Team UAB at Chemotherapy Timelines 2025: Integrating encoders and large language models for chemotherapy timeline generation

Vijay Jain[1], Chris Coffee[1], Kaiwen He[1], Remy Cron[1],
Micah Cochrane[1], Luis Mansilla-Gonzalez[1], Danish Murad[1], Akhil Nadimpalli[2],
John D. Osborne[1]
[1]University of Alabama at Birmingham, Birmingham, Alabama, USA
[2]Independent Researcher, Birmingham, AL, USA
{jainv, chrico, kaiwenhe, rrcron, mdcochra, lmansill, dmurad, ozborn}@uab.edu,
{akhilnadimpalli@gmail.com}

Abstract

Reconstructing the timeline of Systemic Anticancer Therapy (SACT) or "chemotherapy" from heterogeneous Electronic Health Record (EHR) notes is a challenging task. Rapid developments in Large Language Models (LLMs), including a range of architectural improvements and post-training refinements since the 2024 Chemotherapy Timelines Task could make this task more tractable. We evaluated the performance of 4 recently released LLMs (GPT-4.1-mini, Phi4 and 2 Qwen3 models) on this task. Our results indicate that even with a variety of prompt optimization and synthetic data training, more work is still needed to see a useful application of this work.

1 Introduction

Accurately extracting Systemic Anticancer Therapy (SACT) or "chemotherapy" treatment timelines from clinical narratives is essential for conducting retrospective outcome studies, enabling researchers to correlate the sequence and timing of administered regimens with long-term patient outcomes and responses. However, clinical documentation is often scattered between heterogeneous types of note. This makes both manual abstraction of timelines exceptionally laborious and error-prone and increases the complexity of development for systems abstracting these timelines. Continued advances in large language models (LLMs) with improved reasoning capabilities(OpenAI, 2025), larger context window sizes and higher overall performance may enable SACT extraction above baselines seen in the 2024 task(Yao et al., 2024). This includes newer models such as GPT-4.1(OpenAI, 2025), a derivative of the larger proprietary model GPT-4(Achiam et al., 2023) as well as smaller local LLMs such as Phi4(Abdin et al., 2024)and Qwen3(Yang et al., 2025) with reasoning ability. Furthermore, smaller masked language models of which BERT(Devlin et al., 2019) is the canonical example have seen both architectural improvements (Warner et al., 2024) and biomedical finetuning since 2024(Lee et al., 2025). However, efforts have been limited due to a single, domain-specific data set (Yao et al., 2024) that includes only three types of cancer: breast, ovarian, and melanoma. In this work, we assess the ability of recent LLMs to address this problem on the Chemo-Timelines 2025 Shared Task(Yao et al., 2025) for both Task 1 (where additional gold annotations are provided) and Task 2 where input is restricted to clinical notes.

1.1 Related Work

For both Task 1 and Task 2, SACT timelines must be generated and consolidated. A variety of approaches have been used for this, including the use of local LLMs(Yao et al., 2024) which have the ability to create non-extractive timelines that are not present in the original text. A variety of strategies can be deployed for this, of which fine-tuning(Anisuzzaman et al., 2025), retrieval augmented generation (RAG)(Arslan et al., 2024), and prompt engineering(Brown et al., 2020) are popular choices. Fine-tuning is costly, but prompt-engineering is a light-weight strategy for performance improvement. One such modular prompt-engineering framework is DSPy(Khattab et al., 2024) which implements a variety of different prompt engineering strategies including Simba(Lee et al., 2024), MIPROv2(Opsahl-Ong et al., 2024) as well as few-shot selection. Prompts provide an easy mechanism to include relevant temporal events either from the gold information in Task 1 or through encoder-based extraction methods. Fine-tuned encoders may still outperform LLMs in information extraction tasks due to their bidirectional understanding of language (the result of masked language modeling instead of autoregressive training), however, more recent results are mixed(Obeidat

October 30, 2025 ©2025 Association for Computational Linguistics

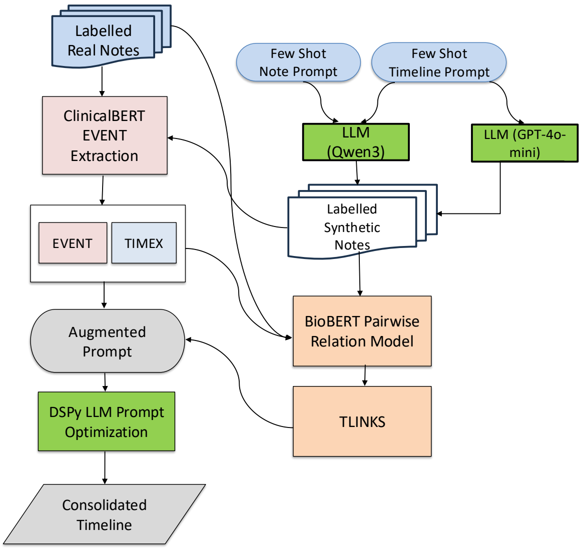


Figure 1: System Overview. Both Task 1 and Task 2 system use DSPy in conjunction with a LLM to generate JSON formatted SACT timeline predictions. The Task 1 system uses gold EVENTS and TIMEX whereas the Task 2 system directly generates those from text.

et al., 2025; Arzideh et al., 2025; Roccabruna et al., 2024). Similarly to others(Bannour et al., 2024; Tan et al., 2024) in previous tasks, we employ encoder models to better provide context for the final timeline.

