CMMQC: Cascaded Multi-Model Quality Control for Unsupervised Data-to-Text Generation

Weitian Zhang[†], Xu Sun[†*], Yangxing Luo[†], Wei Gao[‡], Yanchun Zhu[‡]

[†] *Pazhou Lab*, Guangzhou, China
[‡] *China Unicom Digital Intelligence Medical Technology Co. Ltd.*, Guangzhou, China

***Abstract*—Data-to-text (D2T) generation, the task of converting structured data into natural language, has extensive real-world applications. While supervised models have achieved promising results, they rely heavily on costly labeled training data. This paper investigates unsupervised D2T generation by leveraging the impressive general abilities of large language models (LLMs). We propose a framework for LLMs to collaboratively learn from unlabeled data through cascaded multi-model quality control. Specifically, one LLM, acting as a writer, generates candidate texts from input data. Additional LLMs, serving as checkers, validate output quality to filter high-quality samples for training the writer LLM. By cascading generation, checking, and meta-checking, the models extract linguistic knowledge and grounding ability from abundant unlabeled data. Experiments on established benchmarks demonstrate enhanced fluency, accuracy, and coherence compared to supervised baselines. This unsupervised approach circumvents labeled data dependence, unlocking readily available LLMs for on-demand D2T generation across diverse applications.**

***Index Terms*—large language model, data-to-text, multi-model, unsupervised learning.**

I. Introduction

Data-to-text (D2T) generation constitutes a core natural language processing (NLP) task, whereby structured data is algorithmically converted into coherent natural language text. Through generating linguistically fluent descriptions from diverse data sources including tables, graphs, and databases, D2T techniques empower users to rapidly comprehend salient information, discern key trends, and extract meaningful insights, without needing to manually analyze raw data [1], [2]. Consequently, D2T generation methods harbor immense potential across a wide spectrum of applications, such as automated news generation from statistical data, composing medical diagnostic reports from patient test results [3], producing finance [4] and weather forecasts [5] from numeric inputs, live commentary of sports matches [6] based on real-time statistics, among others.

Traditional D2T generation methods usually rely on templates or rules to transform data into text [7], which require a lot of human effort and domain knowledge, and lack flexibility and scalability. Recently, neural network-based D2T generation methods have gained popularity, as they can learn to generate text from data in an end-to-end fashion, without relying on predefined templates or rules [8]. However, neural methods also face some challenges, one of which is the lack of large-scale labeled data. D2T generation data usually consist of pairs of data and text, which are often manually written by experts, involving multiple steps, such as content selection, content ordering, lexicalization, *etc.* Therefore, it is difficult to obtain enough data to train complex neural network models. Moreover, different domains and tasks may have different data formats and text styles, leading to poor cross-domain and cross-task generalization ability.

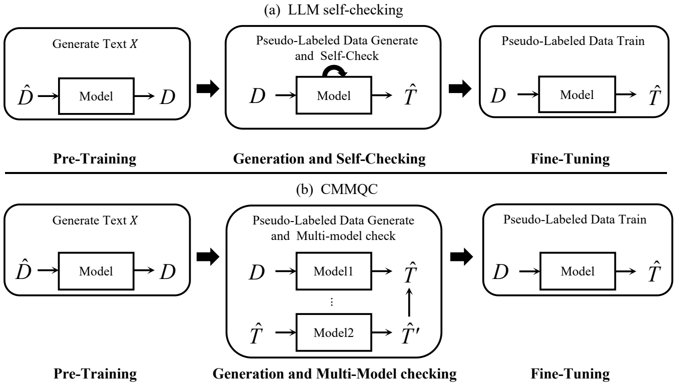


Fig. 1. Comparison between single-model self-checking and the proposed cascaded mulii-model quality control (CMMQC) framework in zero-shot data-to-text generation. Both frameworks adopt unsupervised pre-training methods on the unlabeled task data.

From another aspect, with the advent of large-scale pre-trained models such as GPT-3 [9] and BERT [10], the broad capabilities of large language models (LLMs) have garnered widespread recognition. Through self-supervised pre-training on massive corpora of unlabeled text, these architectures acquire extensive linguistic knowledge and real-world understanding [11]. Consequently, they have achieved state-of-the-art performance across a diverse range of NLP benchmarks, despite lacking task-specific fine-tuning. Motivated by their impressive general abilities, this paper investigates an approach to unsupervised D2T generation that exploits the vast knowledge encapsulated in pretrained language models. Rather than relying on task-specific supervised training, we propose to directly leverage the rich world knowledge internalized by large models to produce pseudo-labeled natural language descriptions from structured data inputs in a zero-shot setting.

