LoRA Fine-Tuning of Qwen2.5-0.5B-Instruct for Symptom Polarity Classification in Chinese Medical Dialogue

PEIYUAN HAN, University of California, San Diego, USA

This project explores the use of parameter-efficient fine-tuning to adapt a lightweight instruction-following language model—Qwen2.5-0.5B-Instruct—to the domain of Chinese medical dialogue, with a focus on symptom polarity classification. Motivated by the need for privacy-preserving, locally deployable AI tools in under-resourced healthcare environments, we apply LoRA on a filtered subset of the PromptCBLUE benchmark. We demonstrate improvements in structure alignment, instruction adherence, and domain-specific output quality compared to the original base model.

CCS Concepts: • **Computing methodologies** → **Natural language processing**; *Machine learning*.

Additional Key Words and Phrases: LoRA, instruction tuning, medical NLP, Qwen2.5, PromptCBLUE, Chinese medical dialogue

ACM Reference Format:
Peiyuan Han. 2025. LoRA Fine-Tuning of Qwen2.5-0.5B-Instruct for Symptom Polarity Classification in Chinese Medical Dialogue. In *Final Project Report, UCSD ECE 284, Spring 2025*. ACM, New York, NY, USA, Article 111, 8 pages. https://doi.org/10.1145/XXXXXXX.XXXXXXX

1 Motivation

Despite the rapid advancements of large language models (LLMs) in natural language processing (NLP), their real-world application in healthcare remains limited. Leading models such as GPT-4o, Claude, and Med-PaLM have demonstrated strong capabilities across tasks like entity recognition, summarization, and question answering. However, these models are typically proprietary, cloud-based, and resource-intensive, making them unsuitable for deployment in under-resourced clinical settings. Furthermore, transmitting patient information to external servers raises severe privacy and compliance concerns.

Existing open-source Chinese medical LLMs, such as ChatMed and FLAT-LLaMA, typically exceed 7B parameters and require multi-GPU setups, creating a practical gap between academic advances and real-world feasibility. This gap is particularly prominent in community hospitals and rural clinics, where access to high-end compute infrastructure is rare.

To address this challenge, we explore a lightweight and privacy-preserving alternative based on `Qwen2.5-0.5B-Instruct`, an open-source, instruction-tuned model with only 0.5B parameters. By applying Low-Rank Adaptation (LoRA), we efficiently fine-tune this model on a carefully selected subset of `PromptCBLUE`—a benchmark suite for Chinese medical NLP tasks.

While `PromptCBLUE` contains 11 heterogeneous tasks, our preliminary experiments showed that multi-task fine-tuning led to inconsistent convergence and format instability. To ensure structural consistency and evaluation reliability, we focus on a single, clinically meaningful task: **symptom polarity classification**. This task involves identifying named symptoms from multi-turn doctor-patient dialogues and assigning each a polarity label (*positive*, *negative*, or *uncertain*), thus closely resembling real-world diagnostic workflows.

By restricting the task scope and introducing an instruction-driven fine-tuning strategy with prompt masking, we demonstrate how a 0.5B general-purpose model can be transformed into a structure-aware, domain-specialized assistant.

2 Related Work

Recent advances in large language models (LLMs) have enabled impressive performance across a variety of medical natural language processing tasks. Notable examples include Med-PaLM [Singhal et al. 2023], which fine-tunes PaLM for USMLE-style question answering, and GPT-4, which has demonstrated zero-shot competence in medical reasoning. However, these models are either proprietary or resource-intensive, making them inaccessible for local deployment in many real-world settings.

In China, the open-source community has made substantial progress in developing Chinese medical LLMs. The *ChatMed* project introduces a series of instruction-tuned models trained on Chinese medical dialogues and knowledge graphs. The *ChatMed-Consult* model is based on over 500K online doctor-patient consultations, while *ChatMed-TCM* focuses on traditional Chinese medicine using an entity-centric self-instruct strategy. These models leverage the LLaMA backbone and utilize LoRA-based parameter-efficient fine-tuning to reduce hardware costs.

Our work complements these efforts by focusing on a smaller yet flexible model, *Qwen2.5-0.5B-Instruct*, and adapting it to a structured subtask using *PromptCBLUE*. PromptCBLUE [Zhu et al. 2022] is a comprehensive Chinese medical NLP benchmark covering 11 distinct tasks, including named entity recognition, relation extraction, sentence similarity, and report generation. Prior work using PromptCBLUE often targets full-task leaderboard optimization. For example, FLAT-LLaMA [Sun et al. 2023] fine-tunes LLaMA on multiple tasks using prompt-formatted examples, and MedPinyin [Zhang et al. 2023] leverages phonetic embeddings to boost medical QA performance across PromptCBLUE tasks.

