StateCloud at CQs-Gen 2025: Prompt Engineering for Critical Questions Generation

Jinghui Zhang and **Dongming Yang** * and **Binghuai Lin**
jinghui-19@tsinghua.org.cn, yangdongming@pku.edu.cn,
linbinghuai@gmail.com
China Telecom Cloud Technology Co., Ltd

Abstract

This paper presents StateCloud's submission to the Critical Questions Generation (CQs-Gen) shared task at the Argument Mining Workshop 2025. To generate high-quality critical questions from argumentative texts, we propose a framework that combines prompt engineering with few-shot learning to effectively guide generative models. Additionally, we ensemble outputs from diverse large language models (LLMs) to enhance accuracy. Notably, our approach achieved 3rd place in the competition, demonstrating the viability of prompt engineering strategies for argumentative tasks.

1 Introduction

Critical Questions (CQs) play a pivotal role in argumentation by challenging the validity, relevance, or sufficiency of claims. Automated generation of CQs from argumentative texts has emerged as a key task in computational argumentation, enabling systems to engage in nuanced discourse. The CQs-Gen shared task at the Argument Mining Workshop 2025 aims to advance this capability by developing a system model to produce high-quality, contextually relevant CQs.

In this paper, we present StateCloud's submission to the CQs-Gen task(Calvo Figueras et al., 2025). Our approach centers on prompt engineering to guide generative LLMs toward producing critical questions that adhere to domain-specific requirements. While fine-tuning LLMs is a common strategy, we prioritize few-shot learning with carefully curated prompts to leverage pre-trained knowledge efficiently. We further enhance accuracy by ensembling outputs from diverse state-of-the-art LLMs.

The main contributions of this work are:

- A systematic framework for prompt engineering tailored to argumentative CQs generation.
- Empirical validation of model ensembling for improving question accuracy.

Our system achieved 3rd place in the competition, demonstrating the effectiveness of prompt-driven strategies.

2 Related Work

2.1 Critical Question Generation in Argument Mining

CQs-Gen is a specialized task in computational argumentation that focuses on identifying and formulating questions that challenge the validity, relevance, or sufficiency of arguments.

With the advent of machine learning, supervised approaches (Nguyen and Litman, 2016); (Opitz and Frank, 2019) emerged, training classifiers or sequence-to-sequence models on annotated datasets. These methods improved generalization but required substantial labeled data, which is costly to obtain for argumentative tasks.

To address this limitation, researchers have explored transfer learning(Dutta et al., 2022); (Hua and Wang, 2022). Dutta et al. used web data for argumentative knowledge, adapting Transformers via Selective MLM (masking discourse markers instead of random tokens) and prompt-based relation prediction, reducing labeled data needs.

2.2 Prompt Engineering for Generative Tasks

Prompt engineering has emerged as a critical methodology for optimizing the performance of LLMs across diverse domains (Zhang et al., 2023);(Brown et al., 2020). Unlike traditional fine-tuning approaches that require extensive parameter updates, prompt engineering operates through carefully designed input formulations that guide LLMs to produce desired outputs without modifying their underlying architecture.

The concept of prompt engineering originated from observations that LLMs are highly sensitive

*Corresponding Author.

