

Controllable and Diverse Text Generation in E-commerce

Huajie Shao*

1. University of Illinois at Urbana
Champaign
2. Alibaba Group
hshao5@illinois.edu

Xuezhou Zhang

University of Wisconsin, Madison
zhangxz1123@cs.wisc.edu

Jun Wang

Alibaba Group
jun.w@alibaba-inc.com

Aston Zhang

University of Illinois at Urbana
Champaign
lzhang74@illinois.edu

Haohong Lin

Zhejiang University
lhh2017@zju.edu.cn

Heng Ji

University of Illinois at Urbana
Champaign
hengji@illinois.edu

Tarek Abdelzaher

University of Illinois at Urbana
Champaign
zaher@illinois.edu

ABSTRACT

In E-commerce, a key challenge in text generation is to find a good trade-off between word diversity and accuracy (relevance) in order to make generated text appear more natural and human-like. In order to improve the relevance of generated results, conditional text generators were developed that use input keywords or attributes to produce the corresponding text. Prior work, however, do not finely control the diversity of automatically generated sentences. For example, it does not control the order of keywords to put more relevant ones first. Moreover, it does not explicitly control the balance between diversity and accuracy. To remedy these problems, we propose a fine-grained controllable generative model, called *Apex*, that uses an algorithm borrowed from automatic control (namely, a variant of the *proportional, integral, and derivative (PID) controller*) to precisely manipulate the diversity/accuracy trade-off of generated text. The algorithm is injected into a Conditional Variational Autoencoder (CVAE), allowing *Apex* to control both (i) the order of keywords in the generated sentences (conditioned on the input keywords and their order), and (ii) the trade-off between diversity and accuracy. Evaluation results on real world datasets¹ show that the proposed method outperforms existing generative models in terms of diversity and relevance. Moreover, it achieves about 97% accuracy in the control of the order of keywords.

Apex is currently deployed to generate production descriptions and item recommendation reasons in Taobao², the largest E-commerce platform in China. The A/B production test results show that our method improves click-through rate (CTR) by 13.17% compared

to the existing method for production descriptions. For item recommendation reason, it is able to increase CTR by 6.89% and 1.42% compared to user reviews and top-K item recommendation without reviews, respectively.

ACM Reference Format:

Huajie Shao, Jun Wang, Haohong Lin, Xuezhou Zhang, Aston Zhang, Heng Ji, and Tarek Abdelzaher. 2021. Controllable and Diverse Text Generation in E-commerce. In *Proceedings of the Web Conference 2021 (WWW '21)*, April 19–23, 2021, Ljubljana, Slovenia. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3442381.3449838>

1 INTRODUCTION

Text generation has been widely used in a variety of natural language processing (NLP) applications, such as review writing assistance [34, 38], production description generation [33], and dialogue generation [19]. In E-commerce, an attractive and accurate product description is crucial to convince customers to click or purchase the recommended items. Thus, how to strike the right balance between diversity and accuracy (relevance)³ of generated text [39] remains a key challenge in text generation. As we know, diversity can help the generated text seem more natural and human-like, while accuracy means the generated text is relevant to the target. In general, techniques that improve diversity reduce accuracy, whereas techniques that improve accuracy often produce the same expressions repeatedly. The contribution of this paper is to develop a controllable generative model that manipulates both the diversity and accuracy of generated sentences to attain both goals at the same time.

Recently, deep generative models have been proposed to diversify generated text on the basis of Variational Autoencoders (VAEs) [16, 35] and generative adversarial networks (GAN) [12, 36, 37, 40]. To better control the relevance of generated results, researchers developed conditional text generators, conditioned on some keywords or attributes of items, user identities, or semantics [16, 26, 38].

Past methods, however, do not manipulate the order of keywords nor accurately diversify the generated text. The order of keywords

*Part of work was completed at Alibaba Group.

¹The sampled data is publicly available at <https://github.com/paper-data-open/Text-Gen-Ecommerce>

²<https://www.taobao.com/>

This paper is published under the Creative Commons Attribution 4.0 International (CC-BY 4.0) license. Authors reserve their rights to disseminate the work on their personal and corporate Web sites with the appropriate attribution.

WWW '21, April 19–23, 2021, Ljubljana, Slovenia

© 2021 IW3C2 (International World Wide Web Conference Committee), published under Creative Commons CC-BY 4.0 License.

ACM ISBN 978-1-4503-8312-7/21/04.

<https://doi.org/10.1145/3442381.3449838>

³relevance and accuracy are interchangeable in this paper

in the generated sentences plays an important role in attracting customers in E-commerce applications. It can provide most important production information to customers, and promote their interests to improve the likelihood of clicking or purchase. For example, when one recommends a skirt to a customer, one can write: “The skirt is very **affordable**, yet **charming**” if the customer prefers price to fashion. Alternatively, one can rephrase: “The skirt is very **charming**, yet **affordable**” if the customer prefers fashion. This observation motivates us to control the order of keywords to generate various product descriptions in order to attract different customers.

