MCPG: A Flexible Multi-Level Controllable Framework for Unsupervised Paraphrase Generation

Yi Chen[1,2], **Haiyun Jiang**[*], **Rui Wang**[1,2], **Lemao Liu, Shuming Shi, and Ruifeng Xu**[1,2,3*]
[1]Harbin Insitute of Technology, Shenzhen, China
[2]Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies
[3]Peng Cheng Laboratory, Shenzhen, China
yichennlp@gmail.com,ruiwangnlp@outlook.com,xuruifeng@hit.edu.cn

Abstract

We present MCPG: a simple and effective approach for controllable unsupervised paraphrase generation, which is also flexible to adapt to specific domains without extra training. MCPG is controllable in different levels: local lexicons, global semantics, and universal styles. The unsupervised paradigm of MCPG combines factual keywords and diversified semantic embeddings as local lexical and global semantic constraints. The semantic embeddings are diversified by standard dropout, which is exploited for the first time to increase inference diversity by us. Moreover, MCPG is qualified with good domain adaptability by adding a transfer vector as a universal style constraint, which is refined from the exemplars retrieved from the corpus of the target domain in a training-free way. Extensive experiments show that MCPG outperforms state-of-the-art unsupervised baselines by a margin. Meanwhile, our domain-adapted MCPG also achieves competitive performance with strong supervised baselines even without training.

1 Introduction

Paraphrase generation aims to restate a given sentence in a way that conveys the same semantic meaning but uses a different expression form. Paraphrasing results benefit lots of downstream tasks, such as text classification (Wang et al., 2022; Chen et al., 2019a), question answering (Dong et al., 2017; Cheng et al., 2021), and semantic matching (Chen et al., 2022). Traditional approaches (Prakash et al., 2016; Gupta et al., 2018) require supervised training on large parallel corpora. However, it is expensive for manual annotation. Thus, unsupervised methods are welcomed in the absence of annotated datasets (Bowman et al., 2016; Wieting et al., 2017; Miao et al., 2019; Hegde and Patil, 2020; Meng et al., 2021; Shen et al., 2022).

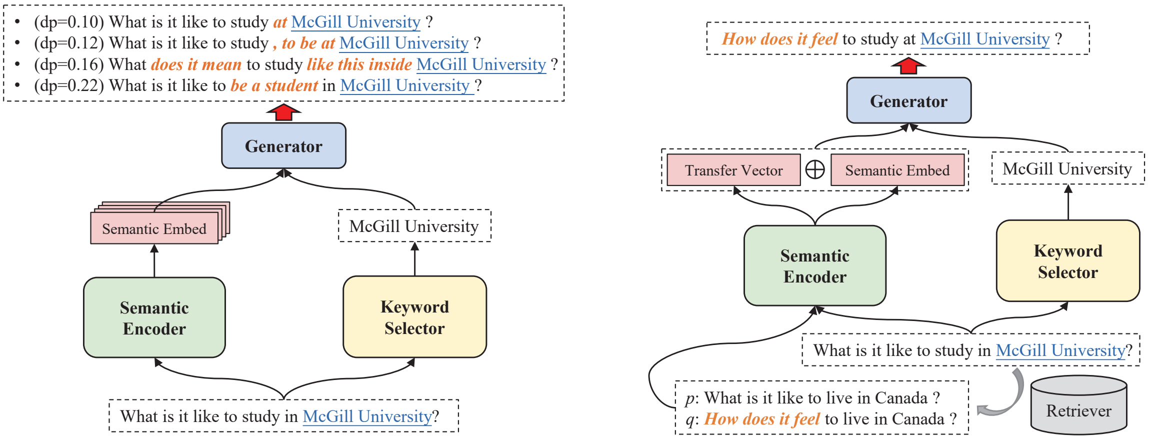
In this paper, we focus on *controllable* unsupervised paraphrase generation, which is a promising direction and some progress has been achieved. However, most existing works are either limited by the supervised settings (Iyyer et al., 2018; Chen et al., 2019b; Kazemnejad et al., 2020; Huang and Chang, 2021; Bandel et al., 2022) or mainly explore to control generation from a single perspective (e.g., syntatic diversity) in the unsupervised condition (Huang and Chang, 2021). *Multi-level* controllable approaches for unsupervised generation are still not well explored. To this end, we propose a simple and effective framework for unsupervised paraphrase generation called **M**ulti-level **C**ontrollable **P**araphrase **G**enerator (MCPG), as shown in Figure1. Overall, MCPG *is controllable in three levels*, i.e., global semantics, local lexicons, and universal styles. This approach is motivated by a commonly observed phenomenon that the human paraphrasing process is decomposable at different levels. For example, given a sentence, we first read it through to capture the general semantic meaning (*global semantics*). Then we read over to identify factual keywords like names of persons and locations, which should be protected from tampering (*local lexicons*). Furthermore, if there are any exemplars for reference, we may optionally get inspirations from them regarding word editing or sentence restructuring (*universal styles*). Finally, we rewrite the remaining parts as much as possible while preserving the original meaning.

Figure1(a) shows the basic paradigm of MCPG tailored for the unsupervised scenario. We employ T5 (Raffel et al., 2020) as the Generator backbone to integrate both global semantic and local lexical constraints. The whole model is trained by reconstructing input text, which is compatible with the denoising objective for pretraining T5.