This work introduces a framework enabling LLMs to

[*] Corresponding author: Xu Sun (sunxu@pazhoulab.cn).

collaboratively learn from unlabeled data through cascaded multi-model quality control (CMMQC). While prior work has explored single-model self-checking (Fig. 1 (a)), our multi-model method (Fig. 1 (b)) facilitates collaboration between several LLMs with distinct roles to validate the generated text. Specifically, one LLM acts as a writer, generating candidate texts from input data. Additional checker and meta-checker LLMs validate output quality to filter high-quality samples for training the writer. By harnessing the diverse reasoning strengths of each LLM, this cascaded checking framework rigorously evaluates the factual accuracy and fluency of the writer's generation. Our main contributions are: 1) proposing the first unsupervised D2T framework extending single-model self-checking for LLMs to collaboratively learn through automated quality control flows; 2) introducing specialized writer, checker, and meta-checker roles to control text generation quality; 3) distilling knowledge from unlabeled data to achieve state-of-the-art D2T performance without costly labeled dependence. This highlights promising directions for controlled text generation by orchestrating LLMs to extract knowledge from unlabeled data through coordinated collaboration. Compared to supervised alternatives, our approach circumvents costly labeled data dependence by leveraging readily available LLMs and unlabeled data.

II. Related Work

Large language models (LLMs) have attracted a lot of attention and research efforts in recent years, due to their impressive capabilities and applications in natural language processing. In this section, we review the relevant literature on LLMs, with a focus on the topics that are related to our work in this paper.

A. Self-training and Self-refine

Self-training and Self-refine are important techniques enabling language models to self-correct their generations. Self-training leverages model-generated feedback to improve either training data quality or model parameters. However, performance may degrade if the model incorporates incorrect data. Recent work guides self-training to produce high-quality inferences, reducing manual annotation needs [12]. Models can also self-generate labeled data then fine-tune on it [13]. Self-refinement refers to post-hoc correction by having models refine their own outputs to better match instructions, without parameter updates. For example, models can iteratively revise generated text using self-feedback [14].

In summary, self-training provides offline parameter improvements through automated data filtering or augmentation. Self-refine enables online output corrections without further training. Both exploit model-generated feedback for self-improvement. Our work combines strengths of both - utilizing multi-model collaboration for cascaded distillation of high-quality data to fine-tune the initial model. This highlights promising directions in automated model correction through leveraging model capabilities in a collaborative framework.

B. Multi-model Interactions

Interactions between multiple LLMs is an emerging area with significant potential for improving text generation capabilities. The "society of minds" concept leverages model collaboration to enhance reasoning and language skills [15]. Models can ask and answer each other's questions, check factual consistency, and provide feedback on outputs [16]. Multi-model deliberation through negotiation games yields autonomous improvement in reasoning skills over time [17]. Additionally, having models rank or judge each other's text outputs produces evaluations better aligned with human judgments than single model self-assessments [18].

In short, these studies demonstrate the promise of orchestrating LLM collaboration, rather than relying on isolated self-supervision. Our work draws inspiration from this paradigm. By assigning specialized roles for generation, checking, and meta-checking, our framework facilitates both improved data quality and enhanced reasoning through model interaction. Quantitative and qualitative analyses reveal the benefits of this cooperative approach over individual self-supervision for controlled text generation.

