TeleAI at SemEval-2026 Task 6: A Confidence-Aware Multi-Stage Reasoning Framework with Chain-of-Thought

Lingling Shi[*1], **Haoyu Jin**[*1], **Shiquan Wang**[1],
Ruiyu Fang[1], **Shuangyong Song**[1], **Xuelong Li**[†2]

[1]Xingchen AGI Lab, China Telecom Artificial Intelligence Technology (Beijing) Co., Ltd
[2]Institute of Artificial Intelligence (TeleAI), China Telecom
jinhaoyu@bupt.edu.cn, shill2,wangsq23,fangry,songshy@chinatelecom.cn, xuelong_li@ieee.org

Abstract

This paper describes our framework for SemEval-2026 Task 6 (CLARITY - Unmasking Political Question Evasions), which focuses on classifying clarity and fine-grained evasion types in political question-answering dialogues. We propose CAMSR-CoT, a **C**onfidence-**A**ware **M**ulti-**S**tage **R**easoning framework with **C**hain-**o**f-**T**hought that unifies the two subtasks through hierarchical label modeling. The framework adopts a confidence-based routing strategy: high-certainty cases are directly resolved, while ambiguous samples are routed to deeper Chain-of-Thought reasoning stages with boundary-aware few-shot exemplars to mitigate label confusion. On the development set, our framework achieves Macro-F1 scores of 0.812 on SubTask 1 and 0.617 on SubTask 2. On the official hidden test set, it ranks 1st in both SubTask 1 (Macro-F1 = 0.89) and SubTask 2 (Macro-F1 = 0.68).

1 Introduction

CLARITY - Unmasking Political Question Evasions (SemEval-2026 Task 6) (Thomas et al., 2026) focuses on classifying the clarity and evasion strategies of answers in English political interviews (Thomas et al., 2024). It includes two subtasks: SubTask 1 is a 3-way classification (Clear Reply / Ambivalent / Clear Non-Reply), and SubTask 2 is a fine-grained 9-way evasion classification. The test set adopts a multi-reference evaluation (each sample has three reference labels). Evasion and equivocation in political Q&A are classic phenomena in discourse analysis (Bull, 1994), and automatically identifying evasion strategies is significant for improving transparency in political accountability.

We propose CAMSR-CoT: a three-stage prompting reasoning framework with confidence-based routing[1], built upon the AI Flow paradigm (An et al., 2026). Analysis of the training set confusion matrix suggests that the three Non-Reply classes (*Declining* / *Claims ignorance* / *Clarification*) have relatively distinct linguistic signals but tend to produce high false positive rates. Meanwhile, the remaining 6 classes show considerable mutual confusion within the 9-class framework. Accordingly, we designed a "divide-and-conquer" strategy: Step 1 uses a lightweight Gate to identify high-confidence Non-Replies and output them directly; medium-to-low confidence suspected Non-Replies enter Step 2 for a 9-class correction review; the remaining samples enter Step 3 for fine-grained classification within the 6-class space. Each stage employs Chain-of-Thought (CoT) reasoning (Wei et al., 2022) with dynamically assembled prompts that include refined label definitions, confusion-pair guidelines, and contrastive few-shot examples mined from the training set confusion matrix.

Technically, CAMSR-CoT builds on in-context learning (Brown et al., 2020) and Chain-of-Thought prompting (Wei et al., 2022). Decomposing the 9-class decision into staged sub-problems reduces the model's tendency to rely on surface cues when labels overlap semantically, in line with the least-to-most decomposition principle (Zhou et al., 2023). On the development set, CAMSR-CoT improves SubTask 2 performance by +0.127 (0.490→0.617) compared to a single-step Chain-of-Thought (CoT) baseline. Ablation studies show that Gate confidence routing and Step 3 few-shot boundaries are the main contributors to the performance gain. The main bottleneck is the *Dodging* class (Recall only 0.179), whose boundaries with *General* and *Deflection* are often ambiguous—human annotators show substantial disagreement on these classes.

[*]Equal contribution.
[†]Corresponding author.

[1]Code available at: https://github.com/ther7777/semeval-2026-task6-camsr-cot

Proceedings of the The 20th International Workshop on Semantic Evaluation (2026), pages 1591–1609

2 Background

2.1 Task Setup and Data Overview

Each sample consists of an interview question, an interview answer, and a simplified sub-question derived from the original question. The framework must classify the answer with respect to the sub-question. There is a deterministic mapping between the two SubTask levels: *Explicit* corresponds to Clear Reply; *Declining to answer*, *Claims ignorance*, and *Clarification* correspond to Clear Non-Reply; the remaining 5 classes (*Implicit*, *Partial/half-answer*, *General*, *Deflection*, *Dodging*) correspond to Ambivalent. Our framework does not predict SubTask 1 independently but derives it from SubTask 2 results via this mapping to ensure consistency. Throughout this paper, we refer to the three SubTask 2 labels corresponding to *Clear Non-Reply* collectively as *Non-Reply*.

The training set contains 3,448 single-label English samples; the development set contains 308 multi-reference samples (each annotated independently by 3 annotators), with a pronounced long-tail distribution (see Appendix Table 5): minority classes such as *Partial/half-answer* (2.3%) and *Clarification* (2.7%) are extremely sparse, and there is a distribution shift in *Dodging* (Train 20.5%→Dev 18.7%) and *General* (11.2%→18.6%).

2.2 Label Ambiguity and Confusion Patterns

Multi-reference annotations in the development set reveal the inherent ambiguity of the task (Plank et al., 2014): only 40.6% of samples have unanimous agreement among three annotators, 48.7% have two different labels, and 10.7% have three different labels. *General* appears in 4 of the top 5 confusion pairs (with General↔Implicit being the highest at 12.0%), indicating that the boundary between "general answers" and other evasion types is blurred. Furthermore, high disagreement rates in *Explicit↔General* (10.7%) and *Explicit↔Implicit* (8.8%) suggest that even on the fundamental judgment of "whether information was provided," human annotators disagree significantly. See Appendix Table 6 for details.

Baseline Analysis. Zero-shot reasoning (Kojima et al., 2022) with DeepSeek-V3 using the original label definitions (Thomas et al., 2024) achieves only 0.421 SubTask 2 Macro-F1 (Table 2), with the model being over-sensitive to refusal cues (misclassifying *Dodging* and *General* as *Declining to answer*) and significant *Implicit↔General* confusion. These patterns motivate two design decisions: (1) mining contrastive few-shot examples from the training set confusion matrix to teach boundary distinctions; (2) staged label-space reduction—isolating the three Non-Reply classes first, then classifying within the reduced 6-class space.

2.3 Related Work

Our work intersects three lines of research. Equivocation typologies for political interviews (Bull, 1994), further refined by Rasiah (2010), inform the response clarity taxonomy underlying our task (Thomas et al., 2024). Methodologically, our framework builds on Chain-of-Thought prompting (Wei et al., 2022) and its multi-step extensions—least-to-most decomposition (Zhou et al., 2023), Tree-of-Thought (Yao et al., 2023), and Plan-and-Solve (Wang et al., 2023a)—which demonstrate that structured staged reasoning improves LLM performance across various complex tasks (Dong et al., 2024; Shen and Zhang, 2025; Wu et al., 2025; Chen et al., 2025; Zhang et al., 2024; Chang et al., 2025; Shi et al., 2026; Li et al., 2025; Xiong et al., 2025a; Okwuonu et al., 2025; Xing et al., 2025; Xiong et al., 2025b). Our confusion-matrix-driven example selection relates to active example selection strategies (Diao et al., 2024); automatic prompt optimization (Khattab et al., 2024) offers a complementary direction we leave to future work. Our multi-reference evaluation draws on work treating annotation disagreement as a legitimate signal rather than noise (Plank et al., 2014; Uma et al., 2021).

3 Framework Overview

3.1 Overall Pipeline

Figure 1 illustrates the three-stage inference pipeline of CAMSR-CoT. In Step 1 (the *Gate*), we predict a label and a confidence level. If the answer is a high-confidence Non-Reply, we output it directly; otherwise, we route it to Step 2 for review or Step 3 for subdivision.

Given an input sample x, let g and c denote the Gate's predicted label and confidence, respectively, and let $\mathcal{S}_{\text{NR}}$ be the set of Non-Reply classes. The

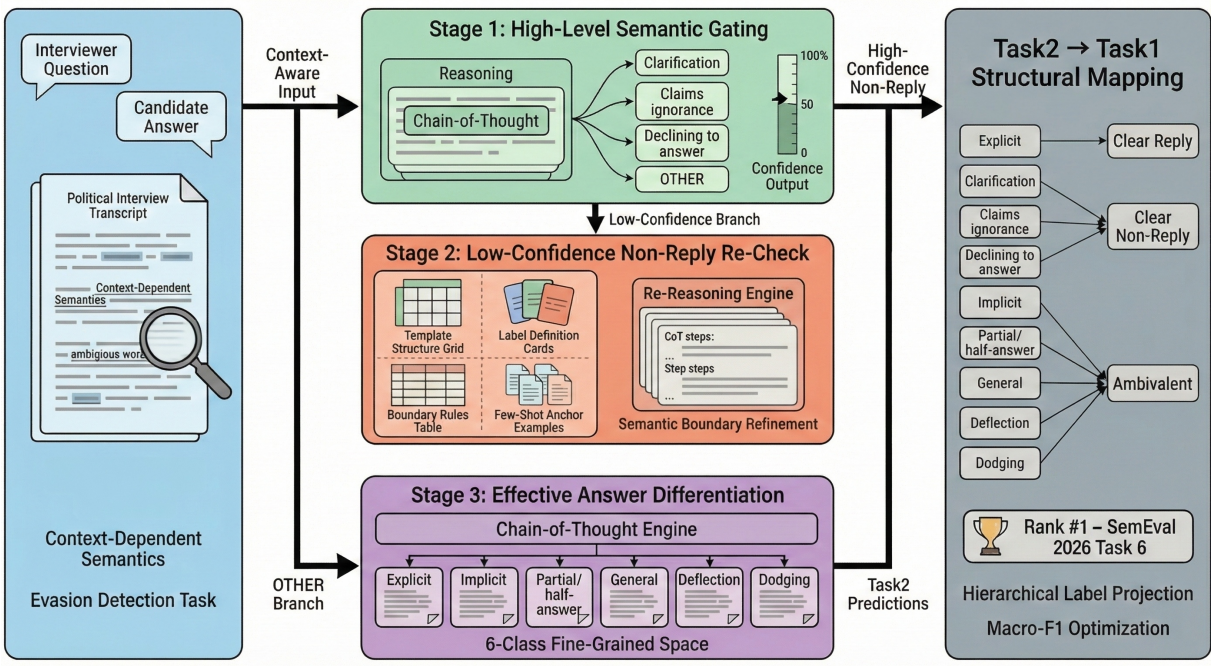

Figure 1: Three-stage routing pipeline of CAMSR-CoT.

routing logic for the SubTask 2 prediction $\hat{y}$ is:

$$\hat{y} = \begin{cases} g & \text{if } g \in \mathcal{S}_{\text{NR}} \wedge c = \text{High} \\ \text{Step2}(x, g) & \text{if } g \in \mathcal{S}_{\text{NR}} \wedge c \neq \text{High} \\ \text{Step3}(x) & \text{if } g = \text{OTHER} \end{cases} \quad (1)$$

This design diverts low-ambiguity samples early and concentrates extra prompt budget on high-risk branches. Final SubTask 1 labels are generated via the fixed mapping described in Section 2.1. Each stage uses dynamically assembled prompts, the components of which are summarized in Table 1.