The global semantics is controlled by the semantic embedding. We show that *enabling dropout in-*

[*]Corresponding Authors

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 5948–5958
December 7-11, 2022 ©2022 Association for Computational Linguistics


(dp=0.10) What is it like to study at McGill University ?
(dp=0.12) What is it like to study , to be at McGill University ?
(dp=0.16) What does it mean to study like this inside McGill University ?
(dp=0.22) What is it like to be a student in McGill University ?
Generator
Semantic Embed
McGill University
Semantic Encoder
Keyword Selector
What is it like to study in McGill University?
How does it feel to study at McGill University ?
Generator
Transfer Vector
⊕
Semantic Embed
McGill University
Semantic Encoder
Keyword Selector
What is it like to study in McGill University?
p: What is it like to live in Canada ?
q: How does it feel to live in Canada ?
Retriever

(a) MCPG-basic

(b) MCPG-adapt

Figure 1: Model overview. (a) MCPG-basic is the basic framework tailored for the unsupervised scenario, which controls the ***global semantics*** through the semantic embedding and the ***local lexicons*** through the pre-specified factual keywords. The semantic embedding is disturbed by the dropout mechanism inside the Semantic Encoder with a probability dp, which controls the output diversity. (b) MCPG-adapt is the refined framework which controls the ***universal style*** through a transfer vector. This version is designed to increase the model's domain adaptability when annotated exemplar paraphrases from the target domain are available in a *training-free* way.

side the Semantic Encoder during inference works well in creating different semantic embeddings for the same sentence while keeping the semantic meaning. This provides a helpful guidance for MCPG to generate paraphrases with various expression forms. Besides, we can further *control the diversity degree* by tuning the dropout probability and *filter bad semantic embeddings* according to the cosine similarity between them and the dropout-disabled standard embedding of the same input.

The local lexicons are controlled by the keywords. Previous works (Zeng et al., 2019; Su et al., 2021; Yang et al., 2022) pre-specify keywords using either rule-based or model-based methods, but neglect the importance of factual entities, which are actually the unchangeable words we must copy during paraphrasing. Towards this, we adopt Named Entities Recognition tools to identify factual entities like names of persons and locations as keywords. Fixing this kind of keywords turns out to be an effective way for improving paraphrase quality.

In Figure 1(b), we offer an alternative option to constrain the universal style of outputs, which helps MCPG adapt to the target domain efficiently without training.

The universal styles are controlled by a transfer vector, which implicitly encodes the lexical and structural mapping mode between a pair of exemplar paraphrases retrieved from the training set of target domain. This mode is similar to style transfer (Riley et al., 2021). We derive the transfer vector via subtracting the semantic embeddings of two parallel paraphrases, which requires no extra training. By adding the transfer vector to the semantic embedding of the input sentence, our model is inspired to paraphrase in a tone more consistent with the target domain.

Our contributions are to: (1) present a multi-level controllable framework for unsupervised paraphrase generation which is flexible for domain adaptation; (2) propose a dropout-diversified semantic encoding method to control both the global semantics and output diversity; (3) explore the importance of factual entities to control local lexicons; (4) use the transfer vector based on target domain exemplars to control universal styles in an efficient way that requires no extra training.

2 Related Work

Supervised Approaches Recent works on supervised paraphrase generation mainly focus on improving paraphrase quality using neural models (Prakash et al., 2016). An important direction is to make the generation process more controllable either from a single perspective or in a hybrid way. One approach is to seek guidance from exemplars (Iyyer et al., 2018; Chen et al., 2019b; Goyal and Durrett, 2020; Kazemnejad et al., 2020; Yang et al., 2021). Another approach is to identify keywords to control lexicons (Zeng et al., 2019; Su et al., 2021). Fu et al. (2019) use predicted neighbors of source words to form target paraphrase. Li et al. (2019) decompose paraphrasing transformation into different granularity levels. Other hybrid methods include (Yang et al., 2022; Bandel et al., 2022). Although these models controlled by various constraints demonstrate promising performance, they require parallel data for training, which is labor-intensive and inextensible to other domains.

Unsupervised Approaches There are two main lines for unsupervised paraphrase generation: optimization-oriented and pivot-based methods. The goal of *optimization-oriented methods* (Miao et al., 2019; Liu et al., 2020; Siddique et al., 2020) is to find the optimal paraphrase by optimizing an objective function that considers semantic fidelity, expression diversity, language fluency, etc. Nonetheless, designing reliable metrics to automatically evaluate the quality of paraphrases has been a long-standing problem. *The pivot-based approaches* aim to first represent the input sentence by some kind of pivot and then reconstruct the input from it (Cai et al., 2021). A popular pivot is the latent semantic representation learned by a variational autoencoder (VAE) (Bowman et al., 2016; Roy and Grangier, 2019), which is mathematically interpretable but hard to train intrinsically. Another method is to use other languages as a pivot (Wieting et al., 2017; Lapata et al., 2017; Wieting and Gimpel, 2018; Guo et al., 2019), which relies on external machine translation systems. The works most similar to us explore to take advantages from large-scale language models and adopt corrupted sequences as paraphrasing pivot (Hegde and Patil, 2020; Niu et al., 2021; Meng et al., 2021). However, these methods lack a global semantic guidance due to the absence of the complete input sentence. In addition to the above methods, Huang and Chang (2021) take the first step towards syntactically controllable unsupervised paraphrasing by manipulating the embedding of the potential constituency parse tree. Whereas, it is time-consuming for constituent parsing, which may increase time complexity for paraphrase generation.