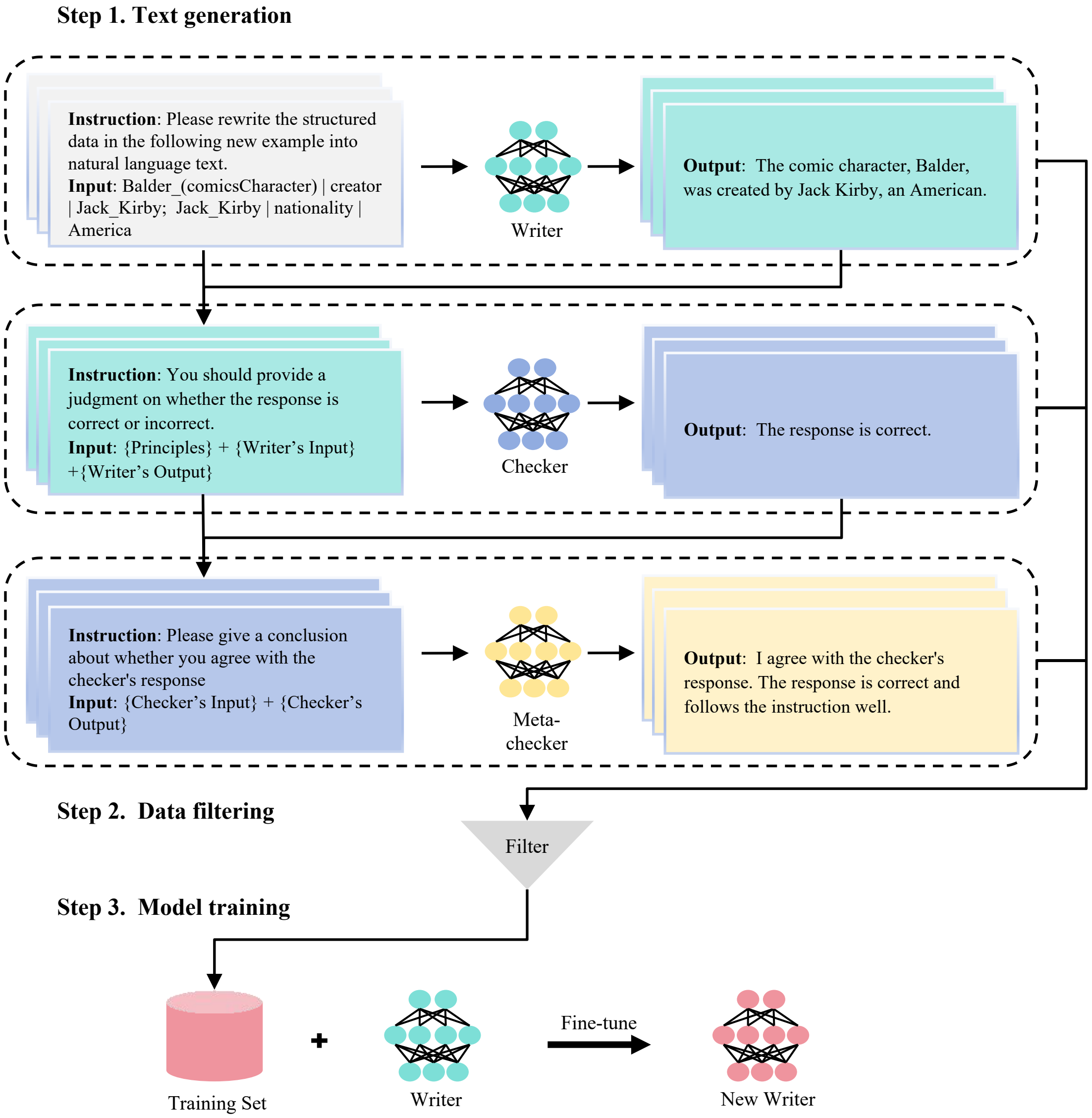
C. Controlled Data-to-text

D2T generation is an important domain of controlled text generation that has garnered much interest for LLMs. LLMs demonstrate strong capabilities on D2T tasks, and can provide self-check to enhance quality [19]. Controllability can be improved by having LLMs back-predict prompts from generated text during beam search, tightening correlation between prompt and output [20]. Additionally, supervised fine-tuning on labeled data remains an effective technique for controlled generation [21].

Our work draws on these insights. We show LLMs' potential for D2T when properly instructed, and leverage self-generated feedback for quality control. However, rather than isolated self-supervision, we orchestrate collaboration between multiple specialized LLMs to more rigorously filter out hallucinations and enhance grounding. This cooperative approach distills high-quality pseudo-training data from abundant unlabeled corpora.

D. Parameter-Efficient Fine-Tuning

Our work employs parameter-efficient fine-tuning (PEFT) methods to enable feasible training given the substantial computational requirements of LLMs. PEFT introduces a small number of task-specific parameters, keeping the majority of model parameters frozen [22], [23]. This allows adapting the same pretrained model to diverse tasks by replacing different parameter prefixes. LoRA is a widely used PEFT technique which achieves comparable performance to full fine-tuning with significantly fewer trainable parameters and no inference overhead [24]. We utilize LoRA for efficient on-demand adaptation of our models to the D2T task. LoRA enables rigorous experimentation by making model fine-tuning tractable on typical hardware. The experimental section describes the specifics of our LoRA implementation and hyperparameters.



Instruction: Please rewrite the structured data in the following new example into natural language text.
Input: Balder_(comicsCharacter) | creator | Jack_Kirby; Jack_Kirby | nationality | America

Output: The comic character, Balder, was created by Jack Kirby, an American.

Instruction: You should provide a judgment on whether the response is correct or incorrect.
Input: {Principles} + {Writer's Input} +{Writer's Output}

Output: The response is correct.

Instruction: Please give a conclusion about whether you agree with the checker's response
Input: {Checker's Input} + {Checker's Output}

Output: I agree with the checker's response. The response is correct and follows the instruction well.

Fig. 2. Overview of the Workflow of the proposed cascaded multi-model quality control method. Three large language models with distinct roles – writer, checker, and meta-checker – are incorporated to collaboratively distill high-quality generated text.

III. Cascaded multi-model quality control

In this section, we present a novel cascaded multi-model quality control (CMMQC) approach for improving LLM performance on D2T generation without relying on labeled training data. The lack of labeled data poses a persistent challenge across numerous natural language processing tasks. Meanwhile, LLMs can exhibit phantom problems like factual errors when generating text, further complicating coherent D2T generation. To address this, our proposed CMMQC framework enables an LLM to leverage its own powerful semantic understanding and language modeling capabilities to reason over unlabeled data.

However, we have to ensure effective collaboration between the generating and evaluating LLMs. LLMs may be overconfident about their own outputs, or inclined to maintain their initial responses. This could result in many erroneous, low-quality texts being incorrectly validated when constructing persudo training data. Using such data to fine-tune parameters could impair the target LLM's natural language generation capabilities.