Diversity makes text appear more human like. A sentence that conveys the intended meaning might be: “The trousers are **comfortable** with **comfortable** fabric, and **comfortable** fit”. A more diverse description might say: “The trousers are **comfortable** with **soft** fabric, and **relaxed** fit”. The key is to diversify while maintaining accuracy.

To this end, we propose a novel controllable text generation framework, called *Apex*. *Apex* combines Conditional Variational Autoencoders (CVAEs), illustrated in Fig. 1, with a linear PI controller (as shown in Fig. 2), a variant of proportional, integral, differential (PID) control algorithm [2], to control the diversity and accuracy of generated text. Specifically, the CVAEs condition the keywords and their order appearing in the reference text on the conditional encoder with Transformer [32] to control the diversity of generated sentences. However, CVAEs often suffer from the KL-vanishing problem for sequence generation models [24, 29, 41]. Namely, the KL-divergence term becomes zero and the decoder completely ignores latent features conditioned on input data. This is because the powerful decoder can directly learn information from other paths, such as the input keywords of the conditional encoder (red arrow) in Fig. 1. On the other hand, KL divergence in CVAEs can affect the diversity and accuracy of generated text. A large KL-divergence can diversify generated text, but lower its accuracy. Hence, controlling the value of KL-divergence is of great importance to text generation.

Prior work, such as cost annealing and cyclical annealing [4, 24], only focus on mitigating the KL-vanishing problem, but cannot explicitly control the value of KL-divergence. Besides, these methods cannot fully avert KL-vanishing because they blindly tune the weight on KL term without using feedback (from output KL-divergence) during model training. In practice, the feedback from KL-divergence can tell us when the KL-vanishing problem occurs so that we can change the weight on the KL term accordingly in the VAE objective function. Specifically, when KL-divergence becomes too small (thus hurting diversity), we need to reduce the weight of the KL term in the objective being optimized, allowing the optimization to favor larger KL-divergence values. Conversely, when KL-divergence becomes too large (thus possibly hurting accuracy), we increase its weight in the objective function, causing the optimization to favor smaller values. This is not unlike control of composition of chemical ingredients to optimize specific product qualities.

Inspired by control literature, we design a PI controller, a variant of PID control algorithm [1, 2] as introduced in Section 2.3, that is novel in using KL-divergence feedback. The basic idea of PID controller is to calculate the error between a desired value (in this case, the desired KL-divergence) and the current actual value, then apply a correction in a direction that reduces that error. By doing this, it stabilizes the actual KL-divergence around a specified value, called *set point*. In this paper, the designed PI controller dynamically

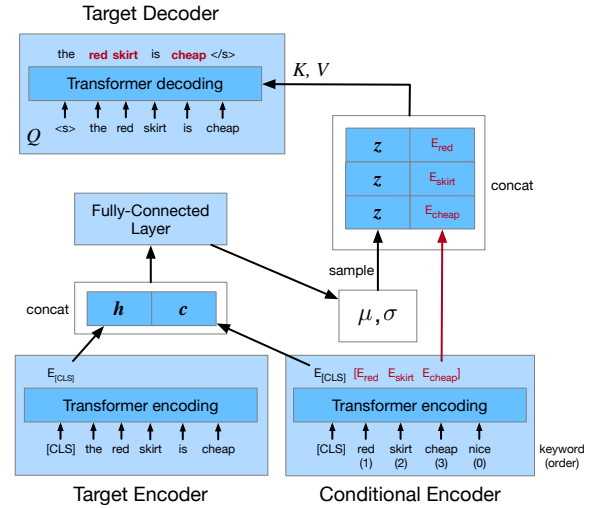


Figure 1: The CVAEs component of the proposed *Apex*.

tunes the weight on KL term in the CVAE objective based on the output KL-divergence to stabilize it to a desired value. In this way, we can not only totally solve KL-vanishing problem, but also achieve an *adjustable trade-off* between diversity and accuracy of generated text via controlling KL-divergence.

We evaluate the performance of *Apex* on the real-world dataset, collected from Taobao, a Chinese E-commerce platform. The experimental results illustrate that *Apex* outperforms the state-of-the-art text generators in terms of diversity and accuracy. It can also control the order of keywords in the generated sentences with about 97% accuracy.

Importantly, *Apex* has been deployed in Taobao E-commerce platform to generate production descriptions and item recommendation reasons. According to A/B tests, our method achieves an improvement of 13.17% over the existing method in terms of CTR for production description generation. For item recommendation reason, it improves CTR by 6.89% and 1.42% compared to user reviews and top-K item recommendation without reviews, respectively.