3 Model

3.1 Overview

Figure1 illustrates the overall architecture of MCPG. The basic framework in Figure1(a) denoted as MCPG-basic is built for the unsupervised condition, which consists of three modules, i.e., Dropout-enabled Semantic Encoder, Factual Keyword Selector and Controllable Generator. Given a source sentence, we first encode its semantic meaning into a dense vector $\boldsymbol{x}$ via the Semantic Encoder, which performs the dropout operation internally to obtain diverse semantic embeddings. In addition to using the dropout disturbed embedding as a global semantic constraint, we also provide the Generator with a few keywords $\mathcal{W} = \{w_i\}_{i=1}^{M}$ as a local lexical constraint to ensure that factual entities are preserved. Figure1(b) shows how to further control the universal expression style with the adapted framework MCPG-adapt when annotated samples from the target domain are available. We seek guidance from the transfer vector $\boldsymbol{v}$ based on parallel exemplars, which are retrieved by the Exemplar Provider from the training set of target domain. By adding the semantic embedding $\boldsymbol{x}$ with the transfer vector $\boldsymbol{v}$, the Generator is allowed to produce paraphrases that are more adaptive with the target domain. Note, the style constraint based on annotated exemplars is not involved during training. We only use it to increase the model's domain adaptability in a plug-and-play way.

3.2 Global Semantic Constraint

The Semantic Encoder is responsible for deriving an embedding for the input sentence, which is helpful for both *preserving global semantics* of the original sentence and *increasing expression diversity*.

SimCSE (Gao et al., 2021) has been proved to be effective for sentence representation. We follow them to perform unsupervised contrastive learning upon BERT and use it as the backbone for our Semantic Encoder. The last-layer [CLS] representation is taken as the semantic embedding. Note, the semantic similarity between two sentences can be measured by the cosine distance of their embeddings. This is helpful for filtering bad semantic embeddings (see the last paragraph of this section) and retrieving exemplar (see Sec 5.1).

Dropout is a useful technique for deep network regularization. Conventionally, we only turn it on during training and disable it during inference. Here, we show that *enabling dropout during inference* is helpful for producing embeddings with subtle difference for the same input while keeping the semantics, thus guiding the model to generate more diverse paraphrases.

Given an input sentence, we obtain the semantic embedding $\boldsymbol{x}$ from the Semantic Encoder by

$$\boldsymbol{x} = \text{SemEncoder}(x, dp, drop = True), \quad (1)$$

where dp denotes the dropout probability and $drop = True$ means the we enable dropout during inference. On the other hand, the conventional standard embedding $\bar{\boldsymbol{x}}$ of the same input is derived without dropout by setting $drop = False$. We only use the build-in dropout mechanism inside BERT and do not add any additional dropout operations.

We assume that the standard embedding $\bar{\boldsymbol{x}}$ conveys the full semantic meaning of input sentence without losing any information about the original expression form. Whereas, $\boldsymbol{x}$ is continuously disturbed by the dropout operation that acts on the intermediate layers of the encoder. Generally, the higher the dropout probability dp, the further away the disturbed embedding $\boldsymbol{x}$ from the standard embedding $\bar{\boldsymbol{x}}$, and the more diverse the generated texts. Along this line, we can tune dp to achieve a *trade-off between semantic preservation and expression diversity*. Besides, we *further control the dropout behaviour* by checking the similarity between $\boldsymbol{x}$ and $\bar{\boldsymbol{x}}$. Only the semantic embeddings $\boldsymbol{x}$ with a cosine similarity larger than a threshold $\lambda(= 0.75)$ will be used for generating paraphrase candidates.

3.3 Local Lexical Constraint

Dense embeddings are effective for encoding global semantics of sentences. However, like many previous models, totally relying on a dense semantic embedding for paraphrasing may *fail to retain important keywords*, such as factual entities (e.g., persons and locations), which we do not hope to change during paraphrase generation. Besides, using a few words as local clues tends to be helpful for producing semantically consistent paraphrases when the input text is long. Therefore, we utilize entity-based keywords to make the model more controllable in the lexical level.

We use the Keyword Selector to extract keywords. We first adopt the high-accuracy version of TexSmart fine-grained NER tool (Zhang et al., 2020; Liu et al., 2021) to recognize a set of named entities $\mathcal{W}_1$ from the input sentence. Then we remove the stop words and randomly sample the remaining non-entity keywords $\mathcal{W}_2$ every $3 \sim 10$ words. $\mathcal{W}_1$ and $\mathcal{W}_2$ are combined to form the final keyword set $\mathcal{W} = \{w_i\}_{i=1}^{M}$. The subscripts i corresponds to the order in which the specific keyword appears in the input sequence. Note that the selected keywords only account for a small proportion of the whole sentence, which does not hurt the output diversity too much.

3.4 Paraphrase Generator

We employ T5 (Raffel et al., 2020) as the Generator backbone, which is based on an encoder-decoder structure and shows strong performance in the text-infilling task (Gao et al., 2022). We follow the standard denoising objective used in the pretraining stage of T5 to reconstruct the input sentence from a list of keywords $\mathcal{W}$. The only difference is that we introduce the semantic embedding $\boldsymbol{x}$ as a global semantic constraint. The generation function is defined as

$$y = \text{GenDecoder}(\boldsymbol{x} \oplus \text{GenEncoder}(\mathcal{W})). \quad (2)$$

where GenEncoder and GenDecoder denote the encoder and decoder of the T5-based Generator respectively and $\oplus$ denotes concatenation.

To construct the input sequence for the GenEncoder, we separate the keywords $\mathcal{W}$ by sentinel tokens [BLANK_i], each of which stands for a consecutive span of missing content. For example in Figure 1(a), the keywords only include "McGill University", thus the input sequence should be

"[BLANK_0] McGill University [BLANK_1] "

Denote the last layer hidden states of the GenEncoder as $[\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_S]$, where $\boldsymbol{h}_i \in \mathbb{R}^H$ and S is the input sequence length. We concatenate the semantic embedding $\boldsymbol{x} \in \mathbb{R}^H$ with them to form $[\boldsymbol{x}, \boldsymbol{h}_1, ..., \boldsymbol{h}_N]$. These vectors are then attended by the Generator decoder in each layer as memory vectors to guide paraphrase generation.