Instead of optimizing across all tasks, we adopt a task-specialized perspective by focusing on symptom polarity classification, a structured classification task with high clinical interpretability. This allows us to more closely examine how a small, general-purpose LLM can be adapted into a reliable assistant under low-resource settings.

Author's Contact Information: Peiyuan Han, University of California, San Diego, La Jolla, California, USA, p3han@ucsd.edu.

Final Project Report, University of California, San Diego

ACM ISBN 978-x-xxxx-xxxx-x/25/06
https://doi.org/10.1145/XXXXXXX.XXXXXXX

From the methodological side, we leverage *LoRA (Low-Rank Adaptation)* [Hu et al. 2022], a parameter-efficient fine-tuning method that freezes the base model and injects trainable rank-decomposed matrices into selected layers, typically the query and value projections in self-attention. Compared to earlier adapter-based methods such as *Houlsby Adapters* [Houlsby et al. 2019], which insert full bottleneck modules after each Transformer block, LoRA introduces significantly fewer parameters and has been widely adopted in the PEFT ecosystem [Dettmers et al. 2023]. LoRA allows domain-specific tuning using small datasets without overfitting or catastrophic forgetting, making it ideal for adapting large models to medical tasks.

In this work, we combine the open instruction-following capability of Qwen with the structure-awareness enabled by LoRA. While ChatMed focuses on instruction generation and pretraining, our approach starts from a pretrained instruction model and emphasizes strict format alignment, evaluation on structured prompts, and targeted adaptation to medically relevant outputs.

3 Project Aim

This project investigates whether a compact instruction-tuned language model—*Qwen2.5-0.5B-Instruct*—can be effectively adapted to perform structured clinical reasoning under real-world constraints, such as limited GPU capacity and strict data privacy requirements in medical environments.

Our initial plan was to fine-tune the model on all 11 tasks from the PromptCBLUE benchmark, aiming to develop a general-purpose Chinese medical assistant. However, early experiments exposed key challenges: inconsistent prompt formats, cross-task interference, and unstable convergence during training. To ensure more interpretable and reliable evaluation, we narrowed our focus to a single representative task: **symptom polarity classification**. This task maintains the core challenge of clinical understanding while allowing for structured evaluation and clear supervision signals.

Specifically, our project pursues the following goals:

- **Unified data structuring**: Convert heterogeneous PromptCBLUE samples into a consistent instruction-tuning format comprising three explicit fields: `instruction`, `input`, and `output`;
- **Efficient model adaptation**: Apply LoRA (Low-Rank Adaptation) to fine-tune only a small fraction of the model parameters, enabling training on consumer-grade hardware (e.g., a single 4070 GPU) while preserving model capacity;
- **Structured output generation**: Train the model to generate clearly formatted outputs (e.g., `symptom: label`) and compare performance against the zero-shot base model;
- **Failure mode analysis**: Identify and categorize common prediction errors—including label drift, repetition, and boundary misidentification—to inform downstream improvements;
- **Task specialization for offline use**: Demonstrate that targeted adaptation on a narrow but clinically relevant task can unlock the utility of small-scale LLMs in privacy-critical and resource-constrained healthcare scenarios.

Ultimately, this work serves as a proof of concept for how domain-aligned prompt design and lightweight fine-tuning strategies can transform a general-purpose LLM into a reliable, structured-output medical assistant—even when deployed fully offline.

4 Methodology

4.1 Overview

Our methodology comprises four key stages: (1) **task curation and preprocessing**, (2) **instruction-style prompt formatting**, (3) **parameter-efficient fine-tuning using LoRA**, and (4) **model merging and structured inference**. Together, these steps aim to adapt a general-purpose instruction-following model—*Qwen2.5-0.5B-Instruct*—into a lightweight, domain-aligned assistant capable of performing symptom polarity classification with high structure fidelity and minimal computational overhead.

By isolating a single representative task from the PromptCBLUE benchmark and aligning both input and output formats with the model's instruction-tuning paradigm, we minimize domain shift and maximize training efficiency. The use of LoRA enables us to fine-tune only a small subset of parameters, making the pipeline suitable for deployment in resource-constrained environments. At inference time, we preserve prompt consistency to ensure reliable and format-compliant generation.