2 Method and Materials

2.1 System Overview

Our system takes as input either gold annotations (EVENTS or TIMEX expressions), and TLINKs from the Task 1 gold standard or generates them using either a BioBERT(Lee et al., 2020) model (TLINKs) or a ClinicalBERT(Alsentzer et al., 2019) (EVENTs) model. An overview is shown in Figure 1. These inputs are integrated into the prompt to guide timeline generation, with the encoder models semantically rich representations that capture long-context clinical dependencies.

2.2 Synthetic Data Generation

Synthetic data was created to assist information extraction for TLINK identification and EVENT identification as shown in Figure 1. The goal is to improve robustness in low resource settings (Li et al., 2021).

TLINK Generation For TLINK synthetic data, an OpenAI GPT-4o-mini model hosted in the UAB's firewalled Azure Enclave was used. This Enclave is approved for PHI by the UAB's Health System. Synthetic data was used to compensate for class imbalance in the original training data, so synthetic examples were generated only to augment minority classes (ENDS-ON and BEGINS-ON). Specific training details can be found at The system is available on `github.com/vijay0019/UAB_ChemoTimelines`.

EVENT Generation Synthetic oncology notes were generated with Qwen3-32B, using a one-shot prompting technique to address the lack of EVENT training data. Each prompt sampled drugs/regimens from a merged lexicon built by uniting entities observed in the training notes with entries from HemOnc(Warner et al., 2015), a curated open regimen vocabulary. The synthetic notes were mixed with the real corpus for a second round of fine tuning. This knowledge-guided augmentation targets the regimen names, abbreviations, and phrasings that are undersampled in the original notes. Notes are generated by randomly sampling drugs and regimens from the merged lexicon, with a constraint that no drug/regimen appears more than twice per note. Each synthetic note is conditioned to match the writing style and structure of real notes including de-identified headers and footers thereby preserving real-world patterns while preventing exposure of Protected Health Information (PHI) (Melamud and Shivade, 2019). We also introduce controlled variation e.g., domain specific abbreviations(Liu et al., 2001) and common misspellings to better reflect noisy clinical text. Entities in the synthetic notes are validated using the given list.

Timeline Generation Synthetic timelines were generated with the goal of creating a more comprehensive synthetic set of notes. An overview of synthetic data generation for EVENTs is shown in Figure 2. Synthetic timelines were also generated by Qwen3-32B (Yang et al., 2025) with reasoning disabled. The model was prompted with both a system prompt and a user prompt to guide synthetic timeline generation. The system prompt described the timeline generation task, including definitions and formatting preferences, i.e. if the model is following the correct format for the output (SACT entity, relation type, TIMEX3 expression) for each timeline triplet and TIMEX3 expressions are formatted correctly. The user prompt gave five example timelines from the training set and asked the model to generate a synthetic timeline for a patient having cancer with a specific primary site,

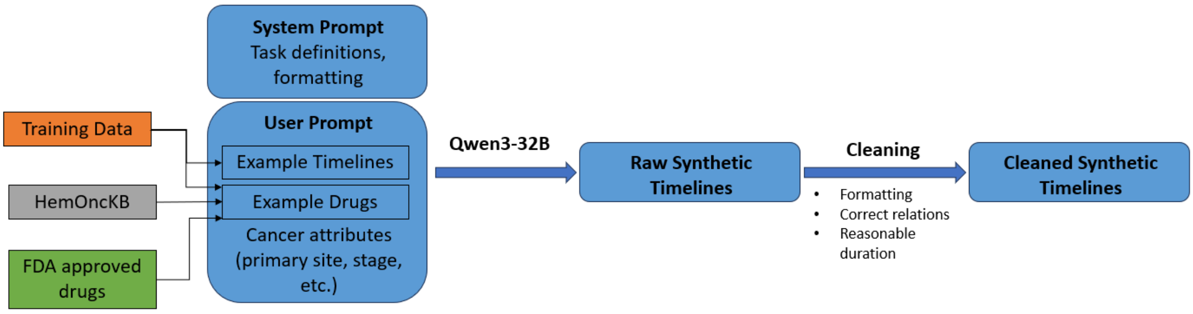


Figure 2: Synthetic Timeline generation

maximum stage of progression, stage at diagnosis, remission and recurrence status, time at start of therapy and therapy duration. The user prompt also included five SACT references for the primary site requested. These SACT references are drawn from a list that includes all SACTs in the training data, SACTs found in the HemOnc Knowledge Base (Warner et al., 2015) and SACTs found in the FDA's approved drugs for cancer lists (National Cancer Institute, 2025). After generation of timelines, there was a cleaning step in which timelines that were incorrectly formatted, had incorrect relations or were five years or longer were removed. An attempt is made to reformat the timeline to a correct format before removal. Part of these cleaned timelines were then used as a prompt for STRS1. An example cleaned synthetic timeline with its system and user prompt, as well as its raw form before reformatting, is shown in Figure 3. The system is available at `https://github.com/vijay0019/UAB_ChemoTimelines`.

