metric is the proportion of gold assignment events that are never correctly detected before the scenario ends.

Completion detection miss rate. This metric is the proportion of gold completion events that are never correctly detected before the scenario ends.

Terminal gaps. As a reference comparison, we report terminal assignment, unit-assignment, completion, and state gaps. These are the absolute differences between the last incremental-prefix metrics and the corresponding `full_transcript` metrics for the same run, scenario, and transcript structure.

Completion outcome. For `completion_outcome`, exact match proved too strict because the model often paraphrases the completion evidence rather than reproducing the gold message string verbatim. We therefore report a retrospective similarity-based reanalysis on gold-completed tasks using ROUGE-L F1 as a secondary reference metric.

6 Results

The main empirical result is straightforward: in this controlled setup, simplifying transcript structure has little effect on aggregate task-state monitoring performance. Across all conditions, assignment, completion, and current-state accuracy remain high. The lowest aggregate metric is unit assignment, but this should be interpreted as a stricter prefix-level score rather than as frequent confusion between responsible units.

6.1 Incremental Accuracy

Incremental inference is the more deployment-relevant setting because the command-support system must update task state while communication unfolds. Table 1 shows that performance remains strong and tightly clustered across transcript structures. The unit-assignment column is consistently lower than the other state-tracking metrics, but inspection of the persisted artifacts suggests that this does not primarily reflect wrong responsible-unit names. Most unit mismatches arise when an incremental prediction has not yet marked a gold-assigned task as assigned, especially for conditional or preparatory tasking, with a smaller number of premature assignments. Thus, the harder part is aligning assignment timing and unit grounding as evidence unfolds, rather than choosing among units once a task is accepted as assigned.

Structure	Assign.	Unit	Complete	Current
Structured dialogue	0.981 ± 0.006	0.864 ± 0.016	0.977 ± 0.008	0.958 ± 0.011
No speaker	0.980 ± 0.007	0.865 ± 0.016	0.978 ± 0.007	0.958 ± 0.009
Continuous transcript	0.981 ± 0.007	0.868 ± 0.014	0.978 ± 0.007	0.959 ± 0.010

Table 1: Incremental monitoring accuracy by transcript structure.

In the offline `full_transcript` reference, assignment and unit-assignment accuracy are 1.000 in all three transcript conditions. Completion and final-state accuracy are 0.933 in all conditions.

6.2 Incremental Latency and Misses

Latency is the second key deployment-oriented result because the dashboard benefits not only from eventually correct updates but from updates that arrive close to the true transition point. We therefore report two complementary quantities in Table 2: conditional latency in prefix steps for events that were correctly detected, and miss rate for gold transitions never correctly detected before the scenario ended. Assignment latency therefore requires the model to mark a task as assigned with the correct unit, while completion latency requires the model to mark the task as completed while preserving coherent assigned state and the correct unit.

The latency results are straightforward. Across all three transcript conditions, correct assignment updates remain well below one prefix step on average and no assignment misses occur. Completion detection is even stronger, with effectively immediate updates and zero completion misses throughout. In practical terms, once the transcript supports a state change, the model usually updates at once or with only a very small delay.

6.3 Terminal Convergence

Table 3 reports absolute terminal gaps between the last incremental prefix metrics and the corresponding full-transcript reference for the same run,

Structure	Assign. lat.	Assign. miss	Comp. lat.	Comp. miss
Structured dialogue	0.256 ± 0.127	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
No speaker	0.256 ± 0.126	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
Continuous transcript	0.278 ± 0.133	0.000 ± 0.000	0.006 ± 0.011	0.000 ± 0.000

Table 2: Incremental correct-detection latency and miss rate by transcript structure.

Structure	Assign. gap	Unit gap	Comp. gap	State gap
Structured dialogue	0.000 ± 0.000	0.000 ± 0.000	0.006 ± 0.011	0.006 ± 0.011
No speaker	0.000 ± 0.000	0.000 ± 0.000	0.006 ± 0.011	0.006 ± 0.011
Continuous transcript	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000

Table 3: Terminal convergence of the last incremental prefix to the offline full-transcript reference.

scenario, and transcript structure.