2 PRELIMINARIES

We first introduce our text generation problem, and then present the background of two generative models, variational autoencoders (VAEs) and conditional variational autoencoders (CVAEs). Finally, we review the general PID control algorithm in automatic control theory.

2.1 Problem

The goal of this work is to generate text descriptions of items given selected keywords and their order. For a given item, let the set $Y_a = \{y_1, y_2, \dots, y_m\}$ denote the set of selected keywords, and $Y_p = \{p_1, p_2, \dots, p_m\}$ denote their corresponding order occurring in the reference text, such that $p_i, i \in \{1, 2, \dots, m\}$, is the order of keyword y_i . Given Y_a and Y_p for the item, the goal is to automatically generate an accurate text description of the item, where the specified keywords are used (in meaningful descriptive sentences) and appear in the specified order.

To meet the above goal, we need to train a text generation model. To train a model, for each set of input keywords Y_a of an item in the training data, we use a corresponding reference text with n words, denoted by $X_{1:n} = \{x_1, x_2, \dots, x_n\}$. The keywords appear in the order, Y_p , in the reference text. Overloading the order notation, let $p_i = 0$ mean that the i -th keyword is *not* in the reference text. As shown in Fig. 1, the keyword, “red”, is the first one among other keywords in the reference sentence. It is thus numbered as “1” in numerical order. For keyword “nice”, it is numbered as “0” because it is not in the reference text. Note that, the reference text is used to train the CVAEs model to learn the conditional distribution of latent variable, denoted by (μ, σ) .

During testing, we use sampled data from the distribution of the latent variable, (μ, σ) , together with the input keywords Y_a and their order Y_p to generate text.

2.2 Background of VAEs and CVAEs

As one of the most popular deep generative models, VAEs [18, 23] have been widely used in various applications, such as image generation and text generation in language modeling. A VAE [18, 23] model includes two main parts: an encoder and decoder. The encoder first learns a conditional distribution of a latent variable z , given observed data x . The decoder then reconstructs data x from the generative distribution of latent code. However, due to intractable posterior inference, researchers often optimize the following evidence lower bound (ELBO).

$$\mathcal{L}_{vae} = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x)||p(z)) \quad (1)$$

where $p_\theta(x|z)$ is a probabilistic *decoder* parameterized by a neural network to reconstruct data x given the latent variable z , and $q_\phi(z|x)$ is a *encoder* network that approximates the posterior distribution of latent variable z given data x . In addition, $p(z)$ is a prior distribution of input data, such as Unit Gaussian.

CVAEs [17] build on the VAEs model by simply conditioning on y (e.g., image generation given a label). CVAEs attempt to approximate the conditional distribution $p(x|y)$ given an input y . The variational lower bound of CVAEs can be expressed as

$$\begin{aligned} \mathcal{L}_{cvae} = & \mathbb{E}_{q_\phi(z|x,y)} \log p_\theta(x|z, y) \\ & - D_{KL}(q_\phi(z|x, y)||p_\theta(z|y)), \end{aligned} \quad (2)$$

where $p_\theta(z|y)$ is the prior probability distribution conditioned on y .

However, optimizing the VAEs and CVAEs model can lead to the KL-vanishing problem that KL-divergence becomes zero during model training in language modeling [4, 24]. Namely, the decoder only learns plain language without using the distribution of latent variable learned from the encoder. This is because the decoder is very powerful so that it can learn information from other paths, such as an attention mechanism [24].

2.3 PID Control Algorithm

PID algorithm has been the most popular feedback control method in control theory, and it has been widely used in both physical systems [2] and computing systems [13]. The basic idea of PID control is to calculate the error between a desired value (in this case, the desired KL-divergence) and the current actual value, then apply a correction in a direction that reduces that error. In its general form, the correction is the weighted sum of three terms; one is proportional

to error (P), one is proportional to the integral of error (I), and one is proportional to the derivative of error (D). The general model of PID controller is defined by

$$w(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}, \quad (3)$$

where $w(t)$ is the output of the controller; $e(t)$ is the error between the actual value and the desired value at time t ; K_p , K_i and K_d denote the coefficients for the P term, I term and D term, respectively. Since the derivative (D) term essentially computes the slope of the signal, when the signal is noisy it often responds more to variations induced by noise. Hence, we do not use D term based on established best practices in control of noisy systems. The resulting specialization of PID is called the PI controller.

In this paper, we propose a CVAEs model with the PI control algorithm to control the diversity and accuracy of generated sentences based on the input keywords and their order.

