PNLP at MEDIQA-OE 2025: A Zero-Shot Prompting Strategy with Gemini for Medical Order Extraction

Parth Mehta
silver_shaw (Competition Team Name)
parthmehta2016@gmail.com

Abstract

Medical order extraction from doctor-patient conversations presents a critical challenge in reducing clinical documentation burden and ensuring accurate capture of patient care instructions. This paper describes our system for the MEDIQA-OE 2025 shared task using the ACI-Bench and PriMock57 datasets, which achieved second place on the public leaderboard with an average score of 0.6014 across four metrics: description ROUGE-1 F1, reason ROUGE-1 F1, order-type strict F1, and provenance multi-label F1. Unlike traditional approaches that rely on fine-tuned biomedical language models, we demonstrate that a carefully engineered zero-shot prompting strategy using Gemini 2.5 Pro can achieve competitive performance without requiring model training or GPU resources. Our approach employs a deterministic state-machine prompt design incorporating chain-of-thought reasoning, self-verification protocols, and structured JSON output generation. The system particularly excels in reason extraction, achieving 0.4130 ROUGE-1 F1, the highest among the top performing teams. Our results suggest that advanced prompt engineering can effectively bridge the gap between general-purpose large language models and specialized clinical NLP tasks, offering a computationally efficient and immediately deployable alternative to traditional fine-tuning approaches with significant implications for resource-constrained healthcare settings.

1 Introduction

Clinical documentation represents a significant burden for healthcare providers, consuming substantial time that could otherwise be devoted to patient care. The accurate extraction of medical orders from doctor-patient conversations is essential for maintaining comprehensive patient records while reducing this documentation overhead. The MEDIQA-OE 2025 (Corbeil et al., 2025b) shared task addresses this challenge by focusing on the automatic identification and structuring of medical orders - including medications, laboratory tests, imaging studies, and follow-up appointments - from conversational transcripts.

Traditional approaches to medical information extraction have relied heavily on fine-tuned domain-specific models, which require substantial computational resources and annotated training data. These methods, while effective, present barriers to deployment in resource-constrained clinical environments where computational resources are limited and rapid deployment is essential.

This paper presents a paradigm shift in clinical NLP: leveraging the reasoning capabilities of large language models through sophisticated prompt engineering to achieve competitive performance without any model training. Our team's submission to MEDIQA-OE 2025 demonstrates that a zero-shot approach using Gemini 2.5 Pro, combined with deterministic chain-of-thought reasoning and self-critique mechanisms, achieves second place on the public leaderboard with an average F1 score of 0.6014. Notably, this performance was attained without GPU requirements or model fine-tuning, making the solution immediately deployable.

Our approach challenges conventional wisdom about the necessity of domain-specific fine-tuning for clinical NLP tasks. The main contributions of this work are: (1) empirical evidence that prompt engineering can match or exceed fine-tuned models for medical order extraction; (2) a structured prompting methodology combining chain-of-thought reasoning with self-verification specifically designed for clinical information extraction, democratizing access to advanced clinical documentation tools.

2 Related Work

Medical order extraction from clinical conversations has evolved significantly, and the MEDIQA-

Proceedings of the 7th Clinical Natural Language Processing Workshop, pages 75–83
October 30, 2025 ©2025 Association for Computational Linguistics

OE shared task series driving innovation in this domain. Early approaches focused on named entity recognition and relation extraction from structured clinical texts. Recent advances have shifted toward utilizing transformer architectures and processing unstructured clinical conversations.

BioBERT (Lee et al., 2019) established domain-adaptive pre-training as a strong baseline for biomedical NER, relation extraction, and QA, making encoder backbones prevalent in clinical NLP pipelines. BioClinical-ModernBERT (Sounack et al., 2025) introduced a domain-adapted long-context encoder for biomedical and clinical NLP, addressing a key limitation of earlier encoders when processing extended conversations. When paired with T5's unified text-to-text framework (Raffel et al., 2019), this approach showed promise for joint classification and generation. Nevertheless, these architectures demanded substantial computational resources for fine-tuning and exhibited inconsistent narrative fidelity when extracting complex order descriptions.

