MAMM-REFINE: A Recipe for Improving Faithfulness in Generation with Multi-Agent Collaboration

David Wan Justin Chih-Yao Chen Elias Stengel-Eskin Mohit Bansal
UNC Chapel Hill
{davidwan,cychen,esteng,mbansal}@cs.unc.edu

Abstract

Multi-agent collaboration among models has shown promise in reasoning tasks but is underexplored in long-form generation tasks like summarization and question-answering. We extend multi-agent multi-model reasoning to generation, specifically to improving faithfulness through refinement, i.e., revising model-generated outputs to remove factual inconsistencies. We investigate how iterative collaboration among multiple instances and types of large language models (LLMs) enhances subtasks in the refinement process, such as error detection, critiquing unfaithful sentences, and making corrections based on critiques. We design intrinsic evaluations for each subtask, with our findings indicating that both multi-agent (multiple instances) and multi-model (diverse LLM types) approaches benefit error detection and critiquing. Additionally, reframing critiquing and refinement as reranking rather than generation tasks improves multi-agent performance. We consolidate these insights into a final "recipe" called **M**ulti-**A**gent **M**ulti-**M**odel **Refine**ment (MAMM-REFINE), where multi-agent and multi-model collaboration significantly boosts performance on three summarization datasets as well as on long-form question answering, demonstrating the effectiveness and generalizability of our recipe.[1]

1 Introduction

Large language models (LLMs) have achieved remarkable performance in natural language generation but still suffer from hallucinations and a lack of faithfulness (Guerreiro et al., 2023; Zhang et al., 2023; Tang et al., 2023, 2024b; Liu et al., 2024), where the generated content is inconsistent with the input source or the world. To address this problem, many studies have developed post-hoc self-refinement techniques (Madaan et al., 2023; Gero et al., 2023; Raunak et al., 2023; Jiang et al., 2023; Gou et al., 2024; Wadhwa et al., 2024). However, these techniques have been found to be less effective without external feedback, as models require external information to identify errors (Huang et al., 2024a). One promising avenue for extending models beyond their inherent capabilities is multi-agent debate (Chen et al., 2024a; Du et al., 2023; Liang et al., 2023), where multiple LLMs improve their answers over the course of a debate or discussion. The agents can be multiple instances of the same model or different models (i.e. multi-model).

While several approaches focus on improving generation faithfulness through refinement, e.g. by breaking down the refinement process into fine-grained subtasks (Liu et al., 2023b; Wadhwa et al., 2024), past work has used a single instance of the same model for each of these subtasks, without multi-agent collaboration. Different models, due to their diverse training data and methods, often exhibit different hallucination patterns (Rawte et al., 2023; Guerreiro et al., 2023; Ye et al., 2023). Therefore, adopting a multi-model, multi-agent framework could help systems achieve higher faithfulness by allowing models to revise their solutions based on diverse answers obtained through collaboration. In such collaborative settings, different models' hallucinations might cancel out.

However, several challenges remain before the promise of multi-agent and multi-model approaches can be realized in generation tasks: First, multi-agent frameworks have shown great promise for reasoning tasks (Chen et al., 2024a; Du et al., 2023; Liang et al., 2023) where the final answers are generally from a closed set and easily verified, leading to easy stopping criteria and enabling voting across agents. Applying multi-agent collaboration to generative tasks such as summary refinement – where final answers are long and difficult to verify – is less straightforward. Additionally, due to the complexity and multitude of the design

[1]Our code is available at https://github.com/meetdavidwan/mammrefine.

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 9882–9901
April 29 - May 4, 2025

choices in generation and refinement tasks, it is not clear which components would benefit from a multi-agent framework. As verified by our empirical results, naively applying multi-agent reasoning to all subtasks might unnecessarily increase cost and could even hurt performance, as agents may lead each other down incorrect paths.

To extend multi-agent approaches to long-form generation, we focus on the task of improving faithfulness through refinement, as it is backed by extensive literature on evaluation metrics and generation strategies. As illustrated in Figure 1, we conduct a comprehensive analysis to determine which refinement subtasks benefit most from a multi-agent pipeline. Focusing on the three-subtask approach from Wadhwa et al. (2024), a state-of-the-art refinement strategy, we consider DETECT, CRITIQUE, and REFINE subtasks, which can be recombined into different pipelines (e.g., DETECT-REFINE, CRITIQUE-REFINE, REFINE only). We apply multi-agent collaboration to these subtasks, framing CRITIQUE and REFINE with two approaches: a discriminative method (RERANK) that selects the best option among multiple candidates, and GENERATE, which updates the answer freely. Our research addresses three core questions: **(1) Which refinement subtasks benefit from a multi-agent approach? (2) Which subtasks benefit from a multi-model approach? (3) For which task type (GENERATE or RERANK) is the multi-agent approach most effective?**

To answer these, we perform an extensive intrinsic analysis to find the optimal setting for each subtask, creating a "recipe", **M**ulti-**A**gent **M**ulti-**M**odel **Refine**ment (MAMM-REFINE), that combines the best configurations. Using TofuEval (Tang et al., 2024c), a dataset with human-annotated sentence-level faithfulness judgments and critiques, we design intrinsic evaluation tasks for each of the three subtasks, DETECT, CRITIQUE, and REFINE, as illustrated in Figure 2. Our findings show that multi-agent approaches generally outperform single-model baselines, with multi-model variants offering further gains, and for CRITIQUE and REFINE, multi-agent methods provide consistent gains with RERANK but not with GENERATE.

Next, after determining the best configuration for each subtask from the intrinsic evaluations, we evaluate end-to-end performance on three summarization datasets with MAMM-REFINE, MediaSum (Zhu et al., 2021), MeetingBank (Hu et al., 2023), and UltraChat (Ding et al., 2023), comparing our approach with other refinement baselines. Across different refinement pipelines that use various combinations of subtasks, we show that selecting the best components from our intrinsic analysis gives us a generalizable recipe for improved refinement that holds across generation tasks and datasets, with gains on all three summarization tasks. We further show that our recipe generalizes to long-form question answering, improving faithfulness in a non-summarization domain.

2 Method

We begin refinement with a model-generated output Y and optionally an input context X (e.g., a summarization document or question-answering context (Xu et al., 2023)). Using a refinement prompt and model M_r, we transform Y into a refined output Y_r. We first outline common refinement pipelines and their corresponding subtasks, and next illustrate how each adapts to our multi-agent setting for generative tasks:

Direct Refinement. A single task directly prompts the refinement model to improve the summary based on the document: $Y_r = M_r(X, Y)$.

DETECT-CRITIQUE-REFINE (DCR). As illustrated in top left section of Figure 1, we follow Wadhwa et al. (2024)'s breakdown of refinement into three steps, covering all components of the refinement process. First, each output sentence $y^i \in \{y^0, \dots, y^N\}$ is evaluated by a detection subtask $M_d(X, y^i)$, DETECT, to produce a binary faithfulness label, indicating whether the sentence requires refinement. We define faithfulness as whether the output is supported by the input, and measure it using a model given the prompt described in Appendix D.[2] In addition to a binary label, the detection step produces a reasoning chain which can be treated as a critique of the sentence (i.e. justifying why the sentence is unfaithful). For each sentence marked as unfaithful, we also optionally employ a critique subtask $M_c(X, y^i)$, CRITIQUE, that generates a critique c^i detailing the error span (i.e. which tokens make the output unfaithful) and suggest a fix. Finally, based on the outputs of DETECT and CRITIQUE, we use REFINE to generate a refined summary $Y_r = M_r(X, Y, C)$, where C is the set of critiques, either directly from DETECT's reasoning or from the explicit CRITIQUE subtask. These three subtasks can be recombined

[2]Note that this prompting process is not the same what is used as the final evaluation metrics described in Section 3.2.

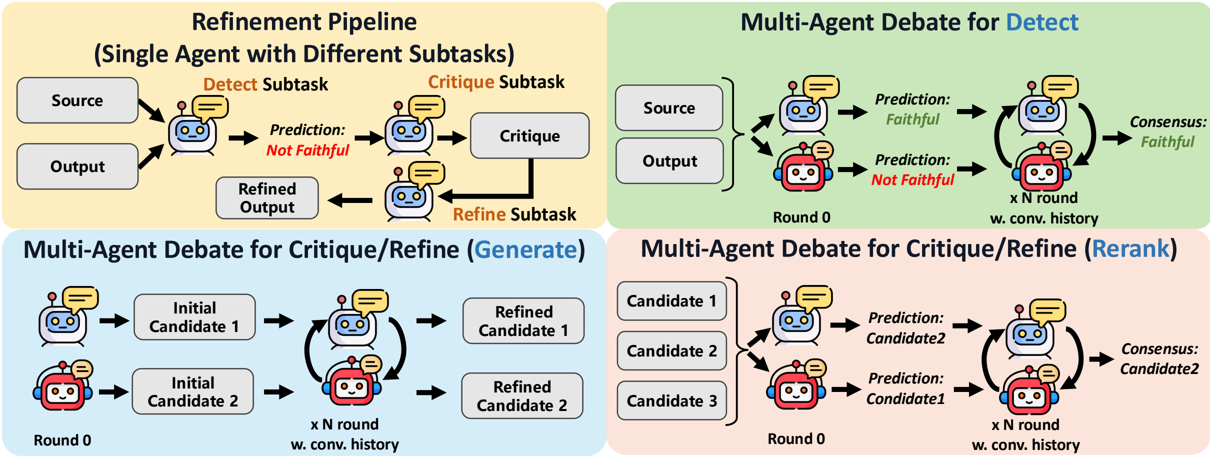


Figure 1: Illustration of the refinement pipeline (top-left) and how multi-agent debate is applied to different subtasks. In the DETECT subtask (top-right), agents collectively choose among a discrete set of options, such as making yes/no decisions or selecting the most faithful candidate. For the CRITIQUE and REFINE subtasks, we explore two approaches. In the bottom-left panel, we frame the task as generative (using GENERATE), where each agent updates its own critique or output based on other agents' responses. In the bottom-right panel, we frame it as a discriminative task using RERANK, where agents choose the best output from the candidates. While discriminative tasks converge to a single solution, generative tasks result in updated responses from each agent.

into other variants, e.g. DETECT-REFINE (refining only on unfaithful generations), and CRITIQUE-REFINE (Chern et al., 2023b) (refining based on critiques for all examples).

Multi-agent Debate. We adapt the multi-agent framework introduced by Chen et al. (2024a), which has shown strong performance on short-form QA tasks such as commonsense and math reasoning. Let $A^1, \ldots, A^n$ be a list of n agents participating in a discussion. In the initial round, we ask each agent to generate its own output g_0^i. For each subsequent round k, we ask each agent to update its answer based on all agents' responses from the last round, i.e., $g_k^i = A^i(g_{k-1}^1, \ldots, g_{k-1}^n)$, forming a conversational state. That is, for each subtask, every agent can view the previous responses of the other agents and update its answer accordingly. Discussion ends when the maximum round is reached or when the agents have reached a consensus.

Extending to Generative Tasks. While adapting multi-agent collaboration to a binary classification task like DETECT is straightforward (see upper right of Figure 2), extending it to long-form tasks like CRITIQUE and REFINE is challenging for two key reasons. First, evaluation is challenging as each agent produces its own answer; past work has addressed this by averaging the performance of individual agents (Du et al., 2023). Secondly, determining a stopping criterion is challenging. Unlike with classification tasks, where it is clear when agents have converged to the same answer, evaluating and matching long-form outputs is a challenging open problem (Huang et al., 2024a). Nevertheless, as shown in the bottom left of Figure 1, we experiment with a generative multi-agent variant (GENERATE) of CRITIQUE and REFINE, where agents read others' answers and update their own.

To better leverage the strength of multi-agent systems on closed-set tasks, we implement an alternative way to combine generations: RERANK, as illustrated in bottom right of Figure 1. Here, we transform open-ended generative tasks into a discriminative ones by asking agents to select the best generation from a set of candidates. Agents in RERANK produce item indices (a closed set), making the task a classification problem and simplifying voting and convergence checks.