3 ARCHITECTURE OF APEX

We introduce the overall architecture of *Apex* for controllable text generation. It combines the CVAEs model (as shown in Fig. 1) with a PI controller (as shown in Fig. 2) to manipulate the diversity and accuracy of generated text.

3.1 Framework of CVAEs

The CVAEs condition on the input keywords of items and their corresponding order, and then generate text to fulfill the keywords as specified in conditional encoder.

3.1.1 Target Encoder and Conditional Encoder. As shown in Fig. 1, our target encoder is to encode the sequence of reference text via the Transformer model [32]. Given a reference text $x = (x_1, x_2, \dots, x_n)$ with n elements, we embed each one in a low dimensional space as $E = (e_1, e_2, \dots, e_n)$, where $e_i \in \mathbb{R}^f$ is the i -th column of an embedding matrix $D \in \mathbb{R}^{f \times V}$, f and V denote the embedding size and the vocabulary size respectively. In addition, motivated by the BERT model [7], we add a special token [CLS] at the beginning of every reference text. The final hidden state corresponding to this token is used as the aggregate sequence representation. Then the word embeddings of the reference text and the special token are fed into a d -dimensional Transformer, and it finally yields the output of hidden state $E_{[CLS]}$, also denoted by h .

For the conditional encoder, we also adopt the Transformer model to encode the input keywords $y_a = (y_1, y_2, \dots, y_m)$ and their corresponding order $y_p = (p_1, p_2, \dots, p_m)$ in the reference text. Similar to the target encoder, a special token [CLS] is added at the beginning of the input keywords to aggregate sequence representation. In this encoder, the selected keywords of an item are first embedded in the low dimensional space, denoted by $U = (u_1, u_2, \dots, u_m)$, where $u_j \in \mathbb{R}^f$ for $j = 1, \dots, m$. Inspired by literature [9], we embed their corresponding orders in the low dimensional space $O = (o_1, o_2, \dots, o_m)$, where $o_j \in \mathbb{R}^f$. Then we combine their embeddings together with an element-wise addition, yielding element representations $C = (u_1 + o_1, u_2 + o_2, \dots, u_m + o_m)$, where $C \in \mathbb{R}^{f \times m}$. Finally, we feed both the embedding of the special token [CLS] and the element representations C into the Transformer to obtain the hidden state c .

After that, we concatenate the output hidden state of the above two encoders, and then feed it into the fully connected (FC) layers to generate the conditional distribution of latent variable, z .

3.1.2 Target Decoder. The target decoder G_θ is to predict the target words via the Transformer given the input keywords and their order. Firstly, we concatenate the sampled data from the distribution of latent variable, (μ, σ) , and the representation of each input keyword and the corresponding order, $[E_1, E_2, \dots, E_m]$, from conditional encoder as the key (K) and value (V) of the Transformer. Namely, $K = V = [z \oplus E_1, z \oplus E_2, \dots, z \oplus E_m]$. In addition, each word of reference sequence in the target decoder is embedded into a low dimensional space as the input query (Q). Finally, we can predict the target words based on Q, K and V in the Transformer decoder [32].

3.2 Controllable KL-divergence using PI Control Algorithm

We propose a linear PI control algorithm to manipulate the value of KL-divergence in Fig.2, which has two advantages below:

- Avert KL-vanishing problem.
- Achieve an adjustable trade-off between diversity and accuracy for generated text.

In language modeling, one challenge is that the VAEs and CVAEs model often suffer from the KL-vanishing problem due to a powerful decoder, as mentioned in Section 2.2. Our model also has the same problem because the Transformer decoder can predict the target words from the input keywords in the conditional encoder and its previous time steps. To mitigate KL-vanishing, some attempts [4, 24] have been made by researchers. One popular way is to add a hyperparameter, w , to the KL term in Eq. (2), yielding:

$$\begin{aligned} \mathcal{L}_{cvae} = & \mathbb{E}_{q_\phi(z|x,y)} \log p_\theta(x|z,y) \\ & - w D_{KL}(q_\phi(z|x,y) || p_\theta(z|y)). \end{aligned} \quad (4)$$

The basic idea is to set the weight w to 0 at the beginning of model training, and then gradually increase it until to 1 with an annealing method. The prior work, such as cost annealing and cyclical annealing [4, 24], vary the weight in an open-loop fashion, without using feedback. When there exists a very powerful decoder such as Transformer, they cannot fully solve the KL-vanishing problem, as illustrated in Section 4. To address this problem, one simple and direct method is to adjust the weight based on the actual output KL-divergence during model training. When the KL-divergence becomes too small, we need to reduce the weight to boost the KL-divergence. The PI control algorithm has such function, which can automatically tune the weight in the above VAE objective to stabilize the KL-divergence to a desired value (set point).