The latest advances in medical-specific large-language models marked another evolutionary step. MedGemma (Google Health AI, 2025) and Lingshu (Team et al., 2025) incorporated extensive clinical knowledge through specialized pre-training on medical corpora. Despite theoretical advantages, practical deployment proved challenging for teams with limited resources—both models experienced out-of-memory errors on commodity hardware even with quantization strategies.

Our approach fundamentally departs from this fine-tuning paradigm. Rather than pursuing increasingly specialized models, we demonstrate that general-purpose LLMs like Gemini 2.5 Pro (DeepMind, 2025), trained on Trillion of tokens when guided by meticulously engineered prompts incorporating chain-of-thought reasoning and self-critique, surpass domain-specific models without any training. This finding aligns with emerging evidence that sophisticated prompt engineering can activate latent clinical reasoning capabilities in large language models, offering a computationally efficient alternative to traditional fine-tuning approaches.

3 System Description

3.1 Model Selection Journey

Our development process involved systematic exploration of increasingly sophisticated approaches, each informing the final solution. This iterative journey revealed crucial insights about the trade-offs between model specialization and prompt engineering.

Early Phase: Rule-based classification attempts. Initial experiment with BioBERT for order-type classification– yielded high precision but missed many orders (recall issues). Attempts to explore joint classification and generation at attribute extraction using DeBERTa-v3 (He et al., 2021). However, the extracted spans were inconsistent, often capturing partial information or including conversational artifacts.

Middle Phase: Joint modeling approach combining a long-text encoder BioClinical ModernBERT with text decoder to handle long transcripts and generate structured output. Resulted in improved order identification but narrative fidelity was low (the generated text did not closely match the conversation details).

Transition: Trials with cutting-edge medical LLMs MedGemma 4B and Lingshu 7B for end-to-end generation. These models promised better medical knowledge and reasoning but were impractical to run on available hardware (out-of-memory issues on Google Colab's free tier, even with 4-bit quantization).

Final Phase: Pivot to a pure prompt-engineering approach using an accessible large model Gemini 2.5 Pro. No fine-tuning, no task-specific training – instead, harness the model's reasoning ability through carefully crafted prompts. This phase yielded a breakthrough: a *+13* F1 point jump in average score over the best fine-tuned attempt, achieving our best results with minimal infrastructure.

3.2 Final LLM-Based Approach

3.2.1 Core Architecture

Our system leverages Gemini 2.5 Pro as the primary inference engine, though the approach generalizes to other large language models including Mistral Medium (Mistral AI, 2025), Qwen3 (Yang et al., 2025). The key innovation lies not in model selection but in the prompt engineering methodology that transforms a general-purpose LLM into a specialized medical order extractor.

Model flexibility. Although instantiated with Gemini 2.5 Pro, the procedure is model-agnostic. Comparable API-served models (e.g., Mistral Medium and Qwen3) can be used with the same prompt structure and post-processing, subject to their context limits and decoding controls.

3.2.2 Prompt Engineering Strategy

The prompt design follows a three-stage cognitive workflow that mirrors clinical reasoning processes:

Stage 1: Chain-of-Thought Analysis. The model first ingests the entire transcript to build contextual understanding. It then performs a chronological sweep, identifying potential orders through explicit doctor statements. Each candidate undergoes systematic evaluation against definitive order criteria, distinguishing actionable orders from tentative recommendations or general advice.

Stage 2: Self-Critique and Verification. Before generating output, the model conducts mandatory self-auditing. This includes schema adherence checking, provenance integrity verification, redundancy elimination, and completeness assessment. If discrepancies are detected, the model must restart its analysis, ensuring only validated orders reach the output stage.

Stage 3: Deterministic JSON Generation. The final stage produces structured JSON output with strict schema compliance. Each order contains four mandatory fields: $order\_type$ (constrained to "medication", "lab", "imaging", or "follow-up"), $description$ (concise clinical summary), $reason$ (medical justification), and $provenance$ (turn IDs providing evidence).

