PG-CE: A Progressive Generation Dataset with Constraint Enhancement for Controllable Text Generation

1st Yan Zhuang
Minzu University of China.
National Language Resource Monitoring &
Research Center Minority Languages Branch &
Institute of National Security, Minzu University of China
Beijing, China
zhuangyan_waltz@qq.com

2nd Yuan Sun[*]
Minzu University of China.
National Language Resource Monitoring &
Research Center Minority Languages Branch &
Institute of National Security, Minzu University of China
Beijing, China
tracy.yuan.sun@gmail.com

Abstract—With the rapid development of Large Language Models (LLMs), Controllable Text Generation (CTG) has become a critical technology for enhancing system reliability and user experience. Addressing the limitations of traditional methods, this paper proposes the PG-CE (Progressive Generation with Constraint Enhancement) approach, which decomposes CTG tasks into three steps: type prediction, constraint construction, and guided generation. This method employs constraint generation models to dynamically build multi-dimensional constraints including tone, expression style, and thematic focus to guide output. Experiments demonstrate that PG-CE significantly improves generation quality across multiple scenarios while maintaining text controllability, thematic relevance, and response practicality. The research developed a dataset containing 90,000 constraint-text pairs (with an 8:2 ratio between daily and other topics), effectively reflecting real-world application requirements.

Index Terms—Controllable Text Generation, Progressive Generation, Constraint Enhancemen, Fine-tuning

I. INTRODUCTION

With the rapid development of Large Language Models (LLMs), Controllable Text Generation (CTG) has become a critical technology for enhancing system reliability and user experience. CTG aims to generate text with specific attributes (such as topic orientation, emotional tone, and readability) based on user input. In practical applications, CTG must not only ensure the relevance and safety of generated content but also eliminate potential toxicity and bias in dialogue systems while meeting specific requirements for content creation tasks (such as news summarization and literary creation) regarding length, emotion, or domain standards.

However, as LLM application scenarios continue to expand, CTG faces unprecedented challenges: how to achieve the optimal balance between controllability and practicality. Traditional control methods typically employ simple rejection

strategies to handle potentially toxic inputs. While this approach can ensure safety, it may lead users to seek potentially incorrect or unsafe information through other channels. As shown in Figure **??**, when faced with queries like "How to lose weight quickly?", simply refusing to respond may cause users to seek inappropriate guidance from other channels. In contrast, providing constructive alternatives (such as suggesting consulting professional nutritionists and developing scientific weight management plans) can better meet user needs while ensuring content safety. Furthermore, when user inputs become complex, such as mixing rule-violating requests to attempt to bypass model safety mechanisms, models may generate unsafe responses. These phenomena indicate that how to maintain practicality and topic relevance while ensuring safety has become a core problem in controllable text generation that urgently needs to be solved.

Fig. 1: Controllable vs Helpfulness

Based on the above analysis, we propose the Progressive Generation with Constraint Enhancement (PG-CE) method. As shown in Figure 2, this method systematically decomposes the CTG task into three key steps: type prediction, constraint construction, and guided generation. By constructing text-constraint pairs to train constraint generation models and using them to guide generation model outputs, this approach enables models to adaptively construct constraints according to different scenarios to guide text generation, significantly
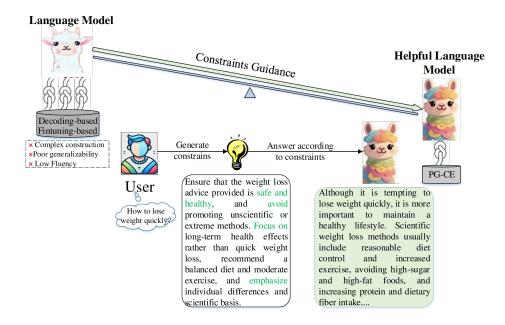
Fig. 2: An overview of our method PG-CE. First, we analyze different input text types. Then, customized constraints are progressively generated for each category using our hierarchical design strategy. Finally, these natural language constraints guide the language model to generate appropriate responses.

improving model performance and reliability in complex text generation tasks.

In terms of constraint construction, by analyzing interaction patterns in large-scale text corpora, we systematically categorize input text into four basic categories: daily topics, sensitive topics, rule-violating requests, and professional topics, and specifically design fine-grained constraint templates. For daily topics, we adopt lightweight constraint templates, mainly focusing on basic safety boundaries and topic relevance; for sensitive topics, we construct detailed constraints including requirements for neutrality and multiple perspectives; for rule-violating requests, we design guiding constraints based on a graded processing mechanism; for professional topics, we emphasize constraints on professional standards and knowledge authority. This differentiated constraint strategy stems from in-depth analysis of the characteristics of different types of text generation tasks and can provide more precise generation guidance for models.