3 Experimental Setup

3.1 Agent Setup

In all of our tasks, we use two strong agents of similar capability – GPT-4o (OpenAI, 2024) and Claude-3.5 Sonnet (Anthropic, 2024) when employing multiple agents; for our main experiments, we limit the number of agents to two to reduce computational cost. In Section B.4, we also illustrate the gains achieved by adding more agents. We use the same prompts for all models, which are shown in Appendix D. We consider the following combinations to evaluate the effectiveness of multi-agent

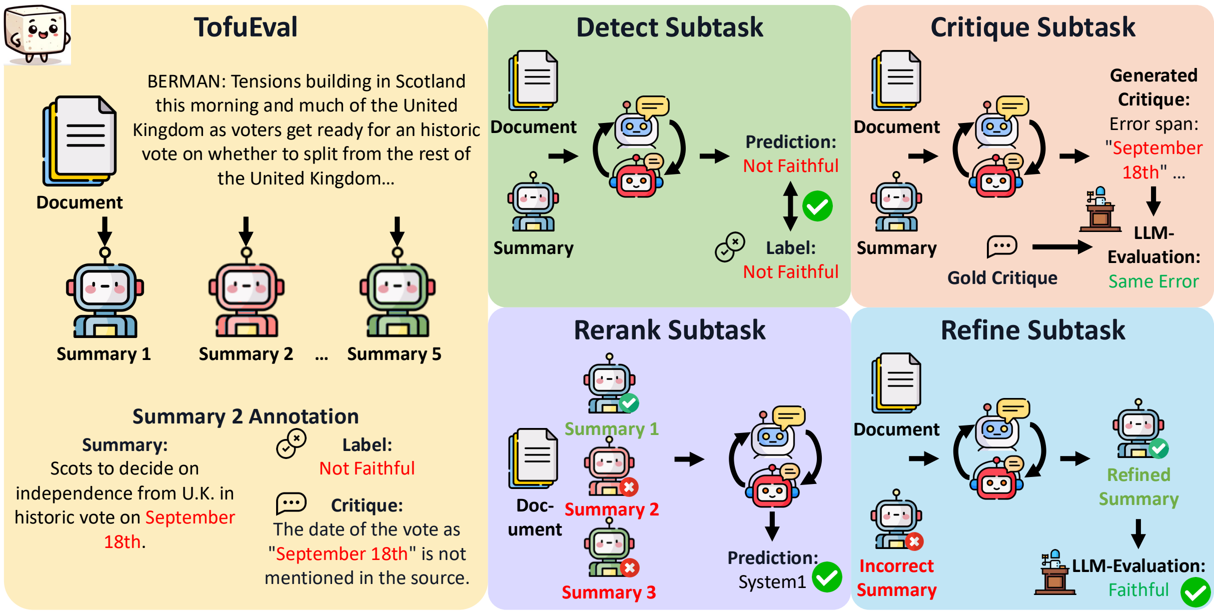


Figure 2: Illustration of our setup for intrinsic evaluations for different subtasks. We convert TofuEval, a dataset of system summaries with human-annotated faithfulness labels and critiques, to tasks for evaluating the performance of different multi-agent setups for DETECT, CRITIQUE, and REFINE subtasks with RERANK and GENERATE.

settings along three axes: agent, model, and task. First, we differentiate between single-agent (SA) and multi-agent (MA) settings based on the number of agents used. Second, within the MA setting, we distinguish between single-model (SM), where multiple instances of the same model are used, and multi-model (MM), where different models serve as agents. From the pipeline perspective, we also consider a single-agent multi-model setting, where different subtasks use different models. Finally, we frame CRITIQUE and REFINE as both generative (via GENERATE) and discriminative tasks (via RERANK). For the two models we employ, this results in two instances of GPT-4o (2xG), two instances of Claude (2xC), and the MAMM setting of using one GPT-4o and one Claude (G+C). For fair comparison between single-agent and multi-agent settings, we report the average performance of the single agents. For GENERATE, which generates multiple outputs, we report the average scores, similar to Du et al. (2023). For all tasks, we set the debate to run for a maximum of 10 rounds.

3.2 Intrinsic Evaluation Tasks

We evaluate our methods using TofuEval (Tang et al., 2024c), a topic-focused summarization task with annotations on two datasets: MediaSum (Zhu et al., 2021) and MeetingBank (Hu et al., 2023). TofuEval contains 50 documents for each dataset, each paired with three topics. It also contains sentence-level faithfulness judgments from annotators for summaries generated by five different systems. For each sentence, annotators were asked to provide a binary faithfulness judgment and, if deemed unfaithful, write a critique explaining the error. We use this dataset to create intrinsic tasks for evaluating DETECT and CRITIQUE. An example is shown on the left side of Figure 2, where summary 2 contains a hallucination regarding the date of the vote. For all four intrinsic evaluations, we use all 150 document-topic pairs (50 documents $\times$ 3 topics). We randomly selected one summary out of the five systems for each document-topic pair. We split the 50 documents into 10 for validation and 40 for testing, resulting in 30 validation and 120 test document-topic pairs. We tune all methods on the validation set. Our main research questions for the intrinsic evaluations are: (1) Does MA improve performance? (2) Does MM improve performance? (3) How do different frameworks affect performance? Appendix B.2 further explores how performance varies over debate iterations.

DETECT Evaluation. We use the human-annotated faithfulness label and evaluate whether our detection model outputs the same label as a discriminative task. This yields 81 validation and 324 sentence-level test examples for MediaSum, and 85 validation and 328 test examples for MeetingBank. Following Laban et al. (2021), we use balanced accuracy (BACC) to account for class imbalance. As baseline, we compare to strong automatic metrics, MiniCheck (Tang et al., 2024a) and AlignScore (Zha et al., 2023), which are trained entailment metrics between document chunks and summary sentences designed for summarization tasks.

RERANK Evaluation. As an alternative to having agents directly debate their summaries, we aim to explore the best methods for reranking generated summaries. To achieve this, we use the human labels for all the different systems. Because ToFuEval contains no gold summaries, we bootstrap this data by identifing cases where only one system's summary is judged by humans to be faithful, treating that summary as "gold". We then create test scenarios where we present this faithful summary along with two to four unfaithful summaries randomly sampled from the remaining summaries, resulting in sets of three to five summaries. We randomly shuffle the candidates to ensure the model is not biased toward any position. For evaluation, we measure the accuracy of the model in ranking the faithful summary highest from the set of candidates. We compare this to two baselines that use MiniCheck and AlignScore for reranking, selecting the output with the highest faithfulness score.

CRITIQUE Evaluation. To evaluate the performance of the critique model, we consider two settings: *Gold* and *Detect*. These settings correspond to generating critiques when the summary sentence is considered unfaithful according to gold labels or model predictions, respectively. In the *Gold* setting, we use the human-provided faithfulness labels, whereas in the *Detect* setting, we use the predicted faithfulness labels from the best model found in the intrinsic DETECT evaluation (G+C). We also evaluate the explanations generated by the DETECT subtask as part of its chain-of-thought. To evaluate our approach, we adopt the methodology outlined by Wadhwa et al. (2024), which we further verify with human evaluations in Section C.1. Specifically, we prompt GPT-4o to assess whether the generated critique aligns with the human-written critique. We instruct the model to select one of the following options: (1) Error Match: The generated critique identifies the same error as described by the human. (2) Error, No Match: The generated critique discusses a different error than the one noted by the human. (3) No Error Detected, No Match: The generated critique states that there is no error, despite the human indicating otherwise.

REFINE Evaluation. We evaluate different methods for the refinement model using the same setup as in the final evaluation. We primarily test the effect of various refinement methods when using the best detector from the intrinsic DETECT evaluation (G+C) and the best critique models for both the Gold critique (2xC) and DETECT settings (2xG). We assess the faithfulness of the summaries using MiniCheck (Tang et al., 2024a) and a GPT-4-based Likert evaluation, following Wadhwa et al. (2024). Both metrics show high correlations with human judgments of faithfulness (Tang et al., 2024a; Liu et al., 2023a; Chiang and Lee, 2023; Gao et al., 2023), which we also verify in Section C.2. We calculate the faithfulness of each summary sentence and then aggregate averaging across all sentences.

3.3 Extrinsic Refinement Setup

While the intrinsic tasks are tuned on the validation set of MediaSum and MeetingBank, we evaluate for the extrinsic evaluation on the test sets of MediaSum and MeetingBank, as well as on UltraChat (Ding et al., 2023) as a held-out, out-of-domain setting. As baselines, we use each single agent individually to perform each component task. We then combine identical models in a multi-agent setting (e.g., 2xG or 2xC) and also explore a multi-model setting by combining Claude and GPT-4o. For tasks where a generative approach is applicable (i.e., critique and refinement), we further investigate GENERATE, as detailed in Section 2. We use MiniCheck and Likert scores for evaluation.

4 Results

We report the intrinsic and extrinsic results, and examine how MAMM-REFINE generalizes to long-form question-answering. We provide additional discussions and show improvement from additional agents and how multi-agent performance changes after each round of discussion in Appendix B.

4.1 DETECT Intrinsic Results

We present the best strategy for DETECT, which identifies hallucinating sentences and thus helps the refinement systems to refine only where needed. We report BACC in the left side of Table 1. We first note that the single agents perform competitively compared to the baseline of using the MiniCheck and AlignScore metrics to detect unfaithful sentences, especially on MeetingBank.

Effect of Multi-Agent. Using the same model does not improve performance, except for a slight im-

Category	Method	DETECT MediaSum	DETECT MeetingBank	RERANK MediaSum 2	3	4	RERANK MeetingBank 2	3	4
Baseline	MiniCheck	72.8	69.8	56.7	46.7	53.3	80.0	53.3	**76.7**
	AlignScore	70.8	71.6	56.7	46.7	46.7	70.0	63.3	70.0
Single Agent	GPT-4o	72.1	76.5	73.1	38.5	53.8	62.5	65.6	53.1
	Claude	72.7	77.7	84.6	50.0	53.8	78.1	56.3	56.3
Multi-Agent Single-Model	2xG	72.5	75.5	83.3	46.2	40.0	62.5	66.7	50.0
	2xC	72.5	76.1	**92.3**	46.2	**58.3**	**81.3**	56.3	56.3
Multi-Agent Multi-Model	**G+C**	**74.3**	**80.2**	**92.3**	**53.8**	45.5	**81.3**	**68.8**	62.5

Table 1: Detection (left) and reranking (right) results. We report balanced accuracy for detection and accuracy of selecting the faithful candidate for reranking (Acc@1). Reranking performance is broken down by number of distractors (columns). We **bold** the method that we select as the best method for DETECT and RERANK.

provement of 0.4% on MediaSum when using 2xG over single GPT-4o. On MeetingBank, we observe a decrease of 1% with both 2xG and Claude.

Effect of Multi-Model. Multi-model improves beyond the two base models. Specifically, G+C outperforms Claude, the best of the two single models, by 1.6% and 2.5% on MediaSum and MeetingBank, respectively. This indicates the effectiveness of multi-model in improving detection accuracy.

Takeaway. Multi-agent single-model does not improve DETECT, but the multi-model variant helps.

4.2 RERANK Intrinsic Results

Next, we determine the best combination for RERANK, since it will be used for both critique and refinement. The accuracy of selecting the most faithful summaries with different numbers of candidates is shown on the right of Table 1.

Effect of Multi-Agent. Here, MA improves over its SA, specifically on MediaSum. However, we only observe an improvement of 3.2% with reranking 2 candidates using 2xC on MeetingBank.

Effect of Multi-Model. Similar to DETECT, we find that MAMM generally achieves high accuracy: it is tied with MASM with 2xC for the highest accuracy when reranking two candidates, and outperforms all other variants when reranking three candidates on MediaSum and MeetingBank. This confirms the importance of having multiple models. Among the different settings, we find that the largest gain occurs when there are only two choices, improving accuracy by 7.7% and 3.2% on MediaSum and MeetingBank, respectively. This aligns with prior works showing that LLMs perform better in pairwise comparisons (Huang et al., 2024b).