The complete prompt implementation (Listing 1 in Appendix A) underwent 12 iterations, each addressing specific failure modes identified through development set analysis. Early versions struggled with multi-order turns and implicit reasons, leading to the incorporation of explicit handling rules. The final prompt incorporates explicit rules handling edge cases: multi-order turns generate separate order objects; implicit reasons are extracted from surrounding context; continuation of existing treatments and conditional orders are excluded. A deterministic seed ensures reproducible outputs across runs.

3.2.3 Prompt Components

The prompt was structured with several key components to guide the model's reasoning process:

- **Role Directive:** The prompt began with a role-setting instruction (e.g., "You are a deterministic, expert-level clinical information extraction engine...") to establish the model's persona and enforce strict adherence to instructions.

- **Definitions:** It provided precise definitions for each required field (`order_type`, `description`, `reason`, `provenance`) and specified the acceptable values or format for each.

- **Rules for Valid Orders:** A set of explicit rules (R1–R6) was enumerated to guide the model's judgment. These rules covered edge cases such as ignoring tentative or hypothetical statements, excluding patient-suggested actions unless confirmed by the doctor, and avoiding duplicate orders.

- **Step-by-step Workflow:** The prompt enforced a structured, internal chain-of-thought process. The model was required to log its steps: first, scanning the entire transcript; second, gathering evidence from specific dialogue turns; third, extracting candidate orders; fourth, validating each candidate against the rules (marking it as "VALID" or "INVALID" with justification); and finally, constructing the structured output only from validated candidates. This workflow significantly improved the model's precision.

- **Example and Format:** A template example of a perfect JSON output was included to demonstrate the exact required format, minimizing structural errors in the final generation.

4 Experiments and Results

4.1 Dataset Description

The MEDIQA-OE 2025 shared tasks dataset is derived from the SIMORD corpus (Corbeil et al., 2025a) and provided transcripts were from two complementary sources. ACI-Bench (Ouyang et al., 2023) contributed naturalistic clinical encounters captured without virtual assistant intervention, preserving the authentic flow of doctor-patient interactions. PriMock57 (Korfiatis et al., 2022) added 57 mock primary care consultations with professionally transcribed dialogues and corresponding clinical notes.

The dataset exhibited significant class imbalance. Training data contained 63 encounters with 143 orders, while development data included 100 encounters with 255 orders. Medication orders dominated (52.4% in training, 45.9% in development), followed by laboratory tests, follow-up appointments, and imaging studies. This distribution reflects typical primary care patterns, where medication management and routine testing predominate.

4.2 Evaluation Metrics

The shared task employed four complementary metrics capturing different aspects of extraction quality:

- **Description ROUGE-1 F1** measures lexical overlap between extracted and reference order descriptions, evaluating the model's ability to capture key clinical terms while maintaining conciseness.
- **Reason ROUGE-1 F1** assesses medical justification extraction, requiring models to identify not just what was ordered but why, often from dispersed conversational context.
- **Order Type Strict F1** evaluates categorical classification accuracy across the four order types, penalizing any deviation from the exact category labels.
- **Provenance Multi-Label F1** measures evidence attribution precision, validating whether extracted orders correctly reference supporting transcript turns.

The final leaderboard score averages these four metrics equally, balancing lexical accuracy, semantic understanding, classification precision, and evidence grounding.

4.2.1 Metric Computation Details

Based on the official evaluation script, each metric is computed as follows:

Description F1 averages three sub-metrics: Match (binary presence), Strict (exact string match), and ROUGE-1 (unigram overlap):

$$\text{Desc}_{\text{F1}} = \frac{1}{3}(\text{Match}_{\text{F1}} + \text{Strict}_{\text{F1}} + \text{R1}_{\text{F1}}) \quad (1)$$

ROUGE-1 F1 (for both description and reason) computes unigram precision and recall after preprocessing (lowercase, punctuation removal):

$$\text{ROUGE-1}_{\text{F1}} = \frac{2 \cdot P \cdot R}{P + R} \quad (2)$$

Order Type Strict F1 counts exact categorical matches across four types (medication, lab, imaging, follow-up).

