Latent Variables Improve Hard-Constrained Controllable Text Generation on Weak Correlation

Weigang Zhu[1], Xiaoming Liu[2], Guan Yang[3], Jie Liu[4], Haotian Qi[5]
School of Computer, Zhongyuan University of Technology, Zhengzhou, Henan 451191, China[1, 2, 3, 5]
Zhengzhou Key Laboratory of Text Processing and Image Understanding, Zhengzhou Henan 450007, China[1, 2, 3]
School of Information Science, North China University of Technology, Beijing 100144, China[4]
Research Center for Language Intelligence of China, Beijing 100089, China[2, 4]

***Abstract*—Hard-constrained controllable text generation aims to forcefully generate texts that contain specified constrained vocabulary, fulfilling the demands of more specialized application scenarios in comparison to soft constraint controllable text generation. However, in the presence of multiple weak correlation constraints in the constraint set, soft-constrained controllable models aggravate the constraint loss phenomenon, while the hard-constrained controllable models significantly suffer from quality degradation. To address this problem, a method for hard-constrained controllable text generation based on latent variables improving on weak correlations is proposed. The method utilizes latent variables to capture both global and local constraint correlation information to guide the language model to generate hard-constrained controllable text at the macro and micro levels, respectively. The introduction of latent variables not only reveals the latent correlation between constraints, but also helps the model to precisely satisfy these constraints while maintaining semantic coherence and logical correctness. Experiment findings reveal that under conditions of weak correlation hard constraints, the quality of text generation by the method proposed exceeds that of the currently established strong baseline models.**

***Keywords*—Latent variables; controllable text generation; weak correlation; hard constraint**

I. Introduction

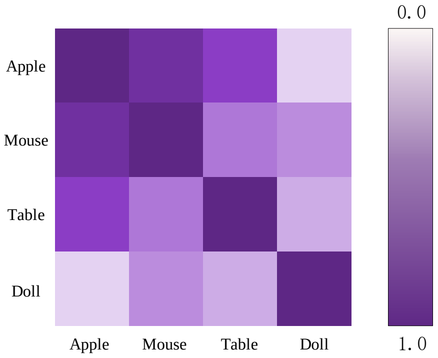
Pre-trained Language Models (PLMs) [1] [2] [3] achieve high-quality text generation through learning from massive corpora and modelling the distribution of natural language. To meet the requirements of specific tasks or scenarios, such as simulating conversations, describing data, editing stories, or auto-generating reports, researchers introduce control mechanisms to ensure that the generated text satisfies given constraints. These constraints can encompass aspects such as sentiment, tone, topic, style, and content.

Constrained controllable text generation can be divided into three rudimentary strategies. The first method [4] [5] [6] [7] usually encompasses constraint controllability during the decoding phase. For example, within each Beam in Beam Search, scores are jointly computed based on the constraints and predicted words, eventually selecting the text route that is both highest scoring and meets the constraints. This method comes with a high decoding cost.

The second method applies a non-autoregressive language model (NAR) [8] [9] based on an Insertion-Transformer. During the text generation process, NAR initially generates words that are bound by constraints, gradually refining the text through insertion operations. These hard-constrained methods require multiple rounds of optimization to generate high-quality text, leading to no significant advantage in terms of generation efficiency and text quality compared to autoregressive models.

The third method is a prompt-based approach [10]. It inputs prompts or a piece of text into the model to guide the model in generating text in line with the prompts. This method offers the advantages of low decoding overheads and high generation quality. However, during the initial phase of generation, the model usually focuses on information that is highly related to the prompt, leaning towards generating text skewed away from prompts with weak correlation.

Fig. 1 delineates the process of generating text from four specific cue words: “Apple”, “Mouse”, “Table”, and “Doll”. The relationship among these words is portrayed through a spectrum of colours where darker shades imply stronger connections, as deduced through Euclidean distance. The diagram reveals a hierarchy change from deep hues in the top-left corner to paler ones in the bottom-right, with the confluence of “Apple” and “Mouse” appearing the most intense, indicating their high correlation. Conversely, the link between “Table” and the rest of the cue words is comparatively weaker, with “Doll” exhibiting the lowest correlation. An in-depth analysis of the generated text content reveals that the model gives precedence to “Apple” and “Mouse” during the composition, demonstrates reduced attention towards “Table”, and entirely excludes “Doll”.



The **Apple Mouse** is a **mouse** that can be used with the iPad, iPhone and iPod touch. It has an **apple**-shaped button on top of the **table** where it sits in your hand.

Fig. 1. Example of the constrained controllable text generation problem.

*Corresponding Author.

This case demonstrates that during the initial stages of generation, language models tend to focus more on constraints with stronger relevance. This results in the model deviating from weaker constraints as the generation progresses, moving towards directions less associated with these weaker constraints.

To address this problem, we propose a method that latent variables improve hard-constrained controllable text generation method on weak correlation. This method introduces a latent variable constraint correlation module which initially captures the semantic context related to constraints and decodes to generate ambiguous text as a global constraint correlation. Subsequently, the module integrates the global constraint correlation text with individual constraint using latent variables to acquire localized constraint correlation text. Ultimately, the model combines both global and local constraint information with the context, steering the text generation towards the constraints. Compared to robust baseline models, our model enhances the connection between weak correlation constraints and the context, generating high-quality text that complies with hard constraints.

The sections that follow are organized as follows: Section II provides an introduction and summary of the works on controllable text generation and latent Transformers. Section III elaborates on the methodology of the model. Section IV gives a concise description of the experimental framework. In Section V, we present an array of experimental outcomes and provide an analysis of these results. Finally, Section VI summarizes the study with a thoughtful conclusion.

II. Related Works

This section chiefly summarizes related works on controllable text generation and latent Transformers. Our mission is to guide the model to generate high-quality text in alignment with constraints, which emphasis is strengthening the correlation information of weak correlation constraints.