Routing Statistics and Inference Cost. On dev, 86.7% of the samples enter the OTHER→Step 3 branch, 7.5% trigger Step 2 correction review, and only 5.8% are direct exits belonging to high-confidence Non-Reply. This results in an average of 1.94 model calls per sample. See Appendix A.5 for detailed cost analysis. A complete end-to-end walkthrough of a development set sample is provided in Appendix A.9.

3.2 Step 1: Gate with Confidence (4-way)

The Gate outputs only three Non-Reply classes (*Declining/Claims/Clarification*) or *OTHER*, along with a *High/Medium/Low* confidence. We adopt a conservative confidence criterion (full prompt in Appendix B) centered on a key-information override rule: if the answer provides any information

Component	Step 1	Step 2	Step 3
Anchors	×	✓(2)	✓(6)
Boundaries	×	✓(3)	✓(10)
Confusion Guide	×	✓	✓
Prior Analysis	×	✓	×
Confidence	✓	×	×
CoT	✓	✓	✓

Table 1: Usage of prompt components in each stage. Numbers in parentheses indicate the count of few-shot examples injected.

requested by the sub-question, the Gate must output *OTHER* regardless of refusal cues. The three confidence levels are defined as follows:

- *High*: the Non-Reply cue is explicit and unambiguous, the answer is short and consists mostly of the refusal/ignorance/clarification itself, and there is no possibility of the answer being *Implicit* or *Partial*.
- *Medium*: the answer fits a Non-Reply definition, but the refusal or ignorance cue is embedded in a longer explanation, followed by pivoting phrases (e.g., "but…", "however…"), or otherwise mixed with additional content that may warrant reclassification.
- *Low*: the answer shows some Non-Reply features, but may arguably be providing partial or implicit information; the Gate flags it as a

potential Non-Reply for downstream verification.

Only High-confidence Non-Replies are output directly; all other cases are routed to Step 2 to reduce false positives.

3.3 Step 2: Non-Reply Correction Review (9-way)

When the Gate predicts Non-Reply with Medium/Low confidence, Step 2 is triggered for a 9-class review. The prompt components are detailed in Table 1. The *Prior Analysis* component unique to Step 2 summarizes the label and confidence determined by the Gate, providing prior context for correction review. Boundary counter-examples (B=3) are all mined from the training set zero-shot confusion matrix, focusing on covering high-frequency misclassification pairs such as *Dodging*→*Declining*.

3.4 Step 3: OTHER Branch Subdivision (6-way)

When the Gate outputs OTHER, the sample may correspond to either Clear Reply or Ambivalent in SubTask 1; Step 3 classifies it within the 6-class set (excluding the three Non-Reply classes), from which the SubTask 1 label is derived via the mapping in Section 2.1. This reduced label space lowers overlap and improves discrimination accuracy. Prompt components are shown in Table 1: Step 3 injects 6 anchors (1 per class) and 10 boundaries (covering the most confusing counter-example pairs). Prior analysis is excluded to prevent Gate priors from influencing subdivision decisions.

4 Experimental Setup

Data and Metrics. Both subtasks are evaluated with Macro-F1 using the official scorer; SubTask 2 uses a multi-reference variant where a prediction is correct if it matches any annotator label (formal definitions in Appendix A.1). Anchors and boundaries are mined from zero-shot reasoning on the training set; main results are reported on the development set.

Model and Inference Configuration. Our main framework (CAMSR-CoT) and its ablation variants use DeepSeek-V3 (DeepSeek-AI et al., 2025) as the backbone, accessed through the DeepSeek API with temperature 0.0 and a maximum generation length of 1,800 tokens to accommodate full CoT reasoning traces.

Method (dev)	SubTask 1 F1	SubTask 2 F1
Prompting Methods (DeepSeek-V3)		
Direct 9-class (No CoT)	0.662	0.421
CoT Single-step 9-class	0.710	0.490
CAMSR-CoT (Ours)	**0.812**	**0.617**
*Alternative Backbones (*CAMSR-CoT *Pipeline)*		
Qwen2.5-7B-Instruct†	0.481	0.288
Qwen3-235B-A22B	0.772	0.556
TeleChat3-36B-Thinking	0.763	0.561
Supervised Fine-Tuning (Qwen2.5-7B)		
Direct SFT	0.587	0.495
SFT + Distillation‡	0.758	0.509

Table 2: Method comparison (dev). †Uses simplified prompts due to context length limits. ‡Teacher is Qwen3-235B-A22B running the CAMSR-CoT pipeline. SubTask 2 uses official multi-reference scorer.

Comparison Methods. Table 2 compares three groups of methods: (1) **Prompting Methods** (DeepSeek-V3): including direct 9-class baseline (no CoT), single-step CoT baseline, and CAMSR-CoT (this paper); (2) **Alternative Backbone Models**: migrating this framework to Qwen3-235B (Qwen Team, 2025), TeleChat3-36B-Thinking (Liu et al., 2025; Wang et al., 2024a,b, 2025; Li et al., 2024; Yao et al., 2024), and Qwen2.5-7B (Qwen Team, 2025) to evaluate framework generality; (3) **Supervised Fine-Tuning** (Qwen2.5-7B) (Ouyang et al., 2022): including direct SFT and knowledge distillation (Hinton et al., 2015) using CAMSR-CoT output as teacher.

5 Results

5.1 Main Results

Table 2 presents the comparison results on the development set. Within the prompting methods group, CAMSR-CoT improves SubTask 2 performance by +0.127 (0.490→0.617) over the single-step CoT baseline, suggesting that the multi-stage routing strategy yields substantial gains. In the alternative backbone experiments, Qwen3-235B (Qwen Team, 2025) and TeleChat3-36B-Thinking (Liu et al., 2025) achieve SubTask 2 F1 scores of 0.556 and 0.561 respectively using the same pipeline, indicating that the CAMSR-CoT framework has strong cross-model transferability, although performance remains lower than DeepSeek-V3 (0.617). Qwen2.5-7B, due to its smaller model size, has limited effectiveness when running the pipeline directly (SubTask 2 F1 = 0.288). In the supervised fine-tuning group, di-

Rank	SubTask 1 Team	F1	SubTask 2 Team	F1
1	**TeleAI (Ours)**	**0.89**	**TeleAI (Ours)**	**0.68**
2	AsymVerify	0.85	ChulaNLP†	0.61
3	CSE-UOI	0.85	CLaC @ CLARITY	0.59
4	Rasende Rakete	0.83	Rasende Rakete	0.59
5	Evaluators	0.83	YNU-HPCC	0.59

Table 3: Official hidden evaluation leaderboard (Top 5). Our framework ranks first in both tasks. †The original Codabench username was *moswisarut*, updated to the team name *ChulaNLP*.

Variant (dev)	SubTask 1 F1	SubTask 2 F1
CoT single-step 9-class (Baseline)	0.710	0.490
+ Gate routing (no confidence)	0.761	0.465
+ Confidence routing (Base Pipeline)	0.811	0.503
+ Refined label definitions	0.800	0.520
+ Step-3 few-shot B=10 (Ours)	**0.812**	**0.617**

Table 4: Ablation study (Dev Macro-F1). Each row adds a component to the previous one. *Base Pipeline* refers to the three-stage framework with basic definitions and no Step-3 few-shots.

rect SFT achieves a SubTask 2 F1 of 0.495, comparable to the single-step CoT baseline (0.490). Knowledge distillation notably improves SubTask 1 (0.587→0.758) with a modest SubTask 2 gain (0.509), suggesting that the teacher's CoT reasoning helps the student better capture coarse-grained clarity distinctions. Both SFT variants, however, remain below large-model prompting approaches. Training details are provided in Appendix A.11.

Fine-grained Performance Analysis. Per-class results (Appendix Table 8) show that Gate routing achieves a Precision of 1.000 for all three Non-Reply classes. The main bottleneck is *Dodging* (Recall = 0.179), whose boundaries with *General* and *Deflection* tend to be ambiguous, consistent with annotator disagreement patterns (Section 2.2).

Official Evaluation Results. On the official hidden evaluation set (Table 3; see Appendix Table 11 for full results), our framework (TeleAI) ranks 1st in both SubTask 1 (Macro-F1 = 0.89) and SubTask 2 (Macro-F1 = 0.68), leading the second place by 4 and 7 percentage points respectively.

5.2 Ablation Study and Analysis

Table 4 shows the contribution of each component to final performance. Key findings: (1) Gate routing + confidence routing improved SubTask 1 from 0.710 to 0.811, mainly contributing to the reduction of Non-Reply false positives; (2) Switching to refined label definitions improved SubTask 2 (+0.017) but had a slight impact on SubTask 1 (−0.011); (3) Injecting few-shot boundaries in Step 3 brought the largest SubTask 2 improvement (+0.097) while maintaining SubTask 1 stability.

Impact of Confidence Routing. The Step 2 correction branch is triggered on only 23/308 dev samples, yet it has a disproportionate influence on Macro-F1: under the multi-reference scorer, an incorrect Non-Reply prediction incurs one FP and multiple FNs (one per reference label), heavily penalizing precision-sensitive classes. By restricting High confidence to short, unambiguous refusals, we route borderline samples into Step 2, where boundary counter-examples help distinguish genuine refusals from strategic deflections. As a result, all three Non-Reply classes achieve a precision of 1.000 on dev (Table 8).