Proceedings of the 12th Argument Mining Workshop, pages 258–264

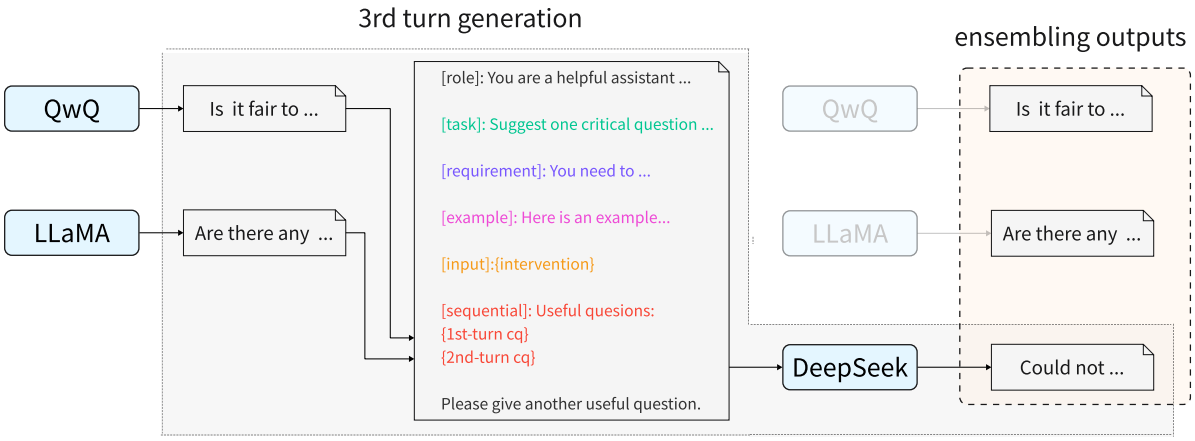

Figure 1: Our framework employs a sequential ensembling approach to integrate the outputs of different LLMs. The output from each preceding model is incorporated into the prompt template for the subsequent generation step.

to input phrasing. Seminal work by (Bsharat et al., 2024) demonstrated how subtle changes in prompt structure could yield dramatically different outputs in creative writing tasks. Their study presents principled guidelines for improving prompt quality, including techniques such as combining affirmative and negative directions, structured formatting, and role assignment.

Subsequent research has introduced more rigorous frameworks for prompt construction, including: Chain-of-Thought (CoT) prompting (Wei et al., 2022); Tree-of-Thoughts (ToT) prompting (Yao et al., 2023); (Long, 2023); Self-Refine prompting (Madaan et al., 2023).

However, robustness issues persist as model outputs remain highly sensitive to subtle prompt variations, necessitating more stable and transferable solutions.

3 Task Description

The CQs-Gen task involves generating meaningful CQs in response to an argumentative text. A dataset of real debate interventions, along with associated CQs, is provided. The validation dataset consists of 186 interventions.

The goal is to develop a system that takes an intervention as input and outputs exactly three CQs, all of which should be effective in challenging the arguments presented.

Each of the three CQs will be independently evaluated by computing the cosine similarity between the embeddings of generated CQs and reference CQs using sentence transformers [1], then assigning one of four labels: Useful, Not-Able-to-Evaluate, Unhelpful, or Invalid. If the similarity score exceeds a predefined threshold, the generated CQ will inherit the same label as its corresponding reference CQ. Otherwise, it will be labeled as Not-Able-to-Evaluate. Finally, the individual question evaluations will be aggregated into an overall score.

4 Methodology

4.1 Framework Overview

Our system generates one question at a time and combines model outputs to produce a specified number of questions. Our scripts are publicly available on GitHub. [2] The framework integrates these key components:

Prompt Engineering We constructed multiple templates with varying structures and linguistic styles, carefully designing the model's role and generation requirements within the prompts. We also evaluated the effectiveness of in-context learning.

Model Ensembling To maintain high-quality question generation while improving output stability, we aggregated outputs from multiple models including QwQ and others. We compared the effects of sequential versus parallel ensemble approaches.

4.2 Prompt Design

We designed four different prompts (shown in Appendix A), all specifying the model's role as a "helpful assistant with critical thinking skills." The variations include:

[1] https://huggingface.co/sentence-transformers/stsb-mpnet-base-v2

[2] https://github.com/QQJellyy/StateCloud-at-CQs-Gen-share-task.git

Models	USE	UNCERTAIN
Llama-3-8B	70.4 %	11.8 %
Llama-3.3-70B	75.8 %	9.1 %
qwen2.5-7B	71.0 %	14.5 %
qwen2.5-14B	64.5 %	18.3 %
qwen2.5-32B	71.0 %	12.9 %
qwen2.5-72B	76.3 %	5.4 %
QWQ-32b-32B	66.7 %	16.1 %
DeepSeek-R1-671B	61.3 %	21.0 %

Table 1: The performance of some open-source general large language models. USE denotes the number of Useful questions. UNCERTAIN denotes the number of Not-Able-to-Evaluate questions.

Zero-shot A zero-shot prompt containing only the target intervention and generation requirements.