A. Controllable Text Generation

Controllable text generation represents a pivotal and challenging branch within the field of Natural Language Processing (NLP), giving rise to a diversity of solutions. Initially, Keskar et al. introduced a novel method by appending a control code (domain, style, theme, etc.) at the beginning of the text corpus, training a language model, CTRL, based on various control codes. Subsequently, Dathathri et al. [11] developed the PPLM model, leveraging an attribute discriminator model to guide the PLM in generating text. Building upon the works of CTRL and PPLM, Chan et al.[12] introduced a conditional control module that facilitates precise control over text generation at the level of words and phrases. Krause et al. [13] employed class-conditional language models as generative discriminators (GeDis) to direct the language generation towards the desired attributes. Yang and Klein [14] proposed the flexible and modular Fudge model, which adds an attribute predictor on top of the original PLM to adjust the probability distribution, achieving improved performance in tasks such as poetry generation, thematic text generation, and machine translation. Pascual et al. [4] introduced a straightforward, efficient, and discriminator-free plug-and-play decoding method, K2T. Other researchers have advanced upon NAR, such as Zhang et al.[8], who proposed Pointer, an insertion-based method for constrained text generation. Miao et al. [15] developed a method known as CGMH, which facilitates the generation of constrained sentences through Metropolis-Hastings sampling. He [9] improved upon CGMH by enabling the model to autonomously learn where to insert, replace, and duplicate content.

To address the escalating costs associated with model training, researchers have proposed the use of Prompts. Li et al. [16] applied Prompts to the domain of controllable text generation, introducing prefixes that guide and constrain the output of generative models to yield desired results. Similarly, Lester et al. [17] employed the model to learn "soft prompts" to adjust a frozen language model for performing specific downstream tasks. Han et al. [18] defined a set of logical rules and used Prompts embedded with these rules as input to generate text related to specified categories as the output. Zou et al. [19] suggested a method known as reverse prompt, which employs candidate texts generated by a PLM to inversely predict prompts. Yang et al. [20] introduced a soft prompt-based method for multi-attribute controllable text generation, which diminishes the impact of prompt placement on text quality. Carlsson et al. [10] presented the use of non-residual prompts for fine-grained control of text generation, addressing the trade-off between fine-grained control and the capability for more expressive advanced instructions.

B. Latent Transformers

Compared to the conventional Transformer models, the latent variable-based Transformer introduces an extra latent variable to capture the semantic information of the input sequence, followed by NAR prediction. The approach of using latent variables in Transformer models was initially proposed by Kaiser et al. [21], who incorporated the concept of discrete latent variables to expedite the decoding process. Expanding on this concept, Shu et al. [22] introduced a NAR neural machine translation et al. [24] proposed a method for non-autoregressive translation by learning target category codes, and later introduced a technique for parallel text generation [25] using method utilizing discrete latent variables. Ma et al. [23] combined generative flows with conditional variational autoencoders to efficiently generate conditional sequences. Based on latent variables, Bao discrete latent variables to capture lexical category information, thus mitigating multimodal issues.

This study makes the following three main contributions to hard-constrained controllable text generation, Specifically, as follows:

1) A novel latent variable constraint controllable strategy is proposed to improve the issue of constraint bias in existing language models.

2) Utilizing latent variables to reveal potential connections among constraints, assisting language models in accurately fulfilling given hard constraints while maintaining semantic coherence and logical correctness.

3) Confirming the effectiveness of this latent variable constraint controllable strategy through experimental results. It demonstrates that this method can effectively satisfy weakly related hard constraint conditions while ensuring the quality of generated text, meeting practical application requirements.

III. Methodology

Humans form sentences based on constraints through rational combinations and skilful utilization. In other words, the intrinsic message contained in a sentence reveals the latent and profound connections among constraints. Present controllable text generation models mainly focus on learning the probability distribution that coexists with text and constraints, while neglecting the correlated information between the constraints. This limitation often leads the model to favour constraints with stronger correlations when faced with weak correlation constraints. Hence, we guide text generation with constraint correlation information, a method more in line with human thinking. By introducing latent variables, we more effectively unearth the latent correlation within sentences, thereby strengthening their generalization capabilities, especially in managing weak correlation constraints. Beyond that, using latent variables to model target sentences helps to reduce the multimodality problem of sentences. Additionally, learning discrete latent variables directly through a Transformer greatly improves the model's overall operational efficiency.

This section is structured into five main components: Part A illustrates the generation of latent constraint correlations. Part B details the embedding of constraints, Part C describes the framework of model, Part D explains the model training process, and finally, Part E summarizes the model inference.

A. Latent Constraint Correlation Generation

To enhance the capacity of model in handling weak correlation constraints and controllable text generation, we introduce a Latent Constraint Correlation Generation (LCCG) module inspired by the concept of VQ-VAE[26]. This module utilizes latent variables to separately process all constraints and individual constraint, thereby obtaining both global and local constraint correlation information. As shown in Fig. 2, based on the foundation of the Vanilla Encoder of Transformers (VET), we add a Constraint Embedding module and a Target Length Prediction (TLP) module. Moreover, the Multi-head Attention layer (MHA) in the Decoder module is employed as the Decoder for LCCG.

The LCCG is responsible for processing an input X of length m, initially transforming it into embedded vectors through the CE layer, and then feeding these vectors into the TLP layer to predict the target sentence length l, akin to a typical classification task. Its prediction loss is as follows:

$$\mathrm{L}_{len} = -\log p_{\theta}(y_{len} \mid X) \tag{1}$$

where, θ represents the model parameters.

After obtaining the length value l, the module adopts the Softcopy mechanism proposed by Wei et al. [27] to match the target sentence length. The hidden layer state $H = \{h_1, h_2, ..., h_j\}$, obtained after the process, is then fed into VET to acquire the continuous latent variable $Z = \{z_1, z_2, ..., z_j\}$. To discretize these continuous latent variables, our study employs the Vector Quantised technique. First, define an embedding space $Q \in \square^{K\times D}$ and denote K is the number of vectors e in the embedding space Q. Then, discrete latent variables are assigned to each continuous latent variable through nearest neighbour lookup. The formula is as follows:

$$\begin{aligned} z_{q,i}^{(k)} &= e_k, \\ k &= \arg\min_{t\in[K]} \left\| z_i - e_t \right\|_2 \end{aligned} \tag{2}$$

where, $i \in \{1, 2, ..., m\}$.