Provenance Multi-Label F1 treats turn IDs as multi-label sets:

$$\begin{aligned} \text{Prov}_{\text{F1}} &= \frac{2 \cdot P_{\text{prov}} \cdot R_{\text{prov}}}{P_{\text{prov}} + R_{\text{prov}}} \\ P_{\text{prov}} &= \frac{|T_{\text{pred}} \cap T_{\text{ref}}|}{|T_{\text{pred}}|} \end{aligned} \quad (3)$$

Average Score:

$$\text{Score}_{\text{avg}} = \frac{1}{4}(\text{Desc}_{\text{F1}} + \text{Rea}_{\text{F1}} + \text{Type}_{\text{F1}} + \text{Prov}_{\text{F1}}) \quad (4)$$

Orders are paired using the Hungarian algorithm maximizing description similarity, with unpaired orders penalized in precision/recall calculations.

Implementation details can be found at organizer github repository: https://github.com/jpcorb20/mediqa-oe

4.3 Results and Analysis

The MEDIQA-OE 2025 shared task attracted participation from **17 teams**, producing a total of **105 submissions**. Our system achieved second place on the public leaderboard shown in Table 1.

The results reveal interesting performance patterns. As shown in Table 1, our system excelled at reason extraction (0.4130), surpassing the first-place team by 40%. The computation of these metrics follows Equations 1–4, ensuring consistent evaluation across all submissions.

4.4 Qualitative Comparative Analysis

To understand the strengths and weaknesses of our approach versus fine-tuned models, we conducted a detailed error analysis on 50 randomly sampled encounters from the development set, examining 187 total orders which is summarized in Table 2, reveals distinct performance patterns between approaches.

Our zero-shot approach particularly excels at: (1) **Multi-sentence reasoning**: Successfully connecting orders with reasons stated 3-5 sentences apart, leveraging the LLM's context window; (2) **Implicit justifications**: Inferring medical reasons from conversational context without explicit linking phrases; (3) **Complex medication orders**: Accurately extracting multi-component dosage instructions with temporal modifications; (4) **Structured follow-ups**: Extracting standardized follow-up patterns learned from it's own knowledge base;

Fine-tuned models perform better on: (1) **Domain-specific abbreviations**: Recognizing

Team	Description ROUGE-1 F1	Reason ROUGE-1 F1	Order Type Strict F1	Provenance Multi-Label F1
1st Place	0.6677	0.2949	0.8145	0.6304
Ours	**0.6406**	**0.4130**	**0.7474**	**0.6044**
3rd Place	0.5799	0.3564	0.7156	0.4838

Table 1: Public leaderboard scores for MEDIQA-OE 2025. Description uses Equation 1; Reason uses Equation 2; Provenance uses Equation 3; Average using Equation 4.

Order Characteristic	Zero-Shot (Ours)	Fine-Tuned (Baseline)
Multi-sentence reasoning	**92%**	71%
Implicit justifications	**88%**	62%
Complex dosage extraction	**85%**	79%
Handling ambiguity	**81%**	68%
Follow-up specifications	**84%**	68%
Lab test abbreviations	72%	**75%**
Imaging details	76%	**77%**

Table 2: Comparative performance on specific order characteristics. Percentages indicate successful extraction accuracy.

specialized lab test acronyms; (2) **Imaging protocols**: Identifying specific imaging modalities and contrast specifications.

Representative examples illustrate the contrasting behaviors:

Example 1: Cross-sentence reasoning (Our approach succeeds)

```
Doctor: "Your blood pressure is still elevated.
        [3 sentences of discussion]. Let's start
        you on lisinopril 10mg daily."
```

Our system correctly links "elevated blood pressure" as the reason despite the intervening sentences. The fine-tuned model extracted "lisinopril 10mg daily" but marked reason as null.

Example 2: Lab panel abbreviations (Fine-tuned succeeds)

```
Doctor: "I'll order a CMP, CBC with diff, and TSH."
```

Fine-tuned model correctly generates three separate lab orders. Our approach incorrectly merged them into a single order: "CMP, CBC with diff, and TSH".