Takeaway. The multi-model multi-agent approach improves reranking accuracy, showing the benefit of such a framework for closed-set tasks.

4.3 CRITIQUE Intrinsic Results

We present the results in Table 2, which analyzes whether the generated critiques identify the same errors as the gold critiques. The critiques that come with the DETECT's CoT are overall worse than those from the dedicated critique subtask, where the highest error matching score with the two-step approach under the *Detect* setting is 12% higher. This underscores the importance of having an additional critique step, so as not to overload LLMs with two tasks at once (Wadhwa et al., 2024).

Effect of Multi-Agent. For *Gold* critique case, we observe that reranking on Claude's critiques performs the best, almost achieving a perfect score. This shows that 2xC can critique the correct problem if there is no error in DETECT. Note that is unrealistic, as the CRITIQUE will not have perfect DETECT predictions and thus will not have 0% "No Error" outputs, where CRITIQUE fails to find errors. Using predictions from DETECT gives us a more realistic idea of what a model will do when the sentence is actually correct and CRITIQUE incorrectly considers it having some faithfulness errors. Interestingly, for the more realistic *Detect* scenario, reranking on 2xG critiques achieves the highest performance. Compared to a single GPT-4o setting, the multi-agent approach improves by 2.4% in terms of capturing the correct error. Multi-agent approach performs the best under the two settings.

Effect of Multi-Model. For both *Gold* and *Detect* settings, G+C is ranked second. As it performs slightly worse than the 2xC for *Gold* and 2xG for *Detect*, MM still demonstrates its generalizability.

Effect of Task Framing. Finally, we also compare

Setting	Category	M_C	EM↑	EMM↓	NE↓
DETECT's CoT	SA	GPT-4o	54.0	7.3	38.8
		Claude	<u>55.9</u>	6.7	37.5
	MASM	2xG	51.9	**8.7**	39.6
		2xC	53.8	**8.7**	37.6
	MAMM	G+C	**57.0**	<u>8.9</u>	**34.1**
Gold	SA	GPT-4o	95.1	5.0	0.0
		Claude	<u>98.5</u>	<u>1.6</u>	0.0
Gold w. RERANK	MASM	2xG	96.8	3.3	0.0
	MASM	**2xC**	**99.2**	**0.8**	0.0
	MAMM	G+C	97.5	2.5	0.0
Detect	SA	GPT-4o	67.1	3.2	29.9
		Claude	68.3	2.5	29.3
Detect w. RERANK	**MASM**	**2xG**	**69.5**	**1.3**	**29.3**
	MASM	2xC	68.3	2.5	**29.3**
	MAMM	G+C	<u>68.9</u>	<u>1.9</u>	**29.3**
Detect w. GENERATE	MASM	2xG	62.1	2.9	35.0
	MASM	2xC	67.5	0.9	31.6
	MAMM	G+C	66.1	2.0	31.9

Table 2: CRITIQUE Results under *Gold* and *Detect* setting, and using DETECT's CoT. EM = Error Match, EMM = Error Mismatch, and NE=No Error Found. We **bold** the best strategy for CRITIQUE for the two settings.

the generative task framing (GENERATE) and the discriminative framing (RERANK) in the bottom section of Table 2. Overall, the best generative approach (2xC) has a lower error matching rate than its reranking counterpart, which is the worst of the three multi-agent systems when reranked.

Takeaway. Though multi-model provides consistent improvement across the two settings, using single-model multi-agent to rerank critiques performs the best compared to other variants. GENERATE does not show improvement from multi-agent.

4.4 REFINE Intrinsic Results

Next, we evaluate which method is best for refinement. We present the results of using 2xC critiques, as they achieve higher faithfulness scores with the validation set in Table 3 and report the results with 2xG critiques in Appendix B.1.[3]

Effect of Multi-Agent. The best setting, 2xG, achieves only a 0.3% gain in MiniCheck. We hypothesize that with good critiques, a strong LLM-based refinement model can perform the task well.

Effect of Multi-Model. For G+C variant with RERANK, we similarly observe that it does not improve beyond the single-model performance. In fact, it achieves faithfulness scores between the two single-agent faithfulness scores.

[3]The result with 2xG critiques show the same trends as with 2xC critiques.

Category	M_R	MCS↑	GL↑
Original	-	78.3	3.8
Single Agent	GPT-4o	84.6	4.2
	Claude	82.8	4.2
MASM w. RERANK	**2xG**	<u>84.9</u>	4.2
MASM w. RERANK	2xC	82.5	4.2
MAMM w. RERANK	G+C	83.4	4.2
MASM w. GENERATE	2xG	**85.2**	4.2
MASM w. GENERATE	2xC	79.1	4.2
MAMM w. GENERATE	G+C	81.4	4.2

Table 3: REFINE results with 2xC critiques with MiniCheck (MCS) and GPT-4o Likert score (GL). We **bold** the method that we select as the best method for REFINE. Full table with scores on MediaSum and MeetingBank separately is shown in Table 6.

Effect of Task Framing. We also compare RERANK with GENERATE and find that the methods using GENERATE further hurt faithfulness when applied to 2xC and G+C, while providing a slight but not significant gain of 0.3% over the method using RERANK.[4] As mentioned in Section 2, GENERATE does not guarantee outputting a single candidate. Considering the limited improvement and the high computational cost of performing multiple rounds of GENERATE (since it requires generating outputs for all agents in each round) compared to only debating on the examples where agents choose different best candidates in RERANK, we opt for 2xG with RERANK.

Takeaway. Overall, we recommend refining using 2xG with RERANK on 2xC critiques, illustrating the need for multi-model approaches from the pipeline perspective, where different models excel at different tasks; that is, Claude excels at generating critiques, and GPT-4o excels at refinement.

4.5 Overall Result with Final Recipe

Finally, we evaluate MAMM-REFINE on MediaSum, MeetingBank, and the held-out dataset, UltraChat. We first focus on applying our best configurations for each subtask to existing refinement pipelines. As shown in Wadhwa et al. (2024), direct refinement even degrades MiniCheck scores on MeetingBank and UltraChat, demonstrating the necessity of a pipeline with more fine-grained subtasks. Nevertheless, we also evaluate direct refinement and pipelines without all the fine-grained subtasks, showing that applying our best configuration of REFINE and DETECT subtasks improves the

[4]We use paired bootstrap test (Koehn, 2004).

Method	M_D	M_C	M_R	MediaSum MCS↑	MediaSum GL↑	MeetingBank MCS↑	MeetingBank GL↑	UltraChat MCS↑	UltraChat GL↑
Original	-	-	-	74.4†	4.1†	82.1†	3.6†	77.6	3.8†
REFINE Only	-	-	GPT-4o	77.6	4.3	76.9†	3.6†	75.8	4.0†
	-	-	2xG	78.3	4.3	77.9†	3.7†	76.7	4.1†
DETECT + REFINE	Claude	-	GPT-4o	77.4	4.3	81.9†	3.7†	78.3	4.0†
	G+C	-	GPT-4o	77.8	4.3	81.1†	3.7†	78.3	4.0†
CRITIQUE + REFINE	-	GPT-4o	GPT-4o	78.9	4.5	85.1	3.9	80.0	4.3
	-	GPT-4o	2xG	78.9	4.5	85.2	4.0	80.6	4.3
	-	2xC	2xG	81.7	4.5	86.7	4.0	80.8	4.3
Single-Agent Single-Model	GPT-4o	GPT-4o	GPT-4o	79.2	4.4	86.6	**4.0**	80.5	4.1†
Single-Agent Multi-Model	Claude	Claude	GPT-4o	78.7	4.4	86.1	**4.0**	81.3	4.2
Multi-Agent Single-Model	2xG	2xG	2xG	79.9	4.4	87.0	**4.0**	79.9	4.2
MAMM-REFINE (Ours)	G+C	2xC	2xG	**82.4**	4.4	**87.4**	3.9	**81.5**	**4.3**

Table 4: Results on MediaSum, MeetingBank and UltraChat with MiniCheck (MCS), GPT-4o Likert score (GL). We show the models used for DETECT (M_D), CRITIQUE (M_C), and REFINE (M_R). † denotes statistically significant improvement by MAMM-REFINE over that entry ($p < 0.05$ using paired bootstrap test).

Method	M_D	M_C	M_R	MCS↑	G-L↑
Original	-	-	-	76.7	3.5†
SASM	G	G	G	80.1	3.9
SAMM	C	C	G	80.9	4.0
MASM	2xG	2xG	2xG	79.1	3.9
MAMM-REFINE	G+C	2xC	2xG	**82.0**	**4.1**

Table 5: Results on Long-form QA with context.

results, with results shown in Table 4. Specifically, using 2xG for M_R improves MiniCheck by 1% on MeetingBank and UltraChat, though still underperforming the original summaries, showing the need for critique-based refinement. Additionally, when we add DETECT, our best MAMM setting (G+C) further improves over direct refinement. Similarly, for the variants of CRITIQUE+REFINE, switching to MA M_R yields a slight gain, as observed in Section 4.4. Specifically, 2xC for M_C and 2xG for M_R provides 2.8%, 0.8%, and 1.0% boosts over using only GPT-4o for M_C and M_R on MiniCheck compared to the CRITIQUE+REFINE baseline.

We finally examine the three-step approach using all of our best configurations. Here, we observe the highest MiniCheck scores. In fact, MAMM-REFINE is the only method among three-step approaches that has a statistically significant ($p < 0.05$) gain over the original summary on both MediaSum and MeetingBank, as measured by both metrics. On the UltraChat dataset, MAMM-REFINE is also the only three-step variant with a statistically significant faithfulness improvement over the original summary according to the GPT-4o Likert score. We also test four settings – applying single or multi-model configurations to single or multi-agent setups – and evaluate these as an ablation study. For MediaSum and MeetingBank, multi-agent is important, while on UltraChat, multi-model is important. Nevertheless, we observe a consistent trend where both multi-agent and multi-model configurations are key to improving faithfulness.

4.6 Extending to Long-form QA

Next, we also explore how the pipeline extends to other generation tasks, such as long-form question answering (LFQA). We use the ELI5 dataset (Fan et al., 2019) collected in WebGPT (Nakano et al., 2021), which includes questions, model-generated answers, and the corresponding supporting context. From this data, we randomly select 100 examples and apply our refinement model. With the supporting context, the task is essentially question-answering with retrieved evidence, i.e. retrieval-augmented generation. Since evaluating the faithfulness of LFQA with context has the same setup as summarization (Xu et al., 2023), we employ the same experimental setup and metrics. The results are shown in Table 5. We observe that multi-model and multi-agent approaches improve the faithfulness of the answers, and our recipe provides the most faithful responses, improving 5.3% on MiniCheck and 0.6 points on the Likert score. We similarly observe as an ablation that multi-model provides a stronger gain than multi-agent. This illustrates that our recipe can not only generalize to a held-out summarization dataset, but to a held-out non-summarization generation task like long-form question answering. We also report the setup and results without the context in Appendix B.5.

5 Related Work

Multi-Agent systems with LLMs. A large body of research focuses on multi-agent systems for reasoning tasks (Du et al., 2023; Liang et al., 2023; Yin et al., 2023; Chen et al., 2024a; Kim et al., 2024; Haji et al., 2024; Tang et al., 2024d; Sun et al., 2024), where multiple LLMs engage in debates or discussions. Recent studies have also proposed multi-agent systems for LLM evaluation, where agents either undergo a peer review process, obtaining a win rate by ranking each other (Li et al., 2023b), or engage in debates to determine the better LLM response (Chan et al., 2023). To address hallucination, Feng et al. (2024) propose using a multi-agent system to identify knowledge gaps between LLMs. The success of this paradigm hinges on the fact that reasoning tasks typically have well-defined solutions. In contrast, multi-agent systems for generation tasks largely focus on enhancing creativity through role-playing (Wang et al., 2024; Lu et al., 2024; Li et al., 2023a), where evaluation metrics are less established. To the best of our knowledge, we are the first to propose a multi-agent long-form generation in the context of improving faithfulness on summarization and long-form question-answering.