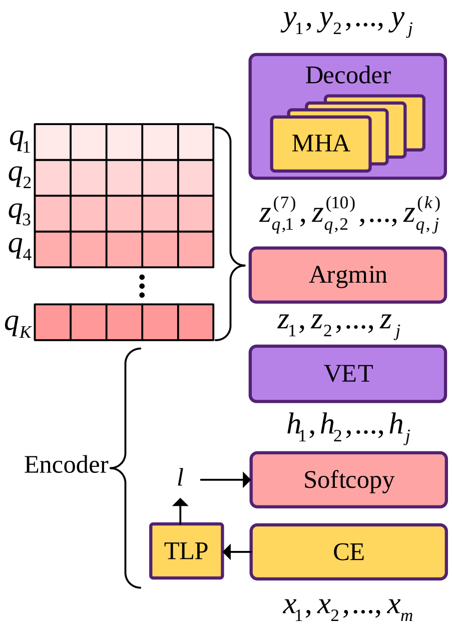


Fig. 2. Latent constraint correlation generation module.

Vector Quantization employs the $\arg\min$ function during forward propagation to obtain discrete latent variables Z_q. As the $\arg\min$ function is non-differentiable, a Straight-Through Estimator is utilized to design the loss, thus the loss for LCCG is as follows:

$$\begin{aligned} &\mathrm{L}_{LCCG} = -\log p_{\theta}(Y \mid Z_q, X) + \alpha \parallel z - sg[z_q] \parallel_2^2 + \beta \mathrm{L}_{len}, \\ &sg(\square) = \begin{cases} x & forward\ pass \\ 0 & backward\ pass \end{cases} \end{aligned} \tag{3}$$

where, $\alpha = 2.5$, $\beta = 2.5$. LCCG updates the embedding space $q_j \in Q$ vectors with an Exponential Moving Average over a small batch of target labels $\{y_1, ..., y_i, ...\}$, which is defined as follows:

$$\begin{aligned} v_j &\leftarrow \lambda v_j + (1-\lambda)\sum_i 1[z_{qi} = j], \\ q_j &\leftarrow \lambda q_j + (1-\lambda)\sum_i \frac{1[z_{qi} = j] y_i}{v_j} \end{aligned} \tag{4}$$

where, v_j represents the count for the group j, $1[\square]$ is the indicator function, and the decay parameter λ is set to 0.9999 following prior work.

B. Constraint Embedding

To deepen our comprehension of constraints and textual characteristics, we integrate a Constraint Embedding layer positioned between the conventional Token Embedding and Position Embedding layers. The specific framework is illustrated in Fig. 3.

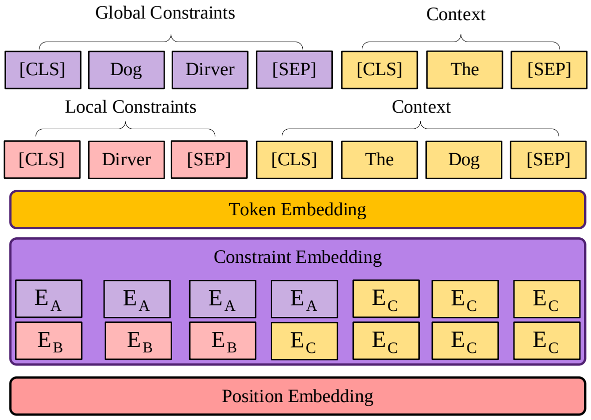


Fig. 3. Constraint embedding layer.

The newly added Constraint Embedding layer embeds three types of input sequences: Global Constraints Sequence, Local Constraints Sequence, and Context Sequence, into three different embedding vectors, namely E_A, E_B, and E_C. In our practical experiments, E_A, E_B, and E_C were set to vectors entirely composed of 0, 1, and 2, respectively. This embedding approach effectively captures the latent information within global constraints, as well as the latent information between local constraints and sentences, thus enhancing the ability of model to understand constraints.

C. Model Framework

Our model adopts an encoder-decoder Transformer architecture similar to BART. As depicted in Fig. 4, the primary function of the encoder is to transform constraints into latent variables. In the decoder section, we have refined the attention mechanism for each layer. The Masked Multi-Head Attention (MMHA) is used exclusively to obscure future information of each token in the context, while the MHA is used for cross-attention between the context and constraint-related information and also as a part of the latent variable decoder. The Fully Connected layer (FC) is utilized to fine-tune the constraint correlation information and text generation. The given constraints $C=\{c_1,c_2,\dots c_m\}$ and the context X, let the constraints c_i denote unmet constraints within the context, where $i\in\{1,2,\dots,m\}$. The satisfaction of fine-grained, controllable text generation requires the following steps:

1) Global constraints text generation: Constraints C are input into the encoder to obtain discrete latent variables Z_G, which then bypass the MMHA module and are passed to the MHA in the decoder, thereby generating global constraint correlation text Y_G:

$$p(Y_G\,|\,C)=\prod_{j=1}^{J}p(y_j\,|\,Z_G,C)\square p(Z_G\,|\,C) \tag{5}$$

2) Local constraints text generation: This step is different from the first one in that it uses Y_G as the context, which is input along with the constraints c_i. The model generates local constraint correlation text Y_L that is relevant to the constraints based on the input information:

$$p(Y_L\,|\,c_i,Y_G)=\prod_{h=1}^{H}p_\theta(y_h\,|\,Z_L,c_i,Y_G)\square p(Z_L\,|\,c_i,Y_G) \tag{6}$$

3) Constraint-guided text generation: This step entails applying Global Average Pooling (GAP) to Y_G, to obtain P_G. The context X, after embedding, is fed into the MMHA of the decoder while masking future information of the token. Thereafter, X, as the Query, engages in cross-attention calculations with both P_G and Y_L to predict the subsequent token.

$$p(y\,|\,X,C)=\prod_{n=1}^{N}p_\theta(y\,|\,X<n,P_g,Y_L) \tag{7}$$

4) Incorporate the predicted token into the context X and reselect constraint c_i. Repeat the operations of the second and third steps until the generated text meets all the given constraint conditions.

Global constraint correlation text provides a macroscopic guiding direction for the language model, while local constraint correlation text serves to refine the relevant constraint text. Considering that the quality requirements for these texts are not high, a non-autoregressive approach is adopted for generation to improve efficiency.

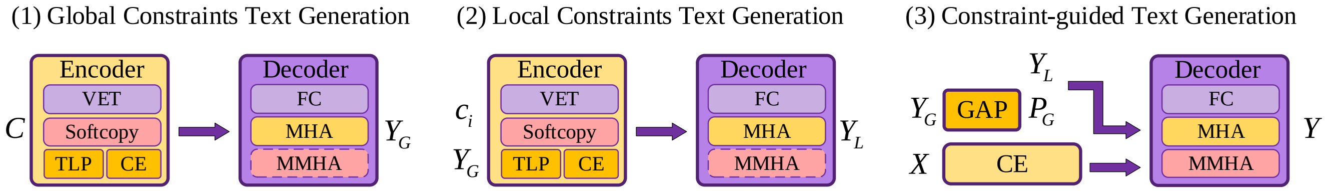


Fig. 4. Procedure for hard-constrained text generation.