4.2 Backbone Architecture: Qwen2.5-0.5B

We use *Qwen2.5-0.5B-Instruct* as our base model for instruction tuning. It is a decoder-only Transformer with 24 layers and approximately 500 million parameters, designed to support long-context generation (up to 32,768 tokens) while remaining lightweight enough for fine-tuning on consumer-grade GPUs.

Qwen incorporates several architectural features optimized for efficiency and expressive power, including Grouped Query Attention (GQA) to reduce memory usage, Rotary Positional Embedding (RoPE) for long-range context modeling, RMSNorm for stable training, and SwiGLU in the feedforward layers to enhance non-linear capacity. Each input token is represented as a 896-dimensional vector, which flows consistently through all layers.

These design choices make Qwen a compelling backbone for parameter-efficient instruction tuning. For reproducibility, a detailed breakdown of its architecture, token shape transformations, and attention mechanism is provided in Appendix **??**.

4.3 LoRA-based Instruction Tuning on Qwen

To adapt Qwen2.5-0.5B-Instruct to our clinical instruction task, we employ *Low-Rank Adaptation (LoRA)* [Hu et al. 2022], a parameter-efficient fine-tuning technique that inserts small trainable matrices into selected submodules while freezing the original model weights. This enables fast adaptation on limited hardware with minimal overfitting.

Adapter Injection. We inject LoRA into the self-attention modules (`q_proj`, `k_proj`, `v_proj`, `o_proj`) across all Transformer blocks. Each adapter adds a residual low-rank update $\Delta W = BA$, with $A \in \mathbb{R}^{r \times d_{\text{in}}}, B \in \mathbb{R}^{d_{\text{out}} \times r}$, scaled by a factor α/r. We use $r = 8$, $\alpha = 16$, and apply dropout $p = 0.05$ to improve generalization.

Training Setup. We format each training sample as a flattened sequence: `instruction + input + output`, and truncate to 1024 tokens. The instruction and input region is label-masked (set to `-100`) so that loss is computed only on the target output. We train for 3 epochs using the Huggingface `Trainer` with `bfloat16` precision,

a batch size of 2 (gradient accumulation = 8), AdamW optimizer with learning rate 2×10^{-4}, and weight decay 0.01. All experiments run on a single NVIDIA RTX 4070 GPU.

Model Export and Inference. After training, we merge the LoRA weights into the frozen base via `merge_and_unload()`, yielding a standalone checkpoint for inference. This ensures compatibility with Qwen runtime and avoids any additional LoRA dependencies at deployment time.

Efficiency and Impact. This configuration updates fewer than 1% of model parameters yet achieves substantial gains in structured output alignment and instruction-following accuracy (see Section 5). The low-rank design enables cost-effective adaptation, allowing us to run experiments end-to-end within 4 GPU hours for the full tasks and only 0.5 GPU hours for the symptom polarity classification.

Table 1. Comparison of training strategies: full finetuning vs. LoRA

Method	Trainable Params	GPU Memory
Full Finetune	500M (100%)	12 GB
LoRA (Ours)	0.8M (<1%)	4.8 GB

4.4 Early Attempts and Failure Modes

Before arriving at our final training pipeline, we explored several alternative strategies that proved suboptimal in practice. These pilot experiments helped us identify the key failure modes in adapting a general-purpose instruction model to complex medical NLP benchmarks.

Multi-task Fine-tuning on Full PromptCBLUE.. Our initial approach attempted to fine-tune Qwen2.5-0.5B-Instruct on all 11 tasks from the PromptCBLUE benchmark. Despite its comprehensiveness, this setup introduced substantial convergence instability due to heterogeneous task formats, conflicting label schemas, and inconsistent prompt templates. The model often overfit to specific task scaffolds and failed to generalize across task boundaries. Training loss was unstable, and generated outputs exhibited high structural drift and noise.

Unmasked Full-Sequence Loss. We also experimented with computing causal LM loss across the entire flattened prompt, including the `instruction` and `input` segments. While this approach conforms to standard language modeling practice, it diluted training signals: the model allocated gradient updates to instructional tokens rather than focusing on output accuracy. This led to slower convergence and weak output alignment.