In addition, KL-divergence plays an important role in the CVAEs because it affects the diversity and accuracy of generated text. When KL-divergence is large, the more diverse the generated text is, the lower its quality. In contrast, the generated text has higher reconstruction accuracy, but it is less diverse when KL-divergence is small. Thus, it is important to control KL-divergence to achieve a good trade-off between diversity and accuracy. This can be realized by PI control algorithm as well.

Fig. 2 illustrates the block diagram of our PI control algorithm. It samples the actual output KL-divergence during model training as

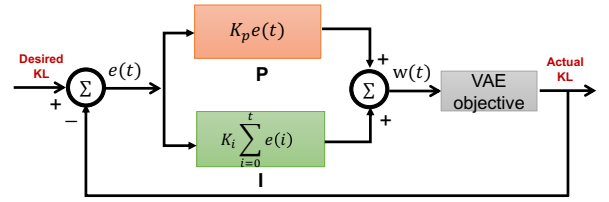


Figure 2: PI controller component of the proposed Apex for controllable KL-divergence.

the feedback to the PI controller. Then we compare the difference, $e(t)$, between the actual KL-divergence and desired one as the input of PI controller to calculate the weight w . According to Eq. (4), when KL-divergence becomes smaller, the controller sets the weight w to a smaller value in the objective function to reduce penalty for higher divergence; otherwise increases it until 1, if KL-divergence becomes larger. During model training, we sample the KL-divergence at each training step in discrete time. Note that, as mentioned earlier, we do not use the full PID control, since the output of KL divergence is not very stable during model training with mini-batch data. Since the D term in PID control is a derivative term, it reacts to signal slope, which is often dominated by noise, and is therefore not productive. In order to improve the response time of control system, we adopt a linear PI control algorithm, as shown in Fig. 2. The P term is proportional to the current error and I term sums (integrates) past errors with a sampling period $T_s = 1$. Finally, the weight $w(t)$ is a sum of P term and I term below

$$w(t) = K_p e(t) + K_i \sum_{i=0}^t e(i), \quad (5)$$

where $0 \leq w(t) \leq 1$ and $e(t)$ is the error between the desired KL-divergence and the output KL divergence at training step t ; K_p and K_i are the *negative* coefficients of the P term and I term, respectively.

Now, we briefly introduce how the designed PI control algorithm works. When error is large and positive (KL-diverge is below set point), both P term and I term becomes negative, so the weight $w(t)$ is limited to its lower bound 0 that encourages KL-divergence to grow. When KL-divergence significantly exceeds the set point (called overshoot), the error becomes negative and large, both P term and I term become positive, so the weight becomes positive, causing KL-divergence to shrink. In both cases, the PI control algorithm dynamically tunes the weight to help KL-divergence approach the set point.

3.3 Set Point Guidelines

One question is: how to choose the desired value (set point) of KL-divergence for text generation? For text generation, the weight w in the VAE objective varies from 0 to 1, so we may choose the intermediate point, $w = 0.5$, to run the CVAE models using cost annealing or cyclical annealing. The KL-divergence of CVAE model would converge to a specific value, v_0 , after training many steps if not suffering from KL-vanishing (almost not). If it does suffer, we may choose a smaller weight w until KL converges to a non-zero value, v_0 . Then we can increase or decrease the desired KL-divergence to some extent based on the benchmark v_0 . Note that, since our PI control algorithm is end-to-end dynamic learning method, users can

Algorithm 1: PI control algorithm.

```

input :Desired KL divergence,  $v_{KL}$ ,  $K_p$ ,  $K_i$ ,  $\alpha$ , training steps  $T$ 
output :weight on KL term  $w(t)$  at training step  $t$ 
1  $I(0) = 0, w(0) = 0;$ 
2 for  $t = 1$  to  $T$  do
3   sampled KL divergence  $\hat{v}_{KL}(t);$ 
4    $e(t) \leftarrow v_{KL} - \hat{v}_{KL}(t);$ 
5    $P(t) \leftarrow K_p e(t);$ 
6   if  $0 \leq w(t-1) \leq 1$  then
7      $I(t) \leftarrow I(t-1) + K_i e(t)$ 
8   else
9      $I(t) \leftarrow I(t-1)$  // Anti-windup
10  end
11  // calculate weight  $w(t)$ 
12   $w(t) \leftarrow P(t) + I(t);$ 
13  if  $w(t) > 1$  then
14     $w(t) \leftarrow 1$ 
15  end
16  if  $w(t) < 0$  then
17     $w(t) \leftarrow 0$ 
18  end
19  return  $w(t)$ 
20 end

```

customize the desired value of KL-divergence to meet their demand for different applications. For example, if users want to improve the diversity of generated text, they could choose a large set point of KL-divergence.