The goal of the GenDecoder is to fill in the blanks (e.g., [BLANK_0] and [BLANK_1] in the example input). That is, the decoder only predicts the missing contexts and the output positions for the given keywords, which are indicated by placeholders [KEY_i]. For the above example input, the target sequence should be

"What is it like to study [KEY_0] ? "

where [KEY_0] indicates the input keyword "McGill University". By replacing the placeholders [KEY_i] with corresponding keywords, we finally obtain the paraphrase sentence y.

Training Objective The whole model is trained in an unsupervised way, where the target sentence

y is exactly the input sentence x itself. The loss function over the whole corpus $\mathcal{D}$ is defined as

$$\mathcal{L} = -\sum_{x \in \mathcal{D}} \log p(x \mid \boldsymbol{x}, \mathcal{W}). \tag{3}$$

3.5 Universal Style Constraint

MCPG-basic trained on the non-parallel corpus is already qualified for paraphrase generation. In this section, we refine it to MCPG-adapt with an additional style constraint when annotated exemplars from target domain are accessible in a training-free way, as shown in Figure 1(b).

Given an input x, we first retrieve the K most similar paraphrase pairs $\{p_k, q_k\}_{k=1}^{K}$ from the training corpus, where p_k and q_k denote the parallel paraphrases. The selection criterion is based on the cosine similarity between the standard semantic embedding of x and the average embedding of $\{p_k, q_k\}$ derived from the Semantic Encoder (3.2). The selected pairs are taken as exemplars to guide paraphrase generation.

Denote the semantic embeddings for a pair of exemplar paraphrases p, q as $\boldsymbol{p}, \boldsymbol{q} \in \mathbb{R}^H$, and the input sentence x as $\boldsymbol{x} \in \mathbb{R}^H$. We model the difference of universal style between p and q by a transfer vector

$$\boldsymbol{v} = \frac{\boldsymbol{q}}{\|\boldsymbol{q}\|} - \frac{\boldsymbol{p}}{\|\boldsymbol{p}\|}, \tag{4}$$

where $\|\cdot\|$ denotes L2 norm. Note that, we assume p is more syntactically similar to x than q based on the Levenshtein distance between corresponding POS tag sequences. Otherwise we just swap $\boldsymbol{p}$ and $\boldsymbol{q}$ in Equation (4). Then we transfer $\boldsymbol{x}$ to the paraphrase embedding $\boldsymbol{y}$ by

$$\boldsymbol{y} = (\frac{\boldsymbol{x}}{\|\boldsymbol{x}\|} + \boldsymbol{v}) \times \|\boldsymbol{x}\|. \tag{5}$$

Finally, we decode the paraphrase sequence y by

$$y = \text{GenDecoder}(\boldsymbol{y}, \text{GenEncoder}(\mathcal{W})), \tag{6}$$

where we just replace the semantic embedding $\boldsymbol{x}$ in Equation (2) with the transferred embedding $\boldsymbol{y}$.

4 Experimental Setup

4.1 Datasets

There are two variants of our proposed model, i.e., the unsupervised MCPG-basic and the domain-adapted MCPG-adapt. We evaluate MCPG-basic on three widely-used datasets: Quora[1], MSCOCO (Lin et al., 2014) and Twitter (Lan et al., 2017) under the unsupervised setting following in Meng et al. (2021). For MCPG-adapt, we compare it with supervised baselines on both Quora and MSCOCO datasets follow the supervised evaluation protocol in Su et al. (2021). The details are as follows.

Quora is grounded from the duplicated question pairs sharing the same answers in the Quora forum. We use 30k instances for testing in the unsupervised experiments. To validate the domain-adapted performance, the size of training, validation and test sets are 100k, 4k and 20k.

MSCOCO was originally constructed for image caption, which contains 5 different captions for each image. We randomly pick one of them as source sentence and the rest as targets. We use 20k test instances under the unsupervised setting. For supervised experiments, we split the dataset into 93k, 4k and 20k for training, validation and testing.

Twitter was built from linked tweets sharing URLs. There are both automatically and human annotated sentence pairs. We only use the pairs that are manually labeled as "paraphrases", which results in 566 instances for unsupervised testing.

4.2 Baselines and Evaluation Metrics

We compare MCPG-basic and MCPG-adapt with unsupervised and supervised models respectively.

The unsupervised baselines include (1) VAE (Bowman et al., 2016); (2) Lag VAE (He et al., 2019); (3) CGMH (Miao et al., 2019); (4) UPSA (Liu et al., 2020); (5) BT (Wieting et al., 2017); (6) Corruption (Hegde and Patil, 2020); (7) ConRPG (Meng et al., 2021). Results for the unsupervised baselines are cited from Meng et al. (2021).

The supervised baselines include (1) ResidualLSTM (Prakash et al., 2016); β-VAE (Higgins et al., 2017); (2) Transformer (Vaswani et al., 2017); (3) DNPG (Li et al., 2019); (4) LBOW-Topk (Fu et al., 2019); (5) LBOW-Gumbel (Fu et al., 2019); (6) IANet+X (Su et al., 2021); (7)IANet+S (Su et al., 2021). Results for the supervised baselines are cited from Su et al. (2021).

We evaluate all models by several automatic metrics: BLEU (Papineni et al., 2002), iBLEU (Sun and Zhou, 2012), and ROUGE (Lin, 2004). The

[1] https://www.kaggle.com/c/ quora-question-pairs

4-gram BLEU and ROUGE scores of both 1 and 2 grams are reported. The iBLEU score is included to penalize trivial outputs which simply repeat the input sentence. We further calculate SelfBleu (Zhu et al., 2018) to measure output diversity in Sec 6.1. Lower SelfBleu score indicates better diverity.