Few-shot A few-shot prompt featuring an example intervention with corresponding helpful and unhelpful CQs.

Oral-expression A version with more conversational requirement phrasing to examine the impact of linguistic style.

Requirements-ahead A structurally modified prompt placing requirements earlier to investigate component ordering effects.

4.3 Model Ensembling

We selected n candidate open-source models and evaluated their performance using a calibration dataset. Two ensembling methods were implemented:

Parallel Ensemble The top 3 models each generate one CQ independently, with results combined directly. This configuration operates under the explicit assumption that model diversity inherently produces distinct question formulations, thus intentionally omitting deduplication steps.

Sequential Ensemble A single model generates an initial CQ. Subsequent CQs are produced iteratively by incorporating all previous results into the prompt (similar to few-shot learning). The sequential template is detailed in Appendix A.

5 Experiment Setup

5.1 Dataset

Our experiments utilized two data components.

- A randomly selected intervention from the sample set, paired with its corresponding useful and unhelpful question pairs for few-shot demonstration.

Prompt versions	USE	UNCERTAIN
zero-shot	74.7%	9.7%
requirements-ahead	75.3%	7.0%
oral-expression	75.8%	8.1%
few-shot	76.3%	5.4%

Table 2: The performance of prompt engineering. Here we use the Qwen2.5-72B model. USE denotes the number of Useful questions. UNCERTAIN denotes the number of Not-Able-to-Evaluate questions.

Prompt versions	Combination Strategy	USE	UNCERTAIN
Qwen + R1	parallel	68.8%	13.2%
Qwen→R1	sequential	74.5%	7.6%
Qwen + Qwen	parallel	76.4%	5.4%
Qwen→Qwen	sequential	77.2%	5.9%

Table 3: The performance of model ensembling. USE denotes the number of Useful questions. UNCERTAIN denotes the number of Not-Able-to-Evaluate questions. Qwen denotes Qwen2.5-72B-Instruct. R1 denotes DeepSeek-R1.

- The full validation set comprising 186 interventions, each annotated with multiple *useful*, *unhelpful*, and *invalid* CQs for evaluation. This structure enabled both effective few-shot learning and comprehensive evaluation of model performance.

5.2 Models

We selected eight state-of-the-art open-source models based on three criteria: model size, training data distribution and reasoning capability. Details of the selected models are provided in Tables 1. All models were inferred using HuggingFace Transformers with default generation configurations (temperature, top-p, etc.).

6 Results and Analysis

6.1 Model Comparison

The performance of general and reasoning LLMs on the validation set are presented in Tables 1, respectively. Since neither Unhelpful nor Invalid labels constitute effective challenges, we focus primarily on Useful and Not-Able-to-Evaluate labels to highlight valid or potentially valid critiques. The top-performing model was Qwen2.5-72B-Instruct, generating 142 useful CQs for 186 interventions, with LLaMA-3.3-70B closely following at 141 useful CQs.

Scaling Effects While larger models produced more useful CQs, consistent with expectations, the marginal gains were surprisingly small: both LLaMA and Qwen models at 7B/8B scales gen-

System	Models	USE(val)	UNCERTAIN(val)	USE(test)	UNCERTAIN(test)
System 1	Qwen2.5-72B→72B→72B	**76.2%**	6.1%	45.1%	5.9 %
System 2	Qwen2.5-72B + 32B +7B	72.8%	**10.9%**	42.2%	15.7 %
System 3	**Qwen2.5-72B + QwQ + DeepSeek-R1**	71.3%	10.4%	**47.1%**	**22.5** %

Table 4: Performance of the three final submitted systems on the validation and test sets. "→" denotes sequential ensembling, while "+" indicates parallel ensembling. **Bold** values indicate best performance across systems.

erated approximately 131 useful CQs, while their 70B/72B counterparts produced only about 10 additional useful CQs.

Reasoning Models Underperform Despite reasoning models' strong performance on benchmark tasks, they did not outperform general LLMs in useful CQ generation on validation set. Notably, reasoning models produced significantly more *Not-Able-to-Evaluate* CQs, suggesting they may generate more novel CQs beyond the annotation scope. This leaves open the possibility that their true capability might be underestimated by current evaluation metrics.