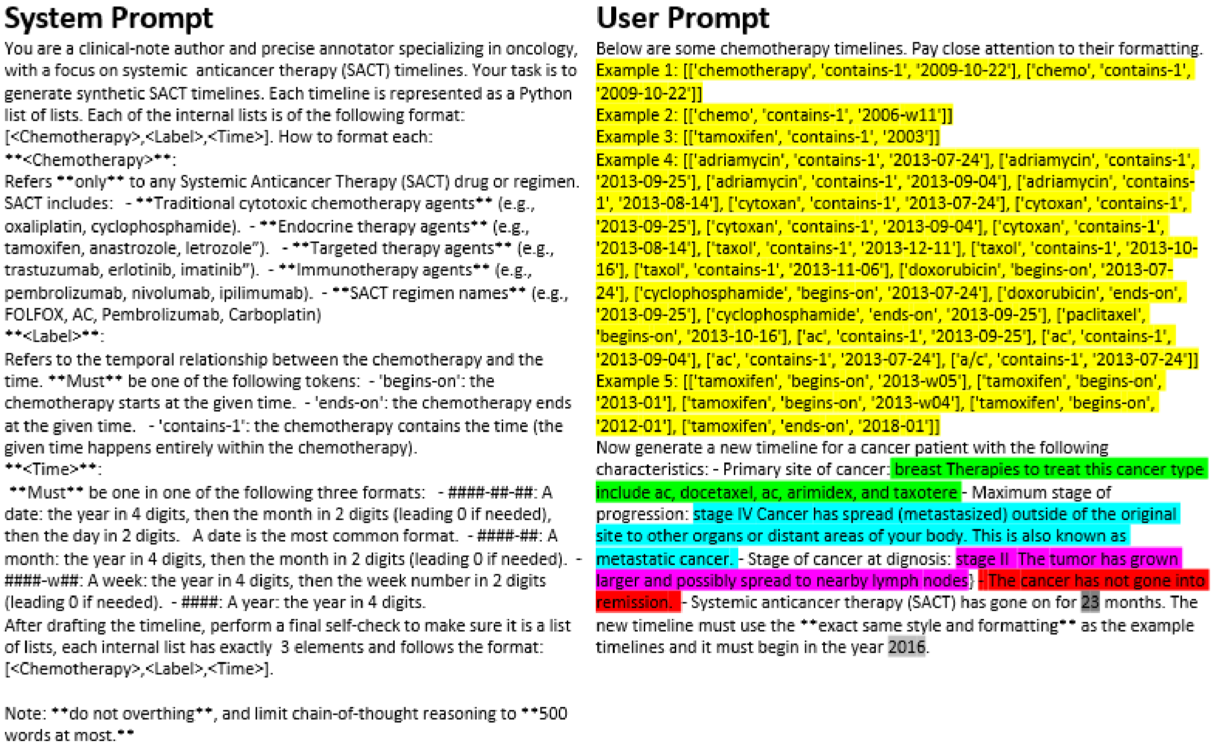
2.3 SACT EVENT Extraction

For Task 1, SACT EVENTS and TIMEX expressions are provided for the test data set, but for Task 2 we developed our own pipeline to extract SACT EVENTS (drug and regimen) mentions and TIMEX3 time expressions in clinical notes. These two outputs form the nodes for the downstream temporal relation prediction where we link each chemotherapy event to its most relevant time reference to build patient level timelines. This design mirrors the task framing: systems must identify events and time expressions, then infer their temporal relations to recover the treatment timeline. Since encoder models tend to perform well for these tasks, therefore we fine-tuned ClinicalBERT (Alsentzer et al., 2019) for NER over the official training notes to detect SACT entities (drug/regimen spans). On our held out dev set, this achieved a high precision but low recall. While precision was strong, the relatively low recall meant that many true SACT mentions were missed. In a timeline-reconstruction setting, recall shortfalls are especially harmful because missing them means removing candidate EVENT–TIMEX3 pairs for the relation prediction, reducing the chance of recovering correct [chemo, relation, time] triplets even when the TIMEX3 recognition is accurate. For TIMEX3, we use TimeNorm for span recognition and normalization(Bethard, 2013; Laparra et al., 2018; Xu et al., 2019).

2.4 SACT TLINK Extraction

The relation classification task involved four categories: CONTAINS, BEGINS-ON, ENDS-ON, and no_relation between SACT and TIMEX entities. BioBERT (Lee et al., 2020) was fine-tuned for span-based relation classification. The model extracts span representations by combining start/end token embeddings, width embeddings and entity type embeddings. For context, between-token context extraction was implemented using average pooling of tokens between subject and object entities. The final classification layer concatenates subject, context, and object representations through a two-layer feedforward network with ReLU activation. Model selection was based on macro-averaged F1-score on validation data. Predictions were filtered using a 0.5 confidence threshold and constraints on relation entity type. For overlapping predictions, more specific relations (BEGINS-ON/ENDS-ON) were prioritized over CONTAINS. Duplicate relations across patient notes were removed.

Figure 3: The system prompt and an example user prompt. Highlighted portions in the user prompt are variable. Text highlighted yellow corresponds to example timelines, green to the primary site and explanation, blue to most advanced stage and explanation, magenta to diagnosis stage and explanation, red to remission and recurrence information, dark gray to timeline duration and light gray to start year. Blue font text in the raw output corresponds to the text that was extracted to create the cleaned output.