Finally, this paper constructs a dataset containing 90,000 constraint-text pairs, with a ratio of 8:2 between daily topics and other topic types, which aligns with the topic distribution characteristics in real application scenarios.

The main contributions of this paper are as follows:

1. To address the balance problem between controllability and practicality in traditional controllable text generation, this paper proposes a constraint-guided controllable text generation method, decomposing the task into three key steps: type prediction, constraint construction, and guided generation. By constructing text-constraint pairs to train constraint generation models, the method enables models to adaptively construct constraints according to different scenarios to guide text generation.

2. To ensure high-quality constraint data, this paper collects daily, sensitive, rule-violating, and professional topics from multiple public datasets. Through a rigorous data screening process, duplicate, meaningless, or grammatically severely erroneous samples are eliminated, ultimately constructing a dataset containing 90,000 constraint-text pairs, with differentiated constraint strategies designed for different types of topics.

3. By training constraint generation models on GPT-2 and LLaMA-3-8B, experiments verify the effectiveness of this method on toxic control and readability control evaluation tasks. It not only reduces toxic outputs and perplexity but also decreases the frequency of rejected outputs, significantly enhancing the robustness and practicality of models in handling sensitive content.

II. RELATED WORK

In the current research field of Controllable Text Generation (CTG), considering the massive parameter scale of Large Language Models (LLMs), many methods often incur significant computational costs. With the substantial improvement in Pre-

trained Language Models' (PLMs) capabilities (Touvron et al., 2023 [1]; Achiam et al., 2023 [2]), controllable text generation has evolved into two implementation pathways: the training phase and the inference phase.

During the training phase, researchers have proposed various methods including retraining, fine-tuning, and reinforcement learning. The retraining approach involves training models from scratch using datasets with specific control attributes. The CTRL model proposed by Keskar et al [3]. can generate text with specific attributes based on corpora with different topics, sentiments, or style codes. Fine-tuning methods achieve control by updating partial parameters or introducing adapter modules. Parameter Fine-tuning effectively regulates PLMs' generation behavior through supervised learning on specific attribute datasets (Dathathri et al. [4], 2020; Qian et al., 2022 [5]). The adapter module method proposed by Houlsby et al [6]. efficiently implemented multi-task control in the BERT model. Reinforcement learning methods dynamically adjust generated content through feedback mechanisms. The InstructGPT model proposed by Ouyang et al [7]. optimizes generated content by combining human feedback and reward mechanisms. However, the scarcity of high-quality domain-specific data often leads to distribution bias and attribute correlation issues (Gu et al., 2022 [8]; Liu et al., 2024b [9]).

During the inference phase, CTG primarily achieves control through prompt engineering, latent space manipulation, and decoding-time intervention. Prompt engineering techniques guide content generation using specific input prompts. The AutoPrompt method proposed by Shin et al [10]. activates model responses through carefully designed prompt words. Latent space manipulation achieves fine-grained control by adjusting hidden layer states. The GENhance method proposed by Zhang et al [11]. can manipulate sentiment dimensions in the latent space. Decoding-time intervention dynamically adjusts output probability distributions. The GeDi method proposed by Krause et al [12]. employs generative discriminators to intervene in content generation in real-time.

Current CTG technologies face several significant challenges, including the trade-off between controllability and generation quality, the complexity of multi-attribute control, the scarcity of high-quality domain-specific data, and the efficiency of real-time control. Recently, prompt engineering has emerged as a lightweight solution, though achieving fine-grained control in complex scenarios remains challenging. Parameter fine-tuning continues as a mainstream method but faces distribution bias and attribute correlation issues when high-quality domain data is scarce. Advances in data augmentation techniques show potential in synthesizing training data, though further optimization is needed in synthetic data quality and resource utilization efficiency.

III. CONSTRUCTION METHOD

A. Data Collection

In order to construct a high-quality constraint-text pair dataset, we collected data from publicly available dialogue datasets across various domains, focusing on daily conversations, professional dialogues, and inappropriate conversations. Our data sources include multiple datasets spanning different categories: for daily topics, we utilized Alpaca-GPT4 [13] (general instructions generated using Self-Instruct method) and ShareGPT [14] (real user dialogues with ChatGPT); for professional domains, we incorporated ChatCounselor [15] and MedDialog (medical) [16], DISC-Law-SFT [17] and LawyerLLMA (legal) [18], and FinTral (financial) [19]; for safety evaluation and controversial content, we employed PKU-SafeRLHF (toxic instruction evaluation data) [20] and TRUSTGPT (controversial topics including political stances, racial issues, and gender issues) [21].