3.4 Algorithm Summary

We summarize the proposed PI controller in Algorithm 1. Our PI algorithm updates the weight, $w(t)$, on the KL term in the objective using the sampled KL divergence at training step t as feedback, as shown in Line 3. Line 4 computes the error $e(t)$ between sampled KL-divergence and set point. Line 5 to 12 is to calculate the weight w based on error $e(t)$. Note that, Line 9 is a popular constraint in PID/PI design, called anti-windup [27]. It effectively disables the integral term of the controller when controller output gets out of range, not to exacerbate the out-of-range deviation.

4 EVALUATION

We evaluate the performance of *Apex* on a real-world dataset. We first verify that the PI controller can totally avert KL vanishing. Then, we compare the performance of *Apex* with baselines using automatic evaluation and human evaluation. We also show that our approach can generate text that obeys the order of keywords given by user input. Finally, we implement online A/B testing to demonstrate the good performance of the proposed *Apex*.

4.1 Datasets

We collect a real world dataset of item descriptions and keywords from the Chinese E-commerce platform, Taobao. This dataset contains 617, 181 items and 927, 670 item text descriptions. Each item description is written by multiple human writers who help sellers write text to attract more customers. According to our statistics, each

item on average has about 1.4 descriptions written by humans and the average number of words in each description is 61.3. In addition, there are about 14.75 keywords per item on average. The total size of the vocabulary used is 88, 156.

4.2 Baselines

We compare the performance of *Apex* with the following baselines:

- Apex-cost: This method uses cost annealing method instead of PI controller for *Apex*.
- Apex-cyclical: This method uses cyclical annealing method instead of PI controller for *Apex*.
- Seq2seq [8]: This is the seq2seq model with a maximum likelihood estimator [31]. This paper adopts the Transformer model instead of LSTM in the encoder and decoder for a fair comparison.
- DP-GAN [36]: This conditional generator builds on the seqGAN, which uses the first sentence as conditional input.
- ExpansionNet [25]: This model conditions on some aspects or keywords of items and user identity to generate personalized and diverse text.

4.3 Experimental Settings

We implement our experiments using Tensorflow through the Texar platform [15]. We use a three-layer Transformer with default eight heads in the encoder and decoder. While transformer design often includes up to six layers, we find that three are sufficient for our purposes and use that number to improve efficiency. Besides, we adopt three-layer fully connected network with hidden size of 400, 200 and 100 to learn the distribution of latent features. In our experiment, we use 80%, 10% and 10% of the above dataset as the training, validation and testing data. We set the maximum number of words for each reference text to 100. The maximum number of input keywords is 50. In addition, we set both the dimension of word embedding and order embedding to 512. The learning rate is set to 0.0001 with the exponential decay by a decay factor 0.9 every 1000 training steps. Also, we set the batch size to 100 for model training. Based on empirical PID tuning rules [30], we set the coefficient of PI algorithm, K_p and K_i , to -0.01 and -0.0001 , respectively. The sampling period, T_s , is set to 1 for our PI controller.

4.4 PI algorithm for KL vanishing

We first conduct an experiment to demonstrate that the proposed PI algorithm (a variant of PID) can totally avert the KL vanishing. We compare it with the two representative approaches below:

- Cost annealing [4]: This method firstly sets the weight of KL divergence term to 0 in the early stage, and then gradually increases it to 1 using Sigmoid.
- Cyclical annealing [24]: This method splits the training process into M (e.g., 5) cycles and each increases the weight from 0 until to 1 using the linear function.

Fig. 3 illustrates that the KL divergence and its weight w vary with the training steps for different methods. Note that, here PI- v means we set the KL divergence to a desired value v (e.g., 2) for our designed PI algorithm. We can observe from Fig. 3(a) that when the training step is close to 100, the KL divergence of CVAEs gradually decreases to 0, which means the model suffers from KL vanishing. At that point, our PI algorithm automatically sets the weight to a