Contribution of Definitions and Boundaries. From the base pipeline to the final framework, SubTask 2 Macro-F1 improved by +0.114. The largest per-class gains are in *General* (+0.334) and *Implicit* (+0.383)—notably the two classes with the highest annotator disagreement (Section 2.2). We attribute this to the refined definitions spelling out the key difference between these two classes ("inferable specific information" vs. "on-topic but vague"), combined with boundary examples that present direct *General*↔*Implicit* contrasts.

Additional ablation dimensions—boundary count sensitivity (B=8−14) and the isolated contribution of the confusion guide (+0.102 F1)—are reported in Appendix A.7. Detailed per-class ablation comparisons for SubTask 1 and SubTask 2 are provided in Appendix Tables 7 and 9.

Summary of Key Drivers. From the ablation experiments, we identify three key performance drivers in decreasing order of impact: (1) confidence-aware Gate routing most significantly reduces Non-Reply false positives, yielding the largest SubTask 1 gain (+0.101 F1); (2) boundary-aware few-shot examples in Step 3 contribute the largest SubTask 2 gain (+0.097 F1), directly addressing confusion among fine-grained evasion types; (3) refined label definitions and confusion guides help resolve the persistent ambiguity between *General* and *Implicit* (+0.017 SubTask 2 F1). All three components are complementary: re-

moving any one degrades overall performance.

5.3 Error Analysis

Of 308 dev samples, 112 (36.4%) are misclassified, with 47 on unanimously annotated samples. Errors concentrate in the *Dodging–General–Deflection* triangle (77.7% of all errors); all 18 direct-exit samples are correct (0% error), while Step 2 and Step 3 error rates are 47.8% and 37.8%. Closer inspection of Step 2 errors reveals a systematic pattern: 9 of the 10 misclassified samples are over-corrected to *Deflection*, with gold labels spanning *General* (7), *Declining* (3), *Dodging* (3), and *Implicit* (2). Because Step 2 receives only borderline Non-Reply samples that the Gate flagged with medium or low confidence, these cases inherently sit at the semantic boundary between refusal and evasion; the 9-class review tends to over-correct toward *Deflection*—the most frequent evasion type in the correction prompt—rather than preserving the original Non-Reply label or selecting other Ambivalent classes.

Two residual patterns dominate: (1) Evasion Triangle Blur: the model defaults to *General* when answers stay on-topic but sidestep the sub-question, while annotators favor *Dodging* or *Deflection*; (2) Hedging Bias: hedging language biases the model toward *Implicit* even when specific information is provided (13 cases crossing the Clear Reply boundary).

The *Dodging* class is the primary bottleneck (Recall = 0.179, F1 = 0.289): we attribute this to both annotation ambiguity and model limitations. On the annotation side, *Dodging* has the lowest inter-annotator agreement among all nine classes, frequently confused with *General* and *Deflection* by human raters themselves (Section 2.2); on the model side, LLMs exhibit a tendency to interpret deliberately vague but on-topic responses as merely general rather than intentionally evasive, making *Dodging* inherently difficult to capture without explicit intent modeling.

See Appendix A.8 for the confusion matrix, per-path statistics, and illustrative case studies.

6 Conclusion

This paper proposes CAMSR-CoT: a multi-stage prompting framework with confidence routing. By analyzing label ambiguity and model confusion patterns (Section 2.2), we designed a routing strategy and dynamic contrastive examples, achieving 1st place in SemEval-2026 Task 6. Ablation studies confirm that Gate confidence routing and few-shot boundaries are key to performance. We did not evaluate majority voting or alternative prompt templates for the Step 2 correction path in the current work. Future directions include exploring more robust correction mechanisms (e.g., self-consistency (Wang et al., 2023b) or multi-sample voting) for the Step 2 branch, automating boundary retrieval and adaptive example selection, and distilling the reasoning pipeline (Hinton et al., 2015) into smaller models to reduce deployment costs.

References

Hongjun An, Wenhan Hu, Sida Huang, Siqi Huang, Ruanjun Li, Yuanzhi Liang, Jiawei Shao, Yiliang Song, Zihan Wang, Cheng Yuan, Chi Zhang, Hongyuan Zhang, Wenhao Zhuang, and Xuelong Li. 2026. Ai flow: perspectives, scenarios, and approaches. *Vicinagearth*, 3(1).

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Peter Bull. 1994. On identifying questions, replies, and non-replies in political interviews. *Journal of Language and Social Psychology*, 13(2):115–131.

Wenhan Chang, Tianrui Zhu, Yue Zhao, Shuangyong Song, Peng Xiong, Wenhao Zhou, and Yongxiang Li. 2025. Chain-of-lure: A synthetic narrative-driven approach to compromise large language models. *Preprint*, arXiv:2505.17519.

Jinyang Chen, Haolun Wu, Jianhong Pang, Yihua Wang, Dell Zhang, and Changzhi Sun. 2025. Tool learning with language models: a comprehensive survey of methods, pipelines, and benchmarks. *Vicinagearth*, 2(1).

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2025. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Shizhe Diao, Pengcheng Wang, Yong Lin, Rui Pan, Xiang Liu, and Tong Zhang. 2024. Active prompting with chain-of-thought for large language models. In

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1330–1350, Bangkok, Thailand. Association for Computational Linguistics.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. 2024. A survey on in-context learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1107–1128, Miami, Florida, USA. Association for Computational Linguistics.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. 2024. Dspy: Compiling declarative language model calls into self-improving pipelines. In *The Twelfth International Conference on Learning Representations*.

Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213. Curran Associates, Inc.

Xiang Li, Yiqun Yao, Xin Jiang, Xuezhi Fang, Chao Wang, Xinzhang Liu, Zihan Wang, Yu Zhao, Xin Wang, Yuyao Huang, Shuangyong Song, Yongxiang Li, Zhongjiang He, Xuelong Li, and 1 others. 2024. Tele-FLM technical report. *Preprint*, arXiv:2404.16645.

Zhongqiu Li, Shiquan Wang, Ruiyu Fang, Mengjiao Bao, Zhenhe Wu, Shuangyong Song, Yongxiang Li, and Xuelong Li. 2025. MR-UIE: Multi-perspective reasoning with reinforcement learning for universal information extraction. *Vicinagearth*, 2(1).

Xinzhang Liu, Chao Wang, Zhihao Yang, Zhuo Jiang, Xuncheng Zhao, Haoran Wang, Lei Li, Dongdong He, Luobin Liu, Kaizhe Yuan, Han Gao, Zihan Wang, Yitong Yao, Sishi Xiong, Wenmin Deng, Haowei He, Kaidong Yu, Yu Zhao, Ruiyu Fang, and 35 others. 2025. Training report of telechat3-moe. *Preprint*, arXiv:2512.24157.

Uche Christina Okwuonu, Helen Kwipnchep Njoya, and Uzoigwe Tracy Peremoboere. 2025. Enhancing math reasoning ability of large language models via computation logic graphs. *Knowledge-Based Systems*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Barbara Plank, Dirk Hovy, and Anders Søgaard. 2014. Linguistically debatable or just plain wrong? In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 507–511, Baltimore, Maryland. Association for Computational Linguistics.

Qwen Team. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Parameswary Rasiah. 2010. A framework for the systematic analysis of evasion in parliamentary discourse. *Journal of Pragmatics*, 42(3):664–680.

Yiqing Shen and Dell Zhang. 2025. A survey of language-guided video object segmentation: from referring to reasoning. *Vicinagearth*, 2(1).

Lingling Shi, Haoyu Jin, Ruiyu Fang, Shuangyong Song, Jinsong Su, Yongxiang Li, and Xuelong Li. 2026. CEMT: Chain-of-thought enhanced machine translation. In *Findings of the Association for Computational Linguistics: ACL 2026*. Association for Computational Linguistics.

Konstantinos Thomas, Giorgos Filandrianos, Maria Lymperaiou, Chrysoula Zerva, and Giorgos Stamou. 2024. "I never said that": A dataset, taxonomy and baselines on response clarity classification. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 5204–5233, Miami, Florida, USA. Association for Computational Linguistics.

Konstantinos Thomas, Giorgos Filandrianos, Maria Lymperaiou, Chrysoula Zerva, and Giorgos Stamou. 2026. Semeval-2026 task 6: CLARITY – unmasking political question evasions. *Preprint*, arXiv:2603.14027.

Alexandra Uma, Tommaso Fornaciari, Dirk Hovy, Silviu Paun, Barbara Plank, and Massimo Poesio. 2021. Learning from disagreement: A survey. *The Journal of Artificial Intelligence Research*, 72:1385–1470.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023a. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2609–2634, Toronto, Canada. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. Self-consistency improves chain of thought reasoning in language models. *Preprint*, arXiv:2203.11171.

Zihan Wang, Xinzhang Liu, Shixuan Liu, Yitong Yao, Yuyao Huang, Zhongjiang He, Xuelong Li, and 1 others. 2024a. Telechat technical report. *Preprint*, arXiv:2401.03804.

Zihan Wang, Xinzhang Liu, Shixuan Liu, Yitong Yao, and 1 others. 2024b. TeleChat: An open-source bilingual large language model. In *Proceedings of the 10th SIGHAN Workshop on Chinese Language Processing (SIGHAN-10)*. Association for Computational Linguistics.

Zihan Wang, Xinzhang Liu, Yitong Yao, Chao Wang, Yu Zhao, Zhihao Yang, Wenmin Deng, Kaipeng Jia, Jiaxin Peng, Yuyao Huang, Sishi Xiong, Zhuo Jiang, Kaidong Yu, Xiaohui Hu, Fubei Yao, Ruiyu Fang, Zhuoru Jiang, Ruiting Song, Qiyi Xie, and 19 others. 2025. Technical report of telechat2, telechat2.5 and T1. *CoRR*, abs/2507.18013.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Di Wu, Ruiyu Fang, Liting Jiang, Shuangyong Song, Xiaomeng Huang, Shiquan Wang, Zhongqiu Li, Lingling Shi, Mengjiao Bao, Yongxiang Li, and Hao Huang. 2025. Multi-intent spoken language understanding: a survey of methods, trends, and challenges. *Vicinagearth*, 2(1).

Hongrui Xing, Xinzhang Liu, Zhuo Jiang, Zhihao Yang, Yitong Yao, Zihan Wang, and 1 others. 2025. LLMSR@XLLM25: A language model-based pipeline for structured reasoning data construction. In *Proceedings of the 1st Joint Workshop on Large Language Models and Structure Modeling (XLLM 2025)*. Association for Computational Linguistics.