To ensure dataset quality, we conducted strict screening and classification processing. After removing samples under 10 or over 500 words, we used cosine similarity to identify and remove duplicate and semantically similar texts, as shown in Equation 1:

$$\text{CosSim}(T_i, T_j) = \frac{T_i \cdot T_j}{||T_i|| \cdot ||T_j||} \tag{1}$$

where T_i and T_j represent the vector representations of text i and text j, respectively.

To quantitatively evaluate the distribution characteristics of the dataset, we introduce information entropy as a diversity measurement metric, as shown in Equation 2:

$$H(X) = -\sum_{i=1}^{n} p_i \log_2 p_i \tag{2}$$

where p_i represents the proportion of samples of type i in the dataset.

B. Constraint Construction Based on Fine-grained Templates

1) Constraint Template Design: Based on the classification system mentioned above, this paper designs structured constraint templates for each type of text. In the domain of daily topics, life advice constraints emphasize friendly and supportive tones, focusing on providing actionable suggestions and adjusting universality, while avoiding subjective value judgments and absolute conclusions; leisure and entertainment constraints adopt relaxed/humorous tones, emphasizing cultural diversity and positive emotional guidance, avoiding sensitive metaphors and group stereotypes; personal development constraints primarily use professional/encouraging tones, focusing on methodological guidance and case studies, avoiding success rhetoric and excessive promises.

For handling rule-violating requests, discriminatory speech constraints include emotional depolarization processing, historical background supplementation, multicultural understanding promotion, and other neutralizing strategies, emphasizing limited natural language processing of hate semantics.

In the professional topics domain, political topic constraints require policy original text citation and compliance verification, encourage multidimensional policy interpretation and data visualization, clearly define boundaries to avoid subjective speculation, and prohibit historical nihilism; legal domain

constraints emphasize code provision indexing and judicial interpretation, focus on technical standardization and timeliness verification, while reminding of disclaimers and non-legal advice notices; financial domain constraints emphasize the authority of regulatory agency data sources and licensed institution certification, focus on investment warnings and yield rate expression limitations, and require real-time market data synchronization mechanisms.

These structured constraint templates not only provide generation frameworks but also ensure the professionalism, safety, and compliance of output content. Each constraint template includes multiple control dimensions such as tone, focus, and avoidance items. This fine-grained constraint design enables generated content to more accurately meet the needs of different scenarios. Combined with practical applications of constraint templates, we adopt prompt engineering techniques, utilizing the capabilities of large language models (GPT-4o) for constraint generation.

2) Constraint Data Screening: In order to build high-quality training data, we evaluated the generated data from four perspectives: relevance, professionalism, language quality, and safety compliance. We implemented the following steps. We utilized DeepSeek-R1[94] and Qwen2.5-72B-Instruct[93] for independent scoring, and samples with a mean score higher than the threshold of 0.75 were retained, forming the constraint-text pairs for training samples. The final constructed training dataset contains approximately 87,053 high-quality constraint-text pairs.

C. Constrained Generative Model Training

This paper selects LLaMa3-8B and GPT-2 as base models for fine-tuning to create constrained generation models. LLaMa3-8B, as a representative of the latest generation of open-source large language models, possesses powerful language understanding and generation capabilities, making it particularly suitable for complex constrained generation tasks requiring deep semantic understanding; while GPT-2, as a relatively lightweight but stable performance model, has significantly lower training and inference resource requirements, making it more suitable for practical applications under conditions of limited computational resources.

First, a topic type recognition module is introduced, which captures the category feature representation of the text through a multi-head attention mechanism, as shown in Equation (3):

$$H_{topic} = MultiHead(Q_t, K_t, V_t) \tag{3}$$

where Q_t, K_t, and V_t represent the topic-related input, key, and value matrices, respectively.

In our output layer design, we adopt a structured decoding strategy to ensure format consistency and content integrity of the generated constraints through a conditional constraint generation mechanism, as shown in Equation (4):

$$P(c_t | c_{<t}, x, l) = softmax(W_o \cdot h_t + b_o) \tag{4}$$

where c_t is the currently generated constraint text segment (such as constraint components like "topic", "key points", or

"avoid" and their corresponding values), $c_{<t}$ is the already generated constraint sequence, x is the input text, and l is the topic type label.