Refinement. Refinement has gained significant focus, including leveraging human feedback (Saunders, 2023) and automatic feedback through self-refinement from the same model (Madaan et al., 2023; Gero et al., 2023; Raunak et al., 2023), other trained models (Xu et al., 2024; Akyurek et al., 2023; Paul et al., 2024; Chern et al., 2023a; Chen et al., 2024b), or external tools (Jiang et al., 2023; Olausson et al., 2024; Gou et al., 2024; Chen et al., 2024c). For improving faithfulness of summarization, many post-processing approaches (Fabbri et al., 2022; Balachandran et al., 2022; Thorne and Vlachos, 2021) focus on training such refinement model, or using human-annotated numeric scores as feedback (Stiennon et al., 2020; Wu et al., 2021; Nguyen et al., 2022; Scheurer et al., 2024). More recently, efforts have concentrated on using LLMs to directly refine generations, such as by utilizing fine-grained feedback from a faithfulness detector at the level of atomic, non-decomposable facts (Wan et al., 2024), or employing a two-step (Liu et al., 2023b) or three-step (Wadhwa et al., 2024) refinement approach. Our work is complementary to past refinement and multi-LLM work, as we measure the effect of multi-agent approaches across the components of the refinement pipeline. By testing MM and MA settings, we create a generalizable refinement recipe across generation tasks.

6 Conclusion

We carefully curate components for incorporating multi-agent collaboration into generation, improving generation faithfulness through refinement. Through intrinsic evaluations, we find that employing multiple agents, particularly multiple models, benefits discriminative tasks like DETECT and RERANK. We then show how to apply RERANK to CRITIQUE and REFINE. In extrinsic evaluations, we find that the best variation for each component improves several refinement methods, and our final recipe shows gains on three summarization benchmarks and transfers to long-form question-answering tasks, showing its generalizability.

Limitations

First, our work primarily focuses on faithfulness, which is crucial to building user trust in LLMs and enabling safe model use. While there are other aspects, such as coherence and relevance, that could contribute to a comprehensive evaluation, we choose to evaluate faithfulness due to its rich literature and extensive experiments using the best automatic evaluation metrics. Regarding evaluation, although the automatic metrics we use have shown high correlations with human judgments of faithfulness, a gap still exists, which could be addressed by conducting human evaluations. However, considering the trade-off between the high cost and unreliability of using Mechanical Turk workers, we opt to report statistical significance based on automatic evaluations for more reliable assessments. Finally, refinement pipelines and multi-agent frameworks involve additional steps that lead to higher computational costs. However, these costs tend to reduce over time, and applying multi-agent reasoning to open-ended generation tasks more broadly is a crucial area for which we lay the groundwork. We do not forsee any particular risks beyond those inherent to any text generation task. Since our work focuses on improving faithfulness, it is aimed at mitigating some of the risks associated with using LLMs for generation.

Acknowledgement

We thank the anonymous reviewers for their helpful comments. This work was supported by Google PhD Fellowship, the Microsoft Accelerate Foundation Models Research (AFMR) grant program, an NSF-CAREER Award 1846185, DARPA ECOLE Program No. HR00112390060, and NSF-AI Engage Institute DRL-2112635. Any opinions, findings, and conclusions or recommendations in this work are those of the author(s) and do not necessarily reflect the views of the sponsors.

References

Afra Feyza Akyurek, Ekin Akyurek, Ashwin Kalyan, Peter Clark, Derry Tanti Wijaya, and Niket Tandon. 2023. RL4F: Generating natural language feedback with reinforcement learning for repairing model outputs. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7716–7733, Toronto, Canada. Association for Computational Linguistics.

Anthropic. 2024. Introducing claude 3.5 sonnet.

Vidhisha Balachandran, Hannaneh Hajishirzi, William Cohen, and Yulia Tsvetkov. 2022. Correcting diverse factual errors in abstractive summarization via post-editing and language model infilling. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9818–9830, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Meng Cao, Yue Dong, and Jackie Cheung. 2022. Hallucinated but factual! inspecting the factuality of hallucinations in abstractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3340–3354, Dublin, Ireland. Association for Computational Linguistics.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2023. Chateval: Towards better llm-based evaluators through multi-agent debate. *arXiv preprint arXiv:2308.07201*.

Justin Chen, Swarnadeep Saha, and Mohit Bansal. 2024a. ReConcile: Round-table conference improves reasoning via consensus among diverse LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7066–7085, Bangkok, Thailand. Association for Computational Linguistics.

Justin Chih-Yao Chen, Archiki Prasad, Swarnadeep Saha, Elias Stengel-Eskin, and Mohit Bansal. 2024b. Magicore: A multi-agent coarse-to-fine refinement framework for reasoning. *arXiv preprint arXiv:2409.12147*.

Xinyun Chen, Maxwell Lin, Nathanael Schärli, and Denny Zhou. 2024c. Teaching large language models to self-debug. In *The Twelfth International Conference on Learning Representations*.

I-Chun Chern, Steffi Chern, Shiqi Chen, Weizhe Yuan, Kehua Feng, Chunting Zhou, Junxian He, Graham Neubig, and Pengfei Liu. 2023a. Factool: Factuality detection in generative ai – a tool augmented framework for multi-task and multi-domain scenarios. *Preprint*, arXiv:2307.13528.

I-chun Chern, Zhiruo Wang, Sanjan Das, Bhavuk Sharma, Pengfei Liu, and Graham Neubig. 2023b. Improving factuality of abstractive summarization via contrastive reward learning. In *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*, pages 55–60, Toronto, Canada. Association for Computational Linguistics.

Cheng-Han Chiang and Hung-yi Lee. 2023. Can large language models be an alternative to human evaluations? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15607–15631, Toronto, Canada. Association for Computational Linguistics.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3029–3051, Singapore. Association for Computational Linguistics.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2023. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*.

Alex Fabbri, Prafulla Kumar Choubey, Jesse Vig, Chien-Sheng Wu, and Caiming Xiong. 2022. Improving factual consistency in summarization with compression-based post-editing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9149–9156, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. Eli5: Long form question answering. *arXiv preprint arXiv:1907.09190*.

Shangbin Feng, Weijia Shi, Yike Wang, Wenxuan Ding, Vidhisha Balachandran, and Yulia Tsvetkov. 2024. Don't hallucinate, abstain: Identifying LLM knowledge gaps via multi-LLM collaboration. In *Proceedings of the 62nd Annual Meeting of the Association* *for Computational Linguistics (Volume 1: Long Papers)*, pages 14664–14690, Bangkok, Thailand. Association for Computational Linguistics.

Mingqi Gao, Jie Ruan, Renliang Sun, Xunjian Yin, Shiping Yang, and Xiaojun Wan. 2023. Human-like summarization evaluation with chatgpt. *Preprint*, arXiv:2304.02554.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, Soroosh Mariooryad, Yifan Ding, Xinyang Geng, Fred Alcober, Roy Frostig, Mark Omernick, Lexi Walker, Cosmin Paduraru, Christina Sorokin, Andrea Tacchetti, Colin Gaffney, Samira Daruki, Olcan Sercinoglu, Zach Gleicher, Juliette Love, Paul Voigtlaender, Rohan Jain, Gabriela Surita, Kareem Mohamed, Rory Blevins, Junwhan Ahn, Tao Zhu, Kornraphop Kawintiranon, Orhan Firat, Yiming Gu, Yujing Zhang, Matthew Rahtz, Manaal Faruqui, Natalie Clay, Justin Gilmer, JD Co-Reyes, Ivo Penchev, Rui Zhu, Nobuyuki Morioka, Kevin Hui, Krishna Haridasan, Victor Campos, Mahdis Mahdieh, Mandy Guo, Samer Hassan, Kevin Kilgour, Arpi Vezer, Heng-Tze Cheng, Raoul de Liedekerke, Siddharth Goyal, Paul Barham, DJ Strouse, Seb Noury, Jonas Adler, Mukund Sundararajan, Sharad Vikram, Dmitry Lepikhin, Michela Paganini, Xavier Garcia, Fan Yang, Dasha Valter, Maja Trebacz, Kiran Vodrahalli, Chulayuth Asawaroengchai, Roman Ring, Norbert Kalb, Livio Baldini Soares, Siddhartha Brahma, David Steiner, Tianhe Yu, Fabian Mentzer, Antoine He, Lucas Gonzalez, Bibo Xu, Raphael Lopez Kaufman, Laurent El Shafey, Junhyuk Oh, Tom Hennigan, George van den Driessche, Seth Odoom, Mario Lucic, Becca Roelofs, Sid Lall, Amit Marathe, Betty Chan, Santiago Ontanon, Luheng He, Denis Teplyashin, Jonathan Lai, Phil Crone, Bogdan Damoc, Lewis Ho, Sebastian Riedel, Karel Lenc, Chih-Kuan Yeh, Aakanksha Chowdhery, Yang Xu, Mehran Kazemi, Ehsan Amid, Anastasia Petrushkina, Kevin Swersky, Ali Khodaei, Gowoon Chen, Chris Larkin, Mario Pinto, Geng Yan, Adria Puigdomenech Badia, Piyush Patil, Steven Hansen, Dave Orr, Sebastien M. R. Arnold, Jordan Grimstad, Andrew Dai, Sholto Douglas, Rishika Sinha, Vikas Yadav, Xi Chen, Elena Gribovskaya, Jacob Austin, Jeffrey Zhao, Kaushal Patel, Paul Komarek, Sophia Austin, Sebastian Borgeaud, Linda Friso, Abhimanyu Goyal, Ben Caine, Kris Cao, Da-Woon Chung, Matthew Lamm, Gabe Barth-Maron, Thais Kagohara, Kate Olszewska, Mia Chen, Kaushik Shivakumar, Rishabh Agarwal, Harshal Godhia, Ravi Rajwar, Javier Snaider, Xerxes Dotiwalla, Yuan Liu, Aditya Barua, Victor Ungureanu, Yuan Zhang, Bat-Orgil Batsaikhan, Mateo Wirth, James Qin, Ivo Danihelka, Tulsee Doshi, Martin Chadwick, Jilin Chen, Sanil Jain, Quoc Le, Arjun Kar, Madhu Gurumurthy, Cheng Li, Ruoxin Sang, Fangyu Liu, Lampros Lamprou, Rich Munoz, Nathan Lintz, Harsh Mehta, Heidi Howard, Malcolm Reynolds, Lora Aroyo, Quan Wang, Lorenzo Blanco, Albin Cassirer, Jordan Griffith, Dipanjan Das, Stephan Lee, Jakub Sygnowski, Zach Fisher, James Besley, Richard Powell, Zafarali Ahmed, Dominik Paulus, David Reitter, Zalan Borsos, Rishabh Joshi, Aedan Pope, Steven Hand, Vittorio Selo, Vihan Jain, Nikhil Sethi, Megha Goel, Takaki Makino, Rhys May, Zhen Yang, Johan Schalkwyk, Christina Butterfield, Anja Hauth, Alex Goldin, Will Hawkins, Evan Senter, Sergey Brin, Oliver Woodman, Marvin Ritter, Eric Noland, Minh Giang, Vijay Bolina, Lisa Lee, Tim Blyth, Ian Mackinnon, Machel Reid, Obaid Sarvana, David Silver, Alexander Chen, Lily Wang, Loren Maggiore, Oscar Chang, Nithya Attaluri, Gregory Thornton, Chung-Cheng Chiu, Oskar Bunyan, Nir Levine, Timothy Chung, Evgenii Eltyshev, Xiance Si, Timothy Lillicrap, Demetra Brady, Vaibhav Aggarwal, Boxi Wu, Yuanzhong Xu, Ross McIlroy, Kartikeya Badola, Paramjit Sandhu, Erica Moreira, Wojciech Stokowiec, Ross Hemsley, Dong Li, Alex Tudor, Pranav Shyam, Elahe Rahimtoroghi, Salem Haykal, Pablo Sprechmann, Xiang Zhou, Diana Mincu, Yujia Li, Ravi Addanki, Kalpesh Krishna, Xiao Wu, Alexandre Frechette, Matan Eyal, Allan Dafoe, Dave Lacey, Jay Whang, Thi Avrahami, Ye Zhang, Emanuel Taropa, Hanzhao Lin, Daniel Toyama, Eliza Rutherford, Motoki Sano, HyunJeong Choe, Alex Tomala, Chalence Safranek-Shrader, Nora Kassner, Mantas Pajarskas, Matt Harvey, Sean Sechrist, Meire Fortunato, Christina Lyu, Gamaleldin Elsayed, Chenkai Kuang, James Lottes, Eric Chu, Chao Jia, Chih-Wei Chen, Peter Humphreys, Kate Baumli, Connie Tao, Rajkumar Samuel, Cicero Nogueira dos Santos, Anders Andreassen, Nemanja Rakićević, Dominik Grewe, Aviral Kumar, Stephanie Winkler, Jonathan Caton, Andrew Brock, Sid Dalmia, Hannah Sheahan, Iain Barr, Yingjie Miao, Paul Natsev, Jacob Devlin, Feryal Behbahani, Flavien Prost, Yanhua Sun, Artiom Myaskovsky, Thanumalayan Sankaranarayana Pillai, Dan Hurt, Angeliki Lazaridou, Xi Xiong, Ce Zheng, Fabio Pardo, Xiaowei Li, Dan Horgan, Joe Stanton, Moran Ambar, Fei Xia, Alejandro Lince, Mingqiu Wang, Basil Mustafa, Albert Webson, Hyo Lee, Rohan Anil, Martin Wicke, Timothy Dozat, Abhishek Sinha, Enrique Piqueras, Elahe Dabir, Shyam Upadhyay, Anudhyan Boral, Lisa Anne Hendricks, Corey Fry, Josip Djolonga, Yi Su, Jake Walker, Jane Labanowski, Ronny Huang, Vedant Misra, Jeremy Chen, RJ Skerry-Ryan, Avi Singh, Shruti Rijhwani, Dian Yu, Alex Castro-Ros, Beer Changpinyo, Romina Datta, Sumit Bagri, Arnar Mar Hrafnkelsson, Marcello Maggioni, Daniel Zheng, Yury Sulsky, Shaobo Hou, Tom Le Paine, Antoine Yang, Jason Riesa, Dominika Rogozinska, Dror Marcus, Dalia El Badawy, Qiao Zhang, Luyu Wang, Helen Miller, Jeremy Greer, Lars Lowe Sjos, Azade Nova, Heiga Zen, Rahma Chaabouni, Mihaela Rosca, Jiepu Jiang, Charlie Chen, Ruibo Liu, Tara Sainath, Maxim Krikun, Alex Polozov, Jean-Baptiste Lespiau, Josh Newlan, Zeyncep Cankara, Soo Kwak, Yunhan Xu, Phil Chen, Andy Coenen, Clemens Meyer, Katerina Tsihlas, Ada Ma, Juraj Gottweis, Jinwei Xing, Chenjie Gu, Jin Miao, Christian Frank, Zeynep Cankara, Sanjay Ganapathy, Ishita Dasgupta, Steph Hughes-Fitt, Heng Chen, David Reid, Keran Rong, Hongmin Fan, Joost van Amersfoort, Vincent Zhuang, Aaron