Sishi Xiong, Dakai Wang, Yu Zhao, Jie Zhang, Changzai Pan, Haowei He, Xiangyu Li, Wenhan Chang, Xuelong Li, and 1 others. 2025a. TableReasoner: Advancing table reasoning framework with large language models. *Preprint*, arXiv:2507.08046.

Sishi Xiong, Dakai Wang, Yu Zhao, Jie Zhang, Changzai Pan, Haowei He, Xiangyu Li, Wenhan Chang, Xuelong Li, and 1 others. 2025b. TeleAI at SemEval-2025 task 8: Advancing table reasoning framework with large language models. In *Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025)*. Association for Computational Linguistics.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, volume 36, pages 11809–11822. Curran Associates, Inc.

Yiqun Yao, Xiang Li, Xin Jiang, and 1 others. 2024. 52B to 1T: Lessons learned via Tele-FLM series. *Preprint*, arXiv:2407.02783.

Wei Zhang, Hongcheng Guo, Jian Yang, and 1 others. 2024. Lemur: Log parsing with entropy sampling and chain-of-thought merging. *Preprint*, arXiv:2402.18205.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. 2024. LlamaFactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 400–410, Bangkok, Thailand. Association for Computational Linguistics.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. 2023. Least-to-most prompting enables complex reasoning in large language models. *Preprint*, arXiv:2205.10625.

A Extended Analysis

A.1 Evaluation Metric

We evaluate our system using Macro-F1, the official metric for both subtasks (Thomas et al., 2024).

SubTask 1 (single-label classification). We compute the standard Macro-F1 score by averaging the class-wise F1 scores.

SubTask 2 (multi-reference classification). The evaluation accommodates multiple valid gold labels per sample. Following the official evaluation scheme, a prediction p is considered a True Positive (TP) for class c if $p = c$ and c is present in the sample's reference set G. Crucially, a False Negative (FN) for a gold class $c \in G$ is only penalized if the model's prediction misses the entire reference set (i.e., $p \notin G$). If the model successfully predicts any valid label from G, no False Negatives are counted for the other unpredicted labels in G. We utilize the official scorer provided by the task organizers to compute the final metrics.

A.2 Label Distribution and Annotator Disagreement

Table 5 shows the label distribution of SubTask 2 classes in the training and development sets. There is a clear distribution shift between the two sets: *Explicit* accounts for 30.5% in the training set but drops to 25.9% in the development set; the proportions of *General* and *Implicit* in the development set are much higher than in the training set (rising from 11.2% and 14.2% to 18.6% and 18.9%, respectively), while *Dodging* slightly decreased (from 20.5% to 18.7%). This distribution shift means that classifiers built on training set frequency distributions may be systematically biased on the development set, especially tending to underestimate the *General* and *Implicit* classes. Furthermore, minority classes such as *Partial/half-answer* (2.0–2.3%) and *Clarification* (1.3–2.7%) are extremely sparse in both sets, posing a challenge for anchor selection in few-shot prompting—typical positive examples available for selection from the pool are highly limited.

Table 6 lists the most frequent annotator disagreement pairs in the development set. *General* appears in 4 of the top 5 disagreement pairs, covering approximately 38% of disagreement samples in total, indicating that the boundary of the *General* category itself lacks consistency among human annotators. Notably, the disagreements for

SubTask 2 Label	Train	Dev
Explicit	30.5%	25.9%
Dodging	20.5%	18.7%
Implicit	14.2%	18.9%
General	11.2%	18.6%
Deflection	11.1%	8.1%
Declining to answer	4.2%	3.6%
Claims ignorance	3.5%	2.9%
Clarification	2.7%	1.3%
Partial/half-answer	2.3%	2.0%

Table 5: SubTask 2 Label Distribution. Dev column aggregates votes from 3 annotators. Percentages may not sum to 100% due to rounding.

Explicit↔*General* (10.7%) and *Explicit*↔*Implicit* (8.8%) cross the boundaries of the SubTask 1 hierarchy (Clear Reply vs. Ambivalent), suggesting considerable subjectivity even on the basic judgment of "whether substantive information was provided." Disagreements in *Deflection*↔*Dodging* (4.9%) and *Dodging*↔*General* (9.1%) also explain why *Dodging* becomes the most difficult class for the system—when an answer neither explicitly refuses nor directly answers, the annotator's judgment between "strategic evasion" and "general answering" often depends on subjective inference of the respondent's intent.

Annotator Disagreement Pair	Sample %
General ↔ Implicit	12.0%
Explicit ↔ General	10.7%
Dodging ↔ General	9.1%
Explicit ↔ Implicit	8.8%
Deflection ↔ General	6.2%
Deflection ↔ Dodging	4.9%

Table 6: Top annotator disagreement pairs on dev (percentage of samples where at least one pair of annotators disagreed).

A.3 SubTask 1 Per-Class Analysis

Table 7 shows per-class F1 for the three SubTask 1 classes across system variants. *Clear Non-Reply* remains stable across all variants (0.851–0.870), due to the Gate routing strategy's high precision on Non-Reply classes. *Ambivalent* is the largest class (206 support samples) and varies only slightly across variants ($\pm$0.005), indicating robust judgments on this class. *Clear Reply* fluctuates more: after introducing Refined label definitions, its F1 drops from 0.699 to 0.652, but recovers to 0.693 in the final CAMSR-CoT version. This fluctuation mainly comes from boundary changes between

SubTask 1 label	Supp.	Base	+ Refined definitions	Full (Ours)
Ambivalent	206	0.883	0.878	0.873
Clear Non-Reply	23	0.851	0.870	0.870
Clear Reply	79	0.699	0.652	0.693
Macro-F1	–	0.811	0.800	**0.812**

Table 7: SubTask 1 per-class F1 on dev. Columns compare our Base pipeline, adding refined definitions, and the full system.

Explicit↔General/Implicit: when more refined definitions shift the classification bias in Step 3, some samples originally correctly predicted as *Explicit* may be reclassified into Ambivalent-side classes, and vice versa. Overall, SubTask 1 Macro-F1 varies within 0.800–0.812, showing that the three-stage routing framework is relatively stable for coarse-grained classification.

A.4 SubTask 2 Per-Class Analysis

Table 8 presents the full Precision, Recall, and F1 breakdown for the final CAMSR-CoT system on dev. Table 9 further compares per-class F1 scores across system variants.

Label	P	R	F1
SubTask 1: Clarity			
Clear Reply	0.716	0.671	0.693
Clear Non-Reply	0.870	0.870	0.870
Ambivalent	0.863	0.883	0.873
Macro-F1	–	–	*0.812*
SubTask 2: Evasion Details			
Explicit	0.824	0.656	0.731
Implicit	0.551	0.606	0.577
Partial/half-answer	0.286	0.333	0.308
General	0.556	0.494	0.523
Deflection	0.395	0.500	0.441
Dodging	0.750	0.179	0.289
Declining to answer	1.000	0.769	0.870
Claims ignorance	1.000	0.692	0.818
Clarification	1.000	1.000	1.000
Macro-F1	–	–	*0.617*

Table 8: Per-class performance of CAMSR-CoT on dev (SubTask 1 & SubTask 2).

Non-Reply classes. *Declining to answer* (0.800→0.870), *Claims ignorance* (0.783→0.818), and *Clarification* (always 1.000) remain strong across variants. These classes exhibit clear linguistic signals (e.g., explicit refusal phrases, clarification questions), which the Gate routing strategy captures effectively.

SubTask 2 label	Base	+ Refined definitions	Full (Ours)
Explicit	0.659	0.620	0.731
Implicit	0.194	0.342	0.577
Partial/half-answer	0.235	0.083	0.308
General	0.189	0.273	0.523
Deflection	0.442	0.466	0.441
Dodging	0.222	0.276	0.289
Declining to answer	0.800	0.833	0.870
Claims ignorance	0.783	0.783	0.818
Clarification	1.000	1.000	1.000
Macro-F1	0.503	0.520	**0.617**

Table 9: SubTask 2 per-class F1 on dev (official multi-reference scorer).

Implicit and General. *Implicit* (F1: 0.194→0.577) and *General* (F1: 0.189→0.523) are the largest beneficiaries. This improvement stems from two complementary factors: (1) refined label definitions operationalize the distinction between "relevant but insufficient information" (*Implicit*) and "general talk that avoids the specific question" (*General*); (2) Step 3 boundary examples present contrastive cases of these two classes, helping the model distinguish "implicit responses" from "surface-related but substantively evasive" answers.

Dodging. *Dodging* remains low (0.222→0.276→0.289), with much smaller gains than *Implicit* and *General*. This bottleneck reflects not only model limitations but also intrinsic annotation ambiguity: in the development set's multi-reference annotations, *Dodging* has very high disagreement rates with *General* (9.1%) and *Deflection* (4.9%). When a respondent neither answers directly nor explicitly refuses, but instead adopts an "indirect strategy", annotators often disagree on the intent.

Partial/half-answer. With only 6 support samples on dev, *Partial/half-answer* is highly sensitive to individual predictions. Its F1 drops from 0.235 to 0.083 after introducing refined label definitions, but recovers to 0.308 in the final system.

A.5 Inference Cost Estimation

Table 10 reports the call count and estimated token consumption of each stage on the 308-sample development set. Token counts are approximate and serve as comparative cost indicators.

The dominant cost driver is Step 3, which handles 86.7% of samples with the longest prompts (due to 6 anchor examples, 10 boundary examples,

Stage	Calls	Rate	Input (tokens)	Output (tokens)
Step 1 (Gate)	308	100.0%	≈3,300	≈210
Step 2 (Correction)	23	7.5%	≈3,000	≈300
Step 3 (Classification)	267	86.7%	≈4,300	≈170
Weighted avg / sample	**1.94**	—	≈7,300	≈380
Total (308 samples)	598	—	≈2.25M	≈0.12M

Table 10: Per-stage API call counts and estimated token consumption on the development set (n=308). 18 samples (5.8%) are direct exits after Step 1 (high-confidence Non-Reply) and incur no further calls.

and 6-class definitions). The Step 2 correction branch, despite being triggered on only 7.5% of samples, has moderate prompt length because it includes 9-class definitions plus a gate-conditioned confusion guide and prior analysis block. Overall, the framework requires approximately 7,700 tokens per sample on average (roughly 1.94× the cost of a single-step 9-class CoT prompt), with the additional cost concentrated in the richer few-shot context of Step 3.