D. Model Training

The model training consists of two stages. The first stage is the training of latent variables, where the task is to enable the model to generate both global and local constraint correlation text given the provided constraints and contextual conditions. During this stage, the MMHA layer is frozen. When the input consists of all constraints, the text is used as the training target. When the input consists of a single constraint or global constraint correlation text, the target text is selected from the beginning of the original text or the position after the previous constraint to the end of the original text or the position before the next constraint.

The second stage is the fine-tuning stage, where the encoder, MHA, and MMHA layers of the model are frozen, and only the FC layer of the decoder is fine-tuned. The specific loss function is as follows:

$$\begin{aligned} \mathrm{L} = -\log p_\theta(y \mid X, P_G, Y_L) - \mu \log p_\theta(Y_G \mid C) - \\ \eta \log p_\theta(Y_L \mid c_i, Y_G) \end{aligned} \tag{8}$$

where $\mu = 0.4$, $\eta = 0.6$.

E. Inference

Cross-attention between constraints and context is an effective soft constraint method for language models. However, it is challenging to train a model to generate text that fully incorporates constraints, often requiring additional processing. Therefore, in the inference stage, we draw inspiration from the work of Pascual et al. [4] and make some improvements to ensure that the model's output meets the constraints.

In concrete terms, we involve treating a subset of words from the correlated text as a set of guiding words, denoted as set W, while disregarding the order of these guiding words. At each decoding step t, a new subset of guiding words, denoted as set W_t, is selected from set W, consisting of guiding words that have not appeared before the current time step. The top-k algorithm is then employed to select the k most likely predicted words from the predicted word set. Subsequently, the similarity between each predicted word y_t and guiding word w is computed, $w \in W_t$. This similarity is then weighted with the probability distribution of each predicted word, resulting in a reweighted probability distribution for the current word. The formula is as follows:

$$\begin{aligned} score(\cdot \mid y_{1\cdots t-1}) &= \log p(\cdot \mid y_{1\cdots t-1}) \\ score'(y_t, W_t \mid y_{1\cdots t-1}) &= score(y_t \mid y_{1\cdots t-1}) + \\ &\lambda\square\max\left\{0, \max_{w \in W_t}\left[\cos(y_t, w)\right]\right\} \end{aligned} \tag{9}$$

where, $score(\cdot \mid y_{1\cdots t-1})$ is the scoring function, and scores are used for sampling. $score'(y_t, W_t \mid y_{1\cdots t-1})$ is the overloaded scoring function, which takes the guiding word set W_t as input. Parameter λ adjusts the transition of tokens generated by the model from being unconstrained to becoming the next guiding word. The calculation for λ is as follows:

$$\lambda_t = \begin{cases} \lambda_0 \exp\left\{\dfrac{c(t - t_n)}{T - |W_t| - t_n}\right\} & t < T - |W_t| \\ \infty & t \geq T - |W_t| \end{cases} \tag{10}$$

where, T represents the length of the text under local constraints, t_n denotes the position where the last guiding word appeared. The hyperparameters λ_0 and c are used to control the initial value and increment of λ. In this context, they are set to 10 and 100, respectively. When $\lambda \to \infty$, the predicted word is forced to be a constrained guiding word, and the current local constraint is terminated. Then, the model enters the next local constraint while updating the guiding word set W.

IV. Experimental Setup

In this section, we list the datasets suitable for the text generation method used in our study, then outline the metrics for both automatic and human evaluations, and finally provide a description of the experimental details.

A. Datasets and Evaluation Metrics

1) Datasets: The experiments consist of two tasks, starting with the model pre-trained on the Wikitext-103-raw-v1 dataset. The first task is focused on constraint-driven controllable text generation, with the objective to evaluate and ascertain if this approach enhances the model's competency in excavating and understanding the latent connections among constraints, as well as if it elevates the quality of text production. The data for the experiments include CommonGen [28], Yelp Reviews [29], and E2ENLG [30], with detailed information presented in Table I.

TABLE I. The Comparison Statistics of Datasets

Datasets	Train	Valid	Test	Total
Wikitext-3-raw-v1	1801.35k	3.76k	4.35k	1805.7k
CommonGen	67.39k	4k	1.5k	68.89k
Yelp Reviews	650k	46.5k	50k	700k
E2ENLG	42.1k	4.67k	4.69k	4.79k

Table I compares the parameters of the training sets, validation sets, and test sets for the Wikitext-103-raw-v1, CommonGen, Yelp Reviews, and E2ENLG datasets.

CommonGen dataset is used for model training in commonsense reasoning benchmark tasks, where the goal is to generate a coherent and commonsense sentence given a set of common concept words. The training, validation, and test sets of CommonGen dataset comprise 67,389, 4,018, and 6,042 sentences respectively. Each sample features has three to five key concepts with an average sentence length of 11 words.

Yelp Reviews dataset contains over fifty million reviews. Our study builds upon the data processing work of He [10] on the Yelp Reviews dataset, who chose a keyword set from a thousand sentences that could not cover the entire text extensively. This work constructs a keyword set based on word frequency, eliminating Stopwords and selecting the top 5,000 most frequent words to assure the quality of the set. Exclusions are made for samples without the keywords. For each case, the corresponding target word is a term from the keyword collection, supplemented by additional words randomly chosen from the sample to enhance the target word's diversity.

E2ENLG dataset is an end-to-end text generation dataset for the restaurant industry, with tasks requiring the generation of descriptions based on multiple key-value pairs. E2ENLG dataset provides a crowdsourced corpus of 50k instances, each with a Meaning Representation (MR) shaped by dialogue acts and accompanied by up to 16 natural language references.