Cohen, Shixiang Shane Gu, Anhad Mohananey, Anastasija Ilic, Taylor Tobin, John Wieting, Anna Bortsova, Phoebe Thacker, Emma Wang, Emily Caveness, Justin Chiu, Eren Sezener, Alex Kaskasoli, Steven Baker, Katie Millican, Mohamed Elhawaty, Kostas Aisopos, Carl Lebsack, Nathan Byrd, Hanjun Dai, Wenhao Jia, Matthew Wiethoff, Elnaz Davoodi, Albert Weston, Lakshman Yagati, Arun Ahuja, Isabel Gao, Golan Pundak, Susan Zhang, Michael Azzam, Khe Chai Sim, Sergi Caelles, James Keeling, Abhanshu Sharma, Andy Swing, YaGuang Li, Chenxi Liu, Carrie Grimes Bostock, Yamini Bansal, Zachary Nado, Ankesh Anand, Josh Lipschultz, Abhijit Karmarkar, Lev Proleev, Abe Ittycheriah, Soheil Hassas Yeganeh, George Polovets, Aleksandra Faust, Jiao Sun, Alban Rrustemi, Pen Li, Rakesh Shivanna, Jeremiah Liu, Chris Welty, Federico Lebron, Anirudh Baddepudi, Sebastian Krause, Emilio Parisotto, Radu Soricut, Zheng Xu, Dawn Bloxwich, Melvin Johnson, Behnam Neyshabur, Justin Mao-Jones, Renshen Wang, Vinay Ramasesh, Zaheer Abbas, Arthur Guez, Constant Segal, Duc Dung Nguyen, James Svensson, Le Hou, Sarah York, Kieran Milan, Sophie Bridgers, Wiktor Gworek, Marco Tagliasacchi, James Lee-Thorp, Michael Chang, Alexey Guseynov, Ale Jakse Hartman, Michael Kwong, Ruizhe Zhao, Sheleem Kashem, Elizabeth Cole, Antoine Miech, Richard Tanburn, Mary Phuong, Filip Pavetic, Sebastien Cevey, Ramona Comanescu, Richard Ives, Sherry Yang, Cosmo Du, Bo Li, Zizhao Zhang, Mariko Iinuma, Clara Huiyi Hu, Aurko Roy, Shaan Bijwadia, Zhenkai Zhu, Danilo Martins, Rachel Saputro, Anita Gergely, Steven Zheng, Dawei Jia, Ioannis Antonoglou, Adam Sadovsky, Shane Gu, Yingying Bi, Alek Andreev, Sina Samangooei, Mina Khan, Tomas Kocisky, Angelos Filos, Chintu Kumar, Colton Bishop, Adams Yu, Sarah Hodkinson, Sid Mittal, Premal Shah, Alexandre Moufarek, Yong Cheng, Adam Bloniarz, Jaehoon Lee, Pedram Pejman, Paul Michel, Stephen Spencer, Vladimir Feinberg, Xuehan Xiong, Nikolay Savinov, Charlotte Smith, Siamak Shakeri, Dustin Tran, Mary Chesus, Bernd Bohnet, George Tucker, Tamara von Glehn, Carrie Muir, Yiran Mao, Hideto Kazawa, Ambrose Slone, Kedar Soparkar, Disha Shrivastava, James Cobon-Kerr, Michael Sharman, Jay Pavagadhi, Carlos Araya, Karolis Misiunas, Nimesh Ghelani, Michael Laskin, David Barker, Qiujia Li, Anton Briukhov, Neil Houlsby, Mia Glaese, Balaji Lakshminarayanan, Nathan Schucher, Yunhao Tang, Eli Collins, Hyeontaek Lim, Fangxiaoyu Feng, Adria Recasens, Guangda Lai, Alberto Magni, Nicola De Cao, Aditya Siddhant, Zoe Ashwood, Jordi Orbay, Mostafa Dehghani, Jenny Brennan, Yifan He, Kelvin Xu, Yang Gao, Carl Saroufim, James Molloy, Xinyi Wu, Seb Arnold, Solomon Chang, Julian Schrittwieser, Elena Buchatskaya, Soroush Radpour, Martin Polacek, Skye Giordano, Ankur Bapna, Simon Tokumine, Vincent Hellendoorn, Thibault Sottiaux, Sarah Cogan, Aliaksei Severyn, Mohammad Saleh, Shantanu Thakoor, Laurent Shefey, Siyuan Qiao, Meenu Gaba, Shuo yiin Chang, Craig Swanson, Biao Zhang, Benjamin Lee, Paul Kishan Rubenstein, Gan Song, Tom Kwiatkowski, Anna Koop, Ajay Kannan, David Kao, Parker Schuh, Axel Stjerngren, Golnaz Ghiasi, Gena Gibson, Luke Vilnis, Ye Yuan, Felipe Tiengo Ferreira, Aishwarya Kamath, Ted Klimenko, Ken Franko, Kefan Xiao, Indro Bhattacharya, Miteyan Patel, Rui Wang, Alex Morris, Robin Strudel, Vivek Sharma, Peter Choy, Sayed Hadi Hashemi, Jessica Landon, Mara Finkelstein, Priya Jhakra, Justin Frye, Megan Barnes, Matthew Mauger, Dennis Daun, Khuslen Baatarsukh, Matthew Tung, Wael Farhan, Henryk Michalewski, Fabio Viola, Felix de Chaumont Quitry, Charline Le Lan, Tom Hudson, Qingze Wang, Felix Fischer, Ivy Zheng, Elspeth White, Anca Dragan, Jean baptiste Alayrac, Eric Ni, Alexander Pritzel, Adam Iwanicki, Michael Isard, Anna Bulanova, Lukas Zilka, Ethan Dyer, Devendra Sachan, Srivatsan Srinivasan, Hannah Muckenhirn, Honglong Cai, Amol Mandhane, Mukarram Tariq, Jack W. Rae, Gary Wang, Kareem Ayoub, Nicholas FitzGerald, Yao Zhao, Woohyun Han, Chris Alberti, Dan Garrette, Kashyap Krishnakumar, Mai Gimenez, Anselm Levskaya, Daniel Sohn, Josip Matak, Inaki Iturrate, Michael B. Chang, Jackie Xiang, Yuan Cao, Nishant Ranka, Geoff Brown, Adrian Hutter, Vahab Mirrokni, Nanxin Chen, Kaisheng Yao, Zoltan Egyed, Francois Galilee, Tyler Liechty, Praveen Kallakuri, Evan Palmer, Sanjay Ghemawat, Jasmine Liu, David Tao, Chloe Thornton, Tim Green, Mimi Jasarevic, Sharon Lin, Victor Cotruta, Yi-Xuan Tan, Noah Fiedel, Hongkun Yu, Ed Chi, Alexander Neitz, Jens Heitkaemper, Anu Sinha, Denny Zhou, Yi Sun, Charbel Kaed, Brice Hulse, Swaroop Mishra, Maria Georgaki, Sneha Kudugunta, Clement Farabet, Izhak Shafran, Daniel Vlasic, Anton Tsitsulin, Rajagopal Ananthanarayanan, Alen Carin, Guolong Su, Pei Sun, Shashank V, Gabriel Carvajal, Josef Broder, Iulia Comsa, Alena Repina, William Wong, Warren Weilun Chen, Peter Hawkins, Egor Filonov, Lucia Loher, Christoph Hirnschall, Weiyi Wang, Jingchen Ye, Andrea Burns, Hardie Cate, Diana Gage Wright, Federico Piccinini, Lei Zhang, Chu-Cheng Lin, Ionel Gog, Yana Kulizhskaya, Ashwin Sreevatsa, Shuang Song, Luis C. Cobo, Anand Iyer, Chetan Tekur, Guillermo Garrido, Zhuyun Xiao, Rupert Kemp, Huaixiu Steven Zheng, Hui Li, Ananth Agarwal, Christel Ngani, Kati Goshvadi, Rebeca Santamaria-Fernandez, Wojciech Fica, Xinyun Chen, Chris Gorgolewski, Sean Sun, Roopal Garg, Xinyu Ye, S. M. Ali Eslami, Nan Hua, Jon Simon, Pratik Joshi, Yelin Kim, Ian Tenney, Sahitya Potluri, Lam Nguyen Thiet, Quan Yuan, Florian Luisier, Alexandra Chronopoulou, Salvatore Scellato, Praveen Srinivasan, Minmin Chen, Vinod Koverkathu, Valentin Dalibard, Yaming Xu, Brennan Saeta, Keith Anderson, Thibault Sellam, Nick Fernando, Fantine Huot, Junehyuk Jung, Mani Varadarajan, Michael Quinn, Amit Raul, Maigo Le, Ruslan Habalov, Jon Clark, Komal Jalan, Kalesha Bullard, Achintya Singhal, Thang Luong, Boyu Wang, Sujeevan Rajayogam, Julian Eisenschlos, Johnson Jia, Daniel Finchelstein, Alex Yakubovich, Daniel Balle, Michael Fink, Sameer Agarwal, Jing Li, Dj Dvijotham, Shalini Pal, Kai Kang, Jaclyn Konzelmann, Jennifer Beattie, Olivier Dousse, Diane Wu, Remi Crocker, Chen Elkind, Siddhartha Reddy