System Prompt

You are a clinical-note author and precise annotator specializing in oncology, with a focus on systemic anticancer therapy (SACT) timelines. Your task is to generate synthetic SACT timelines. Each timeline is represented as a Python list of lists. Each of the internal lists is of the following format: [<Chemotherapy>,<Label>,<Time>]. How to format each:
<Chemotherapy>:
Refers **only** to any Systemic Anticancer Therapy (SACT) drug or regimen. SACT includes: - **Traditional cytotoxic chemotherapy agents** (e.g., oxaliplatin, cyclophosphamide). - **Endocrine therapy agents** (e.g., tamoxifen, anastrozole, letrozole"). - **Targeted therapy agents** (e.g., trastuzumab, erlotinib, imatinib"). - **Immunotherapy agents** (e.g., pembrolizumab, nivolumab, ipilimumab). - **SACT regimen names** (e.g., FOLFOX, AC, Pembrolizumab, Carboplatin)
<Label>:
Refers to the temporal relationship between the chemotherapy and the time. **Must** be one of the following tokens: - 'begins-on': the chemotherapy starts at the given time. - 'ends-on': the chemotherapy ends at the given time. - 'contains-1': the chemotherapy contains the time (the given time happens entirely within the chemotherapy).
<Time>:
Must be one in one of the following three formats: - ####-##-##: A date: the year in 4 digits, then the month in 2 digits (leading 0 if needed), then the day in 2 digits. A date is the most common format. - ####-##: A month: the year in 4 digits, then the month in 2 digits (leading 0 if needed). - ####-w##: A week: the year in 4 digits, then the week number in 2 digits (leading 0 if needed). - ####: A year: the year in 4 digits.
After drafting the timeline, perform a final self-check to make sure it is a list of lists, each internal list has exactly 3 elements and follows the format: [<Chemotherapy>,<Label>,<Time>].

Note: **do not overthing**, and limit chain-of-thought reasoning to **500 words at most.**

User Prompt

Below are some chemotherapy timelines. Pay close attention to their formatting.
Example 1: [['chemotherapy', 'contains-1', '2009-10-22'], ['chemo', 'contains-1', '2009-10-22']]
Example 2: [['chemo', 'contains-1', '2006-w11']]
Example 3: [['tamoxifen', 'contains-1', '2003']]
Example 4: [['adriamycin', 'contains-1', '2013-07-24'], ['adriamycin', 'contains-1', '2013-09-25'], ['adriamycin', 'contains-1', '2013-09-04'], ['adriamycin', 'contains-1', '2013-08-14'], ['cytoxan', 'contains-1', '2013-07-24'], ['cytoxan', 'contains-1', '2013-09-25'], ['cytoxan', 'contains-1', '2013-09-04'], ['cytoxan', 'contains-1', '2013-08-14'], ['taxol', 'contains-1', '2013-12-11'], ['taxol', 'contains-1', '2013-10-16'], ['taxol', 'contains-1', '2013-11-06'], ['doxorubicin', 'begins-on', '2013-07-24'], ['cyclophosphamide', 'begins-on', '2013-07-24'], ['doxorubicin', 'ends-on', '2013-09-25'], ['cyclophosphamide', 'ends-on', '2013-09-25'], ['paclitaxel', 'begins-on', '2013-10-16'], ['ac', 'contains-1', '2013-09-25'], ['ac', 'contains-1', '2013-09-04'], ['ac', 'contains-1', '2013-07-24'], ['a/c', 'contains-1', '2013-07-24']]
Example 5: [['tamoxifen', 'begins-on', '2013-w05'], ['tamoxifen', 'begins-on', '2013-01'], ['tamoxifen', 'begins-on', '2013-w04'], ['tamoxifen', 'begins-on', '2012-01'], ['tamoxifen', 'ends-on', '2018-01']]
Now generate a new timeline for a cancer patient with the following characteristics: - Primary site of cancer: breast Therapies to treat this cancer type include ac, docetaxel, ac, arimidex, and taxotere - Maximum stage of progression: stage IV Cancer has spread (metastasized) outside of the original site to other organs or distant areas of your body. This is also known as metastatic cancer. - Stage of cancer at dignosis: stage II The tumor has grown larger and possibly spread to nearby lymph nodes} - The cancer has not gone into remission. - Systemic anticancer therapy (SACT) has gone on for 23 months. The new timeline must use the **exact same style and formatting** as the example timelines and it must begin in the year 2016.

2.5 SACT Timeline Extraction System (STES)

The STES employs a multi-iteration approach using the DSPy framework with large language models to process clinical reports and construct temporal treatment timelines. The system begins by creating report clumps that fit within the model's context window (typically $\frac{1}{8}$ of the total context size) and groups reports by patient ID to maintain temporal coherence. Each iteration processes these clumps through a `SACTTimelineUpdate` module that extracts drug names exactly as they appear in clinical text. This includes brand names, generic names, abbreviations, and variations—along with their temporal relations (*begins-on*, *ends-on*, *contains-1*) and associated dates with varying levels of specificity (year, month, day, or week).

The system implements an incremental timeline construction strategy where each processed report clump updates the existing timeline by adding new events and removing conflicting ones through an `Update` object containing *add* and *remove* lists. To ensure robustness, the system employs a retry mechanism across multiple language model instances with different temperature settings, falling back to empty updates when all models fail to generate valid responses. The final timeline undergoes deduplication and chronological sorting based on date components (year, month, day, week) followed by drug name and relation type. Date objects are converted to competition-standard string formats, and the system validates date formatting through regex patterns before generating the final JSON output for each patient's treatment timeline.