The pattern suggests our approach excels at leveraging broader context and implicit reasoning, while fine-tuned models better handle domain-specific conventions learned from training data. This aligns with the LLM's strength in general reasoning versus the encoder-decoder's pattern memorization.

Configuration	Avg F1	Δ	Std Dev
Full System	**0.6014**	—	0.0023
Component Removals:			
- w/o Examples	0.5482	-0.0532	0.0037
- w/o Chain-of-Thought	0.5178	-0.0836	0.0041
- w/o Self-Verification	0.5721	-0.0293	0.0029
- w/o Edge Case Rules	0.5789	-0.0225	0.0031
- w/o JSON Schema	0.5623	-0.0391	0.0045
Prompt Structure Variants:			
- Minimal Instructions	0.4723	-0.1291	0.0052
- No Role Definition	0.5843	-0.0171	0.0028
- Single-Stage Process	0.5392	-0.0622	0.0039

Table 3: Ablation results (mean of 3 runs). Statistical significance tested via paired bootstrap ($p < 0.05$ for all except deterministic seed).

4.5 Ablation Studies

We systematically ablated prompt components to understand their individual contributions presented in Table 3. Each variant was evaluated on the full development set with three random seeds.

Key findings from ablations:

Examples are critical: Removing the input-output example causes the second largest performance drop (5.3%). Without examples, the model frequently merges multiple orders and inconsistently formats the provenance field.

Chain-of-thought provides structured reasoning: The largest drop 8.4% without CoT primarily affects reason extraction (drops from 0.4130 to 0.3421) and multi-turn order handling.

Self-verification catches a lot of errors: Manual inspection revealed that without self-verification, 23% of outputs had violations (missing fields, incorrect types, output structure) that weren't caught.

Diminishing returns on complexity: Adding medical persona or conversational tone showed minimal gains, suggesting the model already activates medical knowledge through the task description.

5 Discussion

Our results demonstrate that sophisticated prompt engineering can match or exceed traditional fine-tuning approaches for medical order extraction. The success of this zero-shot strategy challenges prevailing assumptions about the necessity of domain-specific training for clinical NLP tasks.

The system's exceptional performance on reason extraction (0.4130 ROUGE-1 F1) merits particular attention. While competing approaches struggled to connect orders with their medical justifications—often stated sentences apart in natural conversation—our chain-of-thought prompting successfully maintained contextual threads throughout lengthy transcripts. This capability suggests that LLMs possess latent clinical reasoning abilities that can be activated through appropriate prompting rather than requiring explicit training.

The slightly lower performance on description extraction compared to the first-place team reveals an interesting trade-off. Our prompt emphasized extracting verbatim clinical details while avoiding conversational artifacts, occasionally resulting in overly concise descriptions that missed scoreable tokens. Fine-tuned models, trained on specific annotation guidelines, may better calibrate their extraction granularity to match evaluation metrics.

Several limitations warrant consideration. First, the approach depends on API availability and pricing models of commercial LLMs, potentially limiting deployment in settings with restricted internet access or budget constraints. To address API dependency concerns we may utilize open-source models like Mistral Medium, Qwen3, etc. Second, prompt engineering or prompt optimization requires iterative refinement and domain expertise to achieve optimal performance, though this investment is one-time rather than per-dataset. Third, the deterministic generation strategy, while ensuring reproducibility, may miss valid alternative order interpretations that a probabilistic approach might capture.

Future work should explore several directions. Ensemble approaches combining multiple LLMs like 'LLM-as-a-judge' could improve robustness. Prompt optimization techniques, including automated prompt search, might discover more effective instruction formulations. Finally, human-in-the-loop workflows could leverage the model's self-critique capability to flag low-confidence extractions for review.

6 Conclusion

This paper presented a paradigm shift in medical order extraction, demonstrating that zero-shot prompt engineering with large language models can achieve performance competitive with complex fine-tuned systems. Our second-place finish in the MEDIQA-OE 2025 shared task, with an average F1 score of 0.6014, validates this approach's effectiveness while highlighting its practical advantages: no training data requirements, GPU-free deployment, and immediate applicability across clinical settings.