Jonnalagadda, Jong Lee, Dan Holtmann-Rice, Krystal Kallarackal, Rosanne Liu, Denis Vnukov, Neera Vats, Luca Invernizzi, Mohsen Jafari, Huanjie Zhou, Lilly Taylor, Jennifer Prendki, Marcus Wu, Tom Eccles, Tianqi Liu, Kavya Kopparapu, Francoise Beaufays, Christof Angermueller, Andreea Marzoca, Shourya Sarcar, Hilal Dib, Jeff Stanway, Frank Perbet, Nejc Trdin, Rachel Sterneck, Andrey Khorlin, Dinghua Li, Xihui Wu, Sonam Goenka, David Madras, Sasha Goldshtein, Willi Gierke, Tong Zhou, Yaxin Liu, Yannie Liang, Anais White, Yunjie Li, Shreya Singh, Sanaz Bahargam, Mark Epstein, Sujoy Basu, Li Lao, Adnan Ozturel, Carl Crous, Alex Zhai, Han Lu, Zora Tung, Neeraj Gaur, Alanna Walton, Lucas Dixon, Ming Zhang, Amir Globerson, Grant Uy, Andrew Bolt, Olivia Wiles, Milad Nasr, Ilia Shumailov, Marco Selvi, Francesco Piccinno, Ricardo Aguilar, Sara McCarthy, Misha Khalman, Mrinal Shukla, Vlado Galic, John Carpenter, Kevin Villela, Haibin Zhang, Harry Richardson, James Martens, Matko Bosnjak, Shreyas Rammohan Belle, Jeff Seibert, Mahmoud Alnahlawi, Brian McWilliams, Sankalp Singh, Annie Louis, Wen Ding, Dan Popovici, Lenin Simicich, Laura Knight, Pulkit Mehta, Nishesh Gupta, Chongyang Shi, Saaber Fatehi, Jovana Mitrovic, Alex Grills, Joseph Pagadora, Dessie Petrova, Danielle Eisenbud, Zhishuai Zhang, Damion Yates, Bhavishya Mittal, Nilesh Tripuraneni, Yannis Assael, Thomas Brovelli, Prateek Jain, Mihajlo Velimirovic, Canfer Akbulut, Jiaqi Mu, Wolfgang Macherey, Ravin Kumar, Jun Xu, Haroon Qureshi, Gheorghe Comanici, Jeremy Wiesner, Zhitao Gong, Anton Ruddock, Matthias Bauer, Nick Felt, Anirudh GP, Anurag Arnab, Dustin Zelle, Jonas Rothfuss, Bill Rosgen, Ashish Shenoy, Bryan Seybold, Xinjian Li, Jayaram Mudigonda, Goker Erdogan, Jiawei Xia, Jiri Simsa, Andrea Michi, Yi Yao, Christopher Yew, Steven Kan, Isaac Caswell, Carey Radebaugh, Andre Elisseeff, Pedro Valenzuela, Kay McKinney, Kim Paterson, Albert Cui, Eri Latorre-Chimoto, Solomon Kim, William Zeng, Ken Durden, Priya Ponnapalli, Tiberiu Sosea, Christopher A. Choquette-Choo, James Manyika, Brona Robenek, Harsha Vashisht, Sebastien Pereira, Hoi Lam, Marko Velic, Denese Owusu-Afriyie, Katherine Lee, Tolga Bolukbasi, Alicia Parrish, Shawn Lu, Jane Park, Balaji Venkatraman, Alice Talbert, Lambert Rosique, Yuchung Cheng, Andrei Sozanschi, Adam Paszke, Praveen Kumar, Jessica Austin, Lu Li, Khalid Salama, Wooyeol Kim, Nandita Dukkipati, Anthony Baryshnikov, Christos Kaplanis, Xiang-Hai Sheng, Yuri Chervonyi, Caglar Unlu, Diego de Las Casas, Harry Askham, Kathryn Tunyasuvunakool, Felix Gimeno, Siim Poder, Chester Kwak, Matt Miecnikowski, Vahab Mirrokni, Alek Dimitriev, Aaron Parisi, Dangyi Liu, Tomy Tsai, Toby Shevlane, Christina Kouridi, Drew Garmon, Adrian Goedeckemeyer, Adam R. Brown, Anitha Vijayakumar, Ali Elqursh, Sadegh Jazayeri, Jin Huang, Sara Mc Carthy, Jay Hoover, Lucy Kim, Sandeep Kumar, Wei Chen, Courtney Biles, Garrett Bingham, Evan Rosen, Lisa Wang, Qijun Tan, David Engel, Francesco Pongetti, Dario de Cesare, Dongseong Hwang, Lily Yu, Jennifer Pullman, Srini Narayanan, Kyle Levin, Siddharth Gopal, Megan Li, Asaf Aharoni, Trieu Trinh, Jessica Lo, Norman Casagrande, Roopali Vij, Loic Matthey, Bramandia Ramadhana, Austin Matthews, CJ Carey, Matthew Johnson, Kremena Goranova, Rohin Shah, Shereen Ashraf, Kingshuk Dasgupta, Rasmus Larsen, Yicheng Wang, Manish Reddy Vuyyuru, Chong Jiang, Joana Ijazi, Kazuki Osawa, Celine Smith, Ramya Sree Boppana, Taylan Bilal, Yuma Koizumi, Ying Xu, Yasemin Altun, Nir Shabat, Ben Bariach, Alex Korchemniy, Kiam Choo, Olaf Ronneberger, Chimezie Iwuanyanwu, Shubin Zhao, David Soergel, Cho-Jui Hsieh, Irene Cai, Shariq Iqbal, Martin Sundermeyer, Zhe Chen, Elie Bursztein, Chaitanya Malaviya, Fadi Biadsy, Prakash Shroff, Inderjit Dhillon, Tejasi Latkar, Chris Dyer, Hannah Forbes, Massimo Nicosia, Vitaly Nikolaev, Somer Greene, Marin Georgiev, Pidong Wang, Nina Martin, Hanie Sedghi, John Zhang, Praseem Banzal, Doug Fritz, Vikram Rao, Xuezhi Wang, Jiageng Zhang, Viorica Patraucean, Dayou Du, Igor Mordatch, Ivan Jurin, Lewis Liu, Ayush Dubey, Abhi Mohan, Janek Nowakowski, Vlad-Doru Ion, Nan Wei, Reiko Tojo, Maria Abi Raad, Drew A. Hudson, Vaishakh Keshava, Shubham Agrawal, Kevin Ramirez, Zhichun Wu, Hoang Nguyen, Ji Liu, Madhavi Sewak, Bryce Petrini, DongHyun Choi, Ivan Philips, Ziyue Wang, Ioana Bica, Ankush Garg, Jarek Wilkiewicz, Priyanka Agrawal, Xiaowei Li, Danhao Guo, Emily Xue, Naseer Shaik, Andrew Leach, Sadh MNM Khan, Julia Wiesinger, Sammy Jerome, Abhishek Chakladar, Alek Wenjiao Wang, Tina Ornduff, Folake Abu, Alireza Ghaffarkhah, Marcus Wainwright, Mario Cortes, Frederick Liu, Joshua Maynez, Andreas Terzis, Pouya Samangouei, Riham Mansour, Tomasz Kępa, François-Xavier Aubet, Anton Algymr, Dan Banica, Agoston Weisz, Andras Orban, Alexandre Senges, Ewa Andrejczuk, Mark Geller, Niccolo Dal Santo, Valentin Anklin, Majd Al Merey, Martin Baeuml, Trevor Strohman, Junwen Bai, Slav Petrov, Yonghui Wu, Demis Hassabis, Koray Kavukcuoglu, Jeffrey Dean, and Oriol Vinyals. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *Preprint*, arXiv:2403.05530.

Zelalem Gero, Chandan Singh, Hao Cheng, Tristan Naumann, Michel Galley, Jianfeng Gao, and Hoifung Poon. 2023. Self-verification improves few-shot clinical information extraction.

Zhibin Gou, Zhihong Shao, Yeyun Gong, yelong shen, Yujiu Yang, Nan Duan, and Weizhu Chen. 2024. CRITIC: Large language models can self-correct with tool-interactive critiquing. In *The Twelfth International Conference on Learning Representations*.

Nuno M. Guerreiro, Duarte M. Alves, Jonas Waldendorf, Barry Haddow, Alexandra Birch, Pierre Colombo, and André F. T. Martins. 2023. Hallucinations in large multilingual translation models. *Transactions of the Association for Computational Linguistics*, 11:1500–1517.

Fatemeh Haji, Mazal Bethany, Maryam Tabar, Jason Chiang, Anthony Rios, and Peyman Najafirad. 2024.

Improving llm reasoning with multi-agent tree-of-thought validator agent. *Preprint*, arXiv:2409.11527.

Yebowen Hu, Timothy Ganter, Hanieh Deilamsalehy, Franck Dernoncourt, Hassan Foroosh, and Fei Liu. 2023. MeetingBank: A benchmark dataset for meeting summarization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16409–16423, Toronto, Canada. Association for Computational Linguistics.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2024a. Large language models cannot self-correct reasoning yet. In *The Twelfth International Conference on Learning Representations*.

Kung-Hsiang Huang, Philippe Laban, Alexander Fabbri, Prafulla Kumar Choubey, Shafiq Joty, Caiming Xiong, and Chien-Sheng Wu. 2024b. Embrace divergence for richer insights: A multi-document summarization benchmark and a case study on summarizing diverse information from news articles. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 570–593, Mexico City, Mexico. Association for Computational Linguistics.

Shuyang Jiang, Yuhao Wang, and Yu Wang. 2023. Self-evolve: A code evolution framework via large language models. *Preprint*, arXiv:2306.02907.

Kyungha Kim, Sangyun Lee, Kung-Hsiang Huang, Hou Pong Chan, Manling Li, and Heng Ji. 2024. Can llms produce faithful explanations for fact-checking? towards faithful explainable fact-checking via multi-agent debate. *Preprint*, arXiv:2402.07401.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2021. Summac: Re-visiting nli-based models for inconsistency detection in summarization. *Preprint*, arXiv:2111.09525.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023a. Camel: Communicative agents for" mind" exploration of large language model society. *Advances in Neural Information Processing Systems*, 36:51991–52008.

Ruosen Li, Teerth Patel, and Xinya Du. 2023b. Prd: Peer rank and discussion improve large language model based evaluations. *arXiv preprint arXiv:2307.02762*.

Zhen Li, Xiaohan Xu, Tao Shen, Can Xu, Jia-Chen Gu, Yuxuan Lai, Chongyang Tao, and Shuai Ma. 2024. Leveraging large language models for nlg evaluation: Advances and challenges. *Preprint*, arXiv:2401.07103.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Zhaopeng Tu, and Shuming Shi. 2023. Encouraging divergent thinking in large language models through multi-agent debate. *arXiv preprint arXiv:2305.19118*.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023a. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Yixin Liu, Budhaditya Deb, Milagro Teruel, Aaron Halfaker, Dragomir Radev, and Ahmed Hassan Awadallah. 2023b. On improving summarization factual consistency from natural language feedback. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15144–15161, Toronto, Canada. Association for Computational Linguistics.

Yixin Liu, Alexander Fabbri, Jiawen Chen, Yilun Zhao, Simeng Han, Shafiq Joty, Pengfei Liu, Dragomir Radev, Chien-Sheng Wu, and Arman Cohan. 2024. Benchmarking generation and evaluation capabilities of large language models for instruction controllable summarization. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 4481–4501, Mexico City, Mexico. Association for Computational Linguistics.