2) Evaluation metrics: We employ both automatic and manual evaluations to demonstrate the enhanced generative performance of our model in universal text generation. For automatic evaluation of generation quality, the paper employs Perplexity (PPL), BLEU [31], NIST [32], and DIST [33] as indicators to measure the similarity between the generated text and human references. Higher scores in BLEU and NIST denote that the model is capable of crafting sentences closely resembling those made by humans.

a) PPL low value often indicates better linguistic fluency. c, degraded repetition can also result in a reduced perplexity score. Hence, one should not rely solely on perplexity, but should combine it with other metrics and qualitative analysis.

b) BLEU Lower-order assesses word-level accuracy, whereas higher-order BLEUs can gauge sentence fluency. We adopt both BLEU-2 and BLEU-4 for evaluation.

c) NIST is an improvement over the BLEU method. It introduces the concept of the information quantity of each n-gram. The NIST score is derived by accumulating the information quantity and then dividing by the total number of n-grams in the translation, effectively placing more weight on less frequent words.

d) DIST measures diversity by dividing the number of unique n-grams by the total number of n-grams; a higher value indicates greater diversity in the text.

For the human evaluation component, this study expands upon the framework established by He [10], incorporating an additional dimension, semantic consistency. The comparative performance of the models is assessed across three criteria: semantic consistency, the smoothness of the sentences, and the richness of information conveyed. In pursuit of impartiality, a set of 50 sentences is chosen at random, and five evaluators are enlisted to review the sentences produced by varying models. These evaluators are tasked with delivering their assessments premised on the consistency in meaning, the fluidity of the text, and the depth of information presented. In instances where the evaluators find themselves unable to discern a clear winner, the outcome is declared a draw. Prior to the annotation process, the sequence of sentences is shuffled to eliminate any potential for prejudice.

B. Experimental Details

In terms of model parameters, this study adopts the pre-trained parameters from BART to serve as the model's initial parameters for fine-tuning tasks. For the first two rounds of training, the learning rate is set at 1e-3, and it is reduced by 3e-4 in each successive round until it reaches the threshold of 1e-5. The experiments utilize an Nvidia RTX A5000 GPU, and taking into account the experimental hardware and training efficiency, the batch size is determined to be 64. The study sets the character length limit to 128. In addition, the paper introduces a regularization parameter to curb overfitting during the training phase. The regularization parameter is established at 0.04, informed by the training performance. Across all tasks, the AdamW algorithm is employed for model optimization.

V. Results and Discussion

This section comprehensively discusses the experimental and analysis work we have undertaken, divided into six parts: automatic evaluation, human evaluations, weak correlation constraint analysis, ablation study, hyperparameter analysis, and generating instances. The principal aim of both part A and part B, automatic and human evaluations, is the appraisal of our model's text generation calibre. Analysis touching on weak correlation constraint investigates how the quality of text is influenced when this study's model, as well as benchmark models, face several weakly related constraints. Part D, Ablation study, validates whether the integration of a latent constraint-association generation module in our model enhances the handling of weak correlation constraints. Part E is hyperparameter exploration segment, which discusses the model's performance under varying parameter configurations. Part F, the exemplification analysis showcases instances of text generated by our model.

The model proposed is compared with three of the latest strong baseline fine-grained text generation models (Keyword2Text (K2T) [4], NRP [10], CBART [9]) and one traditional baseline (Pointer[8]).

The Pointer utilizes the Insertion Transformer architecture for hard constraint text generation, which still has room for improvement regarding the quality of output. The CBART, using an Encoder-Decoder structure for non-autoregressive hard constraint generation, has enhanced the quality, yet it struggles with quality reduction under weak correlation constraint conditions, similar to the plug-and-play controlled decoding approach of K2T. The NRP, utilizing a non-residual attention mechanism, betters text generation but risks constraint loss within contexts of weak correlation constraints. Our model is capable of generating text that meets weak correlation constraints, thereby enhancing the quality of generation.

A. Automatic Evaluation Results and Analysis

This experiment evaluates the text generation quality of the improved model versus the baseline models on three test sets: CommonGen, Yelp Reviews, and E2ENLG.

As shown in Table II, on the Common Gen test set, our model is slightly inferior to the K2T model in terms of NIST scores, but demonstrates a distinct advantage in BLEU and DIST scores. This is due to the fixed mapping from keywords to text in the K2T model, which thus offers relatively poor text diversity. The performance of our model on DIST-4 is comparable to that of NRP, but slightly superior to NRP on DIST-2. This suggests that the improved model can exhibit more granular controllability when generating high-quality sentences.

TABLE II. AUTOMATIC EVALUATION EXPERIMENTS SCORES COMPARISON

Datasets	Models	BELU↑		NIST↑		DIST↑		PPL↓	Len
		B-2	B-4	N-2	N-4	D-2	D-4		
CommonGen	K2T	19.5	4.25	7.52	7.63	0.72	0.95	25.27	7.1
	CBART	17.44	5.34	5.01	3.15	0.71	0.98	32.62	5.7
	NRP	20.15	7.28	7.43	7.59	0.74	0.99	24.01	6.3
	Pointer	10.18	1.77	2.23	2.4	0.45	0.9	72.82	7.2
	Ours	22.92	9.21	7.22	7.36	0.78	0.99	22.73	6.8
Yelp Reviews	K2T	25.6	8.25	7.53	7.61	0.69	0.89	31.19	17.2
	CBART	18.41	7.4	2.54	2.63	0.48	0.94	50.61	15.7
	NRP	23.52	9.11	8.47	8.66	0.74	0.91	35.78	20.3
	Pointer	11.48	2.46	2.14	2.16	0.35	0.68	101.8	27.2
	Ours	26.25	9.52	8.42	8.51	0.82	0.95	40.23	16.8
E2ENLG	K2T	27.1	9.1	8.44	8.65	0.78	0.92	25.79	12.4
	CBART	20.22	8.06	3.46	3.67	0.82	0.91	34.21	13.4
	NRP	26.33	9.23	8.42	8.7	0.81	0.98	20.18	15.3
	Pointer	12.65	2.98	2.39	2.43	0.55	0.83	60.84	14.2
	Ours	29.01	9.78	8.65	8.81	0.86	0.99	19.86	16.1

[a.] Note: Bold numbers indicate the optimal values under this dataset and evaluation method. B-2, B-4 represent the BLEU evaluation method using 2-gram, 4-gram, respectively, with NIST and DIST following a similar pattern.

Our study further evaluated the performance of our model on the Yelp Reviews test set. The results indicated that our model is comparable to the highly-rated NRP model in terms of NIST score, while it also achieves the highest scores in BELU and DIST metrics. This reflects our model deals with the constraints on latent variables, as well as its inclusion of some extraneous noise in the prompt transformation process, impacting its understanding and generation capabilities. Owing to the learning ability of latent variables, our model still surpasses baseline models in terms of generation quality.