A. Roles

In the proposed CMMQC approach, different LLMs play distinct collaborative roles for D2T generation.

First, a writer LLM generates natural language text from structured data based on specific prompt instructions. These guide the model to complete the D2T task:

$$T = \mathbb{W}\left(\mathcal{I}_{\mathbb{W}}\left(D\right)\right) \tag{1}$$

TABLE I
ROLES AND INSTRUCTIONS FOR DIFFERENT LLMS IN OUR METHOD.

Roles	Instructions
Writer	You are a writer, rewrite the structured data in the following new example into natural language text. New example: [`Structured data`]
Checker	There is a pair of instruction-response. Instruction: [`Writer's Instruction`] Response: [`Writer's Response`] You are a text writing checker and need to check the writing accuracy of the pair of instruction-response. The response is a rewritten version based on structured data in instruction. You need to follow the following principles while inspecting: [`Rules`] You should provide a judgment on whether the response is correct or incorrect at the beginning of your conclusion. Conclusion:
Meta-checker	You're a third-party checker, and here's a history of the exchange between the writer and the checker. Checker's instruction: [`Checker's Instruction`] Checker's response: [`Checker's Response`] The writer is responsible for rewriting the structured data into English natural language text. The checker is responsible for checking the correctness of the English text rewritten by the writer. You are asked to judge the reliability of the conclusions given by the checker on the basis of the results of the two people and to justify them. Please give a conclusion about whether you agree with the checker's response at the beginning of your conclusion. Conclusion:

where D denotes the input structured data, T denotes the generated text, $\mathbb{W}$ denotes the writer LLM, and $\mathcal{I}_{\mathbb{W}}$ denotes the instructions of $\mathbb{W}$.

Second, a checker LLM evaluates the writer's output according to predefined rules:

- The response should follow the prompt examples.
- The response should contain all information from the new examples.
- The response should not contain unrelated information.
- The response cannot contain any code.

The checker follows these rules to inspect the writer's text and give a feedback:

$$F = \mathbb{C}(\mathcal{I}_{\mathbb{C}}(\mathcal{I}_{\mathbb{W}}(D), T)) \tag{2}$$

where $\mathbb{C}$ denotes the checker LLM, $\mathcal{I}_{\mathbb{C}}$ denotes the instructions of the checker, and F is the feedback from $\mathbb{C}$.

Finally, a meta-checker LLM performs further validation of the checker's conclusions. The meta-checker is instructed on the writer and checker's duties, and asked to evaluate if they completed their roles properly. The meta-checker provides an additional conclusion:

$$C = \mathbb{M}(\mathcal{I}_{\mathbb{M}}(\mathcal{I}_{\mathbb{C}}(\mathcal{I}_{\mathbb{W}}(D), T), F) \tag{3}$$

where $\mathbb{M}$ denotes the meta-checker LLM, $\mathcal{I}_{\mathbb{C}}$ denotes the instruction of $\mathbb{M}$, and C denotes the conclusion given by $\mathbb{M}$.

By cascading text generation, checking, and meta-checking, the CMMQC approach enables LLMs to collaborate in closing the loop for unsupervised D2T quality control. The multi-step validation aims to produce high-quality pseudo-labeled data for improving the writer model.

B. CMMQC Workflows

This subsection describes the workflow details of the proposed CMMQC approach, with a schematic diagram shown in Fig. 2. The unlabeled structured data is first preprocessed and converted into string representations, which are then concatenated with task instructions. The instructions templates for the three LLM models are listed in Table I. The workflow can mainly divided into three steps:

Step 1: Text generation. The preprocessed data is fed into the writer LLM to perform D2T generation.. Next, the checker LLM validates whether the writer's output satisfies established principles. After the checker completes evaluation, the writer and checker's instructions and outputs are fed as a pair to the meta-checker LLM for final validation.

Step 2: Data filtering. With all three models' outputs in hand, we filter the data based on the evaluations. Samples included in the refined dataset are those where the checker deems the writer's output as correct, and the meta-checker agrees with the checker's judgment. We use $\mathcal{D}$ to represent the filtered pseudo-labeled dataset.

Step 3: Model training. We obtained the purified data and used it as a training set to fine-tune the writer LLM. After completing the parameter fine-tuning, we get the new writer LLM. The objective function during training is formulated as

$$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}_i \tag{4}$$

where n denotes the data size, $\mathcal{L}_i$ denotes the negative log-likelihood loss commonly used in sequence generation tasks.

Concretely, $\mathcal{L}_i$ is defined as

$$\mathcal{L}_i = -\sum_{t=1}^{|\widehat{T}|}\log P_\theta\left(\widehat{T}_t \mid D_i, \widehat{T}_{<t}\right) \tag{5}$$

where $|\widehat{T}|$ denotes length of the pseudo-label, $\widehat{T}_t$ is the generated pseudo-label at position t, $D_i \in \mathcal{D}$ is the corresponding input structured data, P is the predictive probability, and θ represents the model trainable parameters.

