Additional File 11

Within-Framework Baseline Study

For the within-framework baseline, we collapsed the staged pipeline into single Gemini 2.5 Pro calls on the same 52-note benchmark. One baseline combined Stage 1 event extraction and Stage 2 temporal tagging in a single prompt; the other combined Stages 1–3 in a single prompt to generate the final summary directly. To keep the comparison controlled, the merged prompts were constructed programmatically from the production stage prompts, preserving the original stage instructions except for minimal seam edits needed to make cross-stage references coherent within one call. The user input contained only the anchor dates and discharge note, matching the information available to the staged pipeline. Outputs were formatted to match the corresponding staged outputs, allowing the same postprocessing and Gemini 3.1 Pro judges to be applied across baseline and pipeline conditions.

Table S11.1: Stage 2 Baseline (S1+S2 in one call) vs. our pipeline; scored by Gemini 3.1

Dimension	Baseline: one call	Our pipeline
A Completeness	4.36	5.00
B Semantic faithfulness	5.00	5.00
C Temporal tag accuracy	4.56	4.83
Mean	**4.64**	**4.94**

The one-call baseline performs the same two operations as Stages 1–2, but asks the model to extract events and assign temporal labels in a single generation step. This merged setup lowers the mean LLM-judge score from 4.94 to 4.64, with the largest drops in completeness and temporal tag accuracy. The baseline produced 2,019 fewer events than the staged pipeline, suggesting that combining extraction with temporal reasoning reduces coverage. Its dominant error type was temporal misclassification: 53% of all baseline errors involved temporal labels, often because events without an explicit date in the event text were assigned [EXACT]. Together, these results suggest that the stage boundary is useful not merely for organization, but because it reduces the burden on the model: Stage 1 can prioritize high-recall event extraction, while Stage 2 can make a separate, conservative temporal decision for each extracted event.

Table S11.2: Stage 3 Baseline (S1+S2+S3 in one call) vs. our pipeline; scored by Gemini 3.1

Dimension	Baseline: one call	Our pipeline
A Faithfulness to source	5.00	5.00
B Temporal accuracy	3.81	4.75
Mean	**4.4**	**4.88**

Table S11.2 compares a one-call Stage 3 baseline with the staged CliniCIRCA pipeline. Both systems achieve 5.00 on faithfulness to source, indicating that the baseline generally preserves clinically grounded content. The gap appears in temporal accuracy: the one-call baseline scores 3.81, compared with 4.75 for the staged pipeline, lowering the overall mean from 4.88 to 4.40. This pattern suggests that the main failure is not unsupported clinical content, but temporal rendering. Without an explicit intermediate timeline, the one-call baseline more often places otherwise faithful events at the wrong time or collapses uncertain timing into overly specific chronology, consistent with the Stage 1–2 baseline's tendency to over-commit to temporal precision.

Overall, both baselines where all tasks are combined into a single LLM call (instead of three stages in our pipeline) have suffered performance losses. This finding is consistent with recent work showing that decomposition can improve LLM performance on complex NLP and reasoning tasks by separating a difficult task into simpler subtasks, reducing single-step attention burden, and making intermediate decisions more explicit (1,2).

References

1. Wu Z, Bai RH, Zhang A, Gu J, Vydiswaran VV, Jaitly N, et al. Divide-or-Conquer? Which Part Should You Distill Your LLM? In: Al-Onaizan Y, Bansal M, Chen YN, editors. Findings of the Association for Computational Linguistics: EMNLP 2024 [Internet]. Miami, Florida, USA: Association for Computational Linguistics; 2024. p. 2572–85. Available from: https://aclanthology.org/2024.findings-emnlp.145/ doi:10.18653/v1/2024.findings-emnlp.145

2. Xie Y, Jin X, Xie T, Lin M, Chen L, Yu C, et al. Decomposition for Enhancing Attention: Improving LLM-based Text-to-SQL through Workflow Paradigm. In: Ku LW, Martins A, Srikumar V, editors. Findings of the Association for Computational Linguistics: ACL 2024 [Internet]. Bangkok, Thailand: Association for Computational Linguistics; 2024. p. 10796–816. Available from: https://aclanthology.org/2024.findings-acl.641/ doi:10.18653/v1/2024.findings-acl.641