4.3 Implementation Details

We implement the Generator of MCPG based on the huggin-face T5-base checkpoint[2] on a subset of CommonCrawl containing 243 million sentences. To speed up training, we directly use the open-source checkpoint of SimCSE[3] as our Semantic Encoder and only update the parameters of Generator. During training, we set the learning rate to 1e-3 and the batch size to 1024. During inference, MCPG-basic first increases the dropout probability dp from 0.1 to 0.2 to create different semantic embeddings for the same input and then uses top-k sampling to generate 1-best paraphrase based on each embedding. Following Hegde and Patil (2020), we eliminate the ones that are the same as the input and obtain 10 distinct candidates. The unsupervised results in Table 1 are reported on the best candidates that achieve the top sentence-level iBleu scores. Similarly, MCPG-adapt produces K outputs conditioned on different exemplars retrieved from the training set. Table 2 shows the results using 10, 15 and 20 exemplars respectively.

5 Experiments

5.1 Main Results

Unsupervised Performance From Table 1, we see that our MCPG-basic significantly outperforms most baselines across all three datasets. It is noticeable that our model improves the Bleu scores by a large margin, with 6.53 on Quora, 3.56 on MSCOCO and 6.59 on Twitter respectively. This is because most factual entities appear both in the source and target sentences, which demonstrates the effectiveness of our simple keyword selection strategy in the unsupervised scenario. Directly copying keywords from the source sentence may hurt the diversity of generated paraphrases. Nevertheless, MCPG-basic still shows its superiority in terms of the iBleu score, which considers both the fidelity to reference and the difference from input.

Model	**iBleu**	**Bleu**	**R-1**	**R-2**
		Quora		
VAE	8.16	13.96	44.55	22.64
Lag VAE	8.73	15.52	49.20	26.07
CGMH	9.94	15.73	48.73	26.12
UPSA	12.03	18.21	59.51	32.63
BT	11.64	11.59	58.20	32.04
Corruption	12.32	17.97	59.14	32.44
ConRPG	12.68	18.31	59.62	33.10
MCPG-basic	**13.58**	**24.84**	**60.19**	**36.04**
		MSCOCO		
VAE	7.48	11.09	31.78	8.66
Lag VAE	7.69	11.63	32.20	8.71
CGMH	7.84	11.45	32.19	8.67
UPSA	9.26	14.16	37.18	11.21
BT	9.72	14.36	37.64	11.81
Corruption	10.32	15.60	38.12	12.40
ConRPG	11.17	16.98	**39.42**	13.50
MCPG-basic	**11.99**	**20.54**	38.45	**13.64**
		Twitter		
VAE	2.92	3.46	15.13	3.40
Lag VAE	3.15	3.74	17.20	3.79
CGMH	4.18	5.32	19.96	5.44
UPSA	4.93	6.87	28.34	8.53
BT	5.11	6.99	29.11	8.95
Corruption	5.32	7.11	29.80	9.32
ConRPG	5.83	7.32	30.81	10.08
MCPG-basic	**7.66**	**13.91**	**37.68**	**14.84**

Table 1: Unsupervised performance.

Domain-adapted Performance Table 2 compares the results of our domain-adapted model and the supervised baselines, where the annotated parallel corpus from the target domain is accessible. Note that, our model requires no extra training on the annotated data, whose parameters stay the same with the unsupervised MCPG-basic. Even so, MCPG-adapt achieves competitive performance compared with the baselines which need supervised training. Besides, the models's performance improves as the number of used exemplars increases. Combined with Table 1, we find that, with the help of transfer vectors, MCPG-adapt outperforms MCPG-basic on both datasets. The improvement on Quora is more significant than MSCOCO. A possible reason is the difference of data distribution. During experiments we observe that, in Quora, the retrieved exemplars from the training set are more similar to the input sentence than MSCOCO, which makes the derived transfer vectors more accurate for paraphrasing guidance in Quora.

5.2 Human Evaluation

To further verify the performance of the proposed model, we conduct human evaluation under the unsupervised setting. We randomly sample 100 instances from the Quora test set and ask three an-

[2] https://huggingface.co/t5-base

[3] https://huggingface.co/princeton-nlp/unsup-simcse-bert-base-uncased

Model	*Quora*				*MSCOCO*			
	iBleu	Bleu	R-1	R-2	iBleu	Bleu	R-1	R-2
ResidualLSTM	15.93	23.69	55.10	33.86	18.72	23.66	41.07	15.26
β-VAE, $\beta = 10^{-4}$	10.28	19.73	47.62	25.49	18.34	22.54	40.72	14.75
Transformer	17.98	25.01	57.82	32.58	19.81	24.68	41.49	15.84
DNPG	18.01	25.03	63.73	37.75	-	-	-	-
LBOW-Topk	19.03	26.17	58.79	34.57	21.07	25.27	42.08	16.13
LBOW-Gumbel	18.97	26.14	58.60	34.47	20.92	24.98	42.12	16.05
IANet+X	19.62	26.52	59.33	35.01	21.28	26.06	43.81	16.35
IANet+S	20.11	27.09	59.98	36.02	22.03	26.90	44.66	17.13
MCPG-adapt@10	21.08	33.58	66.78	45.54	14.56	23.06	40.03	14.67
MCPG-adapt@15	22.88	35.88	68.14	47.79	16.01	24.75	40.52	15.38
MCPG-adapt@20	23.93	37.22	68.91	49.11	17.12	26.13	40.83	15.81

Table 2: Domain-adapted performance. *MCPG-adapt@K* denotes the model variation using *K* exemplars.