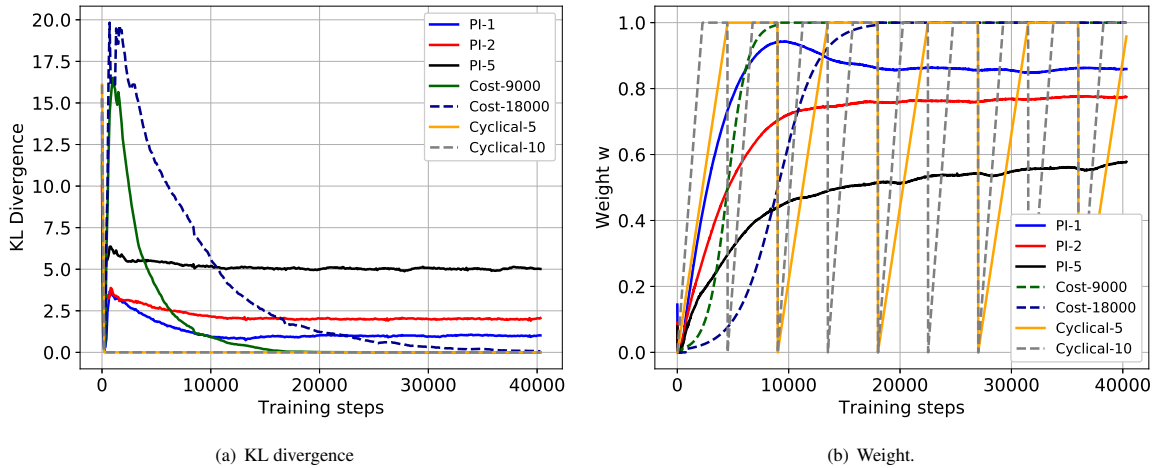


Figure 3: (a) illustrates comparison of KL divergence for different methods. We can observe that both cost annealing and cyclical annealing suffer from KL vanishing problem, while our *Apex* totally averts KL vanishing. (b) Weight varies with the training step.

Table 1: Performance comparison for different methods using automatic metrics averaged over 5 random seeds. For Dis- n , ROUGE-L, METEOR: higher is better. Self-BLEU: lower is better.

Models	Dis-1	Dis-2	Dis-3	ROUGE-L	METEOR	self-BLEU-1	self-BLEU-2	self-BLEU-3
Apex-PI-5	9.266K	127.51K	493.57K	0.2354	0.2593	0.9857	0.9471	0.8747
Apex-PI-2	9.227K	125.66K	477.63K	0.2358	0.2494	0.9841	0.9437	0.8674
Apex-PI-1	9.108K	125.39K	476.40K	0.2364	0.2538	0.9849	0.9454	0.8715
Apex-cost	8.904K	121.06K	456.57K	0.2344	0.2482	0.9853	0.9459	0.8719
Apex-cyclical	8.915K	121.82K	460.24K	0.2347	0.2486	0.9849	0.9444	0.8690
Seq2seq	8.648K	112.31K	427.22K	0.2087	0.2399	0.9819	0.9480	0.8776
ExpansionNet	9.018K	114.96K	473.28K	0.1951	0.1822	0.9854	0.9442	0.8700
DP-GAN	8.124K	102.18K	413.78K	0.1920	0.1949	0.9946	0.9849	0.9698

small value to boost the KL divergence as shown in Fig. 3(b). After 10,000 training steps, the PI controller stabilizes the KL divergence at the customized set points, such as 1, 2 and 5. Most importantly, we do not need to tune the PI parameters again for different set points of KL divergence in our experiment.

For the cost annealing method, we use two different hyper-parameters, 9000 and 18000, in the sigmoid function [4] to gradually increase the weight from 0 until to 1, as shown in Fig. 3(b). From 3(a), we can see that it does not suffer from KL vanishing problem at the beginning of training, because its weight is set to a small value in Fig. 3(b). However, it gradually suffers from KL vanishing after increasing weight from 0 to 1.

For cyclical annealing, we also use different hyper-parameters (e.g., 5 and 10) about the number of cycles to do experiments. We discover that cyclical annealing suffers from KL vanishing a lot, because it blindly changes its weight without using the feedback of KL divergence. In Fig. 3, it can be seen that when the KL divergence is decreasing at some points, its weight still increases, accelerating KL vanishing.

4.5 Performance Comparison

Next, we compare the performance of *Apex* with the baselines above, as shown in Table 1. We adopt the commonly used metrics in NLP:

Dis- n [36], ROUGE-L [14], METEOR [3], and Self-BLEU [42], to evaluate their performance. Here *Apex-PI- v* represents *Apex* with PI algorithm to control the KL divergence to a desired value, such as 5, 2 and 1. Besides, *Apex-cost-anneal* and *Apex-cyclical-anneal* represent our model but replace the PI algorithm with cost annealing and cyclical annealing to generate text, respectively. Table 1 illustrates that *Apex* with PI controller has higher Dis-1, Dis-2 and Dis-3 but lower self-BLEU than the baselines. It means our *Apex* can generate more diverse text than other methods. We also discover that the higher the KL divergence, the more diverse the generated text is. In addition, *Apex* with PI algorithm has higher METEOR and ROUGE-L than the baselines, which means our method can generate accurate text based on keywords. Therefore, the proposed method can achieve a good trade-off between accuracy and diversity using PI control algorithm. We also show more examples in Appendix 7.