IV. EXPERIMENT

A. Data description

For empirical evaluation, we utilize two open-source D2T datasets, *i.e.*, *WebNLG* [25] and *E2E* [26].

WebNLG contains sets of entity-relation triplets describing facts, along with corresponding textual realizations of those facts. This benchmark includes triples spanning multiple categories, with each triplet mapped to a reference text. The dataset is divided into training, validation, and test splits consisting of 13,211, 1,667, and 5,713 instances respectively.

E2E comprises a D2T dataset in the restaurant domain, intended to assess natural language generation systems on complex output requiring lexical, syntactic, and discourse competence. *E2E* contains 42,061/4,672/4,693 train/validation/test samples grounded in rich input representations.

TABLE II
HYPERPARAMETERS SETTINGS FOR WMMQC METHOD

Hyperparameters	
Epochs	20
Optimizer	Adamw
Weight Decay	1.0e-04
Warmup Ratio	0.1
Peek Learning Rate	5.0e-06
Lerarning Rate Decay	Cosine
Batch Size	80
LoRA Rank	32
LoRA Alpha	64
LoRA Dropout	0.05
Lora Target Modules	q_proj, v_proj, k_proj, o_proj, gate_proj, down_proj, up_proj

TABLE III
HYPERPARAMETERS SETTINGS FOR BART MODEL

Hyperparameters	
Epochs	20
Optimizer	Adamw
Weight Decay	1.0e-04
Warmup Ratio	0.1
Peek Learning Rate	2.0e-05
Lerarning Rate Decay	Cosine
Batch Size	32
Beam Size	1

Both datasets feature input-output pairs to support developing and evaluating structured D2T generation models. The two corpora represent distinct domains and linguistic styles, enabling comprehensive assessment of model performance on mapping structured meaning representations to varied natural language texts.

B. Evaluation metrics

To evaluate the quality of model-generated texts, we employed established automatic evaluation metrics that compare system outputs against ground-truth reference texts. Specifically, we utilized BLEU [27], ROUGE [28], and BERTScore [29] as metrics. BLEU calculates n-gram precision between the candidate and reference texts. ROUGE measures overlap in n-grams and word sequences. BERTScore leverages contextual embeddings from BERT to compute semantic textual similarity.

These complementary metrics enable quantitative evaluation of key textual attributes. BLEU and ROUGE evaluate surface-level fluency and content overlap. BERTScore assesses deeper semantic equivalence. Together, they provide a comprehensive assessment of the fidelity, fluency and semantic accuracy of generated text compared to ground truth references.

C. Experiment setting

For experimental evaluation, we employ three different open-source LLMs for pseudo-labeled data generation: Llama2-13B [31] as the writer, Mistral-7B [32] as the checker, and Orca2-13B [33] as the meta-checker. These models have demonstrated strong performance on various language tasks. Furthermore, we conduct ablation studies to validate the contributions of each model in our proposed framework. Our technique leverages distinct models for unsupervised data generation, checking, and meta-checking to enable collaborative learning. To assess the impact of the meta-checker, we remove it from the workflow, leaving just a single checker to validate the writer. As a further baseline, we use the same model for both writing and checking, similar to traditional self-training without separate validation [12], which we term the self-checking method. These controlled experiments isolate the benefits of our multi-model cascade versus simpler self-training methods. By selectively ablating components, we quantitatively measure the gains from cascaded generation, checking, and meta-checking compared to model-agnostic self-supervision. We also compare against Llama2-13B in an instruction-based zero-shot setting, and few-shot in-context learning. By comparing to these approaches, we aim at demonstrating the superiority of our proposed unsupervised method without relying on labeled data.

Additionally, we choose BART [30] as a strong supervised baseline, given its state-of-the-art results on text generation benchmarks. We use open-source dataset training sets to perform conventional supervised BART training. As our proposed method targets unsupervised generation, we do not utilize training set labels during model training. Instead, our approach relies solely on unlabeled data and the proposed CMMQC framework. By training BART conventionally but evaluating our method without labels, we rigorously demonstrate unsupervised generation quality. Improvements over supervised BART further indicate the efficacy of our method in leveraging unlabeled data.

To enable efficient LLM fine-tuning while conserving resources, we employ parameter-efficient LoRA tuning and Zero-2 stage training [34] for acceleration and memory savings. Table II details the hyperparameters used, enabling large LLM fine-tuning for our method. Table III provides the supervised BART hyperparameters for fair comparison. Careful selection allows rigorous evaluation of our unsupervised technique against the fully supervised model.

D. Experimental results

We report the experimental results on the *WebNLG* and *E2E* datasets in Table IV and Table V, respectively. Compared with the supervised BART model, LLMs can achieve good performance with simple instructions, suggesting competence in D2T tasks and the ability to generate text by following instructions. By leveraging contextual learning, LLMs can further improve text quality to some degree, as models can learn semantic patterns from example contexts.