A.6 Extended Official Leaderboard

Table 11 presents the full Top-10 / Top-11 official hidden evaluation results for both SubTask 1 and SubTask 2. Note that the team *ChulaNLP* appears under the username *moswisarut* on the original Codabench leaderboard.

	SubTask 1		SubTask 2	
Rank	Team	F1	Team	F1
1	**TeleAI (Ours)**	**0.89**	**TeleAI (Ours)**	**0.68**
2	AsymVerify	0.85	ChulaNLP†	0.61
3	CSE-UOI	0.85	CLaC @ CLARITY	0.59
4	Rasende Rakete	0.83	Rasende Rakete	0.59
5	Evaluators	0.83	YNU-HPCC	0.59
6	YNU-HPCC	0.83	pressprexx	0.58
7	ChulaNLP†	0.82	CSE-UOI	0.58
8	tahamunawar	0.81	ttda704	0.56
9	CLaC @ CLARITY	0.80	Evaluators	0.54
10	SpinDetector	0.80	gsdeyson	0.52
11	gabriel_stefan	0.80		

Table 11: Extended official hidden evaluation leaderboard. †*ChulaNLP* appears as *moswisarut* on the Codabench leaderboard.

A.7 Additional Ablation: Boundary Count and Confusion Guide

Sensitivity to Boundary Count. We vary the number of boundary counter-examples B in Step 3 while keeping all other components fixed: B=8 yields 0.606, B=10 yields **0.617**, B=12 yields 0.565, and B=14 yields 0.565 (SubTask 2 Macro-F1 on dev). Performance peaks at B=10 and drops sharply beyond, suggesting that additional examples introduce noise or exceed the model's effective in-context utilization window, diluting the most informative contrasts.

Isolating the Confusion Guide. To measure the contribution of the confusion guide (explicit instructions for distinguishing commonly confused label pairs), we compare Step 3 variants: without the guide or few-shots (SubTask 2 F1 = 0.491), adding the guide with refined definitions (0.593), and further adding B=10 boundary examples (0.617). The confusion guide alone contributes +0.102, making it the single most impactful prompt component—larger than boundary examples (+0.024) or gate confidence routing alone.

A.8 Detailed Error Analysis

Confusion Matrix. Figure 2 shows the 9-class confusion matrix on dev (gold labels determined by majority vote among 3 annotators). The dominant error cluster is the *Dodging–General–Deflection* triangle: the system predicts *General* for gold *Dodging* in 18 cases, followed by *Deflection→General* (13), *Implicit→General* (11), and *Deflection→Dodging* (7). Errors involving at least one of these three labels account for 87 of 112 multi-reference errors (77.7%).

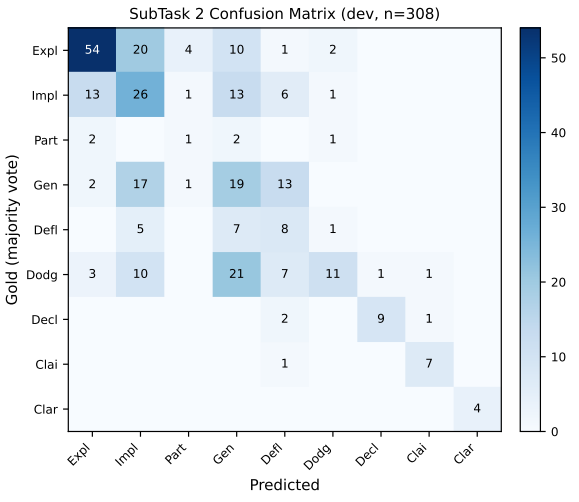

Figure 2: SubTask 2 confusion matrix on dev (n=308). Rows = gold labels (majority vote); columns = predictions. Color intensity is proportional to count.

Error Distribution by Routing Path. All 18 direct-exit samples (Step 1, high-confidence Non-Reply) are classified correctly (0% error rate), con-

firming the Gate's conservative confidence criterion. The Step 2 correction path has an error rate of 47.8% (11/23), reflecting the inherent difficulty of reclassifying borderline Non-Reply candidates in the full 9-class space. The Step 3 branch, handling 86.7% of samples, accounts for 101 of 112 errors (error rate 37.8%).

Case Study 1: *Dodging* Misclassified as *General*.

> **Sub-question:** *Other financial steps the President is prepared to take if the rate cut is deemed insufficient.*
>
> **Answer:** *Roger, I do not comment on the decisions made by the Fed. I will comment on Ben Bernanke; I think he's doing a fine job. The White House and the Congress are responsible for fiscal policy. The worst decision the Congress could make would be to raise taxes during this period. We don't need to raise taxes in order to fund budget priorities.* (truncated)
>
> **Prediction:** General **Gold:** [Deflection, Dodging, Dodging]

Model reasoning (excerpt):

```
The interviewee does not
directly address this specific
query.  Instead, they discuss
fiscal policy, the role of the
White House and Congress, and
their stance on raising taxes.
While the response is related to
the broader economic context, it
does not provide any information
about specific financial steps
the President might take.
```

Analysis: The model correctly identifies that the answer does not address the sub-question, but classifies it as *General* (on-topic but vague) rather than *Dodging* (strategic avoidance). The respondent strategically shifts to their preferred talking point (no tax raises), which 2 of 3 annotators interpret as deliberate evasion.

Case Study 2: *Explicit* Misclassified as *Implicit*.

> **Sub-question:** *What is the pressure point?*
>
> **Answer:** *Yes, I would hope it would be international pressure on not only Hizballah, the group of Hizballah within Lebanon, but also its sponsors. And that's the whole purpose of the United States working with allies and friends, is to send a clear message that sponsoring terror is unacceptable.*
>
> **Prediction:** Implicit **Gold:** [Explicit, Explicit, Explicit] (unanimous)

Model reasoning (excerpt):

```
This provides a specific
mechanism (international
pressure and targeting sponsors)
that can be inferred as the
pressure point, though it is not
explicitly labeled as such.
```

Analysis: The model is misled by the hedging language ("I would hope") into treating the answer as indirect, classifying it as *Implicit*. However, the answer explicitly names the pressure point (international pressure on Hizballah and its sponsors). All three annotators agree this is *Explicit*—the hedging signals politeness, not evasiveness.

A.9 Running Example: End-to-End Pipeline Walkthrough

We illustrate the full CAMSR-CoT pipeline on a development set sample (id=132).

Input.

> **Sub-question:** *What about military options against North Korea once diplomacy and the timetable have been exhausted?*
>
> **Answer:** *Well, diplomacy hasn't run its course. That's what I'm trying to explain to you a la the Sanger question. And we'll continue working to make sure that we give diplomacy a full opportunity to succeed. Yes, David [David Gregory, NBC News].*
>
> **Gold labels:** Deflection, Deflection, Deflection (unanimous)

Step 1: Gate (4-way + Confidence). The Gate produces the following Chain-of-Thought reasoning:

```
Step 1 - Identify sub-question
requirement:  The sub-question
asks about military options
against North Korea once
diplomacy and the timetable
have been exhausted.
Step 2 - Check key-information
override:  The answer discusses
diplomacy not having run its
course and the intention to
give it a full opportunity to
succeed, but it does NOT address
military options.  Key info is
missing.
Step 3 - Check Hard constraints:
The answer does not explicitly
ask for clarification, claim
ignorance, or decline to answer.
It focuses on the current state
of diplomacy.
Step 4 - Apply Tie-breakers:  No
relevant cues for Clarification,
Claims ignorance, or Declining
to answer are present.
```

Output: `<confidence>Medium</confidence>` `<label>OTHER</label>`

Since the Gate predicts `OTHER`, the sample is routed to **Step 3** (6-class subdivision).

Step 3: OTHER Branch Subdivision (6-way). The dynamically assembled prompt includes 6-class definitions, the Ambivalent confusion guide, 6 anchor examples, and 10 boundary counter-examples. The model reasons:

```
The sub-question asks about
military options against North
Korea once diplomacy and the
timetable have been exhausted.
The answer does not directly
address the military options but
instead focuses on the current
state of diplomacy, stating
that it hasn't run its course
yet.  This response does not
provide any information about
military options, nor does it
imply or infer anything about
them.  The answer stays within
the broader topic of North Korea
but shifts the focus to the
ongoing diplomatic efforts,
which is a different specific
point than the sub-question's
core focus on military options.
```

Output: `<label>Deflection</label>`

Final Result. SubTask 2 prediction: *Deflection* → SubTask 1 (via mapping): *Ambivalent*. Both match the gold labels. ✓

Routing Summary. This sample exemplifies the majority path (86.7% of dev samples): Step 1 identifies that the answer provides no Non-Reply signals (no refusal, no ignorance claim, no clarification request), routes to Step 3, where the boundary examples help distinguish *Deflection* (topic shift within the broader domain) from *Dodging* (avoidance without topical engagement) and *General* (on-topic but vague).

A.10 Pipeline Algorithm

Algorithm 1 provides pseudocode for the full CAMSR-CoT inference pipeline, including Gate routing logic and dynamic prompt assembly for each branch.

Key notation: $\mathcal{S}_{\text{NR}} = \{\text{Declining, Claims ignorance, Clarification}\}$; def_k = k-class label definitions; $\text{guide}(g)$ = confusion guide conditioned on Gate label; $\text{anc}_n/\text{bnd}_n$ = n anchor/boundary few-shot examples; $\text{prior}(g, c)$ = Gate label and confidence as prior context for Step 2. BUILD dynamically assembles these components (see Appendix B.2), and CoT-REASON performs Chain-of-Thought inference with the assembled prompt.