An enhanced version of the SACT timeline extraction system (task1_v2_summaries_plus) incorporates running summaries to maintain contextual information across report processing iterations. The system generates and updates a comprehensive treatment summary that captures key treatment phases, medication regimens, temporal milestones, and treatment response indicators mentioned in the reports. This summary serves as persistent memory between iterations, allowing the model to maintain coherence when processing large patient records that exceed context window limitations. The summary is structured to include treatment overviews, detailed medication histories with both generic and brand names, protocol documentation, timeline reconciliation notes, and clinical observations, effectively creating a condensed narrative of the patient's treatment journey that informs subsequent timeline extraction decisions.

Task 2 implements a fundamentally different architecture designed for chemotherapy event extraction from concatenated clinical chunks rather than structured report processing. Unlike Task 1's report-centric approach, Task 2 employs a three-stage pipeline consisting of `ChemoNotesTimeline` for initial event extraction, `ChemoTimelineUpdate` for incremental timeline construction, and `ChemoTimelineCleanup` for deduplication and conflict resolution. The system intelligently manages token usage by applying a summarization step (`ChemoNotesTimeline`) when content exceeds a configurable threshold (typically 25% of context window), and implements dynamic timeline cleanup when the number of events surpasses a specified limit. Task 2 also incorporates dual model configurations with different repetition penalty settings to handle various text patterns and includes more sophisticated chunk concatenation strategies that respect document boundaries and optimize context window utilization. For the ablation study, we used a slightly modified version of our submission model; whereas the submission version forced the model to choose from a hard-coded list of both train and dev gold terms, for the ablation study the requirement was relaxed to a generic string. The full prompt is shown in Appendix A.

2.6 Chemotherapy Timeline Experiments

An early system (Version 1) and a later updated system were used for test submission results for Task 1. The updated system differed only in terms of slight variations to the zero-shot (manually generated) prompt and some post-processing steps related to pruning timeline entities based on multiple passes through the reports. A newer system (Version 2) is used on the development set. This features additional changes to the prompt, including the addition of a running summary, a better timeline example, and an LLM-generated example summary. It also included minor improvements such as the report date and restricting to a single pass through the notes in chronological (rather than random) order. The EVENT and TLINK performance is reported on the dev data set only, as runs on the test set were not completed prior to the task deadline. Local models were run on A100 40GB VRAM GPUs, GPT-4.1-mini was running in the same Azure Enclave as the GPT-4o-mini model used for synthetic TLINK generation.

3 Results

3.1 Official System Results

Official system results are shown in Table 1. As expected, the smaller Phi4:14B is outperformed by GPT-4.1-mini.

Table 1: Version 1.1 prompt provides additional instructions to avoid ungrounded temporal relations and logic to remove timelines that show up infrequently in iterations. Entities are usd as input for all Subtask 1 systems.

System	Task	LLM	Brca	Mela	Ovca	Avg
UABv1	1	Phi4:14B	0.310	0.160	0.217	0.229
UABv1.1	1	Phi4:14B	0.259	0.333	0.244	0.279
UABv1.1	1	GPT-4.1-mini	0.418	0.308	0.296	0.341
UABv0	2	Phi4:14B	0.232	0.265	0.188	0.228

Table 2: Token level EVENT recognition on the test set using ClinicalBERT trained on Actual, Synthetic, and Actual+Synthetic notes.

Training data	Prec.	Recall	F1	Acc.
Actual	**95.6**	83.0	88.9	99.76
Synthetic	64.2	81.8	71.9	99.26
Actual+Synthetic	95.2	**87.8**	**91.4**	**99.81**

3.2 EVENT Extraction Results

We fine-tuned ClinicalBERT (Alsentzer et al., 2019) for token-level EVENT classification on CoNLL-style inputs under three training regimes: real (human-authored) notes, synthetic (LLM-generated) notes, and their mixture . On the test set, the mixture attained 91.4 F1 (P=95.2, R=87.8), a 2.8% relative F1 increase over real-only, accompanied by a 5.8% relative recall increase and a 0.4% relative precision decrease. In comparison, synthetic-only yielded 71.9 F1, a 19.0% relative decrease vs. real-only, with precision 32.8% lower and recall 1.5% lower. Augmenting real notes with synthetic text yielded SACT NER P=0.9524, R=0.8781, F1=0.9137. The higher recall expands the pool of EVENT candidates available to downstream event–time relation classification. These results are consistent with evidence that LLM-generated, ontology-guided synthetic text can improve clinical NER by increasing coverage of rare surface forms without materially degrading precision (Dao et al., 2025). Overall token accuracy is

≈99% across settings so we de-emphasize accuracy given severe class imbalance and instead focus on precision, recall, and F1 for the EVENT class.

Impact of Synthetic Notes on EVENT Extraction Replacing real notes with non-timeline sourced synthetic notes reduces recall and increases false positives: TPs decrease to 9,927 (from 10,076), FNs increase to 2,209 (from 2,060), and FPs increase to 5,534 (from 466). By contrast, training on the mixture improves recall with a small precision cost: TPs increase to 10,657, FNs fall to 1,479 (≈28% fewer than real-only), and FPs increase modestly to 533 (vs. 466). Overall, synthetic-only induces a high false-positive rate, whereas the mixture identifies 581 additional true-positive EVENT tokens relative to real-only. Timeline-sourced notes were judged to be poorer in quality and ultimately were not used for EVENT extraction.