Models	Mean Rank	Agreement
UPSA	3.31	90%
Corruption	2.87	70%
MCPG-basic	**2.07**	96%
Reference	1.78	82%

Table 3: Human evaluation results.

notators to evaluate the output results from UPSA, Corruption, MCPG-basic as well as the reference [4]. Each model is given a rank from 1 (best) to 4 (worst) for each input sentence by comparing their overall output quality w.r.t. semantic fidelity, language fluency and surface-form diversity following (Li et al., 2019). We report the mean rank of three annotators over all the evaluation instances in Table 3. Lower ranks are better. The agreement is the rate that at least two out of three annotators give the same rank to each model. The average Spearman's correlation coefficient between any two annotators is 0.488. It is observed that MCPG-adapt reaches the highest rank compared with UPSA and Corruption with high agreement, which demonstrates the effectiveness of our method.

6 Detailed Analysis

6.1 Analysis of Semantic Constraint

During inference, we tune the dropout probability of the Semantic Encoder to control the global semantics and increase generation diversity (Sec 3.2). We provide both qualitative and quantitative analysis about how this works in Figure 2 and Figure 3. In Figure 2, as the dropout probability *dp* increases from 0.10 to 0.28, the input and output sentences generally become more and more different. When *dp* gets moderately larger (e.g., less than or equal to 0.18), the model produces paraphrases with more diverse surface forms without losing the original semantic meaning. However, if *dp* keeps increasing,

Input: *you just lived through the hottest year on our planet ever recorded.*

0.10	you just lived the hottest year on our planet ever.
0.12	you just lived the hottest year *we have ever had*.
0.14	you just lived through the hottest year *on earth so far* recorded.
0.16	you *guys can live by* the hottest year ever recorded on *this* planet.
0.18	*we* just lived the hottest year *you can* ever *feel* on the planet.
0.20	you lived through the year *we* just *registered* the hottest *time* on the planet.
0.22	you recorded *us for* the hottest *era* ever lived.
0.24	you just *sweltered* the hottest *we have* ever recorded year *of* our *lives*.
0.26	*in any case we* recorded 'the hottest planet ever lived '*right now*.
0.28	you *got* the *record of* the year through the *smallest possible time in* our planet.

Figure 2: Example outputs of MCPG-basic conditioned on different dropout probabilities (ranging from 0.10 to 0.28). Purple parts are new lexical expressions that do not show up in the input sentence. Orange parts are old words in the input but are used in a different way in the output sentences.

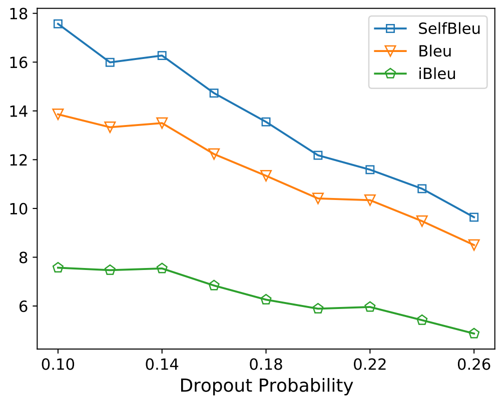


Figure 3: Effect of the dropout probability.

[4] Since ConRPG (Meng et al., 2021) do not release their code, we do not include it for human evaluation.

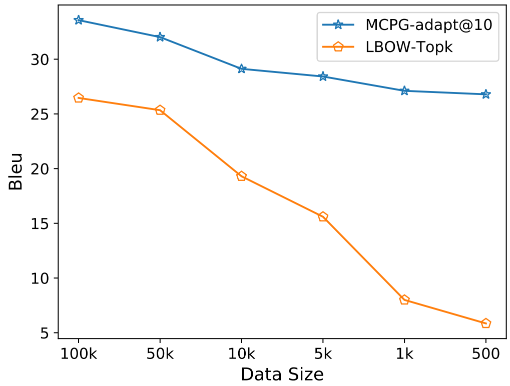


Figure 4: Effect of the parallel data size.

it may slightly hurt the fluency of output sentences. Besides, when *dp* becomes too large, the output sentence may fail to keep the core semantics of the input. The same trend can be statistically demonstrated by Figure 3. We plot the metrics (SelfBleu, Bleu, and iBleu) of MCPG-basic with different dropout probabilities on Twitter. SelfBleu is used to measure the similarity between input and output. The smaller the SelfBleu, the more diverse the output sentence. The results showcase that dropout enables a new way to control paraphrase generation, where we can tune the dropout probability to strike a balance between semantic fidelity and output diversity.

6.2 Analysis of Style Constraint

In Sec 5.1, we have proven that, retrieving exemplars from the annotated parallel corpus as a universal style constraint is helpful for domain adaptation, where our model achieves competitive results with the supervised models even without training. Here, we show that our model is also more robust to the size of annotated corpus than the traditional supervised models. Since IANet+S (Su et al., 2021) does not release codes, we reimplement LBOW-Topk (Fu et al., 2019) and compare our model with them on different parallel data sizes.

Figure 4 shows a dramatic drop in the Bleu score of LBOW-Topk as the size of Quora parallel corpus decreases from 100K to 500. On the contrary, the Bleu score of MCPG-adapt@10 still remains above 25.00 when there are only 500 candidates for retrieval. One reason is that our model only uses the retrieved exemplars from the parallel corpus as an assistant for paraphrase generation, but does not rely on them to update model parameters. The other reason is that, although the retrieved exemplars may become less similar to the input sentence in the semantic level when the annotated data size shrinks, our model can still benefit from mimicking the universal style mapping mode of exemplars through the transfer vector. Figure 5 lists some example outputs guided by different exemplars. The similarity between the input sentence and the exemplars decreases from top to bottom. Nevertheless, our model is able to generate reasonable paraphrases under different conditions.