Subjective Evaluation of Failure. Although we did not fully retrain these baselines to completion, we conducted qualitative probing by feeding the same test prompts to partially trained models from the early experiments. The responses consistently exhibited critical failure modes such as:

- **Label conflicts:** The same symptom appearing multiple times with contradictory labels;
- **Symptom duplication:** Repetition of the same label with no added value;
- **Format drift:** Outputs shifting into free-form natural prose;
- **Non-symptom hallucinations:** Generic phrases or treatment questions mistaken for symptoms.

Lessons and Final Design. These failure modes motivated a shift to a more constrained and reliable setup:

- Focus on a single, structurally well-defined task: symptom polarity classification;
- Normalize all samples into a consistent three-field instruction format;
- Apply output-only loss masking to focus training on answer generation.

This redesign led to substantial improvements in format consistency and prediction reliability. Quantitative evaluation of the final model is presented in Section 5.

4.5 Task Selection and Data Filtering

The PromptCBLUE benchmark [Zhu et al. 2022] includes 11 diverse Chinese medical NLP tasks, ranging from entity recognition and relation extraction to report generation. We initially attempted multi-task fine-tuning on all task types, but encountered severe instability due to inconsistent formats, label schema conflicts, and overfitting to task-specific templates.

To improve convergence and evaluation reliability, we narrowed our focus to a single structured task: **symptom polarity classification**. This task requires reading a doctor-patient dialogue and classifying whether each mentioned clinical finding is *present*, *absent*, or *uncertain*. We selected samples with `task_type = attr_cls` and further filtered instances whose `output` matched a fixed label set to reduce ambiguity.

Each data point was reformatted into a three-field instruction format:

```
Instruction: Describe the classification task
Input: Dialogue + Entity List + Label Options
Output: Symptom1: Label1
Symptom2: Label2 ...
```

This structure allows the model to follow explicit instructions and generate structured outputs aligned with clinical conventions. Although the Qwen model supports long contexts, we truncate each tokenized prompt to a maximum of 1024 tokens to fit within GPU memory constraints.

To enable efficient training under a causal language modeling (CLM) objective, we concatenate all three fields into a flat text sequence and compute cross-entropy loss over the output portion only. A left-shifted token prediction mechanism is used, with prompt tokens masked to avoid gradient leakage. This ensures that the model learns to generate structured symptom labels in response to the instruction and dialogue, without overfitting to prompt scaffolding.

4.6 Instruction-Based Prompt Formatting

To align with the pretraining objective of `Qwen2.5-0.5B-Instruct`, which is optimized for instruction-style input-output pairs, we reformatted each training sample into a unified three-field structure: `instruction`, `input`, and `output`. This format mirrors the instruction-tuning paradigm used in recent LLMs and provides a clear semantic boundary between task description, contextual information, and expected prediction.

For our selected task—symptom polarity classification—we implemented a prompt construction pipeline that dynamically generates task-specific instructions. Each prompt begins with an instruction template describing the classification objective, followed by the dialogue context, and finally a structured output. A representative example is shown in Table 2.

Table 2. Example of structured instruction-style prompt format

`Instruction:` Determine whether each clinical finding listed is *present*, *absent*, *other*, or *uncertain* based on the dialogue.
`Input:` *Doctor: Have you had a fever recently?* *Patient: No, but I've been coughing a lot.*
`Output:` Fever: Negative Cough: Positive

All samples follow the same structural template, which improves training stability and output consistency. By preserving this instruction-based format, we reduce the risk of prompt drift and allow the model to more effectively leverage its instruction-following pretraining.

5 Evaluation Metrics and Objective Function

We evaluate our model from three complementary perspectives: (1) token-level causal language modeling loss, (2) structured prediction accuracy over `symptom: label` pairs, and (3) output format consistency.

5.1 Objective Function: Causal LM Loss

During fine-tuning, we optimize the causal language modeling (CLM) loss over the output-only region. For a tokenized sequence $x = (x_1, x_2, \ldots, x_T)$, the loss is:

$$\mathcal{L}_{\text{CLM}} = -\sum_{t=1}^{T} \log P(x_t \mid x_{<t}) \tag{1}$$

Prompt tokens are masked from loss via `-100` labels to avoid overfitting on template text.

5.2 Token-Level F1 for Structured Output

Each prediction is parsed into a set of `symptom: label` pairs. We compare these sets against gold annotations and compute:

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{2}$$

where TP, FP, FN are computed based on exact match of both entity and polarity.