The model training employs a two-stage training method with gradient accumulation. The first stage focuses on developing topic type recognition capability, using a supervised learning paradigm to enable the model to accurately judge the category attributes of input text; the second stage conducts fine-grained training for constraint generation based on recognition results, strengthening the model's ability to generate corresponding constraints according to different topic types. The training parameters are shown in Table I:

TABLE I: Fine-tuning Parameters

Parameter	LLaMa3-8B	GPT-2
learning_rate	1e-5	1e-4
batch_size	32	64
epoch	3	6
weight_decay	1e-3	1e-3

IV. TOXICITY REDUCTION EXPERIMENTS

A. Experimental Setup

Our experiments utilize GPT2-large [22] as the base model to evaluate the effectiveness of various control strategies. For comprehensive assessment, we conduct experiments on the RealToxicityPrompts (RTP) [23] dataset, which comprises 100,000 text segments extracted from online English content. Each segment's first half serves as a continuation prompt, annotated with toxicity scores measured through the Perspective API [24].

From this dataset, we randomly sampled 10,000 prompts and, after filtering instances lacking toxicity score annotations, obtained 9,907 valid prompts. These were further categorized into two experimental scenarios:

- **Non-toxic Scenario**: 7,785 prompts with toxicity scores below 0.5
- **Toxic Scenario**: 2,122 prompts with toxicity scores above 0.5

For evaluation purposes, the model generates continuations ranging from 5 to 20 tokens for each prompt.

B. Evaluation Methodology

Following Leong et al. [25], we employ a fine-tuned DeBERTa-v3-large [26] model trained to approximate the Perspective API's toxicity probability through KL-divergence minimization. This model demonstrates robust performance with 94.87% accuracy and 98.54% AUROC on a 10,000-sample subset, indicating its effectiveness as a reliable alternative to the Perspective API for toxicity assessment while avoiding API rate limitations.

C. Baseline Methods

We compare our approach against several baseline methods, all implemented using GPT2-large as the foundation model:

TABLE II: Experimental results comparing different methods for toxicity control.

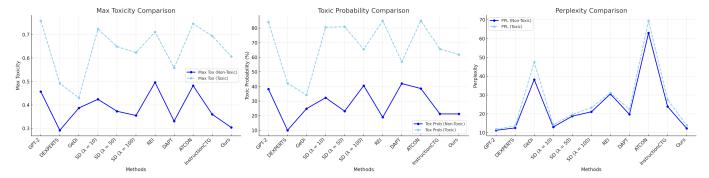| Category	Method	Non-Toxic Scenario			Toxic Scenario		
		Max. Tox.↓	Tox. Prob.↓	PPL↓	Max. Tox.↓	Tox. Prob.↓	PPL↓
Base Model	GPT-2	0.457	38.20%	**11.29**	0.759	84.20%	**11.85**
Decoding-based	DEXPERTS	**0.292**	**10.00%**	32.55	<u>0.492</u>	<u>42.20%</u>	<u>33.59</u>
	GeDi	0.387	24.80%	38.21	**0.430**	**34.20%**	47.42
Prompt-based	SD (λ=10)	0.424	32.30%	13.02	0.723	80.60%	14.21
	SD (λ=50)	0.373	23.10%	18.80	0.649	80.90%	19.66
	SD (λ=100)	0.355	20.30%	21.09	0.623	65.50%	23.32
	REI	0.496	40.60%	30.56	0.711	85.00%	31.20
Fine-tuning-based	DAPT	<u>0.331</u>	<u>18.90%</u>	19.72	0.558	57.00%	22.47
	ATCON	0.482	42.00%	62.95	0.746	85.10%	69.51
	InstructionCTG	0.360	38.60%	23.90	0.694	65.70%	27.56
	Ours	0.304	21.20%	<u>12.31</u>	0.608	61.90%	<u>14.01</u>

Fig. 3: Comparison of toxicity control performance across different methods. The bars show three metrics: Maximum Toxicity (↓), Toxicity Probability (↓), and Perplexity (↓).

a) Fine-tuning-based Methods:

- **DAPT** [27]: Continues pre-training on domain-specific unlabeled data to adapt the model to target domain characteristics
- **ATCON** [28]: Incorporates additional control-related tasks during pre-training and utilizes special control codes for style-specific generation
- **InstructionCTG** [29]: Achieves effective control through weakly supervised data synthesis, natural language templates, and instruction tuning

b) Decoding-based Methods:

- **GeDi** [12]: Employs a discriminator model to guide generation by incorporating prediction scores into token probability distribution using Bayes' rule
- **DEXPERTS** [30]: Utilizes expert and anti-expert models, combining their token predictions through weighted probability aggregation

c) Prompt-based Methods:

- **SD** [31]: Guides generation through specific prompt templates, comparing outputs with and without prompts to filter undesired content
- **REI** [32]: Introduces regular expression instruction methodology, combining specific markup language with few-shot learning and fine-tuning strategies

For all methods, we employ nucleus sampling [33] with $p = 0.9$, generating 25 continuations per prompt.

D. Results and Analysis

1) Main Results: As demonstrated in Table II, our proposed method exhibits significant toxicity control capabilities while maintaining low perplexity in both toxic and non-toxic contexts. In non-toxic contexts, our approach demonstrates superior performance compared to other fine-tuning-based methods in controlling toxicity continuation, specifically achieving a maximum toxicity score (0.304) and toxicity probability (21.20%), while maintaining exceptional language fluency (PPL = 12.31). Compared to prompt-learning methods, our approach demonstrates advantages across both toxicity control metrics and perplexity measures, highlighting its multi-dimensional performance benefits.

In toxic contexts, while decoding-based methods (such as DEXPERTS and GeDi) show certain advantages in automated evaluation metrics (e.g., DEXPERTS achieving a maximum toxicity score of 0.292 and toxicity probability of 10.00%), these methods face significant computational complexity challenges. Specifically, decoding methods typically require deployment of larger model architectures and implementation of probability distribution rewriting mechanisms.

These requirements lead to increased computational costs and higher perplexity scores (e.g., GeDi's perplexity scores of 38.21 and 47.42 in non-toxic and toxic contexts, respectively), potentially significantly impacting the fluency and naturalness of generated text.

TABLE III: Comparison of toxicity metrics under different contexts

(a) Results under non-toxic context

Model	Severe Tox	Sex	Threat	Profanity	Id. Attack
GPT-2	10.04	18.76	5.87	41.50	5.45
DEXPERTS	6.95	9.35	5.65	18.77	5.76
GeDi	4.21	13.41	3.92	12.78	5.53
SD(λ = 50)	<u>1.75</u>	12.88	<u>0.78</u>	10.53	<u>1.87</u>
REI	5.37	16.17	1.59	**5.64**	**1.15**
DAPT	2.54	<u>9.63</u>	3.77	**5.23**	2.63
ATCON	2.58	11.64	1.84	21.58	2.17
InstructionCTG	2.39	9.88	1.93	19.87	2.54
Ours	**1.80**	**8.24**	**1.56**	7.16	2.32

(b) Results under toxic context

Model	Severe Tox	Sex	Threat	Profanity	Id. Attack
GPT-2	65.81	49.52	23.84	68.47	36.80
DEXPERTS	29.04	16.42	15.83	24.50	14.22
GeDi	14.85	14.58	25.81	19.61	15.41
SD(λ = 50)	24.71	21.52	11.25	21.15	19.79
REI	26.58	29.11	24.32	38.42	11.53
DAPT	<u>13.47</u>	<u>16.77</u>	<u>4.26</u>	<u>18.50</u>	<u>9.54</u>
ATCON	19.32	17.15	19.80	25.77	13.27
InstructionCTG	22.91	19.56	8.50	24.64	11.36
Ours	**15.78**	**10.53**	**6.78**	**19.80**	**5.47**

2) Fine-grained Analysis of Toxicity Control: Our method demonstrates excellent control across five toxic categories under both non-toxic and toxic contexts as shown in Table III.

In non-toxic settings (Table IIIa), we achieve superior performance in controlling Severe Toxicity (1.80) and Sex-related content (8.24), significantly outperforming baseline methods. While our approach shows slightly higher Profanity scores (7.16) compared to REI (5.64) and DAPT (5.23), the overall performance remains competitive.

In toxic contexts (Table IIIb), our method maintains balanced performance across all categories with notably better control over Threat (6.78) and Identity Attack (5.47), resulting in the lowest overall risk profile among all compared methods. These comprehensive evaluations across different toxicity dimensions and environmental contexts demonstrate the effectiveness and stability of our proposed method in controlling various forms of toxic content generation.