Input: *what is the best way to improve your chess skills ?*

p: what is the best way to teach chess to children ?
q: what is the best way and right age to introduce chess to a child ?
y: what is the best way to improve your chess skills *and the right age to enter* ?

p: which age is the best to start teaching my little son to play chess ?
q: what are the benefits of teaching children to play chess and what is a good age to start ?
y: *what are the benefits of playing chess and what is a good way to* improve your skills ?

p: what is the best method of learning to speak a language ?
q: which are the best methods to learn a language ?
y: *which are the best methods* to help you improve your skills ?

p: what is the best thing we learn in business school ?
q: what is the most important thing to learn in a business school ?
y: what is *the most important way* to improve your skills in chess ?

p: what do you think is the most difficult language for anyone to learn ?
q: which language do you think is the most difficult all over the world ?
y: *which way is the best way* to improve your chess skills *around the world* ?

Figure 5: Example outputs of MCPG-adapt conditioned on different exemplars. Purple parts are new expressions in the output y inspired by the exemplar pair (p, q). Orange parts are old expressions that both appear in the input and output sentences.

Metrics	Quora	MSCOCO	Twitter
iBleu	11.84 (-1.74)	11.29 (-0.70)	4.80 (-2.86)
Bleu	21.31 (-3.53)	17.75 (-2.79)	8.05 (-5.86)
R-1	55.27 (-4.92)	35.92 (-2.53)	28.74 (-8.94)
R-2	31.29 (-5.35)	12.02 (-1.62)	12.02 (-2.82)

Table 4: Performance of MCPG-basic with random keywords. The 'blue' parts indicate the performance degradation compared with the results in Table 1.

6.3 Analysis of Lexical Constraint

Table 4 reveals the importance of using factual keywords as a local lexical constraint. Replacing the factual entities with random keywords negatively affects the results on three datasets in different degrees. This effect is most significant on Twitter, where many sentences are about politics and are more likely to contain the names of politicians, countries, etc. Nonetheless, this effect is less obvious on MSCOCO, since image captions tend to describe images in a more general way regardless of the specific names of objects.

7 Conclusion

In this paper, we propose a simple and effective model MCPG for more controllable unsupervised paraphrase generation which is also flexible for domain adaptation. MCPG decomposes the generation constraints into multiple levels: the global semantics, the local lexicons, and the universal styles in a human-like manner. Particularly, we manipulate a transfer vector derived from the semantic embeddings of exemplars to control output styles, which provides a new possibility for domain adaptation with annotated samples in a training-free way. In future works, we will explore to (1) control the model from more perspectives such as the syntactic structure, (2) generate paraphrases with higher quality than the input, i.e., to polish the input texts.

Limitations

In this work, we investigate a multi-level controllable approach for unsupervised and domain-adaptive paraphrase generation. Despite the promising experimental results, there are still some limitations of our work:

- We use a default dropout probability during training but tune it during inference, which may cause a little inconsistency.
- We do not shuffle the factual keywords and force the model to copy them, which may slightly hurt the output diversity.
- We use the transfer vector based on parallel exemplars to help the model in domain adaptation. Although the whole process is training-free, it would be better if we could obtain exemplar sentences without a parallel corpus.

Acknowledgements

This work was partially supported by the National Natural Science Foundation of China 62006062 and 62176076, Shenzhen Foundational Research Funding JCYJ20200109113441941, JCYJ20210324115614039, The Major Key Project of PCL2021A06, Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies 2022B1212010005.

References

Elron Bandel, Ranit Aharonov, Michal Shmueli-Scheuer, Ilya Shnayderman, Noam Slonim, and Liat Ein-Dor. 2022. Quality controlled paraphrase generation. In *ACL*.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Józefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *CoNLL*.

Yitao Cai, Yue Cao, and Xiaojun Wan. 2021. Revisiting pivot-based paraphrase generation: Language is not the only optional pivot. In *EMNLP*.

Jindong Chen, Yizhou Hu, Jingping Liu, Yanghua Xiao, and Haiyun Jiang. 2019a. Deep short text classification with knowledge powered attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6252–6259.

Mao Yan Chen, Haiyun Jiang, and Yujiu Yang. 2022. Context enhanced short text matching using click-through data. *arXiv preprint arXiv:2203.01849*.

Mingda Chen, Qingming Tang, Sam Wiseman, and Kevin Gimpel. 2019b. Controllable paraphrase generation with a syntactic exemplar. In *ACL*.

Jiayang Cheng, Haiyun Jiang, Deqing Yang, and Yanghua Xiao. 2021. A question-answering based framework for relation extraction validation. *arXiv preprint arXiv:2104.02934*.

Li Dong, Jonathan Mallinson, Siva Reddy, and Mirella Lapata. 2017. Learning to paraphrase for question answering. *ArXiv*, abs/1708.06022.

Yao Fu, Yansong Feng, and John P. Cunningham. 2019. Paraphrase generation with latent bag of words. *ArXiv*, abs/2001.01941.

Jun Gao, Changlong Yu, Wei Wang, Huan Zhao, and Ruifeng Xu. 2022. Mask-then-fill: A flexible and effective data augmentation framework for event extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2022*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *ArXiv*, abs/2104.08821.

Tanya Goyal and Greg Durrett. 2020. Neural syntactic preordering for controlled paraphrase generation. *ArXiv*, abs/2005.02013.

Yinpeng Guo, Yi Liao, Xin Jiang, Qing Zhang, Yibo Zhang, and Qun Liu. 2019. Zero-shot paraphrase generation with multilingual language models. *ArXiv*, abs/1911.03597.

Ankush Gupta, Arvind Agarwal, Prawaan Singh, and Piyush Rai. 2018. A deep generative framework for paraphrase generation. *ArXiv*, abs/1709.05074.