3.3 TLINK Extraction Results

The TLINK extraction was assessed on the development data set, since no TLINKS were provided as part of the Task 1 test data set. The model achieved 89.4% accuracy with a macro F1-score of 0.889, performing particularly well on positive relations (F1: 0.944).

3.4 Zero-Shot and Few-Shot Local LLM Evaluation

In addition to the official test results, we included an updated set of results on the dev set with additional LLMs in Table 3.

4 Discussion

Our results indicate that despite recent LLM improvements in a range of tasks, the identification of chemotherapy timelines is not a task that can be done well "out of the box" without significant engineering. Only our encoder models, using fine-tuning on a sufficient amount (supplemented with synthetic data) of training data generated reliable performance improvements without significant human intervention. Of interest in the development evaluation, that ablating the chain-of-thought for Qwen3:32B did drop performance, but we lacked time to assess if this generalized to other models.

Larger local LLMs could have been fine-tuned (at greater cost or time) to improve performance similar to previous work fine-tuning Flan-T5 in the 2024 task (Haddadan et al., 2024). The incorporation of additional Performance Efficient Fine-Tuning, Retrieval Augmented Generation, soft prompting and utilization of the existing modifier information in the gold standard likely could have improved results.

We evaluated DSPy's suite of primarily bootstrapping-based methods including Simba(Lee et al., 2024), MIPROv2(Opsahl-Ong et al., 2024) and GEPA(Agrawal et al., 2025), but all either failed to complete with DSPy related errors and/or yielded preliminary results that discouraged debugging. Silver-quality examples of individual timeline chronological updates congruent with context window size would perhaps have been more useful. We are given gold timelines, but not gold timeline updates, and it is a non-trivial task to generate useful examples of correct updates. Additionally, generating timelines and summaries separately for each report and then iteratively pooling them, rather than our cumulative approach, may yield better results and will be explored in future work.

5 Conclusion

Overall, this task remains challenging even with the use of LLMs such as GPT-4.1-mini suggesting that currently, specialized training is required to achieve results comparable to humans. Current effort to create synthetic timelines do not improve performance. We found it was substantially easier through fine-tuning to obtain reliable, fast results with encoder models than to fine-tune prompts for LLMs.

Limitations

Due to the sensitive nature of the data, a Data Use Agreement is required to obtain the data needed to replicate our results. A more complete evaluation of modern LLMs was not feasible due to cost, so GPT-4.1-mini was the only large model fully evaluated.

Acknowledgments

We acknowledge UAB Research Computing for access to the High Performance Computing Infrastructure. Claude Code was used for the generation some of the code by 3 of the developers.

System	Subtask	LLM	Input	CoT	Learning	Breast	Melanoma	Ovarian
UABv1	1	Phi4:14B	Entities	Y	Zero-Shot	0.286	0.145	0.191
UABv1.1	1	Phi4:14B	Entities	Y	Zero-Shot	0.248	0.156	0.196
UABv2	1	Qwen3:32B	Entities	Y	Zero-Shot	0.535	0.593	0.260
UABv2	1	Qwen3:32B	Entities	N	Zero-Shot	0.300	0.314	0.297
UABv2	1	Qwen3:30B:3A	Entities	Y	Zero-Shot	0.659	0.507	0.266
UABv2	1	Qwen3:30B:3A	Entities	Y	Few-Shot	0.520	0.230	0.284
UABv0	2	Phi4:14B	None	Y	Zero-Shot	0.286	0.530	0.159

Table 3: Version 2 includes many changes to the prompt and the addition of summaries.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, and 1 others. 2024. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, and 1 others. 2025. Gepa: Reflective prompt evolution can outperform reinforcement learning. *arXiv preprint arXiv:2507.19457*.

Emily Alsentzer, John R. Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew B. A. McDermott. 2019. Publicly available clinical bert embeddings. *Preprint*, arXiv:1904.03323.

D.M. Anisuzzaman, Jeffrey G. Malins, Paul A. Friedman, and Zachi I. Attia. 2025. Fine-tuning large language models for specialized use cases. *Mayo Clinic Proceedings: Digital Health*, 3(1):100184.

Muhammad Arslan, Hussam Ghanem, Saba Munawar, and Christophe Cruz. 2024. A survey on rag with llms. *Procedia Computer Science*, 246:3781–3790. 28th International Conference on Knowledge Based and Intelligent information and Engineering Systems (KES 2024).

Kamyar Arzideh, Henning Schäfer, Héctor Allende-Cid, Giulia Baldini, Thomas Hilser, Ahmad Idrissi-Yaghir, Katharina Laue, Nilesh Chakraborty, Niclas Doll, Dario Antweiler, and 1 others. 2025. From bert to generative ai-comparing encoder-only vs. large language models in a cohort of lung cancer patients for named entity recognition in unstructured medical reports. *Computers in Biology and Medicine*, 195:110665.

Nesrine Bannour, Judith Jeyafreeda Andrew, and Marc Vincent. 2024. Team NLPeers at chemotimelines 2024: Evaluation of two timeline extraction methods, can generative LLM do it all or is smaller model fine-tuning still relevant ? In *Proceedings of the 6th Clinical Natural Language Processing Workshop*, pages 406–416, Mexico City, Mexico. Association for Computational Linguistics.