Algorithm 1 CAMSR-CoT Inference Pipeline

Require: Sample $x = (\text{sub-question}, \text{answer})$
Ensure: SubTask 2 label $\hat{y}$, SubTask 1 label $\hat{y}_1$
// Step 1: Gate with Confidence
$(g, c) \leftarrow \text{GATE}(x)$ ▷ 4-way + H/M/L
if $g \in \mathcal{S}_{\text{NR}}$ **and** $c = \text{High}$ **then**
  $\hat{y} \leftarrow g$ ▷ Direct exit
else if $g \in \mathcal{S}_{\text{NR}}$ **and** $c \neq \text{High}$ **then**
// Step 2: Non-Reply Correction (9-way)
  $P \leftarrow \text{BUILD}(\text{def}_9, \text{guide}(g), \text{anc}_2, \text{bnd}_3, \text{prior}(g, c))$
  $\hat{y} \leftarrow \text{CoT-REASON}(x, P)$
else ▷ $g = \text{OTHER}$
// Step 3: OTHER Subdivision (6-way)
  $P \leftarrow \text{BUILD}(\text{def}_6, \text{guide}_{\text{amb}}, \text{anc}_6, \text{bnd}_{10})$
  $\hat{y} \leftarrow \text{CoT-REASON}(x, P)$
end if
$\hat{y}_1 \leftarrow \text{MAP}(\hat{y})$ ▷ SubTask 2→1
return $(\hat{y}, \hat{y}_1)$

A.11 Supervised Fine-Tuning Details

This section describes the two supervised fine-tuning (SFT) approaches reported in Table 2: *Direct SFT* and *SFT + Distillation*. Both use Qwen2.5-7B-Instruct (Qwen Team, 2025) as the base model and are trained with LoRA (Hu et al., 2022) via the LLaMA-Factory framework (Zheng et al., 2024).

Training Configuration. Table 12 lists the shared training hyperparameters. Both variants use the same configuration to ensure a fair comparison.

Hyperparameter	Value
Base model	Qwen2.5-7B-Instruct
Fine-tuning method	LoRA
LoRA rank / alpha / dropout	8 / 16 / 0.05
LoRA target modules	all linear layers
Learning rate	1×10^{-4}
LR scheduler	Cosine (warmup ratio 0.1)
Training epochs	3
Effective batch size	32 (multi-GPU)
Precision	FP16 mixed precision
Optimizer	AdamW
Max sequence length	3,072 tokens

Table 12: Shared training hyperparameters for all SFT experiments.

Direct SFT. The direct SFT variant trains the model to predict a single digit (0–8), where each index maps to one of the 9 SubTask 2 labels. Training data is constructed from the 3,448 training samples using majority-vote labels from the original annotations. The instruction template (shown below) provides the index-to-label mapping alongside the classification directive. At inference, the model's

output digit is mapped back to the corresponding label name.

Instruction template:

```
Classify the interviewee's response to the Sub-
    question into ONE label index (0-8).
Judge relevance ONLY against the Sub-question.
Output ONLY ONE digit from 0 to 8. No other text.

0 Explicit
1 Implicit
2 Dodging
3 Deflection
4 Partial/half-answer
5 General
6 Declining to answer
7 Claims ignorance
8 Clarification
```

The input is formatted as a structured block containing the interview question, interview answer, and sub-question (see our code repository for the full template).

Knowledge Distillation via CoT Fusion. The distillation variant uses the CAMSR-CoT pipeline outputs as teacher data, generated via a two-step process:

1. **Teacher Inference.** We run the full CAMSR-CoT pipeline (Step 1 → Step 2 or Step 3) using Qwen3-235B-A22B (Qwen Team, 2025) as the backbone on all 3,448 training samples. For each sample, we collect per-step labels, confidence scores, and Chain-of-Thought reasoning traces.

2. **CoT Fusion.** The multi-step traces are consolidated into a single reasoning paragraph using a dedicated fusion prompt (shown below). The fusion model (also Qwen3-235B-A22B) produces a concise 2–6 sentence justification that is consistent with the predicted label, references specific content from the interview answer, and does not expose the pipeline structure. Only samples where the teacher's prediction matches the gold label are retained for training.

The fused CoT paragraphs paired with their corresponding labels form the distillation training set. The student model learns to generate the reasoning trace before producing the final label, allowing it to internalize the teacher's multi-stage decisions within a single inference pass.

CoT Fusion Prompt (abbreviated; sample-specific fields omitted):

```
You are given one sample with Step1/Step2/Step3
    traces and the final decided labels.
Write ONE concise reasoning paragraph that
    JUSTIFIES the final labels.

Hard constraints:
1) Output ONLY:
<analysis>...</analysis>

2) The reasoning MUST be consistent with:
final_task2={final_task2}
final_task1={final_task1}
You MUST NOT say the final label is wrong,
    misleading, or should be another label.

3) Do NOT mention the pipeline, steps, traces, "
    the model", "guidelines",
   "few-shot", or any external reminders.
   Do NOT mention "Step1/Step2/Step3", "
    decided_step", or "OTHER".

4) If step1_label is OTHER, treat it as:
   "non-non-reply labels (Explicit/Implicit/
    Partial/General/Deflection/Dodging)".
   (But do NOT mention this mapping in the output.)


5) Keep it concise: 2-6 sentences total.
   At least one sentence must point to specific
    content from the interview_answer
   (quote a short phrase or clearly paraphrase a
    concrete part).

6) Optional: you may include ONE brief contrast
    with a nearby label
   (e.g., "This is X rather than Y because ..."),
   but it must still support the final_task2.
```

B Prompt Templates and Few-Shot Library

This appendix provides the full prompt templates, label definitions, confusion guides, and few-shot libraries used in each stage of CAMSR-CoT. All components are dynamically assembled during inference based on the current branch (Step 1/2/3) and the Gate's output.