On the E2ENLG test set, our model scored the highest across all evaluation metrics. It exceeded the CBART model by 0.04 points in DIST score and the K2T model by 0.2 points in NIST-2 score. This suggests that the model also slightly outperforms baseline models in terms of text diversity and coherence. According to the assessment data from CBART and Pointer, it can be observed that the quality of non-autoregressive generation is slightly lower than that of autoregressive generation.

B. Human Evaluation Analysis

Table III shows that our model outperforms the baselines in terms of semantic consistency, fluency, and informativeness, which is even comparable to human levels in sentence fluency and semantic consistency. In text fluency, our model slightly exceeds the baseline models, and considerably surpasses the baselines in both semantic consistency and sentence informativeness. However, our model still falls behind humans in terms of sentence informativeness, this is attributed to the model's excessive focus on text fluency, leading to the generation of sentences that are shorter and less informative than those referenced by humans.

In summary, the evaluations demonstrate that the improved model excels in text generation quality, surpassing other baseline models. This also validates the superior performance and generalization capability of our model in the domain of controlled text generation.

C. Weak Correlation Constraint Analysis

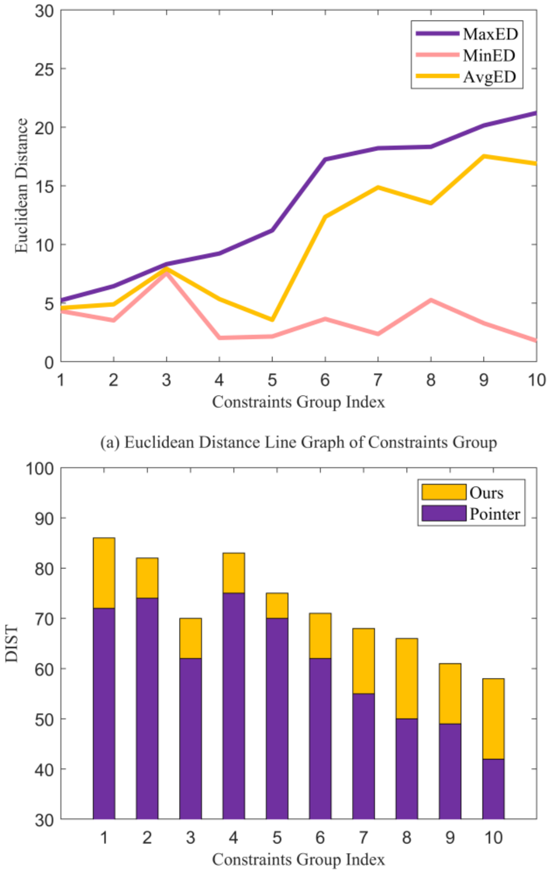
This section is dedicated to analyzing the impact of weak correlation constraints on the model. Existing pre-trained language model is black-box model, and the features they learn from constraints lack interpretability. Therefore, we assess the strengths and weaknesses of the relationships between constraints based on their Euclidean distance, allowing for a more precise measurement of the constraints' impact on the model. Compared to simply measuring the strength of relationships between constraints based on co-occurrence frequency, Euclidean distance can more effectively evaluate the similarity of features among constraints, making this method more comprehensive and accurate.

TABLE III. HUMAN EVALUATION SCORES COMPARISON: %

Metrics	**Model A won**		**Tied**	**Model B won**	
Semantic Consistency	Ours	63.5	15.2	21.3	K2T
	Ours	55.6	10.3	34.1	NRP
	Ours	32.6	21.4	46	Human
Sentence Fluency	Ours	47.5	20.3	32.2	K2T
	Ours	42.7	14.8	42.5	NRP
	Ours	30.1	29.7	40.2	Human
Sentence Informativeness	Ours	71.6	10.4	18	K2T
	Ours	64.3	7.5	28.2	NRP
	Ours	23.5	12.9	63.6	Human

[b.] Note: “Consistency” represents which sentence is more consistent; “Fluency” stands for which sentence is more fluent; “Informativeness” indicates which sentence is more informative?

Within an individual sample, for a given set of constraints $C=\{c_m, m\in ℕ_+\}$, $((m-1)*m)/2$ Euclidean distances can be computed. To better evaluate the model under the influence of constraints, we examined the maximum Euclidean distance (MaxED), the minimum Euclidean distance (MinED), and the average Euclidean distance (AvgED) separately. MaxED can assess the model's capability to handle weak correlation constraints, while MinED can evaluate its ability to deal with strong constraints. These two values also provide insights into the dispersion among the constraints within the set. AvgED offers a more comprehensive metric, presenting an overview of the overall distribution of the constraint set.



(a) Euclidean Distance Line Graph of Constraints Group

(b) DIST Score Bar Chart

Fig. 5. Weak correlation constraint analysis chart.

Subfigure (a) of Fig. 5 reveals an inverse relationship between MaxED among constraints and the quality of sentence generation, indicating that the weaker the relevance of the constraints, the more challenging it is for the model to generate high-quality sentences. Furthermore, if MinED is close to AvgED, it suggests that the constraints are generally weakly correlated, and the quality of the sentences generated by the model is primarily influenced by AvgED. However, when MinED greatly differs from both MaxED and AvgED, strongly related constraints have a minor impact on the quality of sentence generation by the model. According to subfigure (b) of Fig. 5, when the constraints are weaker in correlation, our improved model consistently outperforms the comparison model Pointer in terms of generation quality, and its rate of decline is also slower than that of Pointer. The analysis verifies that the model ensures quality generation when facing weakly related constraints and that the improvement method can effectively handle weak correlation constraints.

D. Ablation Study

Reflecting on the analysis of weak correlation constraints from the, it’s evident that these constraints largely influence the quality of the model's output. To substantiate the preceding section advancements of our model in managing weak correlation constraints, we conducted an ablation study. The study was structured such that each set of constraints included two strongly related constraints, with a progressive addition of weak correlation constraints to discern the disparity in output quality between models applying the LCCG module and those without it, referred to as Non-LCCG.

According to the data presented in Table IV, the assessments indicate enhancements in models incorporating the LCCG module compared to those which do not include it. More specifically, the inclusion of LCCG led to an increase of 4 to 7 percentage points in BLEU-2 scores, a rise of 2 to 3 points in NIST-2, and a significant enhancement of 15 to 20 percentage points in DIST-2. The evidence suggests that with the addition of weak correlation constraints, the gap in generative quality between the two approaches diminishes.