Steven Bethard. 2013. A synchronous context free grammar for time normalization. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 821–826, Seattle, Washington, USA. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

An Dao, Hiroki Teranishi, Yuji Matsumoto, Florian Boudin, and Akiko Aizawa. 2025. Overcoming data scarcity in named entity recognition: Synthetic data generation with large language models. In *Proceedings of the 24th Workshop on Biomedical Language Processing*, pages 328–340, Viena, Austria. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Shohreh Haddadan, Tuan-Dung Le, Thanh Duong, and Thanh Thieu. 2024. Lailab at chemotimelines 2024: Finetuning sequence-to-sequence language models for temporal relation extraction towards cancer patient undergoing chemotherapy treatment. In *Proceedings of the 6th Clinical Natural Language Processing Workshop*, pages 382–393.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Saiful Haq, Ashutosh Sharma, Thomas T Joshi, Hanna Moazam, Heather Miller, and 1 others. 2024. Dspy: Compiling declarative language model calls into state-of-the-art pipelines. In *The Twelfth International Conference on Learning Representations*.

Egoitz Laparra, Dongfang Xu, and Steven Bethard. 2018. From characters to time intervals: New paradigms for evaluation and neural parsing of time normalizations. *Transactions of the Association for Computational Linguistics*, 6:343–356.

Hojoon Lee, Dongyoon Hwang, Donghu Kim, Hyunseung Kim, Jun Jet Tai, Kaushik Subramanian, Peter R Wurman, Jaegul Choo, Peter Stone, and Takuma Seno. 2024. Simba: Simplicity bias for scaling up parameters in deep reinforcement learning. *arXiv preprint arXiv:2410.09754*.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Simon A Lee, Anthony Wu, and Jeffrey N Chiang. 2025. Clinical modernbert: An efficient and long context encoder for biomedical text. *arXiv preprint arXiv:2504.03964*.

Jianfu Li, Yujia Zhou, Xiaoqian Jiang, Karthik Natarajan, Serguei Vs Pakhomov, Hongfang Liu, and Hua Xu. 2021. Are synthetic clinical notes useful for real natural language processing tasks: A case study on clinical entity recognition. *Journal of the American Medical Informatics Association*, 28(10):2193–2201.

Hongfang Liu, Yves A. Lussier, and Carol Friedman. 2001. A study of abbreviations in the UMLS. In *Proceedings of the AMIA Symposium*, pages 393–397.

Oren Melamud and Chaitanya Shivade. 2019. Towards automatic generation of shareable synthetic clinical notes using neural language models. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 35–45, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

National Cancer Institute. 2025. Drugs approved for different types of cancer. `https://www.cancer.gov/about-cancer/treatment/drugs/cancer-type`. Accessed: 2025-08-04.

Motasem S Obeidat, Md Sultan Al Nahian, and Ramakanth Kavuluru. 2025. Do llms surpass encoders for biomedical ner? In *2025 IEEE 13th International Conference on Healthcare Informatics (ICHI)*, pages 352–358. IEEE.

OpenAI. 2025. Introducing GPT-4.1 in the API. Online blog post (OpenAI). Retrieved from OpenAI.

OpenAI. 2025. Introducing openai o3 and o4-mini. `https://openai.com/index/introducing-o3-and-o4-mini/`. Accessed: 2025-08-20.

Krista Opsahl-Ong, Michael J Ryan, Josh Purtell, David Broman, Christopher Potts, Matei Zaharia, and Omar Khattab. 2024. Optimizing instructions and demonstrations for multi-stage language model programs. *arXiv preprint arXiv:2406.11695*.

Gabriel Roccabruna, Massimo Rizzoli, and Giuseppe Riccardi. 2024. Will llms replace the encoder-only models in temporal relation classification? *arXiv preprint arXiv:2410.10476*.

Yukun Tan, Merve Dede, and Ken Chen. 2024. KCLab at chemotimelines 2024: End-to-end system for chemotherapy timeline extraction – subtask2. In *Proceedings of the 6th Clinical Natural Language Processing Workshop*, pages 417–421, Mexico City, Mexico. Association for Computational Linguistics.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. 2024. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory-efficient, and long-context fine-tuning and inference. *arXiv preprint arXiv:2412.13663*.

Jeremy L. Warner, Andrew J. Cowan, Aric C. Hall, and Peter C. Yang. 2015. Hemonc.org: A collaborative online knowledge platform for oncology professionals. *Journal of oncology practice*, 11(3):e336–e350.

Dongfang Xu, Egoitz Laparra, and Steven Bethard. 2019. Pre-trained contextualized character embeddings lead to major improvements in time normalization: a detailed analysis. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 68–74, Minneapolis, Minnesota. Association for Computational Linguistics.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Jiarui Yao, Harry Hochheiser, WonJin Yoon, Eli Goldner, and Guergana Savova. 2024. Overview of the 2024 shared task on chemotherapy treatment timeline extraction. In *Proceedings of the 6th Clinical Natural Language Processing Workshop*, pages 557–569.

Jiarui Yao, Harry Hochheiser, WonJin Yoon, Eli Goldner, and Guergana Savova. 2025. Overview of the 2025 shared task on chemotherapy treatment timeline extraction. In *Proceedings of the 7th Clinical Natural Language Processing Workshop*.