B.1 Step 1: Confidence-Aware Gate Prompt

The Step 1 Gate prompt is responsible for the preliminary 4-class decision (3 Non-Reply + OTHER) and outputting confidence (High/Medium/Low). Key design points include: (1) defining necessary and sufficient conditions for each Non-Reply class (e.g., *Declining to answer* requires an explicit refusal or deferral); (2) operationalizing confidence levels—High is permitted only when the answer is short, unambiguous, and contains no substantive information; (3) requiring Chain-of-Thought analysis before label output to reduce premature judgments based on surface cues.

```
You are given an interview segment (interviewer
    question(s) + interviewee answer) and a
    specific sub-question extracted from it.

Task:
Decide whether the answer is a Non-Reply of one of
     these types.
```

```
If not, output OTHER.

IMPORTANT:
- Judge relevance ONLY against the Sub-question (ignore other questions in the interview).
- Think step by step and write your reasoning inside <analysis>...</analysis>.
- Then output a confidence level inside <confidence>...</confidence>.
- Finally output the final label inside <label>...</label>.
- The <label> must contain ONLY the label text (no punctuation, no extra words).
- Output EXACTLY one label from:
Clarification | Claims ignorance | Declining to answer | OTHER

Hard constraints (must follow):
A) Clarification:
- You may output Clarification ONLY if the answer explicitly asks for clarification / says it does not understand the sub-question / asks a counter-question about a specific component of the question.
- Typical cues include: "What do you mean?", "Which one?", "Clarify", "I don't understand", "You mean X?", "On X?", "Are you asking about X?"

B) Claims ignorance:
- You may output Claims ignorance ONLY if the answer explicitly states lack of knowledge / lack of information / inability to recall / not having reviewed.
- Typical cues include: "I don't know", "I'm not sure", "I can't recall", "I haven't seen/reviewed", "I don't have the details", "I need to check".

C) Declining to answer:
- You may output Declining to answer ONLY if the answer explicitly refuses OR explicitly defers answering now.
- Typical cues include: "No comment", "I can't comment", "I won't discuss that", "Not at liberty", "Too early to say", "I'll let X address that", "I'm not going to tell you", "wait for the report".

Key-information override (CRITICAL):
- If the answer provides ANY of the key information required by the Sub-question (even partially), you MUST output OTHER.
  (In that case it is not a Non-Reply label.)

Tie-breakers (strict):
1) If Clarification cue is present and key info is missing -> Clarification.
2) Else if refusal/deferral cue is present and key info is missing -> Declining to answer.
3) Else if lack-of-knowledge cue is present and key info is missing -> Claims ignorance.
4) Else -> OTHER.

Confidence Level Guide:
<confidence>High</confidence>:
- The cue is explicit and unambiguous (e.g., "No comment", "I don't know").
- The answer is SHORT and consists MOSTLY of the refusal/ignorance/clarification.
- You are 100% sure it fits the definition and contains NO key info.
- There is NO possibility of it being Implicit or Partial answer.

<confidence>Medium</confidence>:
- It fits the definition, but the phrasing is less direct or mixed with other content.
- The refusal/ignorance is embedded in a longer explanation, or followed by "but...", "however...", or acts as a pivot.
- It might be a border-line case (e.g., "It's hard to say right now").

<confidence>Low</confidence>:
- It has some features of a Non-Reply, but you are hesitant.
- It might arguably be providing some information (Partial/Implicit).
- Use this if you want to flag it as a potential Non-Reply but are not sure.

### Examples ###

Example 1 (Short explicit refusal):
Interview Question: Q. You talked about middle class tax cuts on the campaign trail. How will you get Democrats to support that policy?
Interview Answer: You have to ask them.
Sub-question: How will you get Democrats to support that policy?
<analysis>
Step 1 - Identify sub-question requirement: The sub-question asks HOW the speaker will get Democrats to support a policy.
Step 2 - Check key-information override: The answer "You have to ask them" provides NO information about how to get Democratic support. Key info is missing.
Step 3 - Check Hard constraints: The phrase "You have to ask them" redirects the question to others, which is a deferral cue (C).
Step 4 - Apply Tie-breakers: Refusal/deferral cue is present + key info is missing -> Declining to answer.
</analysis>
<confidence>High</confidence>
<label>Declining to answer</label>

Example 2 (Refusal embedded in longer answer):
Interview Question: Q. Mr. President, we know that you talked about Iran and North Korea. Did you ask Russia to take specific steps, for example with Iran, to agree to U.N. sanctions?
Interview Answer: We strategized on both issues... yes, we talked about the U.N. Security Council resolution. And, no, I'm not going to tell you the particulars about the conversation. I will tell you, however, that there is common agreement that we need to get something done at the U.N...
Sub-question: Did you ask Russia to take specific steps, for example with Iran, to agree to U.N. sanctions?
<analysis>
Step 1 - Identify sub-question requirement: The sub-question asks whether SPECIFIC STEPS were requested of Russia regarding Iran/U.N. sanctions.
Step 2 - Check key-information override: The answer mentions "we talked about the U.N. Security Council resolution" and "common agreement" but does NOT reveal what specific steps were asked. Key info is missing.
Step 3 - Check Hard constraints: The phrase "I'm not going to tell you the particulars" is an explicit refusal cue (C).
Step 4 - Apply Tie-breakers: Refusal cue is present + key info is missing -> Declining to answer.
Note: The answer is long and provides context/pivot ("I will tell you, however..."). Although it is Declining, the extra context makes it safer to flag as Medium confidence so downstream models can double-check for Deflection.
</analysis>
<confidence>Medium</confidence>
<label>Declining to answer</label>

Example 3 (Deferral to future report):
Interview Question: Q. Are you prepared to pass on the fate of the war to the next President?
Interview Answer: David Petraeus, the general on the ground, will be bringing his recommendations back to the Congress on or about September the 15th. And I think it's going to be very important for all of us to wait for him to report... I don't want to prejudge what David is going to say.
Sub-question: Are you prepared to pass on the fate of the war to the next President?
```

```
<analysis>
Step 1 - Identify sub-question requirement: The
     sub-question asks whether the speaker is
     PREPARED to pass the war's fate to the next
     President (yes/no + reasoning).
Step 2 - Check key-information override: The
     answer discusses waiting for Petraeus's
     report but does NOT state whether the speaker
      is prepared to pass on the war. Key info is
     missing.
Step 3 - Check Hard constraints: The phrases "wait
      for him to report" and "I don't want to
     prejudge" are deferral cues (C).
Step 4 - Apply Tie-breakers: Deferral cue is
     present + key info is missing -> Declining to
      answer.
Note: Discussing related context (Petraeus report)
      makes this a contextual refusal.
</analysis>
<confidence>Medium</confidence>
<label>Declining to answer</label>

Example 4 (Explicit lack of knowledge):
Interview Question: Q. ----but shouldn't the
     Iranian regime know that there are
     consequences?
Interview Answer: I answered the question, Chuck,
     which is that we don't yet know how this is
     going to play out. Okay? Jake Tapper [ABC
     News].
Sub-question: Shouldn't the Iranian regime know
     that there are consequences?
<analysis>
Step 1 - Identify sub-question requirement: The
     sub-question asks whether Iran SHOULD know
     about consequences (opinion/stance).
Step 2 - Check key-information override: The
     answer does NOT provide a stance on whether
     Iran should know. Key info is missing.
Step 3 - Check Hard constraints: The phrase "we
     don't yet know how this is going to play out"
      is a lack-of-knowledge cue (B).
Step 4 - Apply Tie-breakers: Lack-of-knowledge cue
      is present + key info is missing -> Claims
     ignorance.
</analysis>
<confidence>High</confidence>
<label>Claims ignorance</label>

Example 5 (Deferral to future announcement - mixed
      cues):
Interview Question: Q. So you could be going to
     Charlotte?
Interview Answer: Anybody have any ideas? We'll be
      announcing it very soon.
Sub-question: Could you be going to Charlotte?
<analysis>
Step 1 - Identify sub-question requirement: The
     sub-question asks whether the speaker COULD
     be going to Charlotte (yes/no/maybe).
Step 2 - Check key-information override: The
     answer does NOT confirm or deny going to
     Charlotte. Key info is missing.
Step 3 - Check Hard constraints: "Anybody have any
      ideas?" suggests lack of knowledge (B). "We'
     ll be announcing it very soon" is a deferral
     cue (C).
Step 4 - Apply Tie-breakers: Both B and C cues
     present. Per tie-breaker order, check refusal
     /deferral first -> but the primary cue here
     is lack of knowledge/uncertainty implied by "
     Anybody have any ideas?".
</analysis>
<confidence>Medium</confidence>
<label>Claims ignorance</label>

Example 6 (Request to repeat - didn't hear):
Interview Question: Q. So you're inclined to pull
     the troops out?
Interview Answer: Say it?
Sub-question: Inclination towards pulling the
     troops out.
<analysis>
Step 1 - Identify sub-question requirement: The
     sub-question asks about the speaker's
     inclination on troops.
Step 2 - Check key-information override: The
     answer provides NO information about troops.
     Key info is missing.
Step 3 - Check Hard constraints: "Say it?" is a
     request to repeat the question, a
     clarification cue (A).
Step 4 - Apply Tie-breakers: Clarification cue is
     present + key info is missing ->
     Clarification.
</analysis>
<confidence>High</confidence>
<label>Clarification</label>

Example 7 (Couldn't hear):
Interview Question: Q. With the NHS? With the NHS?
      Should the NHS be on the table, sir?
Interview Answer: I can't hear him. What?
Sub-question: Should the NHS be on the table?
<analysis>
Step 1 - Identify sub-question requirement: The
     sub-question asks whether the NHS should be
     on the table (yes/no + reasoning).
Step 2 - Check key-information override: The
     answer provides NO information about the NHS.
      Key info is missing.
Step 3 - Check Hard constraints: "I can't hear him
     . What?" explicitly indicates the speaker
     didn't hear, a clarification cue (A).
Step 4 - Apply Tie-breakers: Clarification cue is
     present + key info is missing ->
     Clarification.
</analysis>
<confidence>High</confidence>
<label>Clarification</label>

Example 8 (Short counter-question/topic
     clarification):
Interview Question: Q. In your previous
     conversations with Prime Minister Maliki,
     have you been confident that he shares your
     view on Iraq?
Interview Answer: On Iran?
Sub-question: Did you feel confident that he
     shares your view on Iraq?
<analysis>
Step 1 - Identify sub-question requirement: The
     sub-question asks about confidence regarding
     Maliki's view on Iraq.
Step 2 - Check key-information override: The
     answer provides NO information about Iraq or
     the speaker's confidence. Key info is missing.

Step 3 - Check Hard constraints: "On Iran?" is a
     counter-question used to clarify the subject
     of the sub-question, which is a clarification
      cue (A).
Step 4 - Apply Tie-breakers: Clarification cue is
     present + key info is missing ->
     Clarification.
</analysis>
<confidence>High</confidence>
<label>Clarification</label>

### Part of the interview ###
{interview_question}

{interview_answer}

### Sub-question ###
{sub_question}

Output format (must follow exactly):
<analysis>...</analysis>
<confidence>High/Medium/Low</confidence>
<label>LABEL</label>
```

B.2 Base Dynamic Prompt Template (Step 2/3)

Step 2 and Step 3 share a base dynamic template structured as: System Instructions → Label Definitions → Confusion Guide → Few-shot Examples (Anchors + Boundaries) → Input Sample → CoT

Reasoning Format Requirement. During assembly, the Prompt Builder performs the following substitutions based on branch type: (1) Label Space—Step 2 uses full 9-class definitions, Step 3 uses 6-class definitions (excluding the three Non-Reply classes); (2) Confusion Guide—Step 2 dynamically selects discrimination points relevant to the Gate label, Step 3 uses a generic confusion guide for the Ambivalent branch; (3) Few-shot Quantity—Step 2 sets B=3 boundaries, Step 3 sets B=10 boundaries to cover more confusion pairs.

```
You are given an interview segment (interviewer
    question(s) + interviewee answer) and a
    specific sub-question extracted from it.

Task:
Classify the answer into one of the evasion types
    listed in the definitions below (9 for Step 2
     / 6 for Step 3, as specified by the
    definitions block).

{prior_analysis_block}

{definitions_block}

{confusion_guidelines_block}

{fewshots_block}

### Part of the interview ###
{interview_question}

{interview_answer}

### Sub-question ###
{sub_question}

Output format (must follow exactly):
<analysis>
Step 1 - Identify what the sub-question is asking
    for (key information required).
Step 2 - Check if the answer provides that key
    information (fully, partially, or not at all).

Step 3 - If key info is provided: determine
    Explicit vs Implicit vs Partial.
Step 4 - If key info is missing: determine which
    evasion type best fits.
</analysis>
<confidence>High/Medium/Low</confidence>
<label>LABEL</label>

Confidence Level Guide:
- **High**: The label clearly fits; no ambiguity.
- **Medium**: The label is reasonable, but there's
     some overlap with another category (e.g.,
    General vs Deflection, Implicit vs Partial).
- **Low**: Uncertain; could arguably be multiple
    labels.
```

B.3 Label Definitions (9-class / 6-class)

Label definitions are the core component of the prompt. We significantly expanded upon the concise definitions in the original paper (Thomas et al., 2024) to form a set of refined, operational label definitions. Principles for expansion include: (1) Providing necessary and sufficient conditions rather than just descriptive text, e.g., the definition of *Implicit* explicitly requires "the answer provides information from which a response to the sub-question can be reasonably inferred, but does not state it directly using the same terms or structure as the question"; (2) Appending "Distinction from Similar Classes" notes at the end of each definition, e.g., emphasizing in the *General* definition that "if the answer contains specific inferable information, it should be classified as Implicit even if phrased generally"; (3) Providing more specific criteria for minority classes (e.g., *Partial/half-answer*) to reduce misclassification. The full 9-class refined definitions used in Step 2 are listed below. The 6-class definitions (Step 3) follow the same structure with the three Non-Reply classes removed; the complete version is available in our code repository.

```
### 9-Class Evasion Type Definitions ###

Output EXACTLY one label from:
Explicit | Implicit | Partial/half-answer |
    General | Deflection | Dodging | Declining to
     answer | Claims ignorance | Clarification

 Operational decision tree (MUST follow):

Step 1) Identify the Sub-question's core target
    and required information.
- State in one short sentence: "What key info is
    the Sub-question asking for?"
- If multiple components are required (A and B /
    multiple items), list the components.

Step 2) Topic alignment check.
- Does the answer address the Sub-question's core
    target (same entity/aspect) at all?
- If NO -> output Dodging (and stop).

Step 3) Check whether the answer provides ANY key
    information.
- If the answer provides NONE of the required key
    information -> go to Step 5 (Deflection vs
    Dodging).
- If the answer provides AT LEAST SOME on-target
    information relevant to the asked aspect ->
    go to Step 4.

Step 4) Choose among Explicit / Implicit / Partial
    /half-answer / General.

4a) Partial/half-answer:
- If multiple components are required AND the
    answer clearly covers at least one component
    but misses at least one other -> Partial/half
    -answer.

4b) Explicit:
- Output Explicit ONLY if the requested key
    information is explicitly stated in the
    expected form.
- For Yes/No questions: Must contain "Yes", "No",
    "Certainly", "Absolutely", "I don't think so"
     OR a direct statement of the answer (e.g., "
    He has not changed").
  * EXCEPTION: If "Yes/No" is followed by a
    condition that effectively revokes the answer
     ("Yes, if X happens..."), treat as Partial/
    General.
  * EXCEPTION: If "Yes/No" responds to a rephrased
     question ("Let me ask: is X true? Yes..."),
    treat based on relevance to the ORIGINAL Sub-
    question.
- For WH-questions: Must provide the specific name
    , date, number, or entity asked for.

4c) Implicit (STRICT):
```

```
- Output Implicit if the answer does not use the
    explicit form but logically implies a
    specific answer to the Sub-question.
- You must be able to restate the implied answer
    in ONE short sentence with high confidence.
- Example: Q="Has he changed?" A="He is the same
    person." -> Implies "No". (This is Implicit,
    not Explicit).
- If the implication depends on complex
    interpretation or is weak -> do NOT choose
    Implicit.

4d) General (DEFAULT when on-topic but vague):
- Output General if the answer stays on the asked
    aspect (same entity/aspect),
  provides some on-target information, but is too
    vague/high-level and lacks the requested
    specificity.
- There is NO clear pivot away from the asked
    aspect.

Step 5) Deflection vs Dodging (when key info is
    missing- This step applies to cases not
    covered by Non-Reply check):
- Output Deflection if the answer engages the core
     target but then pivots to a different point/
    aspect/frame/object
  instead of answering what is asked (no key
    information provided).
  Pivot signals may include: "but", "however", "
    what matters is", "the real issue is", "let
    me say", or a clear content shift.
  Necessary condition: you can point out what it
    pivoted to. If you cannot identify a clear
    pivot -> prefer Dodging (if unaligned) or
    General (if aligned).
- Otherwise output Dodging.

Tie-breakers (reduce Deflection/Implicit overuse):
- If unsure between Deflection vs General ->
    choose General unless the pivot is
    unmistakable.
- If unsure between Implicit vs General -> choose
    General unless you can restate a concrete
    implied answer.

Non-Reply Labels (Check these first if explicit
    cues exist):
- Declining to answer: Explicit refusal/deferral.
- Claims ignorance: Explicit lack of knowledge.
- Clarification: Explicit request to clarify/
    repeat.
```