5.3 Revised Scoring Protocol for CHIP-MDCFNPC

The official evaluation script treats each `(entity, attr)` pair as an extraction unit. However, we identify several limitations:

- **Ambiguity:** Gold labels are descriptive texts that require mapping.
- **Mixed objectives:** Entity errors confound polarity accuracy.
- **Low coverage:** The dev set contains only 5 examples.

To address this, we introduce a cleaner classification-style evaluation:

- Canonicalize all labels into {`positive, negative, uncertain, none`};
- Align entities and fill missing predictions with `none`;
- Use `sklearn.metrics` to compute Accuracy, Macro-F1, Micro-F1;
- Supplement with confusion matrix and failure trace.

This setup decouples polarity classification from entity extraction and enables fairer comparison across formats.

5.4 LoRA Fine-tuning Configuration

We adopt Low-Rank Adaptation (LoRA) to fine-tune Qwen2.5-0.5B-Instruct with minimal computational overhead. The base model remains frozen, and only a small number of rank-decomposed adapters are trained. Our final configuration, implemented via the `peft` library, is summarized below:

- **Rank:** 8
- **Amplification factor (`lora_alpha`):** 16
- **Dropout:** 0.05
- **Target modules:** `q_proj`, `k_proj`, `v_proj`, `o_proj`
- **Bias:** None
- **Task type:** Causal language modeling

This setup updates fewer than 1% of model parameters and requires less than 5 GB of GPU memory. We selected $r = 8$ based on prior benchmarks balancing performance and efficiency, and applied dropout to mitigate overfitting on a narrow task. The selected modules align with the standard attention backbone, allowing effective adaptation without architectural modifications.

5.5 Training Procedure and Monitoring

We fine-tuned the model using the HuggingFace `Trainer` API with the following configuration:

- **Epochs:** 3
- **Effective batch size:** 16 (2 per device × 8 accumulation steps)
- **Optimizer:** `adamw_torch_fused`
- **Learning rate:** 2×10^{-4}
- **Precision:** bfloat16 mixed precision
- **Logging:** evaluation every 10 steps; checkpoint saving every 50 steps

To monitor convergence, we implemented a custom callback that records training and validation loss across steps. This enabled early detection of instabilities such as overfitting or format drift.

Round 1: Multi-task fine-tuning. We first trained on all 11 PromptCBLUE tasks simultaneously. Although the training loss converged to 0.75, the validation loss plateaued around 2.7, and manual inspection

revealed hallucinated outputs, missing labels, and structural collapse. The dev set failed to provide usable feedback due to format inconsistency and noisy annotations.

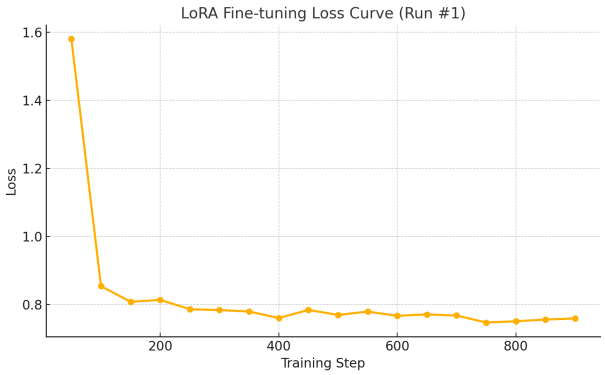

Fig. 1. Training Loss (Round 1, all tasks)

Round 2: Clean task training, noisy dev contamination. Next, we fine-tuned only on the `symptom polarity classification` task, but accidentally included dev samples from a different but structurally similar task. Although training behavior improved, the evaluation loss remained erratic. Post-hoc analysis revealed that mismatched label formats in the dev set (e.g., different entity-label schema) confused the model and inflated validation loss.

Fig. 2. Training vs Evaluation Loss (Round 2, single task unclean dev set)

Round 3: Task-specific adaptation. After restricting training to the filtered symptom-polarity task, structure alignment improved noticeably. Although evaluation loss remained flat (due to residual template artifacts), outputs became more reliable in syntax and polarity labeling. This subjective improvement—confirmed via prompt probing—motivated our decision to discard multi-task learning in favor of a single-task specialization.