The convergence pattern is very tight. By the end of the scenario, the incremental setup almost exactly recovers the performance profile of the offline `full_transcript` reference. This makes the reference useful as a sanity check for interpreting the incremental results.

6.4 Completion-Outcome Reanalysis

The original exact-match `completion_outcome` metric remains 0.000 in all three transcript conditions and is therefore not useful on its own. Inspection of the outputs, however, suggests that this mostly reflects paraphrastic variation: the model often paraphrases the completion evidence instead of reproducing the gold completion string verbatim. We therefore ran a retrospective similarity-based comparison on the already collected outputs for gold-completed tasks only.

Table 4 reports ROUGE-L F1 for this reanalysis. The resulting scores are clearly above zero but remain broadly similar across transcript conditions. This makes the `completion_outcome` field somewhat more informative than exact match suggested, while leaving the main conclusion unchanged: transcript structure still has only small effects in this bounded monitoring setup. At the same time, the scores remain well below the stronger assignment and completion-state results, so we interpret them conservatively as evidence of partial string-level overlap rather than robust semantic understanding.

7 Conclusion

We studied task-state monitoring in firefighter incident command as a bounded coordination-support capability for an incident-command dashboard.

The central system-design question was whether this capability depends on speaker identities and explicit utterance boundaries, both of which can be

Structure	ROUGE-L F1
Structured dialogue	0.625 ± 0.007
No speaker	0.657 ± 0.015
Continuous transcript	0.638 ± 0.017

Table 4: Retrospective `completion_outcome` reanalysis on gold-completed tasks only.

expensive or difficult to obtain robustly from radio communication. For the controlled setting studied here, the answer is largely no: reduced transcript structure does not materially change the overall task-monitoring performance profile. The main remaining weaknesses are unit-related assignment timing in incremental prefixes and capturing the reported completion result rather than detecting assignment or completion itself, and both remain broadly similar across transcript conditions.

More broadly, this suggests that some coordination-support functions may be feasible without heavy transcript preprocessing. Methodologically, the paper also shows how a source-grounded synthetic dataset construction and validation pipeline can support controlled comparison of transcript-structure conditions while holding operational content fixed. At the same time, the present evidence remains deliberately narrow: it comes from a controlled synthetic dataset, from text rather than raw audio, and from a closed-world monitoring task. Future work should therefore test whether the same pattern holds under noisier audio pipelines, less linear communication, and stronger ambiguity, while improving unit-assignment grounding, evidence presentation, and human-in-the-loop integration into command workflows. A further direction is to move from closed-world monitoring toward mixed closed- and open-world coordination support, where systems track predefined tasks while also identifying additional operational tasks that emerge from the incident record. Another direction is to connect evolving coordination state to relevant supporting knowledge resources, so that a shared dashboard can surface applicable procedures, plans, or reference documents without turning the system into an autonomous decision maker.

References

Chris Anthony, Beth A. Bechky, and Anne-Laure Fayard. 2023. "collaborating" with AI: Taking a system view to explore the future of work. *Organization Science*, 34(5):1672–1694.

Vevake Balaraman, Seyedmostafa Sheikhalishahi, and Bernardo Magnini. 2021. Recent neural methods on dialogue state tracking for task-oriented dialogue systems: A survey. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 239–251.

Sarah Bankins, Alannah C. Ocampo, Maurizio Marrone, Simon L. D. Restubog, and Sang Eun Woo. 2024. A multilevel review of artificial intelligence in organizations: Implications for organizational behavior research and practice. *Journal of Organizational Behavior*, 45(2):159–182.

Robert O. Briggs, Gert-Jan de Vreede, and Jay F. Nunamaker. 2003. Collaboration engineering with thinklets to pursue sustained success with group support systems. *Journal of Management Information Systems*, 19(4):31–64.