TABLE IV. ABLATION STUDY SCORES COMPARISON

Count	Models	Automatic Evaluation Metrics		
		BLEU	NIST	DIST
0	Non-LCCG	18.47	4.32	0.76
	LCCG	35.46	7.13	0.98
1	NON-LCCG	12.25	3.67	0.69
	LCCG	26..16	6.21	0.87
2	NON-LCCG	10.07	2.93	0.53
	LCCG	15.24	5.21	0.75
3	NON-LCCG	4.16	2.05	0.41
	LCCG	8.2	4.33	0.56

[c] Note: The count refers to the number of newly added weak correlation constraints. BLUE, NIST, and DIST indicate that the evaluation method uses 2-grams.

The experimental outcomes emphatically confirm the noteworthy efficacy and superiority of our proposed technique in handling weak correlation constraints. The research, underscored by its experimental design and data interpretation, verifies the method’s precision and robustness when confronting issues related to weak correlation constraints.

E. Hyperparameter Analysis

Hyperparameters are significantly influential in both the performance and training process of the model. After completing training in the initial phase, the model has adeptly learned the art of autoregressive text generation and the ability to infer text that adheres to latent variable constraints.

Therefore, in the second phase, we also fine-tuned the loss weight associated with the latent variable constraints. Excessively high loss from latent variable constraints may hinder the language model's ability to find the optimal solution for incorporating latent variable constraints, while too low a loss might fail to ensure that the generated text complies with the constraints. As depicted in Fig. 6, based on feedback from experimental results and our experience, we adjusted the latent variable constraint loss weight μ to 0.4 and η to 0.6 to strike a better balance.

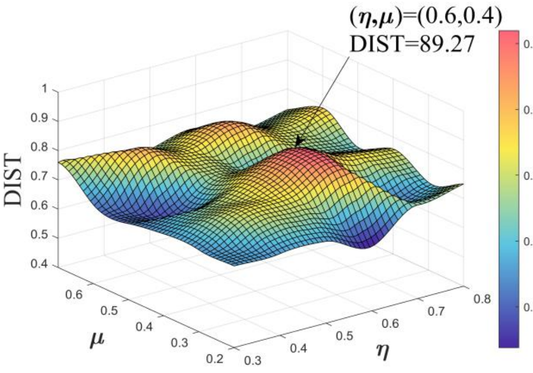


Fig. 6. Hyperparameter analysis.

F. Instance Analysis

In Table V, Case 1, “Basketball” “Forest” and “Paper” are weak correlation constraints. The K2T model generates semantically inconsistent sentences, while the NRP model focuses on the constraints of “Forest” and “Paper” which are Constraints with a strong correlation, neglecting “Basketball” However, our model using latent variables, finds the potential connections between these three constraints. It interprets “Forest” as the venue and “Paper” as paper packaging, and uses “Basketball” to link them together.

TABLE V. GENERATING INSTANCES WITH WEAK CORRELATION CONSTRAINTS

Constraint Case	Models	Instance Output
Basketball, Forest, Paper	K2T	When playing basketball in the forest, I use paper to achieve floating flight.
	NRP	In the forest, I found a paper airplane, which made me feel the flying green leaves and fresh air.
	Ours	We were playing basketball in the forest when we accidentally spilled our paper wrapped lunch.
Phone, Rocket, Eat, Floor	K2T	Eating rocket shaped mobile phones on the floor feels really delicious.
	NRP	The floor of the Rockets is very smooth, and the players are eating cake while making phone calls
	Ours	While I was eating, my phone suddenly slipped from my hand like a rocket and fell onto the floor.
Apple, Mouse, Table, Doll	K2T	I placed a mouse and a doll next to the apple, hoping that they could entertain each other.
	NRP	The Apple Mouse is a mouse that can be used with the iPad, iPhone and iPod touch. It has an apple-shaped button on top of the table where it sits in your hand.
	Ours	The doll was sitting on the table with an apple beside it, while a mouse scurried across the floor.

[d] Note: In the output sentence, words that are constraints are highlighted in bold.

Similarly, in Case 2 of Table V, the four constraints have a weak correlation. The K2T model poorly handles the constraints, resulting in sentences with illogical constructions. The NRP model still focuses on the more related constraints, leading to sentences with less information. However, our model did not encounter such issues, instead, it makes a reasonable arrangement based on the latent characteristics of these four constraints, forming a semantically coherent sentence.

In Case 3 of Table V, NRP primarily focuses on “Apple” and “Mouse” generating a sentence related to the technology field, consequently overlooking “Doll.” K2T considers constraints more comprehensively than NRP but also experiences issues with constraint loss. In contrast, our proposed model didn't lose any constraints and didn't simply interpret “Apple” and “Mouse” as the company brand and technology product, respectively. By thoroughly considering the potential relationships between these four constraints, an optimal solution was found, and a high-quality sentence was successfully generated while satisfying all the constraints.

VI. Conclusion

In our study, we conduct an extensive study on the problem of hard-constrained controllable text generation, and propose a novel latent variable constraint-controllable strategy. The strategy effectively deals with the existence of multiple weak correlation constraints in the text generation process from the language model. Through a series of experiments, the results confirm that the strategy significantly improves the quality of controllable text generation and satisfies the weak correlation constraints.

This study makes significant progress in the direction of hard-constrained controlled text generation, there are still many areas to be explored and deepened, such as the excessive decoding time. In our future work, we intend to further optimize the latent variable constraint-controllable strategy and endeavor to adjust the initialization and capture of latent variables to more accurately reveal the relations between constraints and generate constraint-compliant text more quickly and efficiently.

Acknowledgment

National Natural Science Foundation of China (Project Nos. 62076167 And 61772020).

Henan Key Scientific Research Project of Higher Education Institutions (Project Nos. 24A520058, 24A520060, And 23A520022).

Henan Postgraduate Education Reform and Quality Improvement Project (Project No. YJS2024AL053)

References

[1] M Lewis, Y Liu, N Goyal, M Ghazvininejad, A Mohamed, O Levy, ... & 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. Association for Computational Linguistics, pp. 7871–7880, 2020.

[2] A Radford, J Wu, R Child, D Luan, D Amodei, & I Sutskever, “Language models are unsupervised multitask learners,” OpenAI blog, vol. 1, no. 8, pp. 9, 2019.