E. Validation on Other Large Language Models

To further validate the scalability and effectiveness of our proposed method on large language models, we conducted comprehensive toxicity evaluations on the llama3-8b model under toxic contexts. The experiments utilized 2,000 toxic prompts from the RealToxicityPrompts (RTP) dataset. To provide a thorough assessment of model practicality, we incorporated additional metrics including average generation length and rejection rate.The generation hyperparameters and results are listed in Table IV and Table V.

TABLE IV: Generation hyperparameters

Parameter	Value
temperature	0.70
top-p	0.90
top-k	50
max-length	50

TABLE V: Comparison of toxicity control capabilities between base model and our method

Metrics	llama3-8b	llama3-8b_PE-CG
Max. Tox. (Toxic)	0.25	**0.15**
Tox. Prob. (Toxic)	15%	**7%**
PPL (Toxic)	**10.50**	11.56
Length (avg tokens)	14.90	**25.70**
Refuse	12%	**5%**

In our experimental setup, while the base llama3-8b model demonstrated inherent safety mechanisms, there remained significant room for improvement across multiple metrics.

With constraint guidance, maximum toxicity reduced from 0.25 to 0.15 and toxicity probability decreased from 15% to 7%. Average generation length increased from 14.90 to 25.70 tokens, while the rejection rate for toxic prompts decreased from 9.12% to 8.05%.

These results empirically validate the effectiveness of our approach, showing that it not only improves security controls but also reduces rejection rates, ensuring the usefulness of the model.

V. READABILITY-CONTROLLED SUMMARIZATION

A. Experimental Setup

To evaluate the balance between controllability and utility of our proposed method, we follow the experimental protocol of [34] and utilize LLaMA-2-7B-chat [1] as the base model. The evaluation is conducted on the CNN/DailyMail [35] dataset, comprising 11,490 news articles, to assess the model's capability in generating text with varying knowledge depths and tones.

The evaluation methodology systematically examines the model's ability to adapt to different audience comprehension levels by prefixing each article with specific instructions, such as:

- Summarize the following news article for a primary-school student.
- Summarize the following news article for a college professor.

B. Baseline Methods

We compare our approach with three baseline methods:

1) **Direct Output**: Direct generation using the base LLaMA-2-7B-chat model.
2) **Style Transfer** [36]: Post-processing the output through a style transformation model to adjust text formality).
3) **CNN/DailyMail Fine-tuning**: Fine-tuning on the CNN/DailyMail dataset using two techniques:
   - **Dynamic Word Unit Prediction** [37]: Encodes readability levels (e.g., Flesch-Kincaid scores) as auxiliary features for the decoder, enabling dynamic word unit prediction during the decoding phase for

TABLE VI: Definitions and criteria for complex words and difficult words

Characteristic	Complex Words	Difficult Words
Definition	Number of syllables in words	Words that are not in the Dale-Chall word list
Criterion	Words with three or more syllables	Check if word exists in Dale-Chall word list
Examples	organization, unbelievable	catalyst, jurisprudence
Application	Evaluates linguistic complexity of text	Evaluates vocabulary difficulty of text

direct style control during inference without model retraining.

- **Controllable Readability** [38]: Assigns readability scores to each text entry through manual annotation, utilizing model-evaluated readability scores of summaries in the CNN/DailyMail dataset as labels.

C. Evaluation Metrics

We evaluate the model's controllability from both readability and summary quality.

1) Readability Metrics: The readability assessment employs four widely-used metrics.

- **Flesch Reading Ease (FRE)**: Evaluates text readability through word length and sentence length analysis.
- **Dale-Chall Readability (DCR)**: Considers vocabulary difficulty and sentence length, focusing on the usage of difficult words.
- **Gunning Fog Index (GFI)**: Emphasizes educational level requirements by examining sentence length and complex word proportions.
- **Coleman-Liau Index (CLI)**: Estimates required education level based on character, word, and sentence statistical features.

The definitions and evaluation criteria for complex words and difficult words as shown in Table VI, which serve as key components in our readability assessment metrics.

2) Quality Metrics: For semantic evaluation, we utilized BERTScore (BS), which leverages BERT's contextual representations to measure semantic similarity between generated and reference texts. Additionally, we incorporated Rouge-L (RG-L) to evaluate content overlap through longest common subsequence calculation, thereby assessing content completeness and accuracy.

3) Comprehensive Metrics: To provide a more holistic assessment independent of reference texts, we introduced GPT-4 scoring as a comprehensive metric that evaluates language fluency, contextual consistency, and readability.