Junxian He, Daniel Spokoyny, Graham Neubig, and Taylor Berg-Kirkpatrick. 2019. Lagging inference networks and posterior collapse in variational autoencoders. *ArXiv*, abs/1901.05534.

Chaitra Hegde and Shrikumar Patil. 2020. Unsupervised paraphrase generation using pre-trained language models. *ArXiv*, abs/2006.05477.

Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*.

Kuan-Hao Huang and Kai-Wei Chang. 2021. Generating syntactically controlled paraphrases without using annotated parallel pairs. *ArXiv*, abs/2101.10579.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *NAACL*.

Amirhossein Kazemnejad, Mohammadreza Salehi, and Mahdieh Soleymani Baghshah. 2020. Paraphrase generation by learning how to edit from samples. In *ACL*.

Wuwei Lan, Siyu Qiu, Hua He, and Wei Xu. 2017. A continuously growing dataset of sentential paraphrases. *ArXiv*, abs/1708.00391.

Mirella Lapata, Rico Sennrich, and Jonathan Mallinson. 2017. Paraphrasing revisited with neural machine translation. In *EACL*.

Zichao Li, Xin Jiang, Lifeng Shang, and Qun Liu. 2019. Decomposable neural paraphrase generation. *ArXiv*, abs/1906.09741.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *ACL 2004*.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *ECCV*.

Lemao Liu, Haisong Zhang, Haiyun Jiang, Yangming Li, Enbo Zhao, Kun Xu, Linfeng Song, Suncong Zheng, Botong Zhou, Dick Zhu, Xiao Feng, Tao Chen, Tao Yang, Dong Yu, Feng Zhang, Zhanhui Kang, and Shuming Shi. 2021. Texsmart: A system for enhanced natural language understanding. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*.

Xianggen Liu, Lili Mou, Fandong Meng, Hao Zhou, Jie Zhou, and Sen Song. 2020. Unsupervised paraphrasing by simulated annealing. *ArXiv*, abs/1909.03588.

Yuxian Meng, Xiang Ao, Qing He, Xiaofei Sun, Qinghong Han, Fei Wu, Chun Fan, and Jiwei Li. 2021. Conrpg: Paraphrase generation using contexts as regularizer. In *EMNLP*.

Ning Miao, Hao Zhou, Lili Mou, Rui Yan, and Lei Li. 2019. Cgmh: Constrained sentence generation by metropolis-hastings sampling. In *AAAI*.

Tong Niu, Semih Yavuz, Yingbo Zhou, Nitish Shirish Keskar, Huan Wang, and Caiming Xiong. 2021. Unsupervised paraphrasing with pretrained language models. In *EMNLP*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*.

Aaditya Prakash, Sadid A. Hasan, Kathy Lee, Vivek Datla, Ashequl Qadir, Joey Liu, and Oladimeji Farri. 2016. Neural paraphrase generation with stacked residual lstm networks. In *COLING*.

Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *ArXiv*, abs/1910.10683.

Parker Riley, Noah Constant, Mandy Guo, Girish Kumar, David C. Uthus, and Zarana Parekh. 2021. Textsettr: Few-shot text style extraction and tunable targeted restyling. In *ACL*.

Aurko Roy and David Grangier. 2019. Unsupervised paraphrasing without translation. In *ACL*.

Lingfeng Shen, Haiyun Jiang, Lemao Liu, and Shuming Shi. 2022. Revisiting the evaluation metrics of paraphrase generation. *arXiv preprint arXiv:2202.08479*.

A.B. Siddique, Samet Oymak, and Vagelis Hristidis. 2020. Unsupervised paraphrasing via deep reinforcement learning. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

Yixuan Su, David Vandyke, Simon Baker, Yan Wang, and Nigel Collier. 2021. Keep the primary, rewrite the secondary: A two-stage approach for paraphrase generation. In *FINDINGS*.

Hong Sun and M. Zhou. 2012. Joint learning of a dual smt system for paraphrase generation. In *ACL*.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *ArXiv*, abs/1706.03762.

Chao Wang, Haiyun Jiang, Tao Chen, Jingping Liu, Menghui Wang, Sihang Jiang, Zhixu Li, and Yanghua Xiao. 2022. Entity understanding with hierarchical graph learning for enhanced text classification. *Knowledge-Based Systems*, 244:108576.

John Wieting and Kevin Gimpel. 2018. Paranmt-50m: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. *ArXiv*, abs/1711.05732.

John Wieting, Jonathan Mallinson, and Kevin Gimpel. 2017. Learning paraphrastic sentence embeddings from back-translated bitext. *ArXiv*, abs/1706.01847.

Haoran Yang, Wai Lam, and Pijian Li. 2021. Contrastive representation learning for exemplar-guided paraphrase generation. *ArXiv*, abs/2109.01484.

Kexin Yang, Dayiheng Liu, Wenqiang Lei, Baosong Yang, Haibo Zhang, Xue Zhao, Wenqing Yao, and Boxing Chen. 2022. Gcpg: A general framework for controllable paraphrase generation. In *FINDINGS*.

Daojian Zeng, Haoran Zhang, Lingyun Xiang, Jin Wang, and Guoliang Ji. 2019. User-oriented paraphrase generation with keywords controlled network. *IEEE Access*, 7:80542–80551.

Haisong Zhang, Lemao Liu, Haiyun Jiang, Yangming Li, Enbo Zhao, Kun Xu, Linfeng Song, Suncong Zheng, Botong Zhou, Jianchen Zhu, Xiao Feng, Tao Chen, Tao Yang, Dong Yu, Feng Zhang, Zhanhui Kang, and Shuming Shi. 2020. Texsmart: A text understanding system for fine-grained ner and enhanced semantic analysis. *ArXiv*, abs/2012.15639.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*.