Nathanael Chambers and Dan Jurafsky. 2011. Template-based information extraction without the templates. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 976–986.

Dominik Dellermann, Adrian Calma, Nicola Lipusch, Tim Weber, Sascha Weigel, and Patrick Ebel. 2019. The future of human-AI collaboration: A taxonomy of design knowledge for hybrid intelligence systems. In *Proceedings of the Hawaii International Conference on System Sciences (HICSS)*.

Alan R. Dennis, Anjali Lakhiwal, and Akanksha Sachdeva. 2023. AI agents as team members: Effects on satisfaction, conflict, trustworthiness, and willingness to work with. *Journal of Management Information Systems*, 40(2):49–54.

Alan R. Dennis and Joseph S. Valacich. 1993. Computer brainstorms: More heads are better than one. *Journal of Applied Psychology*, 78(4):531–537.

Alan R. Dennis, Barbara H. Wixom, and Robert J. Vandenberg. 2001. Understanding fit and appropriation effects in group support systems via meta-analysis. *MIS Quarterly*, 25(2):167–197.

Joseph B. Lyons, Katia Sycara, Michael Lewis, and Alexandra Capiola. 2021. Human-autonomy teaming: Definitions, debates, and directions. *Frontiers in Psychology*, 12:589585.

Alexander Maedche, Christine Legner, Alexander Benlian, Benedikt Berger, Henner Gimpel, Thomas Hess, Oliver Hinz, Stefan Morana, and Matthias Söllner. 2019. Ai-based digital assistants. *Business & Information Systems Engineering*, 61(4):535–544.

Ambuj Mehrish, Navonil Majumder, Rishabh Bhardwaj, and Soujanya Poria. 2023. A review of deep learning techniques for speech processing. *Information Fusion*, 99:101869.

Daniel Michelsanti, Zheng-Hua Tan, Shi-Xiong Zhang, Yong Xu, Meng Yu, Dong Yu, and Jesper Jensen. 2021. An overview of deep-learning-based audio-visual speech enhancement and separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:1368–1396.

Fiona F.-H. Nah, Ruilin Zheng, Jian Cai, Keng Siau, and Langtao Chen. 2023. Generative AI and ChatGPT: Applications, challenges, and AI-human collaboration. *Journal of Information Technology Case and Application Research*, 25(3):277–304.

Jay F. Nunamaker, Alan R. Dennis, Joseph S. Valacich, Douglas R. Vogel, and Joey F. George. 1991. Electronic meeting systems to support group work: Theory and practice at arizona. *Communications of the ACM*, 34(7):40–61.

Thomas O'Neill, Nathan McNeese, Anthony Barron, and Brandon Schelble. 2022. Human-autonomy teaming: A review and analysis of the empirical literature. *Human Factors*, 64(5):904–938.

Paul Röttger, Fabio Pernisi, Bertie Vidgen, and Dirk Hovy. 2024. SafetyPrompts: a systematic review of open datasets for evaluating and improving large language model safety. *arXiv preprint arXiv:2404.05399*.

Isabella Seeber, Eva Bittner, Robert O. Briggs, Triparna de Vreede, Gert-Jan de Vreede, Aaron Elkins, Ronald Maier, Alexander B. Merz, Sarah Oeste-Reiß, Niels Randrup, Gerhard Schwabe, and Matthias Söllner. 2020. Machines as teammates: A research agenda on ai in team collaboration. *Information & Management*, 57(2):103174.

Garold Stasser and William Titus. 1985. Pooling of unshared information in group decision making: Biased information sampling during discussion. *Journal of Personality and Social Psychology*, 48(6):1467–1478.

Wei Xiang and Bang Wang. 2019. A survey of event extraction from text. *IEEE Access*, 7:173111–173137.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Advances in Neural Information Processing Systems 36: Datasets and Benchmarks Track*.