5.6 Inference and Model Merging

After fine-tuning, we merged the LoRA-adapted weights into the base Qwen model using `merge_and_unload()` from the `peft` library. This operation produces a single standalone checkpoint,

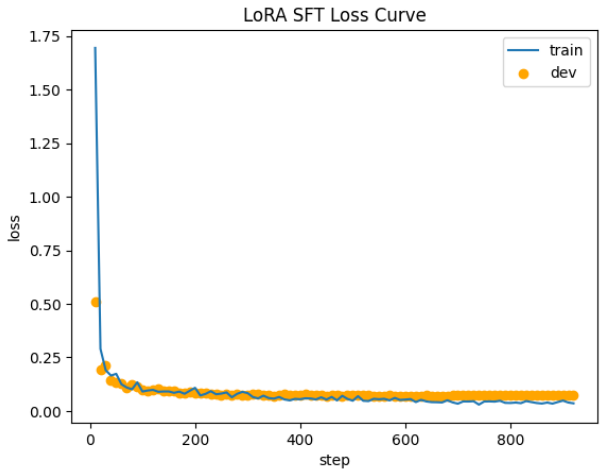

Fig. 3. Training vs Evaluation Loss (Round 2, single task)

eliminating the need for runtime LoRA adapters and simplifying downstream deployment. The resulting model maintains the same architecture as Qwen2.5-0.5B-Instruct, with modified attention projections reflecting the learned task specialization.

Inference Setup. To assess inference performance, we ran the merged models and the original zero-shot Qwen2.5-0.5B-Instruct on 10 representative prompts drawn from training and development distributions. Each prompt was fed in instruction format, and the model's output was evaluated on the following criteria:

- **Structural fidelity:** Does the output follow the `symptom: label` format?
- **Label correctness:** Are predicted polarities consistent with ground truth?
- **Instruction adherence:** Does the output respect the instruction (e.g., avoids free-form generation)?

Subjective Comparison. Our instruction-tuning process involved three distinct rounds of training, each highlighting different failure modes and lessons.

In the first round, we fine-tuned on all 11 tasks from PromptCBLUE without strict filtering. This led to severe degradation: the model lost its basic instruction-following behavior and even failed to carry normal conversations. Outputs were frequently off-topic, structurally incoherent, or collapsed into repeated phrases, suggesting catastrophic interference from conflicting task formats.

In the second round, we correctly restricted the training data to the target task—symptom polarity classification—but mistakenly included a development set drawn from a similar yet incompatible task. This caused the dev loss to remain flat (~2.7) despite clean training convergence, which we initially interpreted as overfitting. However, due to the lack of proper evaluation samples, the true performance remained uncertain, and this version was never quantitatively benchmarked.

In the third and final round, we applied strict task filtering and cleaned both the training and evaluation sets. This resulted in a

fine-tuned model that clearly outperformed the base model. It consistently followed the instruction format, produced well-structured `symptom: label` outputs, and maintained high fidelity to ground-truth labels. In contrast, the zero-shot base model frequently defaulted to natural language summaries or omitted labels altogether.

Practical Outcome. This experiment confirms that even a small model like Qwen2.5-0.5B, when adapted via LoRA and clean task supervision, can be transformed into a reliable, format-following assistant—without incurring additional runtime overhead or memory cost.

6 Results

We evaluate the fine-tuned model on a held-out set of over 600 examples using structure-aware metrics. Each output is parsed into `symptom: label` pairs and compared to the ground-truth annotations. Figure 4 and 5 show the confusion matrices before and after applying synonym canonicalization.

6.1 Subjective Comparison with the Base Model

Before presenting quantitative scores, we highlight key differences observed during manual evaluation. The original Qwen2.5-0.5B-Instruct model, when used in zero-shot mode, produces fluent and semantically relevant outputs. However, its responses often take the form of free-text sentences (e.g., "The patient likely has a fever") rather than structured symptom-label pairs. As a result, the outputs are difficult to parse and cannot be evaluated reliably using automated scripts.

By contrast, early-stage fine-tuned models (from Round 1 and Round 2) performed worse than the base model in both formatting and content fidelity. Outputs were frequently malformed, repetitive, or completely off-task. In Round 1, multi-task interference appeared to damage the base model's instruction-following capability. In Round 2, although the training data was task-specific, an incorrect validation set introduced mismatched schema and inconsistent feedback, making loss metrics misleading.

Only in Round 3—after careful task filtering and format alignment—did the model consistently outperform the base. It produced outputs in the desired `symptom: label` format, adhered to instructions, and returned labels consistent with the input context. This illustrates the value of instruction fine-tuning not for improving fluency, but for enforcing structure and output consistency in domain-specific applications.