The key technical contribution—a structured prompt combining chain-of-thought reasoning with self-critique and deterministic generation—offers a template for similar clinical NLP tasks. Our results suggest that the future of medical information extraction may lie not in increasingly specialized models but in more sophisticated ways of eliciting knowledge from general-purpose language models.

As healthcare systems worldwide grapple with documentation burden and the need for accurate clinical information capture, approaches that minimize technical barriers while maintaining high performance become increasingly valuable. Our work demonstrates that such solutions are not only possible but can rival state-of-the-art alternatives, potentially accelerating the adoption of AI-assisted clinical documentation tools where they are needed most.

References

Jean-Philippe Corbeil, Asma Ben Abacha, George Michalopoulos, Phillip Swazinna, Miguel Del-Agua, Jérôme Tremblay, Akila Jeeson Daniel, Cari Bader, Yu-Cheng Cho, Pooja Krishnan, Nathan Bodenstab, Thomas Lin, Wenxuan Teng, Francois Beaulieu, and Paul Vozila. 2025a. Empowering healthcare practitioners with language models: Structuring speech transcripts in two real-world clinical applications. *CoRR*.

Jean-Philippe Corbeil, Asma Ben Abacha, Jérôme Tremblay, Phillip Swazinna, Akila Jeeson Daniel, Miguel Del-Agua, and François Beaulieu. 2025b. Overview of the mediqa-oe 2025 shared task on medical order extraction from doctor-patient conversations. In *Proceedings of the 7th Clinical Natural Language Processing Workshop*. Association for Computational Linguistics.

DeepMind. 2025. Gemini 2.5 pro. https://deepmind.google/models/gemini/pro/.

Google Health AI. 2025. Medgemma 4b model card. https://developers.google.com/health-ai-developer-foundations/medgemma/model-card.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *CoRR*, abs/2111.09543.

Nikolaos Korfiatis et al. 2022. Primock57: A dataset of primary care mock consultations. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 783–791.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *CoRR*, abs/1901.08746.

Mistral AI. 2025. Models overview. https://docs.mistral.ai/getting-started/models/models_overview/.

Dan Ouyang et al. 2023. Aci-bench: a novel ambient clinical intelligence dataset for clinical workflow analysis and conversational ai. *JAMIA open*, 6(3):ooad062.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

Thomas Sounack, Joshua Davis, Brigitte Durieux, Antoine Chaffin, Tom J. Pollard, Eric Lehman, Alistair E. W. Johnson, Matthew McDermott, Tristan Naumann, and Charlotta Lindvall. 2025. Bioclinical modernbert: A state-of-the-art long-context encoder for biomedical and clinical nlp.

LASA Team, Weiwen Xu, Hou Pong Chan, Long Li, Mahani Aljunied, Ruifeng Yuan, Jianyu Wang, Chenghao Xiao, Guizhen Chen, Chaoqun Liu, Zhaodonghui Li, Yu Sun, Junao Shen, Chaojun Wang, Jie Tan, Deli Zhao, Tingyang Xu, Hao Zhang, and Yu Rong. 2025. Lingshu: A generalist foundation model for unified multimodal medical understanding and reasoning.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. 2025. Qwen3 technical report.

Acknowledgements

I would like to express our gratitude to the organizers of the MEDIQA-OE 2025 shared task for creating a valuable benchmark and fostering research in clinical NLP. We also thank the creators of the ACI-Bench and PriMock57 datasets for providing the high-quality data that made this work possible. This research was facilitated by access to the Gemini 2.5 Pro model, and Google Colab and we acknowledge the developers for making such powerful tools accessible to the research community.

A Full System Prompt

The complete prompt provided to Gemini 2.5 Pro is detailed below. Hyperparameters: temperature=0.1, top_p=0.97, seed=42. This prompt was engineered to enforce a deterministic, multi-step reasoning process for accurate medical order extraction.