6.2 Prompt Design Analysis

Using Qwen2.5-72B-Instruct, we evaluated various prompt designs (Table 2), revealing several insights. The performance variation between prompts proved minimal (142 vs. 139 useful CQs, Δ=2%), significantly overshadowed by model selection impacts. Few-shot prompting demonstrated a clear trade-off: while increasing useful CQ counts, it simultaneously caused a 44% reduction in uncertain questions (from 18 to 10) while increasing unhelpful and invalid outputs.

6.3 Model Ensembling

Our framework implements model ensembling to generate multiple CQs for each intervention. Comparative results for generating two CQs are presented in Table 3, demonstrating the superior performance of sequential ensembling over parallel approaches. The sequential method shows particular effectiveness when applied to reasoning models such as R1, yielding a statistically significant increase in useful CQs generation (114 vs. 135 useful CQs, Δ=18%). Notably, this approach maintains its advantage even when employing identical models, suggesting that the contextual incorporation of previously generated CQs enhances subsequent question quality. This phenomenon indicates that exposing the model to its own outputs creates a beneficial self-refinement mechanism, where each generated question informs and improves subsequent outputs.

7 Submission

We evaluated multiple systems and selected the top three for final submission based on the number of Useful and Not-Able-to-Evaluate CQs generated. The results are presented in Table 4. System 1 employed a sequential approach, where Qwen2.5-72B generated three CQs in succession. System 2 used a parallel ensemble of Qwen2.5-72B, 32B, and 7B models. System 3 combined Qwen2.5-72B, QwQ, and DeepSeek R1 in parallel.

On the validation set, System 1 produced the highest number of Useful CQs, while Systems 2 and 3 generated more Not-Able-to-Evaluate CQs, indicating greater potential for diverse questioning. However, on the test set, System 3 achieved the highest counts for both types of CQs, demonstrating superior overall performance.

8 Conclusion

This paper presented StateCloud's comprehensive framework for the CQs-Gen shared task, integrating innovative prompt engineering with model ensemble techniques. We submitted three different systems, with System 3 emerging as our top performer, ultimately achieving 3rd place in the competition.

Our systematic evaluation yielded several key insights: (1) While larger models (e.g., Qwen2.5-72B-Instruct) achieved marginally better performance, the scaling benefits diminished significantly beyond 7B parameters; (2) Sequential model ensemble demonstrated superior effectiveness over parallel approaches, particularly for reasoning models, which presents a promising direction for enhancing question quality without additional supervision.

9 Limitations

Our study was constrained by the fixed annotation scope of the validation set, which may not fully capture the models' reasoning capabilities. Due to the limited number of systems submitted, we did not evaluate the performance of sequential ensembles with reasoning models.

References

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, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Sondos Mahmoud Bsharat, Aidar Myrzakhan, and Zhiqiang Shen. 2024. Principled instructions are all you need for questioning llama-1/2, gpt-3.5/4.

Blanca Calvo Figueras, Jaione Bengoetxea, Maite Heredia, Ekaterina Sviridova, Elena Cabrio, Serena Villata, and Rodrigo Agerri. 2025. Overview of the critical questions generation shared task 2025. In *Proceedings of the 12th Workshop on Argument Mining (ArgMining 2025)*, Vienna, Austria. Association for Computational Linguistics.

Subhabrata Dutta, Jeevesh Juneja, Dipankar Das, and Tanmoy Chakraborty. 2022. Can unsupervised knowledge transfer from social discussions help argument mining? In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7774–7786, Dublin, Ireland. Association for Computational Linguistics.

Xinyu Hua and Lu Wang. 2022. Efficient argument structure extraction with transfer learning and active learning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 423–437, Dublin, Ireland. Association for Computational Linguistics.

Jieyi Long. 2023. Large language model guided tree-of-thought.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback.

Huy Nguyen and Diane Litman. 2016. Context-aware argumentative relation mining. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1127–1137, Berlin, Germany. Association for Computational Linguistics.