B.4 Confusion Guides

Confusion guides provide structured discrimination rules targeting high-frequency misclassification pairs found in training set zero-shot reasoning. Each guide typically contains 3–5 rules, describing a condition "If ..., then classify as X instead of Y". For example, the confusion guide for *Declining to answer* targets common misclassifications such as *Declining→Dodging*, emphasizing "If the answer provides any content related to the sub-question topic (even indirectly), do not classify as Declining". Confusion guides are dynamically selected in Step 2 based on the Gate label to ensure only relevant discrimination rules are injected, avoiding excessive context length. As a representative example, the confusion guide for *Declining to answer* is shown below. The remaining confusion guides (*Claims ignorance*, *Clarification*, and the *Ambivalent* branch) follow the same structure and are available in our code repository.

```
### Confusion Guidelines: Declining to answer ###

"Declining to answer" requires an EXPLICIT refusal
     or deferral. Watch out for these common
    confusions:

**Declining vs Dodging**:
- Declining: "I won't comment on that" / "No
    comment" / "I'm not going to tell you" (
    EXPLICIT refusal)
- Dodging: Answering a completely different
    question without refusing (NO explicit
    refusal words)

**Declining vs Deflection**:
- Declining: "I'll let the report speak for itself
    " / "Wait for the announcement" (EXPLICIT
    deferral)
- Deflection: "That's a great question, but what's
     really important is..." (Pivot without
    refusing)

**Declining vs General**:
- Declining: "I can't discuss ongoing
    investigations" (EXPLICIT refusal with reason
    )
- General: "We're looking into it" / "It's a
    complex situation" (Vague but not refusing)

Rule of thumb: If there's no explicit refusal
    phrase, it's probably NOT Declining.

**Deferral of Details (Common Confusion)**:
- If a speaker says "I'll let [Someone else]
    discuss the details," but then provides a **
    general answer** or a **timeframe** himself
    in the same response, it is **NOT Declining**.
     It is likely **General** or **Explicit**.
- Only label as Declining if the *entire* relevant
     part of the answer is a referral/refusal.

**Special Note: Declining vs. Claims Ignorance (
    The "Report" boundary)**:
- **Declining**: "I am waiting for the report
    before I comment" / "I don't want to prejudge
     the outcome." -> **Focus is on the decision
    to WAIT/NOT COMMENT.**
- **Claims Ignorance**: "I haven't seen the report
     yet" / "We don't have that information." ->
    **Focus is on the ABSENCE of knowledge.**
- Even if both mention a report, look for the '
    refusal' vs 'lack of info' priority.
```

B.5 Few-Shot Library Used by the Prompt Builder

The few-shot library is divided into two categories: **Anchors** (Positive Examples) and **Boundaries** (Hard Negative Counter-Examples).

We mine **anchors** by selecting representative training samples that DeepSeek-V3 classifies correctly in a zero-shot setting. Selection prioritizes: (1) high confidence; (2) moderate length (avoiding too short or too long); (3) clear embodiment of core class features.

We mine **boundaries** from the confusion matrix of training set zero-shot reasoning. For each high-frequency misclassification pair (e.g., *Dodging→Declining*), we select misclassified samples along with their correct labels to form "Predicted Label vs. True Label" contrastive examples. Dur-

ing inference, the Prompt Builder annotates "Predicted: X, but correct label is Y" before each boundary example to provide an explicit correction signal. This "teaching by error" strategy ensures few-shot budget is precisely invested in decision boundaries prone to mistakes, rather than evenly distributed across all classes.

At runtime, the Prompt Builder selects from the library based on priority according to the current step configuration (K anchors, B boundaries). Step 2 uses K=2, B=3 (focusing on Non-Reply correction), Step 3 uses K=6, B=10 (broad coverage of confusion pairs among 6 classes). As a representative example, the anchor and boundary pools for *Declining to answer* are listed below. The complete libraries for all classes are available in our code repository.


```
{"id": "declining_anchor_1", "label": "Declining
    to answer", "type": "anchor", "
    interview_question": "Q. Could you give us a
    hint about the region or the countries?", "
    interview_answer": "Well, no, you're going to
    --it's going to be announced very shortly.
    Okay?", "sub_question": "Request for a hint
    about the region or the countries.", "
    reasoning": "Explicit deferral: 'no' + 'it's
    going to be announced very shortly' =
    refusing to give info now, deferring to
    future announcement."}
{"id": "declining_anchor_2", "label": "Declining
    to answer", "type": "anchor", "
    interview_question": "Q. It sounds like you'
    re moving closer to eliminating the
    filibuster. Is that correct?", "
    interview_answer": "I answered your question
    .", "sub_question": "It sounds like you're
    moving closer to eliminating the filibuster.
    Is that correct?", "reasoning": "Explicit
    refusal to elaborate: 'I answered your
    question' shuts down further inquiry without
    providing the requested confirmation."}
{"id": "declining_anchor_3", "label": "Declining
    to answer", "type": "anchor", "
    interview_question": "Q. Mr. President, we
    know that you talked about Iran and North
    Korea. Did you ask Russia to take specific
    steps?", "interview_answer": "We strategized
    on both issues... And, no, I'm not going to
    tell you the particulars about the
    conversation.", "sub_question": "Did you ask
    Russia to take specific steps?", "reasoning":
     "Explicit refusal: 'I'm not going to tell
    you the particulars' is a clear declining
    phrase."}
{"id": "declining_anchor_4", "label": "Declining
    to answer", "type": "anchor", "
    interview_question": "Q. Why would you agree
    with Senator McCain or Senator Obama?", "
    interview_answer": "No, I appreciate you
    trying to drag me in the '08 race... what I'm
     not going to do is jump right in the middle
    of a Presidential campaign. We'll let the
    candidates argue out their ideas.", "
    sub_question": "Why would you agree with
    Senator McCain or Senator Obama?", "reasoning
    ": "Explicit refusal to participate in the
    '08 race discussion. Uses 'not going to do'
    and 'let the candidates argue' as deferral."}
{"id": "declining_anchor_5", "label": "Declining
    to answer", "type": "anchor", "
    interview_question": "Q. Are you prepared to
    pass on the fate of the war to the next
    President?", "interview_answer": "[Long
    response] David Petraeus... will be bringing
    his recommendations... I think it's going to
    be very important for all of us to wait for
    him to report... I don't want to prejudge
    what David is going to say.", "sub_question":
     "Are you prepared to pass on the fate of the
     war to the next President?", "reasoning": "
    Timing deferral: Explicitly state 'wait for
    him to report' and 'don't want to prejudge'.
    This is choosing to decline a current answer
    in favor of a future event."}
```

```
{"id": "declining_boundary_1", "gold_label": "
    Dodging", "confused_as": "Declining", "type":
     "boundary", "interview_question": "Q. Can
    you back your statement up and provide
    specifics about how that would work?", "
    interview_answer": "Did you say, That's
    impressive? Did you actually use that term?",
     "sub_question": "The interviewer's request
    for President Trump to back up his statement
    and provide specific details.", "reasoning":
    "NOT Declining: There is no explicit refusal
    phrase. The speaker deflects by asking a
    counter-question, which is Dodging (ignoring
    the question entirely)."}
{"id": "declining_boundary_2", "gold_label": "
    Deflection", "confused_as": "Declining", "
    type": "boundary", "interview_question": "Q.
    So you believe they are aiding Syria?", "
    interview_answer": "It's a general statement
    that we expect them not to be proliferating
    .", "sub_question": "Do you believe they are
    aiding Syria?", "reasoning": "NOT Declining:
    No refusal phrase. The answer pivots to a
    general policy statement without addressing
    the yes/no question. This is Deflection."}
{"id": "declining_boundary_3", "gold_label": "
    General", "confused_as": "Declining", "type":
     "boundary", "interview_question": "Q. What
    about the payment to the Treasury?", "
    interview_answer": "Well, we're going to see
    about that. Amazingly, I find that you're not
     allowed to do that...", "sub_question": "
    What about the payment to the Treasury?", "
    reasoning": "NOT Declining: 'We're going to
    see about that' is vague but not a refusal.
    The speaker then provides context. This is
    General (on-topic but lacking specific answer
    )."}
{"id": "declining_boundary_4", "gold_label": "
    General/Explicit", "confused_as": "Declining
    ", "type": "boundary", "interview_question":
    "Q. When will we see this resolution?", "
    interview_answer": "I'll let Condi talk about
     the details of what she's going to do today
    ... we will work with our partners to get the
     resolution laid down as quickly as possible
    .", "sub_question": "When will we see this
    resolution?", "reasoning": "NOT Declining:
    Even though the speaker defers to someone
    else for 'details', they provide a general
    answer ('as quickly as possible') for the
    timeframe. A deferral of 'details' does not
    make the whole response a Declining if an on-
    topic general answer is provided."}
```