Listing 1: The full prompt used for the Gemini 2.5 Pro model.

```
You are a deterministic, expert-level clinical information extraction engine. Your sole function is to receive a JSON object
      representing a medical encounter and return a JSON object containing extracted medical orders with zero defects. You must
      operate as a state machine, following a fixed workflow with the highest level of precision and strictly adhere to all
      instructions. Failure to adhere to the output format is not an option. Deviating from these instructions is a protocol
      violation.

### Core Directive ###
Analyze the provided transcript and extract all medical orders. An order is defined by four attributes: order_type, description,
      reason, and provenance.

### Attribute Definitions ###
  - order_type: (String) MUST be one of four exact strings: "medication", "lab", "imaging", "follow-up".
  - description: (String) The specific service or product ordered. This should be a direct, non-conversational summary. Extract
        verbatim details like dosage, frequency, and location. For example, from "I'm going to prescribe some Lasix, 40 milligrams
         a day," the description is "lasix 40 milligrams a day". Another example, from "increase lasix from twenty milligrams to
        sixty milligrams for the next four days", the description is "lasix sixty milligrams four days pill". Another example,
        from "use albuterol and atrovent inhalers", the order is repeated twice having one orders description as "albuterol" and
        the other orders description as "atrovent inhalers".
  - reason: (String) The medical justification for the order. This should also be a direct summary. For "For your shortness of
        breath... I want to... put you on some Lasix," the reason is "shortness of breath". If no reason is explicitly stated, use
         the most relevant diagnosis mentioned in connection with the order else an empty string "".
  - provenance: (List of Integers) A JSON list of integer turn_ids. These turns are the absolute proof for the extracted order.
        Every piece of information (type, description, reason) must be traceable to the turn_ids listed here.

### Processing Workflow ###
Execute the following nine-step process. This entire process must be logged within tags before the final JSON output. This log is
      a mandatory component of the operation.
  1. Context Ingestion: Read, Scan and Analyze the entire transcript first to build a complete contextual model of the encounter.
  2. Evidence Gathering: Identify and list all turn_id where potential order candidates stated by doctor or any turn where a
        doctor issues a command or action plan.
  3. Chronological Sweep & Extraction: Iterate through the evidence gathered one by one.
      - Focus exclusively on the "DOCTOR" speaker. Orders are only valid if stated or confirmed by the doctor.
      - Apply the "Definitive Order" Test:
          a. EXTRACT: Clear, direct, undeniable statements of action. (e.g., "I am ordering...", "We will get a...", "I'm going to
              prescribe...", "Make sure you schedule...").
          b. IGNORE: Tentative, conditional, recommended actions or exploratory language. (e.g., "We could think about...", "An
              option might be...", "If it gets worse, we might need...", "we might consider...", "I'd recommend...").
          c. IGNORE: Orders mentioned only by the PATIENT and not confirmed by the DOCTOR.
          d. IGNORE: General advice that is not a specific order (e.g., "You should drink more water").
          e. IGNORE: If a phrase is ambiguous, and it is not a specific, actionable order (e.g., "we need to watch your blood
              pressure...").
          f. IGNORE: Continuations of existing treatments (e.g., "continue taking...", "continue on medication...").
          g. IGNORE: If needed order (e.g., "use medication if needed...", "take medication only as needed for...", "take this
              medication which is stronger than medication only if needed...").
      - Handle Multi-Order Turns: If a single turn contains multiple distinct orders or actions, generate a separate order object
            for each.
  4. Candidate Auditing: For each candidate, audit it against the Core Directives. State explicitly whether it is VALID or INVALID
         and provide a brief justification referencing the rule violated (R1, R2, etc) or not meeting the validation based on the
        JSON Order Schema. This analysis is mandatory.
      - Example Invalid Justification: "INVALID: Violates Rule R2 - Conditional Language."
      - Example Invalid Justification: "INVALID: Violates Rule R3 - This is an instruction for the scribe, not the patient."
  5. Data Structured Extraction: For each VALID candidate identified, systematically extract the four fields and construct the
        order object with meticulous adherence to the JSON Order Schema and populate those four fields.