Li-Chun Lu, Shou-Jen Chen, Tsung-Min Pai, Chan-Hung Yu, Hung yi Lee, and Shao-Hua Sun. 2024. LLM discussion: Enhancing the creativity of large language models via discussion framework and role-play. In *First Conference on Language Modeling*.

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. In *Advances in Neural Information Processing Systems*, volume 36, pages 46534–46594. Curran Associates, Inc.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. 2021. Webgpt: Browser-assisted question-answering with human feedback. *CoRR*, abs/2112.09332.

Duy-Hung Nguyen, Nguyen Viet Dung Nghiem, Bao-Sinh Nguyen, Dung Tien Tien Le, Shahab Sabahi, Minh-Tien Nguyen, and Hung Le. 2022. Make the most of prior data: A solution for interactive text summarization with preference feedback. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1919–1930, Seattle, United States. Association for Computational Linguistics.

Theo X. Olausson, Jeevana Priya Inala, Chenglong Wang, Jianfeng Gao, and Armando Solar-Lezama. 2024. Is self-repair a silver bullet for code generation? In *The Twelfth International Conference on Learning Representations*.

OpenAI. 2024. Hello gpt-4o.

Debjit Paul, Mete Ismayilzada, Maxime Peyrard, Beatriz Borges, Antoine Bosselut, Robert West, and Boi Faltings. 2024. REFINER: Reasoning feedback on intermediate representations. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1100–1126, St. Julian's, Malta. Association for Computational Linguistics.

Vikas Raunak, Amr Sharaf, Yiren Wang, Hany Awadalla, and Arul Menezes. 2023. Leveraging GPT-4 for automatic translation post-editing. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12009–12024, Singapore. Association for Computational Linguistics.

Vipula Rawte, Swagata Chakraborty, Agnibh Pathak, Anubhav Sarkar, S.M Towhidul Islam Tonmoy, Aman Chadha, Amit Sheth, and Amitava Das. 2023. The troubling emergence of hallucination in large language models - an extensive definition, quantification, and prescriptive remediations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2541–2573, Singapore. Association for Computational Linguistics.

Jarem Saunders. 2023. Improving automated prediction of English lexical blends through the use of observable linguistic features. In *Proceedings of the 20th SIGMORPHON workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 93–97, Toronto, Canada. Association for Computational Linguistics.

Jérémy Scheurer, Jon Ander Campos, Tomasz Korbak, Jun Shern Chan, Angelica Chen, Kyunghyun Cho, and Ethan Perez. 2024. Training language models with language feedback at scale. *Preprint*, arXiv:2303.16755.

Yixiao Song, Yekyung Kim, and Mohit Iyyer. 2024. Veriscore: Evaluating the factuality of verifiable claims in long-form text generation. *Preprint*, arXiv:2406.19276.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems*, volume 33, pages 3008–3021. Curran Associates, Inc.

Xiaoxi Sun, Jinpeng Li, Yan Zhong, Dongyan Zhao, and Rui Yan. 2024. Towards detecting llms hallucination via markov chain-based multi-agent debate framework. *Preprint*, arXiv:2406.03075.

Liyan Tang, Tanya Goyal, Alex Fabbri, Philippe Laban, Jiacheng Xu, Semih Yavuz, Wojciech Kryscinski, Justin Rousseau, and Greg Durrett. 2023. Understanding factual errors in summarization: Errors, summarizers, datasets, error detectors. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11626–11644, Toronto, Canada. Association for Computational Linguistics.

Liyan Tang, Philippe Laban, and Greg Durrett. 2024a. Minicheck: Efficient fact-checking of llms on grounding documents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Liyan Tang, Igor Shalyminov, Amy Wong, Jon Burnsky, Jake Vincent, Yu'an Yang, Siffi Singh, Song Feng, Hwanjun Song, Hang Su, Lijia Sun, Yi Zhang, Saab Mansour, and Kathleen McKeown. 2024b. TofuEval: Evaluating hallucinations of LLMs on topic-focused dialogue summarization. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4455–4480, Mexico City, Mexico. Association for Computational Linguistics.

Liyan Tang, Igor Shalyminov, Amy Wong, Jon Burnsky, Jake Vincent, Yu'an Yang, Siffi Singh, Song Feng, Hwanjun Song, Hang Su, Justin Sun, Yi Zhang, Saab Mansour, and Kathleen McKeown. 2024c. Tofueval: Evaluating hallucinations of llms on topic-focused dialogue summarization. In *NAACL 2024*.

Ziyi Tang, Ruilin Wang, Weixing Chen, Keze Wang, Yang Liu, Tianshui Chen, and Liang Lin. 2024d. Towards causalgpt: A multi-agent approach for faithful knowledge reasoning via promoting causal consistency in llms. *Preprint*, arXiv:2308.11914.

James Thorne and Andreas Vlachos. 2021. Evidence-based factual error correction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3298–3309, Online. Association for Computational Linguistics.

Liam van der Poel, Ryan Cotterell, and Clara Meister. 2022. Mutual information alleviates hallucinations in abstractive summarization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5956–5965, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Manya Wadhwa, Xinyu Zhao, Junyi Jessy Li, and Greg Durrett. 2024. Learning to refine with fine-grained natural language feedback. *Preprint*, arXiv:2407.02397.

David Wan, Koustuv Sinha, Srini Iyer, Asli Celikyilmaz, Mohit Bansal, and Ramakanth Pasunuru. 2024. ACUEval: Fine-grained hallucination evaluation and correction for abstractive summarization. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10036–10056, Bangkok, Thailand. Association for Computational Linguistics.

Zhenhailong Wang, Shaoguang Mao, Wenshan Wu, Tao Ge, Furu Wei, and Heng Ji. 2024. Unleashing the emergent cognitive synergy in large language models: A task-solving agent through multi-persona self-collaboration. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 257–279, Mexico City, Mexico. Association for Computational Linguistics.

Jeff Wu, Long Ouyang, Daniel M. Ziegler, Nisan Stiennon, Ryan Lowe, Jan Leike, and Paul Christiano. 2021. Recursively summarizing books with human feedback. *Preprint*, arXiv:2109.10862.

Fangyuan Xu, Yixiao Song, Mohit Iyyer, and Eunsol Choi. 2023. A critical evaluation of evaluations for long-form question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3225–3245, Toronto, Canada. Association for Computational Linguistics.

Wenda Xu, Daniel Deutsch, Mara Finkelstein, Juraj Juraska, Biao Zhang, Zhongtao Liu, William Yang Wang, Lei Li, and Markus Freitag. 2024. LLMRefine: Pinpointing and refining large language models via fine-grained actionable feedback. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1429–1445, Mexico City, Mexico. Association for Computational Linguistics.

Hongbin Ye, Tong Liu, Aijia Zhang, Wei Hua, and Weiqiang Jia. 2023. Cognitive mirage: A review of hallucinations in large language models. *Preprint*, arXiv:2309.06794.

Zhangyue Yin, Qiushi Sun, Cheng Chang, Qipeng Guo, Junqi Dai, Xuanjing Huang, and Xipeng Qiu. 2023. Exchange-of-thought: Enhancing large language model capabilities through cross-model communication. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15135–15153, Singapore. Association for Computational Linguistics.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. AlignScore: Evaluating factual consistency with a unified alignment function. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11328–11348, Toronto, Canada. Association for Computational Linguistics.

Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A. Smith. 2023. How language model hallucinations can snowball. *Preprint*, arXiv:2305.13534.

Chenguang Zhu, Yang Liu, Jie Mei, and Michael Zeng. 2021. MediaSum: A large-scale media interview dataset for dialogue summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5927–5934, Online. Association for Computational Linguistics.

A Experimental Setup Details

In our experiments, we first test multi-agent and multi-model approaches to each component separately using intrinsic evaluations, and then combine these components and measure end-to-end refinement performance. Here, we describe the setup of our intrinsic evaluations for the different subtasks, as shown in Figure 2, and then detail our final evaluation setup on three summarization benchmarks.

A.1 Models

We use the latest versions of GPT-4o and Claude as of October 12, 2024. The number of parameters for these models has not been disclosed. We use the default decoding parameters for all models. For sentence splitting for DETECT, we utilize NLTK's library (Bird et al., 2009).

A.2 Datasets

We use annotations from TofuEval on the MediaSum and MeetingBank, released under the MIT-0 license. UltraChat and WebGPT are released under the MIT license. We follow the authors' instructions to process the data. To our knowledge, the authors of the datasets ensured that there are no harmful data. All datasets are in English.

A.3 Metrics

We use MiniCheck and AlignScore, following the authors' original repositories. For GPT-4 Likert, we use the *gpt-4-0125* version of GPT-4. For VeriScore, we use the authors' original code[5] and employ GPT-4o for extracting and verifying claims.

B Results Details

B.1 Full REFINE results

We report the full results with 2xG and 2xC critiques in Table 6. With 2xG critiques, we also observe that RERANK improves performance over

[5] https://github.com/Yixiao-Song/VeriScore

Method	M_R	2xG Critiques MediaSum MCS↑	G↑	MeetingBank MCS↑	G↑	Average MCS↑	G↑	2xC Critiques MediaSum MCS↑	G↑	MeetingBank MCS↑	G↑	Average MCS↑	G↑
Original	-	74.4	4.1	82.1	3.6	78.3	3.8	74.4	4.1	82.1	3.6	78.3	3.8
Single Agent	GPT-4o	79.6	4.4	86.9	4.0	<u>83.2</u>	4.2	82.1	4.4	87.0	3.9	<u>84.6</u>	4.2
	Claude	79.7	4.4	84.0	3.9	81.8	4.2	80.7	4.4	85.0	4.0	82.8	4.2
MASM w. RERANK	2xG	79.9	4.4	**87.0**	4.0	**83.5**	4.2	<u>82.4</u>	4.4	<u>87.4</u>	3.9	<u>84.9</u>	4.2
MASM w. RERANK	2xC	**81.0**	4.4	84.5	4.0	82.8	4.2	80.4	4.4	84.7	4.0	82.5	4.2
MAMM w. RERANK	G+C	<u>80.0</u>	4.4	85.9	4.0	83.0	4.2	81.9	4.4	85.0	3.9	83.4	4.2
MASM w. GENERATE	2xG	79.9	4.4	<u>86.5</u>	4.0	<u>83.2</u>	4.2	**82.5**	4.4	**87.8**	4.0	**85.2**	4.2
MASM w. GENERATE	2xC	76.9	4.3	80.2	3.9	78.5	4.1	76.7	4.4	81.6	4.0	79.1	4.2
MAMM w. GENERATE	G+C	78.0	4.3	82.6	3.9	80.3	4.1	78.9	4.4	83.9	4.0	81.4	4.2

Table 6: Full refine results with 2xG and 2xC critiques.

the single-agent baseline. Interestingly, in this setting, we do not observe any average improvement across both metrics using GENERATE. This highlights the effectiveness and reliability of RERANK compared to GENERATE.

B.2 Intrinsic Results on Multiple Iterations

DETECT. For the multi-agent models, we also analyze the balanced accuracy across multiple iterations, as the agents update their answers based on the other agent's response. As shown in Figure 3, one round of discussion provides the largest improvement, as the models show improvement in all cases except when using 2xG on MeetingBank. The largest improvement is with the multi-model setting, showcasing the benefit of having diverse responses. Additional rounds only address a few examples (e.g., 10/324 examples) and do not necessarily improve performance. We find that these are the harder examples on which the models have difficulty converging to an answer. See Appendix B.3 for more details.

RERANK. Examining how accuracy changes over multiple rounds of debate, as shown on the right of Figure 3, we find that agents improve the most during the second round and converge by then, except for the multi-model multi-agent G+C method. In this case, on MediaSum, we observe additional improvement at round 3.

CRITIQUE. We further show the performance of the two frameworks across multiple iterations on the left of Figure 4. While reranking improves with further iterations, asking the models to continue refining their generations degrades performance.