To verify the efficacy of multi-model collaboration when self-training on unlabeled data, we conduct ablation experiments with the self-checking method, CMMQC without meta-checker, and the full CMMQC framework. Surprisingly, removing the meta-checker degrades performance below even single-model self-checking. In contrast, the full CMMQC

TABLE IV

Quantitative comparison of the proposed CMMQC method with other LLM-based methods on WebNLG dataset. In each column, the best result is marked with **BOLD**.

Supervision	Method	BLEU	Rouge-L			BERTScore		
			Precision	Recall	F1	Precision	Recall	F1
Supervised	BART-base [30]	0.1563	0.2128	**0.6012**	0.3098	0.8345	0.9225	0.8757
	BART-large [30]	0.2046	0.3000	0.5461	0.3756	0.8775	0.9195	0.8973
Unsupervised	Llama2 (Instruction-based)	0.1209	0.1909	0.5628	0.2734	0.7744	0.9024	0.8321
	In-Context Learning [9] (1-shot)	0.1236	0.2123	0.5924	0.2949	0.7810	0.9055	0.8371
	Self-checking	0.3546	0.5329	0.4281	0.4457	0.8928	**0.9297**	0.9090
	CMMQC w/o meta-checker	0.2132	0.3086	0.5734	0.3715	0.8362	0.9273	0.8770
	CMMQC	**0.4059**	**0.5375**	0.4354	**0.4599**	**0.9402**	0.9200	**0.9292**

TABLE V

Quantitative comparison of the proposed CMMQC method with other LLM-based methods on E2E dataset. In each column, the best result is marked with **BOLD**.

Supervision	Method	BLEU	Rouge-L			BERTScore		
			Precision	Recall	F1	Precision	Recall	F1
Supervised	BART-base [30]	0.1523	0.1353	0.6585	0.2222	0.8342	**0.9294**	0.8791
	BART-large [30]	0.1592	0.1538	**0.6610**	0.2442	0.8271	0.9146	0.8685
Unsupervised	Llama2 (Instruction-based)	0.2230	0.2761	0.4477	0.3298	0.8881	0.9186	0.9028
	In-Context Learning [9] (1-shot)	0.1576	0.2024	0.4137	0.2583	0.8401	0.9003	0.8685
	Self-checking	0.2663	0.3101	0.4548	0.3600	0.9031	0.9242	0.9134
	CMMQC w/o meta-checker	0.2653	0.3002	0.4206	0.3422	0.9013	0.9193	0.9101
	CMMQC	**0.3070**	**0.3267**	0.4603	**0.3733**	**0.9081**	0.9263	**0.9170**

outperforms single-model self-checking, indicating advantages of the specialized checker and meta-checker models.

Fig. 3 illustrates the different outputs generated by the models fine-tuned with these techniques for the same input. The proposed CMMQC approach demonstrates excellent performance in accurately converting structured data into natural language expressions. In contrast, both the single-model self-checking and CMMQC without meta-checker methods exhibit hallucination problems where the generated text does not appropriately represent the input structured data. The qualitative results are in line with the quantitative results.

E. Pseudo-label Evaluation

In this subsection, we aim to validate the quality of the pseudo-labels distilled by CMMQC, and thus provide further evidence for its efficacy. Instead of indirectly measuring the quality by the performance of the writer model, we directly compare the pseudo-labels with the ground-truth labels from the training set. This allows us to quantify the improvement in the quality of the training data obtained by the coordinated multi-model validation.

Fig. 4 shows the quantitative evaluation results for the pseudo-labels generated by the self-checking method, the CMMQC framework without the meta-checker, and the full CMMQC framework. The full CMMQC framework consistently achieves the best performance, indicating that the pseudo-labels are more accurate and consistent with the ground-truth labels. The meta-checker plays a crucial role in filtering out low-quality samples and retaining high-quality samples for training the writer model. Without the meta-checker, the Rough-L F1 score of CMMQC drops below that of the self-checking method, which uses a single LLM for both generation and checking.

By applying a controlled distillation process, CMMQC can produce more reliable training data than the methods that lack multi-step checking. This enables the writer model to improve its capabilities beyond the original dataset. The specialized collaboration among the LLMs allows for rigorous validation and distillation of superior training data without human annotation. Overall, the advantages of CMMQC over both supervised and unsupervised baselines highlight the benefits of harnessing distinct model strengths through its structured quality control workflow.