  6. Mandatory Final Quality (Self-Correction): Before generating the output, Perform a final check on all your extracted valid
        orders. conduct this final check:
      - Schema Adherence: Is every field present and correctly typed in every order object?
      - Provenance Integrity: Read the text at the provenance turn(s). Does it unambiguously support the extracted description and
             order_type? Is reason set to null when no explicit justification was given? Is every single order from the transcript
             captured?
      - Redundancy Check: Is every single order from the transcript captured? Is the same order listed multiple times? Consolidate
             if necessary into the most complete description.
      - Completeness Check: Confirm that no valid orders have been missed.
      - JSON Syntax Validation: Is the final string a single, perfectly formed JSON object? Ensure they are complete, correct, and
             fully compliant with all directives?
  7. Verification Protocol: If any check fails, you must restart and redo from start and correct your draft JSON along and re-
        verify. Log any corrections made during this audit. If no corrections are needed, state "Integrity audit passed."
  8. Final JSON Assembly: Assemble the audited, corrected data into the final, single JSON object according to the JSON Order
        Schema. This JSON object is the only and final output of your response final JSON for output.

### Critical Rules & Edge Cases ###
  - (R1) No Orders Rule: If the transcript contains no identifiable medical orders, the value for the encounter id key MUST be an
        empty list: [].
  - (R2) Multiple Orders in One Turn Rule: If a single turn contains multiple distinct orders, create a separate order object for
        each one. The turn_id can be reused in the provenance for each of these orders.
  - (R3) Implicit Reasons Rule: If a reason is not stated in the same sentence as the order, look at the immediately preceding
        sentences in the conversation for the relevant diagnosis or justification.
  - (R4) Do Not Infer Rule: Do not invent orders or reasons that are not supported by the text. If you cannot find a piece of
```

```
       information for a field, you must do your best to populate it with the closest available information. All fields are
       mandatory.
  - (R5) No-Hallucination Rule: Do not infer, add, or embellish any information not explicitly present in the transcript. The
       extraction must be a literal representation of the doctor's plan.
  - (R6) JSON Rule: The JSON object's key is the encounter_id, and its value is a list of order objects. Your final output must be
        the JSON object and nothing else. No introductory text, no apologies, no explanations.

### JSON Order Schema ###
  - order_type: (String) The high-level clinical category. It must be one of: "medication", "lab", "imaging", "follow-up".
  - description: (String) The formal, clean, accurate and most concise non-conversational summary or action of the order excluding
        conversational filler. Contains only 1 thing. If number are digits then digits else words.
  - reason: (String) The direct, concise, explicit stated medical justification for the order. If no reason is explicitly stated
       in the transcript before or after the order for that specific order, then it must be null. Do not infer or guess a reason
       from general context. Do not alter or paraphrase or phrase or change a reason. Keep it same as in the transcription. Short
        phrase the reason.
  - provenance: (List of Integers) A list of the turn_id(s) that provide the most direct and concise evidence for the order.

### Example of Perfection ###
Input:
$$$
{
    "id": "acibench_D2N122_aci_clinicalnlp_taskB_test1",
    "transcript": [
        { "turn_id": 2, "speaker": "PATIENT", "transcript": "...they did that chest x-ray...and they found this lung nodule...
              referred me here to you..." },
        { "turn_id": 27, "speaker": "DOCTOR", "transcript": "...you do have an incidentally found right upper lobe lung nodule...
              I'm also going to schedule a pet ct this is gon na help to determine if that nodule is metabolically active... for
              your secondary concern of your rheumatoid arthritis i want you to continue to follow up with your rheumatologist..."
              }
    ]
}
$$$

Your Required Output:
$$$
{
    "acibench_D2N122_aci_clinicalnlp_taskB_test1": [
        {
            "order_type": "imaging",
            "description": "pet ct",
            "reason": "to determine if that nodule is metabolically active",
            "provenance": [
                2,
                27
            ]
        },
        {
            "order_type": "follow-up",
            "description": "follow up with your rheumatologist",
            "reason": "rheumatoid arthritis",
            "provenance": [
                27
            ]
        }
    ]
}
$$$
```