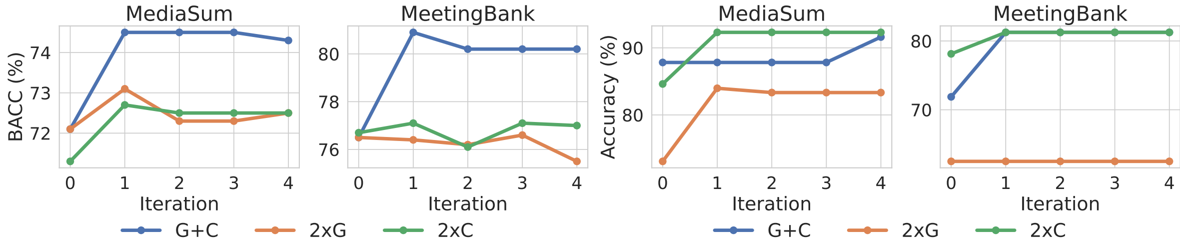
REFINE. When looking at the improvement across iterations on the right of Figure 4, 2xG consistently improves slightly across multiple iterations. However, both multi-agent approaches (G+C and 2xC) performance decreases. With GENERATE, we observe a large decrease in faithfulness score, highlighting the more reliable performance of RERANK.

Round	# Converged	BACC
0	751	76.2
1	50	52.7
2	8	50.0
3	3	50.0
4	1	0.0

Table 7: Number of converged examples for each round and the corresponding BACC for the subset.

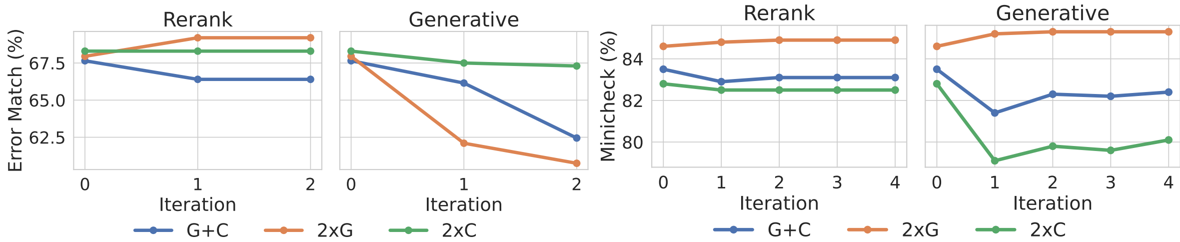
B.3 Analysis on Multi-Round for DETECT

To investigate whether the subsequent rounds involve harder examples that the agents have difficulty agreeing on, we calculate the balanced accuracy on the subset of examples, which the answer from the two agents finally converges for each round. We hypothesize that the model is unable to converge because both agents do not know the correct answer and thus the correct reasoning, making them incapable of convincing each other. We present the number of examples and the corresponding BACC for this subset in Table 7. We observe that for the 50 examples on which the agents converge in the first iteration, the BACC is already reduced from 76.2 to 52.7, and the remaining examples in the subsequent rounds only achieve accuracy at random chance levels. This indicates that the multi-agent approach can help improve performance on more examples but cannot improve cases where both agents are not confident.



(a) DETECT (b) RERANK

Figure 3: Detect and rerank multi-agent performance across multiple iterations.



(a) CRITIQUE (b) REFINE

Figure 4: Error match rate for CRITIQUE and faithfulness score for REFINE across multiple iterations.

M_D	M_C	M_R	MCS↑	GL↑
G+C	2xC	2xG	84.9	4.2
G+C	G+C	G+C	83.5	4.2
G+C+B	G+C+B	G+C+B	84.2	4.3
G+C+B	2xC	2xG	84.7	4.2

Table 8: Results with using three agents: GPT-4o (G), Claude (C), and Gemini (B). MCS=MiniCheck, GL=GPT4o 1-5 point Likert score.

B.4 Refinement with more Agents

We also explore the use of three agents to assess the effect of increased agent diversity. We include Gemini-1.5-flash (Gemini Team et al., 2024) as the third model, which performs similar to GPT4-o and Claude. Since the subtask that benefits the most from a multi-agent, multi-model approach is DETECT, we experiment by adding the third agent to DETECT only, as well as adding it to both DETECT and REFINE (and using the RERANK on the generations from the three agents). We present the results in Table 8. To measure the effect of adding a third agent for all subtasks, we compare the performance of using two agents across all subtasks with that of using three agents. We observe that using three agents provides additional gains in both the MiniCheck and Likert scores, indicating that having more models can indeed help. The three-agent version also achieves competitive scores with the variant where we use three agents for DETECT and the best configuration from our recipe for CRITIQUE and REFINE, indicating that having more agents may reduce the need for comprehensive testing to identify the best combination of subtasks.

The improvement observed with more agents can be attributed to more diverse outputs, each potentially containing different hallucinations due to the training paradigm. Alternatively, it can be thought of as the issue of hallucinations correlating with low confidence: Individual agents may produce hallucinations when they are less confident (Cao et al., 2022; van der Poel et al., 2022). However, the multi-agent framework mitigates hallucinations by enabling agents to collaborate and reach a consensus agreed upon by all (i.e., achieving high confidence), thereby improving faithfulness.

B.5 LFQA Results without Context

For the case without context, the model must retrieve information from its own parametric knowledge. Here, we use VeriScore, a state-of-the-art verification model from Song et al. (2024).

For the "no context" setting, which is reported in Table 9, though single agent and single model performs the best, our recipe improves over the original answers by 6.3%. Among the different variations, single-agent multi-model outperforms multi-agent single-model, indicating that there is still benefit of using multiple models.

Method	M_D	M_C	M_R	No Context VeriScore	With Context MCS↑	G-Likert↑
Original	-	-	-	62.8	76.7	3.5[†]
Single-Agent Single-Model	GPT-4o	GPT-4o	GPT-4o	**71.9**	80.1	3.9
Single-Agent Multi-Model	Claude	Claude	GPT-4o	71.4	80.9	4.0
Multi-Agent Single-Model	2xG	2xG	2xG	71.0	79.1	3.9
MAMM-REFINE (Ours)	G+C	2xC	2xG	70.2	**82.0**	**4.1**

Table 9: Results on Long-form QA for both with and without context.

Method	Detect BACC	Critique EM↑	EMM↓	NE↓
GPT-4o	72.1	95.1	5.0	0.0
Llama3.1-8B	62.2	67.2	32.8	0.0
MAMM w. Llama3.1-8B	71.1	93.8	6.25	0.0

Table 10: MediasSum results with using smaller model, Llama3.1-8b on DETECT and CRITIQUE.

B.6 Additional Analysis

We have observed that the initial performance of each agent before the debate is crucial, as the debate outcomes are heavily influenced by these starting points. Specifically, when there is a large discrepancy between the agents' performances, combining them can help improve the weaker agent while maintaining similar (or slightly worse) performance for the better agent. This dynamic explains why certain settings work better for specific subtasks. For instance, in the critique task, GPT-4o and Claude single agents differ by 3.4 EM points, and MAMM averages their performances. However, SMMA effectively enhances the performance of both agents, and since Claude performs better initially, the MASM achieves the highest overall score by leveraging its stronger baseline.

In contrast, the refine task presents a scenario where GPT-4o outperforms Claude as a single agent. Here, SMMA benefits more from GPT-4o's higher baseline, allowing it to refine and further improve its responses. Meanwhile, MAMM struggles due to Claude's relatively lower performance, which drags down the combined results. These observations demonstrate that SMMA is better suited for tasks where one agent consistently outperforms the other, as it capitalizes on the stronger model's ability to refine its outputs during debate.

Qualitatively, we also find that generations from the same models exhibit little variation, so MA of the same model does not significantly aid in providing options for reranking. Additionally, RERANK can make mistakes when faced with two choices of differing quality, especially in cases where the topics are marginal in TofuEval - when dealing with less frequently mentioned information.

To verify this, we also test on MediaSum by running both larger and smaller models, where the performance discrepancy is more pronounced. Specifically, we employ the Llama3.1-8B model and GPT-4o as two agents and evaluate them using DETECT and the gold setting of CRITIQUE. The results are shown in Table 10. We observe that the smaller 8B model significantly underperforms compared to GPT-4o. When combined in the MAMM setting, the overall performance is slightly lower than that of GPT-4o alone. In cases where the two models disagree, the smaller model agrees with the larger model about 82% of the time, thus failing to contribute to performance improvements. In the remaining 18% of cases, the larger model is persuaded to accept the incorrect answer from the smaller model. These findings underscore the importance of using agents with similar performance levels to achieve further improvements on subtasks.

C Human Evaluations

C.1 Critique Evaluation

We sampled 50 examples and asked two authors to annotate the data using the same instructions provided to GPT, i.e., selecting from three choices. Annotators did not see the generated scores for any of the examples. We observed an inter-annotator agreement of 0.80 using macro-F1 and the average IAA between the GPT-predicted labels and our annotations is 0.61. This demonstrates that the GPT-based metric is an efficient and effective automatic evaluation method.

C.2 Faithfulness Metric Correlations

We conducted a blind, Likert scale human evaluation on 25 samples from MediaSum with MAMM-generated summaries, using the same prompt as for the GPT-based metric. Our annotated Likert scores achieved a Kendall correlation of 0.46 with

Method	Prompt
Direct Refinement	I summarized the following document on the topic ‘{Topic}’: {Document} Summary of the above document on topic ‘{Topic}’: {Summary} If there are any factual inconsistencies in the summary then edit the summary such that the refinement doesn’t have any inconsistencies. Consistency in this context implies that all information presented in the summary is substantiated by the document. If the summary is consistent, then just the copy the same summary with no changes. When refining, make the minimum number of changes.
Detect	Document: {Document} Sentence: {Sentence} Determine if the sentence is factually consistent with the document provided above. A sentence is factually consistent if it can be entailed (either stated or implied) by the document. Please briefly explain the reason within 50 words. Output your answer in json format, with the format as follows: {{“reasoning”: “”, “answer”: “”}}. Please strictly output in JSON format. Only answer yes or no in the “answer” field.
Rerank	Document: {Document} Summarize the provided document focusing on "Topic". The summary should be less than 50 words in length. ### Summary 1: {Summary1} ### Summary 2: {Summary2} ... Select the best summary that contains the least amount of factual inconsistencies. Consistency in this context implies that all information presented in the summary is substantiated by the document. Please briefly explain the reason within 50 words. Output your answer in json format, with the format as follows: {{“reasoning”: “”, “answer”: “”}}. Please strictly output in JSON format. Only answer numbers in the "answer" field.
Critique	I summarized the following document on the topic: ‘{Topic}’: {Document} Summary of the above document on topic ‘{Topic}’: {Summary} Reason about the factually inconsistent span in the sentence. A span is factually inconsistent if it cannot be substantiated by the document. Give reasons for the factual inconsistency, point to the error span by stating “The error span: ⟨span from sentence⟩” and end your answer with a suggested fix to the summary.
Refine	I summarized the following document on the topic ‘{Topic}’: {Document} Summary of the above document on topic ‘{Topic}’: {Summary} Feedback for the above summary: {Feedback} Edit the user response such that the refinement doesn’t have any errors mentioned in the feedback. Make the minimum number of changes when doing the refinement. Do not include a preamble.
Multi-Agent Debate	{Initial Prompt} Carefully review the following solutions from other agents as additional information, and provide your own answer and step-by-step reasoning to the question. One agent’s answer: {{“reasoning”: {}, “answer”: {}}} One agent’s answer: {{“reasoning”: {}, “answer”: {}}}

Table 11: Prompts for different subtasks and multi-agent debate.

the GPT Likert scores, which is comparable to the correlation reported in G-Eval (Liu et al., 2023a), a SOTA, Likert-based evaluation metric (0.43).

D Prompts

We show the prompts for different pipelines in Table 11. We use the same 1-5 Likert prompt by Wadhwa et al. (2024), which contains a detailed rubric (Li et al., 2024).