Juri Opitz and Anette Frank. 2019. Dissecting content and context in argumentative relation analysis. In *Proceedings of the 6th Workshop on Argument Mining*, pages 25–34, Florence, Italy. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed 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*, volume 35, pages 24824–24837. Curran Associates, Inc.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models.

Jinghui Zhang, Dongming Yang, Siyu Bao, Lina Cao, and Shunguo Fan. 2023. Emotion classification on code-mixed text messages via soft prompt tuning. In *Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis*, pages 596–600, Toronto, Canada. Association for Computational Linguistics.

A Prompts

We developed four distinct prompt configurations, with key differentiators highlighted in bold.

Here is the zero-shot prompt:

(1) """You are a helpful assistant with critical thinking skills.

Suggest one critical question that directly challenges an argument in this text:

<text>

{intervention}

</text>

Requirements for the question:

<requirement>

1. Keep the question simple—no explanations or justifications.

2. Ensure logical reasoning aligns with the text.

3. Focus exclusively on content within the provided text.

4. Avoid introducing new concepts or external ideas.

5. Make it specific to the arguments in the text (not generic).

6. Target a single argument critically (e.g., a precise reading-comprehension critique).

</requirement>

"""

Here is the few-shot prompt:

(2) """You are a helpful assistant with critical thinking skills.

Suggest one critical question that directly challenges an argument in this text:

<text>

{intervention}

</text>

Requirements for the question:

<requirement>

1. Keep the question simple—no explanations or justifications.

2. Ensure logical reasoning aligns with the text.

3. Focus exclusively on content within the provided text.

4. Avoid introducing new concepts or external ideas.

5. Make it specific to the arguments in the text (not generic).

6. Target a single argument critically (e.g., a precise reading-comprehension critique).

</requirement>

Here is an example.

<text>

{Intervention example}

</text>

Useful questions:

{Useful question 1}

...

Unhelpful questions:

{Unhelpful question 1}

...

Please give an useful question.

"""

Here is the oral-expression prompt:

(3) """You are a helpful assistant with critical thinking skills.

Suggest one critical question that directly challenges an argument in this text:

<text>

{intervention}

</text>

Requirements for the question:

<requirement>

1. Be useful (challenge one of the arguments in the text).

2. The reasoning should be right.

3. Be related to the text.

4. Do not introduce new concepts not present in the text.

5. Avoid being too general that could be applied to any text.

6. Be critical with one of the argument in the text (e.g. a reading-comprehension question).

</requirement>

Here is an example.

<text>

{Intervention example}

</text>

Useful questions:

{Useful question 1}

...

Unhelpful questions:

{Unhelpful question 1}

...

Please give an useful question.

"""

Here is the requirements-ahead prompt:

(4) """You are a helpful assistant with critical thinking skills.

Suggest one critical question that directly challenges an argument in the given text.

Requirements for the question:

<requirement>

1. Be useful (challenge one of the arguments in the text).

2. The reasoning should be right.

3. Be related to the text.

4. Do not introduce new concepts not present in the text.

5. Avoid being too general that could be applied to any text.

6. Be critical with one of the argument in the text (e.g. a reading-comprehension question).

</requirement>

Here is an example.

<text>

{Intervention example}

</text>

Useful questions:

{Useful question 1}

...

Unhelpful questions:

{Unhelpful question 1}

...

<text>

{intervention}

</text>

Please give an useful question.

"""

1. Be useful (challenge one of the arguments in the text).

2. The reasoning should be right.

3. Be related to the text.

4. Do not introduce new concepts not present in the text.

5. Avoid being too general that could be applied to any text.

6. Be critical with one of the argument in the text (e.g. a reading-comprehension question).

</requirement>

Here is an example.

<text>

{Intervention example}

</text>

Useful questions:

{Useful question 1}

...

Unhelpful questions:

{Unhelpful question 1}

...

<text>

{intervention}

</text>

Useful questions:

{cq}

Please give another useful question.

"""

Here is the prompt for sequential model ensembling:

(5) """You are a helpful assistant with critical thinking skills.

Suggest one critical question that directly challenges an argument in the given text.

Requirements for the question:

<requirement>