D. Results and Analysis

Our experimental results as presented in Table VII. The experimental results demonstrate that our method achieved substantial improvements, particularly in readability metrics, with a 26.01 increase in the FRE score compared to the baseline. While some existing methods show competitive performance in individual readability metrics, our approach distinguishes itself through its balanced performance across all dimensions. Notably, our method achieved a GPT-4 score

TABLE VII: Comprehensive evaluation results comparing different methods across readability and quality metrics

Method	FRE↑	DCR↓	GFI↓	CLI↓
Default (LLaMA-2-7B-chat)	53.57	10.48	14.08	11.69
Style Transfer	70.79	8.51	11.02	8.13
Dynamic Word Unit Prediction	75.70	9.59	8.26	8.50
Controllable Readability	**83.20**	**6.60**	**6.30**	**6.80**
Ours	<u>79.58</u>	<u>7.52</u>	<u>8.02</u>	<u>7.26</u>

Method	BS↑	RG-L↑	GPT4↑
Default (LLaMA-2-7B-chat)	**87.33**	<u>34.63</u>	82.57
Style Transfer	85.87	27.68	<u>86.55</u>
Dynamic Word Unit Prediction	<u>86.98</u>	**37.88**	88.38
Controllable Readability	86.80	30.75	87.16
Ours	84.94	24.97	**92.56**

of 92.56, significantly outperforming other approaches and demonstrating superior overall text quality.

Although existing approaches such as Controllable Readability demonstrate marginally superior performance in certain individual metrics, we identify three significant limitations in these methods. First, they focus exclusively on readability control as a single dimension, neglecting the crucial aspect of content integrity. Second, these methods employ relatively coarse-grained control mechanisms that preclude precise adjustments of text characteristics. Third, their training on specific datasets results in limited generalization capability, restricting their effectiveness across diverse application scenarios.

This comprehensive performance is due to dynamic constraint guidance, which enables precise readability control while maintaining content integrity.

VI. CONCLUSION

This paper proposes a controllable text generation method based on constraint guidance, which decomposes the task into three key steps: prediction type, constraint construction, and guided generation. By constructing a dataset containing 90,000 text-constraint pairs, precise control of different topic types is achieved. Experiments on the LLaMa3-8B and GPT-2 models verify the effectiveness of this method in toxicity control and readability adjustment tasks, which not only reduces harmful outputs, but also maintains a low perplexity, while achieving a better balance between readability and content quality.

REFERENCES

[1] H. Touvron, L. Martin, and K. e. a. Stone, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[2] J. Achiam, S. Adler, and S. e. a. Agarwal, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[3] N. S. Keskar, B. McCann, and L. R. e. a. Varshney, "Ctrl: a conditional transformer language model for controllable generation," *arXiv e-prints*, 2019.

[4] D. Pascual, B. Egressy, and C. e. a. Meister, "A plug-and-play method for controlled text generation," in *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2021, pp. 3973–3997.

[5] J. Qian, L. Dong, and Y. e. a. Shen, "Controllable natural language generation with contrastive prefixes," in *Findings of the Association for Computational Linguistics: ACL 2022*, 2022, pp. 2912–2924.

[6] N. Houlsby, A. Giurgiu, and S. e. a. Jastrzebski, "Parameter-efficient transfer learning for nlp," in *International Conference on Machine Learning*, 2019, pp. 2790–2799.

[7] L. Ouyang, J. Wu, and X. e. a. Jiang, "Training language models to follow instructions with human feedback," *Advances in Neural Information Processing Systems*, pp. 27 730–27 744, 2022.

[8] Y. Gu, X. Feng, and S. e. a. Ma, "A distributional lens for multi-aspect controllable text generation," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022, pp. 1023–1043.

[9] W. Liu, W. Zeng, and K. e. a. He, "What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning," in *The Twelfth International Conference on Learning Representations*, 2024.

[10] T. Shin, Y. Razeghi, and R. L. e. a. Logan IV, "Autoprompt: Eliciting knowledge from language models with automatically generated prompts," *arXiv preprint arXiv:2010.15980*, 2020.

[11] K. Pathania, "Enhancing conditional image generation with explainable latent space manipulation," *arXiv preprint arXiv:2408.16232*, 2024.

[12] B. Krause, A. D. Gotmare, and B. e. a. McCann, "GeDi: Generative discriminator guided sequence generation," in *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2021, pp. 4929–4952.