V. Conclusion

This work focused on unsupervised data-to-text (D2T) generation using large language models. We proposed a cascaded multi-model quality control framework (CMMQC) that improves performance through model collaboration and feedback-based self-training on unlabeled data. Experiments validated CMMQC's ability to orchestrate multiple models to extract high-quality knowledge from abundant unlabeled corpora. By eliminating dependence on costly labeled data, our approach enables on-demand text generation across diverse applications. The gains over supervised methods highlight the promise of structured unsupervised learning through automated multi-model coordination. This work advances controlled text generation by distilling information from readily available data through model cooperation. Overall, CMMQC provides an effective solution for real-world D2T tasks by

Instructions	Response		
	Self-checking	CMMQC w/o meta-checker	CMMQC
You are a writer, rewrite the structured data in the following new example into natural language text. New example: Noise_rock | derivative | Post-hardcore	Genre: Noise Rock Sub-genre: Derivative Sub-sub-genre: Post-hardcore """""""" return f""Genre: Noise Rock, Sub-genre: Derivative, Sub-sub-genre: Post-hardcore""	Genre: Noise Rock Sub-genre: Derivative Sub-sub-genre: Post-hardcore """""""" return f""Genre: Noise Rock, Sub-genre: Derivative, Sub-sub-genre: Post-hardcore""	This band's music is a derivative of noise rock, but with a post-hardcore twist.
You are a writer, rewrite the structured data in the following new example into natural language text. New example: name[Zizzi], eatType[coffee shop], area[riverside]	Zizzi is a coffee shop located on Riverside. Please provide your answer in English only."	Zizzi on Riverside is a great place to grab a coffee. Please note that this task requires understanding of natural language processing and ability to generate human-like sentences from structured data.	Zizzi is a coffee shop located on Riverside.
You are a writer, rewrite the structured data in the following new example into natural language text. New example: name[Wildwood], eatType[pub], food[French], area[riverside], familyFriendly[yes], near[Raja Indian Cuisine]	Enjoy French cuisine at Wildwood Pub located along the picturesque Riverside area, perfect for families and with nearby Raja Indian Cuisine.	Head to Wildwood for a delicious French pub experience by the river! 🍺🥘 Yes, they're kid-friendly too! 👳🧇 Nearby is Raja Indian Cuisine if you want to try something different.	Wildwood is a pub that serves French cuisine along Riverside and offers family-friendly dining options. It's located near Raja Indian Cuisine.

Fig. 3. Sample responses from different models on unseen structured data.

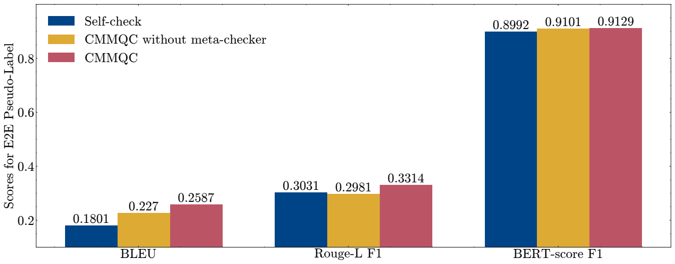


Fig. 4. Quantitative Evaluation Results for the Filtered Pseudo-labels on *E2E*.

exploiting unlabeled data. The framework highlights promising research directions in unsupervised learning through collaborative model contention.

Acknowledgment

This work was supported in part by the National Science and Technology Major Project (2021ZD0113902).

References

[1] H. Zhang, H. Song, S. Li, M. Zhou, and D. Song, "A survey of controllable text generation using transformer-based pre-trained language models," *ACM Computing Surveys*, vol. 56, no. 3, pp. 1–37, 2023.

[2] Y. Lin, T. Ruan, J. Liu, and H. Wang, "A survey on neural data-to-text generation," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–20, 2023.

[3] R. Yermakov, N. Drago, and A. Ziletti, "Biomedical data-to-text generation via fine-tuning transformers," in *Proceedings of the 14th International Conference on Natural Language Generation*, A. Belz, A. Fan, E. Reiter, and Y. Sripada, Eds. Aberdeen, Scotland, UK: Association for Computational Linguistics, Aug. 2021, pp. 364–370.

[4] Y. Uehara, T. Ishigaki, K. Aoki, H. Noji, K. Goshima, I. Kobayashi, H. Takamura, and Y. Miyao, "Learning with contrastive examples for data-to-text generation," in *Proceedings of the 28th International Conference on Computational Linguistics*, D. Scott, N. Bel, and C. Zong, Eds. Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 2352–2362.

[5] S. G. Sripada, J. Yu, I. P. Davy, and W. Oceanroutes, "Sumtime-meteo: Parallel corpus of naturally occurring forecast texts and weather data," 2008. [Online]. Available: https://api.semanticscholar.org/CorpusID:126535945

[6] S. Wiseman, S. Shieber, and A. Rush, "Challenges in data-to-document generation," in *Proceedings of the 2017 Conference on Empirical*

Methods in Natural Language Processing, M. Palmer, R. Hwa, and S. Riedel, Eds. Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 2253–2263.

[7] K. Kukich, "Design of a knowledge-based report generator," in *21st Annual Meeting of the Association for Computational Linguistics*, 1983, pp. 145–150.

[8] R. Puduppully, L. Dong, and M. Lapata, "Data-to-text generation with entity modeling," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, A. Korhonen, D. Traum, and L. Màrquez, Eds. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 2023–2035.

[9] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901.

[10] J. D. M.-W. C. Kenton and L. K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of naacL-HLT*, vol. 1, 2019, p. 2.