6.2 Evaluation Before Synonym Canonicalization

Figure 4 presents the confusion matrix for the final model's predictions without any synonym normalization. The model achieved a micro-averaged precision of 0.660, recall of 0.657, and F1 score of 0.658. The predictions are skewed toward the `positive` label (class 1), and significant confusion is observed between the `positive` and `uncertain` classes (class 3), which often co-occur in ambiguous clinical contexts.

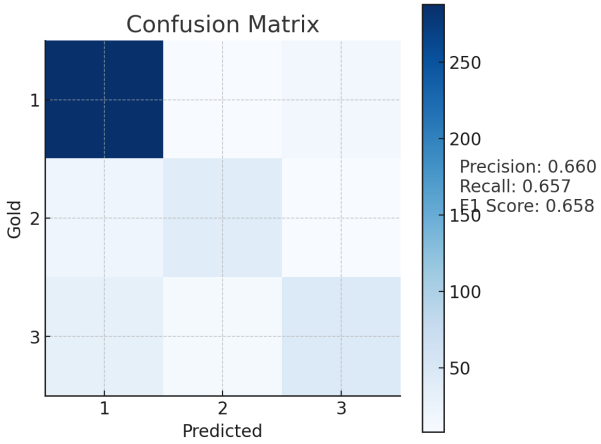

Fig. 4. Confusion matrix before synonym canonicalization.

6.3 Improvement with Synonym Canonicalization

To account for lexical variation in symptom expression (e.g., "cough" vs "dry cough" vs "persistent cough"), we apply a synonym canonicalization step prior to evaluation. This involves mapping expression variants to their canonical forms using string similarity and semantic grouping.

After canonicalization, as shown in Figure 5, performance improves substantially: precision rises to 0.736, recall to 0.728, and F1 to 0.732. Off-diagonal confusion is visibly reduced, especially between similar classes. This confirms that lexical normalization is critical for fair and faithful evaluation in medical NLP settings.

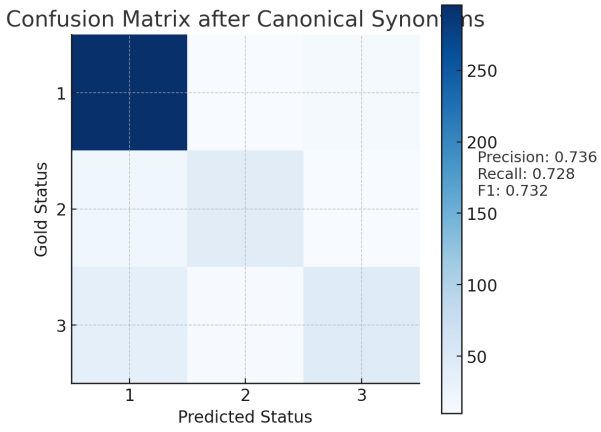

Fig. 5. Confusion matrix after canonical synonym mapping.

6.4 Summary

These findings highlight three key takeaways:

- **Instruction tuning improves structure alignment:** The base model is fluent but unstructured; fine-tuning guides it to produce machine-readable outputs in the required `symptom: label` format.

- **Lexical normalization enhances evaluation quality:** Without synonym mapping, many correct predictions are penalized due to surface-level variation. Canonicalization enables more faithful performance estimates by aligning variants to unified forms.
- **Actual accuracy may be underestimated:** While our synonym table covers a broad range of common variants, it is not exhaustive. Some semantically equivalent expressions may still be counted as mismatches, meaning the true model accuracy could be even higher than reported.

Together, these results demonstrate the importance of structure-aware instruction tuning and robust evaluation design when adapting lightweight LLMs to medical tasks.

7 Discussion

While the LoRA-adapted Qwen model demonstrated promising improvements in structured output generation and label awareness, several challenges emerged throughout the development process. These insights offer a deeper understanding of the trade-offs involved in adapting general-purpose LLMs to structured medical tasks under resource constraints.

7.1 Evaluation Instability and Overfitting

In early experiments, the model quickly reached low training loss but failed to generalize. This was largely due to training across heterogeneous task types, which encouraged memorization of template phrases rather than the development of reasoning capabilities. At the same time, we observed that the development set was unintentionally contaminated by unrelated tasks, breaking instruction-output alignment and inflating validation loss. Together, these factors led to misleading learning curves and obscured the true behavior of the model.

7.2 Output Errors: Conflict, Redundancy, and Format Drift

Despite improved structure adherence, the model occasionally produced multiple, conflicting labels for the same symptom, especially in multi-turn dialogue contexts involving partial recovery or historical mentions. Redundant repetition of entities was also observed, often without added semantic value. Moreover, the model sometimes deviated from the expected instruction-following format and instead reverted to free-text outputs, especially when input prompts were underspecified or indirectly phrased.