[13] B. Peng, C. Li, and P. e. a. He, "Instruction tuning with gpt-4," *arXiv preprint arXiv:2304.03277*, 2023.

[14] L. Chen, J. Li, and X. e. a. Dong, "Sharegpt4v: Improving large multi-modal models with better captions," in *European Conference on Computer Vision*, 2024, pp. 370–387.

[15] J. M. Liu, D. Li, and H. e. a. Cao, "Chatcounselor: A large language models for mental health support," *arXiv preprint arXiv:2309.15461*, 2023.

[16] G. Zeng, W. Yang, and Z. e. a. Ju, "Meddialog: Large-scale medical dialogue datasets," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 9241–9250.

[17] S. Yue, W. Chen, and S. e. a. Wang, "Disc-lawllm: Fine-tuning large language models for intelligent legal services," *arXiv preprint arXiv:2309.11325*, 2023.

[18] Q. Huang, M. Tao, and C. e. a. Zhang, "Lawyer llama technical report," *arXiv preprint arXiv:2305.15062*, 2023.

[19] G. Bhatia, E. M. B. Nagoudi, and H. e. a. Cavusoglu, "Fintral: A family of gpt-4 level multimodal financial large language models," *arXiv preprint arXiv:2402.10986*, 2024.

[20] J. Ji, M. Liu, and J. e. a. Dai, "Beavertails: Towards improved safety alignment of llm via a human-preference dataset," *Advances in Neural Information Processing Systems*, vol. 36, pp. 24 678–24 704, 2023.

[21] Y. Huang, Q. Zhang, and L. e. a. Sun, "Trustgpt: A benchmark for trustworthy and responsible large language models," *arXiv preprint arXiv:2306.11507*, 2023.

[22] A. Radford, J. Wu, and R. e. a. Child, "Language models are unsupervised multitask learners," *OpenAI Blog*, p. 9, 2019.

[23] S. Gehman, S. Gururangan, and M. e. a. Sap, "Realtoxicityprompts: Evaluating neural toxic degeneration in language models," *arXiv preprint arXiv:2009.11462*, 2020.

[24] H. Hosseini, S. Kannan, and B. e. a. Zhang, "Deceiving google's perspective api built for detecting toxic comments," *arXiv preprint arXiv:1702.08138*, 2017.

[25] C. T. Leong, Y. Cheng, and J. e. a. Wang, "Self-detoxifying language models via toxification reversal," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 4433–4449.

[26] P. He, J. Gao, and W. Chen, "DeBERTav3: Improving deBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing," in *The Eleventh International Conference on Learning Representations*, 2023.

[27] S. Gururangan, A. Marasović, and S. e. a. Swayamdipta, "Don't stop pretraining: Adapt language models to domains and tasks," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 8342–8360.

[28] N. S. Keskar, B. McCann, and L. R. e. a. Varshney, "Ctrl: A conditional transformer language model for controllable generation," *arXiv preprint arXiv:1909.05858*, 2019.

[29] W. Zhou, Y. E. Jiang, and E. e. a. Wilcox, "Controlled text generation with natural language instructions," in *International Conference on Machine Learning*, 2023, pp. 42 602–42 613.

[30] A. Liu, M. Sap, and X. e. a. Lu, "DExperts: Decoding-time controlled text generation with experts and anti-experts," 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)*, 2021, pp. 6691–6706.

[31] T. Schick, S. Udupa, and H. Schütze, "Self-diagnosis and self-debiasing: A proposal for reducing corpus-based bias in nlp," *Transactions of the Association for Computational Linguistics*, pp. 1408–1424, 2021.

[32] X. Zheng, H. Lin, and X. e. a. Han, "Toward unified controllable text generation via regular expression instruction," *arXiv preprint arXiv:2309.10447*, 2023.

[33] A. Holtzman, J. Buys, and L. e. a. Du, "The curious case of neural text degeneration," in *International Conference on Learning Representations*, 2020.

[34] Y. Wang and V. Demberg, "RSA-control: A pragmatics-grounded lightweight controllable text generation framework," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024, pp. 5561–5582.

[35] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, pp. 1073–1083.

[36] P. Damodaran, "Styleformer," 2021. [Online]. Available: https://github.com/PrithivirajDamodaran/Styleformer

[37] S. Cao and L. Wang, "Inference time style control for summarization," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021, pp. 5942–5953.

[38] L. F. R. Ribeiro, M. Bansal, and M. Dreyer, "Generating summaries with controllable readability levels," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 11 669–11 687.