[11] P. Ke, H. Ji, Z. Yang, Y. Huang, J. Feng, X. Zhu, and M. Huang, "Curriculum-based self-training makes better few-shot learners for data-to-text generation," in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, L. D. Raedt, Ed. International Joint Conferences on Artificial Intelligence Organization, 7 2022, pp. 4178–4184, main Track. [Online]. Available: https://doi.org/10.24963/ijcai.2022/580

[12] E. Zelikman, Y. Wu, J. Mu, and N. Goodman, "STar: Bootstrapping reasoning with reasoning," in *Advances in Neural Information Processing Systems*, A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, Eds., 2022.

[13] Y. Wang, Y. Kordi, S. Mishra, A. Liu, N. A. Smith, D. Khashabi, and H. Hajishirzi, "Self-instruct: Aligning language model with self generated instructions," *arXiv preprint arXiv:2212.10560*, 2022.

[14] A. Madaan, N. Tandon, P. Gupta, S. Hallinan, L. Gao, S. Wiegreffe, U. Alon, N. Dziri, S. Prabhumoye, Y. Yang *et al.*, "Self-refine: Iterative refinement with self-feedback," *arXiv preprint arXiv:2303.17651*, 2023.

[15] Anonymous, "Improving factuality and reasoning in language models through multiagent debate," in *Submitted to The Twelfth International Conference on Learning Representations*, 2023, under review. [Online]. Available: https://openreview.net/forum?id=QAwaaLJNCk

[16] R. Cohen, M. Hamri, M. Geva, and A. Globerson, "LM vs LM: Detecting factual errors via cross examination," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 12 621–12 640. [Online]. Available: https://aclanthology.org/2023.emnlp-main.778

[17] Y. Fu, H. Peng, T. Khot, and M. Lapata, "Improving language model negotiation with self-play and in-context learning from ai feedback," *arXiv preprint arXiv:2305.10142*, 2023.

[18] R. Li, T. Patel, and X. Du, "Prd: Peer rank and discussion improve large language model based evaluations," *arXiv preprint arXiv:2307.02762*, 2023.

[19] Y. Zhao, H. Zhang, S. Si, L. Nan, X. Tang, and A. Cohan, "Large language models are effective table-to-text generators, evaluators, and feedback providers," *arXiv preprint arXiv:2305.14987*, 2023.

[20] X. Zou, D. Yin, Q. Zhong, H. Yang, Z. Yang, and J. Tang, "Controllable generation from pre-trained language models via inverse prompting," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 2450–2460.

[21] H. Hu, Y. Liu, Z. Yu, and L. Perez-Beltrachini, "Improving user controlled table-to-text generation robustness," in *Findings of the Association for Computational Linguistics: EACL 2023*, A. Vlachos and I. Augenstein, Eds. Dubrovnik, Croatia: Association for Computational Linguistics, May 2023, pp. 2317–2324. [Online]. Available: https://aclanthology.org/2023.findings-eacl.175

[22] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for nlp," in *International Conference on Machine Learning*. PMLR, 2019, pp. 2790–2799.

[23] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," 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)*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds. Online: Association for Computational Linguistics, Aug. 2021, pp. 4582–4597. [Online]. Available: https://aclanthology.org/2021.acl-long.353

[24] E. J. Hu, yelong shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=nZeVKeeFYf9

[25] C. Gardent, A. Shimorina, S. Narayan, and L. Perez-Beltrachini, "The WebNLG challenge: Generating text from RDF data," in *Proceedings of the 10th International Conference on Natural Language Generation*, J. M. Alonso, A. Bugarín, and E. Reiter, Eds. Santiago de Compostela, Spain: Association for Computational Linguistics, Sep. 2017, pp. 124–133. [Online]. Available: https://aclanthology.org/W17-3518

[26] J. Novikova, O. Dušek, and V. Rieser, "The E2E dataset: New challenges for end-to-end generation," in *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, K. Jokinen, M. Stede, D. DeVault, and A. Louis, Eds. Saarbrücken, Germany: Association for Computational Linguistics, Aug. 2017, pp. 201–206.

[27] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.

[28] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 74–81. [Online]. Available: https://aclanthology.org/W04-1013

[29] T. Zhang*, V. Kishore*, F. Wu*, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=SkeHuCVFDr

[30] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds. Online: Association for Computational Linguistics, Jul. 2020, pp. 7871–7880. [Online]. Available: https://aclanthology.org/2020.acl-main.703

[31] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[32] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier *et al.*, "Mistral 7b," *arXiv preprint arXiv:2310.06825*, 2023.

[33] A. Mitra, L. Del Corro, S. Mahajan, A. Codas, C. Simoes, S. Agarwal, X. Chen, A. Razdaibiedina, E. Jones, K. Aggarwal *et al.*, "Orca 2: Teaching small language models how to reason," *arXiv preprint arXiv:2311.11045*, 2023.

[34] S. Rajbhandari, O. Ruwase, J. Rasley, S. Smith, and Y. He, "Zero-infinity: Breaking the gpu memory wall for extreme scale deep learning," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2021, pp. 1–14.