7.3 Prompt Design and Instruction Engineering

One of the key insights from this project is the critical role of prompt formatting. By unifying all inputs into a consistent `instruction + input + output` schema, we enabled the model to better understand task boundaries and expected output formats. Manual prompt engineering—such as standardizing instruction templates and explicitly listing available label options—proved essential for controlling generation behavior.

7.4 Subjective Observations and Human-in-the-Loop Feedback

While early fine-tuned models achieved low training loss, they subjectively underperformed compared to the base model. In fact, the first LoRA adaptation even degraded the base model's general dialogue capabilities, suggesting catastrophic interference. The second attempt, though trained on the correct task, included mismatched samples in the dev set and lacked quantitative evaluation. Only the final round of fine-tuning—with strict task filtering and prompt alignment—outperformed the base model both qualitatively and structurally.

7.5 Generalization and Synonym Coverage

The model performed well on in-distribution samples but remained sensitive to minor variations in wording, symptom names, or instruction phrasing. Although we constructed a manually curated synonym dictionary to align semantically equivalent entities, it is likely that not all cases were fully captured. Therefore, the true performance of the model may in fact be higher than measured by current exact-match metrics.

7.6 Future Directions

To further enhance robustness and task fidelity, several directions can be explored:

- Enforce stricter prompt templates with clear section headers.
- Introduce synthetic samples with varied phrasing to improve generalization.
- Post-process outputs to remove duplicates and enforce label consistency.
- Reframe the task as token-level classification rather than text generation.
- Apply ensemble decoding or majority voting for ambiguous cases.

Together, these improvements may lead to more stable and clinically usable instruction-following models, even under constrained deployment conditions.

8 Conclusion

In this work, we explored the use of Low-Rank Adaptation (LoRA) to adapt an open-source, instruction-tuned language model—Qwen2.5-0.5B-Instruct—to a clinically relevant structured task: symptom polarity classification. Motivated by the need for deployable, privacy-preserving AI tools in low-resource medical environments, we focused on building a lightweight and format-aligned assistant capable of producing structured outputs from raw dialogue text.

Our approach involved a series of iterations across data formatting, task selection, and training configurations. Initial attempts to fine-tune across all PromptCBLUE tasks failed due to format inconsistency and label conflicts, highlighting the difficulty of multitask adaptation in small models. By narrowing our focus to a single task and adopting a unified instruction-input-output schema, we were able to stabilize training and improve structural consistency.

Throughout the process, we encountered and addressed practical issues such as evaluation contamination, overfitting to template language, and output redundancy. The final model, trained on filtered

and reformatted data using LoRA, outperformed the base Qwen model in both subjective and structure-aware evaluation. Despite the absence of full benchmark coverage, our targeted metrics and qualitative comparisons confirmed the effectiveness of task-specific fine-tuning under resource constraints.

This project provides a blueprint for how small language models can be reliably adapted to structured clinical tasks using parameter-efficient tuning. It demonstrates that careful task scoping, prompt engineering, and human-in-the-loop iteration are critical to making instruction-tuned LLMs practically useful. As future work, we aim to expand synonym coverage, explore token-level formulations, and integrate postprocessing for more robust deployment.

References

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient Finetuning of Quantized LLMs. *arXiv preprint arXiv:2305.14314* (2023). arXiv:2305.14314

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, et al. 2019. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning (ICML)*.

Edward Hu, Yelong Shen, Phil Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations (ICLR)*.

Karan Singhal, Shekoofeh Azizi, Tania Tu, et al. 2023. Large language models encode clinical knowledge. *Nature* 620 (2023), 472–477.

Xiaolong Sun et al. 2023. FLAT-LLaMA: Instruction-tuned Open Chinese Medical Foundation Model. *arXiv preprint arXiv:2307.11484* (2023). arXiv:2307.11484

Yu Zhang et al. 2023. MedPinyin: Boosting Chinese medical LLMs with phonetic memory. *arXiv preprint arXiv:2310.11734* (2023). arXiv:2310.11734

Wanjun Zhu, Wei Xu, et al. 2022. PromptCBLUE: A Prompt-based Chinese Biomedical Language Understanding Benchmark. *arXiv preprint arXiv:2204.09610* (2022). arXiv:2204.09610