A System Prompt

Figure 4: Full System Prompt used for SACT timeline generation (Part 1).

```
Update SACT timeline based on patient reports with running summary.

SACT is defined as follows:
"Systemic anticancer therapy (SACT), which includes traditional cytotoxic
    chemotherapy, endocrine therapy, targeted therapy, and immunotherapy, has
    both a low therapeutic index as well as synergistic potential when agents
    are given in combination."

Drug Names: Extract drug names EXACTLY as they appear in the clinical text (
    except be sure to put them in lowercase). Do NOT normalize or convert to
    generic names.
Include ALL variations found in the text:
- Brand names (cytoxan, taxotere, abraxane)
- Generic names (cyclophosphamide, docetaxel, paclitaxel)
- Abbreviations (tc, ac, a/c)
- Generic terms (chemotherapy, chemo)
- Slight variations/typos (docetaxol for docetaxel)

If the text mentions both "cyclophosphamide" and "cytoxan", include both as
    separate entries.
If the text mentions both "chemotherapy" and specific drug names, include both.
Only include drugs that have a temporal relation in the text.

Ignore references to cancer/neoplasms, genetic variations (e.g. HER2), and non-
    SACT procedures such as radiation therapy.

Relations:
- 'begins-on': treatment/medication starts
- 'ends-on': treatment/medication ends
- 'contains-1': treatment occurred within timeframe
'begins-on' and 'ends-on' supersede 'contains-1' for the same drug/date
    combination. Only use them if the text explicitly states the start or end
    date of the treatment.

Acceptable date formats:
1. Specify year, month, and day.
2. Specify year and week.
3. Specify year and month.
4. Specify year only.
Try to be as specific as possible, but do not invent dates that are not
    mentioned in the text.

Keep in mind that the reports are only a subset of the full timeline, so there
    may be events in the timeline that are not mentioned in the reports. Do not
     remove events simply because they are not mentioned in the reports.

If a report doesn't have temporal relations, that likely means the report does
    not contain any relevant information for the timeline. Avoid adding events
    based solely on hypothetical or planned mentions without temporal grounding
    .

Running Summary: Maintain a concise summary of the patient's SACT treatment
    journey, including:
- Key treatment phases and regimens
- Major treatment changes or progressions
- Important temporal milestones
- Treatment response indicators mentioned in reports
Update the summary to reflect new information from current reports while
    preserving important historical context.

Example output format:
[[ ## updated_summary ## ]]
**Treatment Overview:** Chemotherapy begins week 32 of 2013, with documented
    treatment from August 8 - October 10, 2013


"""
```

Figure 5: Full System Prompt used for SACT timeline generation (Part 2).

```
**Medications Administered:**
- **Cyclophosphamide (Cytoxan):** August 8 - October 10, 2013
  - Individual doses: August 29, September 19, October 10
- **Docetaxel (Taxotere):** August 8 - October 10, 2013
  - Individual doses: August 29, September 19, October 10
  - *Note: "Docetaxol" also documented (likely transcription error)*

**Protocol Documentation:**
- **TC regimen** active during August 2013
- **Chemotherapy** contains specific date August 10, 2013
- Standard 3-week cycling pattern evident

**Timeline Reconciliation:**
- Week 32 of 2013 corresponds to ~August 5-11 timeframe
- Treatment initiation August 8 aligns with weekly scheduling
- 4 documented treatment cycles completed over 9-week period

**Clinical Notes:** Complete documentation includes both generic (
    cyclophosphamide, docetaxel) and brand names (Cytoxan, Taxotere) with minor
     spelling variant recorded.
[[ ## timeline_update ## ]]
Update(
    add=[
        ('chemotherapy', 'begins-on', Date(year=2013, month=None, day_of_month=
    None, week_of_year=32)),
        ('chemotherapy', 'contains-1', Date(year=2013, month=8, day_of_month
    =10, week_of_year=None)),
        ('cyclophosphamide', 'begins-on', Date(year=2013, month=8, day_of_month
    =8, week_of_year=None)),
        ('cyclophosphamide', 'ends-on', Date(year=2013, month=10, day_of_month
    =10, week_of_year=None)),
        ('cytoxan', 'begins-on', Date(year=2013, month=8, day_of_month=8,
    week_of_year=None)),
        ('cytoxan', 'contains-1', Date(year=2013, month=8, day_of_month=29,
    week_of_year=None)),
        ('cytoxan', 'contains-1', Date(year=2013, month=9, day_of_month=19,
    week_of_year=None)),
        ('cytoxan', 'contains-1', Date(year=2013, month=10, day_of_month=10,
    week_of_year=None)),
        ('docetaxel', 'begins-on', Date(year=2013, month=8, day_of_month=8,
    week_of_year=None)),
        ('docetaxel', 'ends-on', Date(year=2013, month=10, day_of_month=10,
    week_of_year=None)),
        ('docetaxol', 'begins-on', Date(year=2013, month=8, day_of_month=8,
    week_of_year=None)),
        ('taxotere', 'begins-on', Date(year=2013, month=8, day_of_month=8,
    week_of_year=None)),
        ('taxotere', 'contains-1', Date(year=2013, month=8, day_of_month=29,
    week_of_year=None)),
        ('taxotere', 'contains-1', Date(year=2013, month=9, day_of_month=19,
    week_of_year=None)),
        ('taxotere', 'contains-1', Date(year=2013, month=10, day_of_month=10,
    week_of_year=None)),
        ('tc', 'contains-1', Date(year=2013, month=8, day_of_month=None,
    week_of_year=None))
    ],
    remove=[
        ('cytoxan', 'begins-on', Date(year=2013, month=8, day_of_month=1,
    week_of_year=None))
    ]
)
[[ ## completed ## ]]
"""
```