[3] C Raffel, N Shazeer, A Roberts, K Lee, S Narang, M Matena, ... & P. J. Liu, “Exploring the limits of transfer learning with a unified text-to-text transformer,” Journal of Machine Learning Research, vol. 21, no. 1, pp. 5485-5551, 2020.

[4] D Pascual, B Egressy, C Meister, R Cotterell, & R Wattenhofer, “A plug-and-play method for controlled text generation,” In Findings of the

Association for Computational Linguistics: EMNLP, Association for Computational Linguistics, pp. 3973-3997, 2021.

[5] C. Hokamp, Q. Liu, "Lexically constrained decoding for sequence generation using grid beam search," In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pp. 1535–1546, 2017.

[6] M Post, D Vilar, "Fast lexically constrained decoding with dynamic beam allocation for neural machine translation," In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 1314-1324,2018.

[7] P Anderson, B Fernando, M Johnson, S Gould, "Guided open vocabulary image captioning with constrained beam search," In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 936–945, 2017.

[8] Y Zhang, G Wang, C Li, Z Gan, C Brockett, & B Dolan, "POINTER: Constrained progressive text generation via insertion-based generative pre-training," in Proc of the 2020 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2021, pp. 3045-3059.

[9] X He, "Parallel refinements for lexically constrained text generation with BART," In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, pp. 8653-8666, 2021.

[10] F Carlsson, J Öhman, F Liu, S Verlinden, J Nivre, & M Sahlgren, "Fine-grained controllable text generation using non-residual prompting," In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, pp. 6837-6857, 2022.

[11] S Dathathri, A Madotto, J Lan, J Hung, E Frank, P Molino, ... & R Liu. "Plug and play language models: a simple approach to controlled text generation," International Conference on Learning Representations, ICLR, 2020.

[12] A Chan, Y.S. Ong, B Pung, A Zhang, & J Fu, "CoCon: A self-supervised approach for controlled text generation," International Conference on Learning Representations, ICLR, 2021.

[13] B Krause, A.D. Gotmare, B Mccann, N.S. Keskar, & N.F. Rajani, "GeDi: Generative discriminator guided sequence generation," In Findings of the Association for Computational Linguistics: EMNLP, Association for Computational Linguistics, pp. 4929-4952, 2021.

[14] K Yang, D Klein, "FUDGE: Controlled text generation with future discriminators," in Proc of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 3511-3535, 2021.

[15] N Miao, H Zhou, L Mou, R Yan, & L Li, "CGMH: Constrained sentence generation by metropolis-hastings sampling," In Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, AAAI Press, pp. 6834-6842, 2019.

[16] X Li & 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, Association for Computational Linguistics, pp. 4582-4597, 2021.

[17] B Lester, R Al-Rfou & N Constant, "The power of scale for parameter-efficient prompt tuning," In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, pp. 3045-3059, 2021.

[18] X Han, W Zhao, N Ding, Z Liu, & M Sun, "PTR: Prompt tuning with rules for text classification," In AI Open, vol. 3, pp. 182-192, 2022.

[19] X Zou, D Yin, Q Zhong, M Ding, Z Yang, & 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, Association for Computing Machinery, pp. 2450-2460, 2021.

[20] K Yang, D Liu, W Lei, B Yang, M Xue, B Chen, & Xie, "Tailor: A Soft-Prompt-Based Approach to Attribute-Based Controlled Text Generation," In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, pp. 410-427, 2023.

[21] Ł Kaiser, A Roy, A Vaswani, N Parmar, S Bengio, J Uszkoreit, & N Shazeer, "Fast decoding in sequence models using discrete latent variables," In International Conference on Machine Learning, PMLR, pp. 2390-2399, 2018.

[22] R Shu, J Lee, H Nakayama, & K Cho, "Latent-variable non-autoregressive neural Machine Translation with Deterministic Inference Using a Delta Posterior," In Proceedings of the AAAI Conference on Artificial Intelligence, AAAI Press, pp. 8846-8853, 2020.

[23] X Ma, C Zhou, X Li, G Neubig, & E Hovy, "FlowSeq: Non-autoregressive conditional sequence generation with generative flow," In Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pp. 4282-4292, 2019.

[24] Y Bao, S Huang, T Xiao, D Wang, X Dai, & J Chen, "Non-autoregressive translation by learning target categorical codes," In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, pp. 5749-5759, 2021.

[25] Y Bao, H Zhou, S Huang, D Wang, L Qian, X Dai, …, & L Li, "latent-GLAT: Glancing at Latent Variables for Parallel Text Generation," In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, pp. 8398-8409, 2022.

[26] A.V.D Oord, O Vinyals, K Kavukcuoglu, "Neural discrete representation learning," In Proceedings of the 31st International Conference on Neural Information Processing Systems, Curran Associates, pp. 6309-6318, 2017.

[27] B Wei, M Wang, Hao Zhou, J Lin, & X Sun, "Imitation learning for non-autoregressive neural machine translation," In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, pp. 1304-1312, 2019.

[28] B Lin, W Zhou, M Shen, P Zhou, C Bhagavatula, Y Choi, & X Ren, "CommonGen: A constrained text generation challenge for generative commonsense reasoning," Findings of the Association for Computational Linguistics: EMNLP, Association for Computational Linguistics, pp. 1823-1840, 2020.

[29] W.S. Cho, P Zhang, Y Zhang, X Li, M Galley, C Brockett, …, & J Gao, "Towards coherent and cohesive long-form text generation," In Proceedings of the First Workshop on Narrative Understanding, Association for Computational Linguistics, pp. 1-11, 2019.

[30] O Dušek, J Novikova, & V Rieser. "Evaluating the state-of-the-art of end-to-end natural language generation: the E2ENLG challenge," In Computer Speech & Language, vol. 59, pp. 123-156, 2020.

[31] K Papineni, S Roukos, T Ward, & W Zhu, "Bleu: A method for automatic evaluation of machine translation," In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, pp. 311-318, 2002.

[32] G Doddington. "Automatic evaluation of machine translation quality using n-gram co-occurrence statistics," In Proceedings of the Second International Conference on Human Language Technology Research, Margan Kaufmann, pp. 138-145, 2002.

[33] J Li, M Galley, C Brockett, J Gao, & B Dolan, "A diversity-promoting objective function for neural conversation models," In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, pp. 110-119, 2016.