4.6 Accuracy on Controlling The Order of Keywords

We then compare the accuracy in the control of the order of keywords for different methods. Our experimental results show the proposed *Apex* achieves about 97% accuracy for controlling the order of keywords in the generated sentences. The accuracy for

Table 2: Generated text by different models with different order of input keywords. Case 1 and case 2 show that *Apex* can totally control the order of keywords. Case2-1 and 2-2 show that *Apex* can generate diverse text with the same order of input keywords. However, the baselines cannot control the order of keywords.

Item: 半身裙 (skirt)	
Input keywords: 撞色, 流苏, 迷人, 裙摆 (Contrasting color, Tassel, Charming/Charm, Skirt's hemline)	
Model: Apex-PI-2	
Case 1: 撞色(1), 流苏(2), 迷人(3), 裙摆(4) Contrasting color(1), Tassel(2), Charming(3), Skirt's hemline(4)	这款半身裙,采用了 撞色流苏 的设计,丰富了裙身的细节,增添了几分灵动的气息,散发出 迷人 的气息,而 裙摆 的流苏设计,更是增添了几分俏皮感,更显别致。 This skirt is designed with contrasting-color tassel , which enriches its details, adds a bit of instant style and shows a charming fashion. Plus, the skirt's hemline is decorated with tassel, which makes it more playful and chic.
Case 2-1 裙摆(1), 流苏(2), 迷人(3), 撞色(4) Skirt's hemline(1), Tassel(2), Charming(3), Contrasting color(4)	裙摆 处的 流苏 装饰,为整体增添了几分灵动的美感,行走间摇曳出灵动的 迷人 姿态.而裙身的 撞色 线条,则是将活力的气息展现的淋漓尽致,让人看起来更加的与众不同。 The tassel decoration on the skirt's hemline adds a little of instant style, which helps people show a glamorous and charming gesture when walking. Also, the contrasting color of this skirt shows the vitality, making it out of the ordinary.
Case 2-2 (same as above) 裙摆(1), 流苏(2), 迷人(3), 撞色(4) Skirt's hemline(1), Tassel(2), Charm(3), Contrasting color(4)	这款半身裙的 裙摆 采用 流苏 的设计,展现出一种灵动飘逸的美感,行走间展现出女性 迷人 的魅力. 裙身 撞色 的设计,丰富了视觉效果,展现出女性十足的青春活力,彰显出女性的个性与柔美。 The skirt's hemline is designed with tassel , showing an instant style and charm of ladies when walking. Its contrasting color enriches the visual effects, fully showing the vitality and youth of women as well as their personality and beauty.
Model: Seq2seq	
撞色,迷人,裙摆,流苏 Tassel, Contrasting color, Charming, Skirt's hemline	这款半身裙的亮点之处在于 流苏 的设计,流苏的装饰,不仅展现出了与众不同的时尚品味,还增添了几分灵动的美感,而 撞色 的格纹,则展现出了一股复古的气息,彰显出了非凡的时尚品味, 迷人 至极。 The fantastic part of this skirt is tassel . With tassel as the decoration, the skirt not only shows distinctive fashion, but also provides a feeling of instant style. Moreover, its check pattern designed in contrasting color shows vintage style and extraordinary fashion, which is very charming and fantastic.
Model: ExpansionNet	
撞色,迷人,裙摆,流苏 Tassel, Contrasting color, Charming, Skirt's hemline	上半身的 流苏 带设计,让裙身的艺术感更加上升,同时整体的裙身上又展示出秀气利落之感,同时大摆裙的 裙摆 行走间随风摇曳飘逸,行走间凸显潮流气息。 The design of tassel can improve the artistic sense of the skirt. The entire skirt also shows a sense of elegance. Moreover, the skirt's hemline sways with the wind when the wearer is walking, making it looks more fashionable.
Model: DP-GAN	
撞色,迷人,裙摆,流苏 Tassel, Contrasting color, Charming, Skirt's hemline	这质感的 丝绒 面料,让你在行走间散发出浪漫的气息,而裙身的 撞色 设计,更是将女性的优雅气质展现的淋漓尽致.而裙身的 流苏 装饰,则是以撞色的方式呈现出了一种独特的美感,让你在行走间散发出 迷人 的气息。 This velvet fabric can make you send out a romantics taste. The design of contrasting color on the skirt can totally show the elegance of ladies. In addition, its tassel decoration shows a unique aesthetic feeling, making you look very charming when walking.

seq2seq, DP-GAN and ExpansionNet, are 40%, 16% and 15%, respectively. Thus, our method significantly outperforms the baselines in controlling the order of keywords. Table 2 illustrates an example of product descriptions generated by *Apex* and baselines⁴. For case 1 and case 2, we can see that our *Apex* can totally control the orders of keywords in the generated text. For case 2-1 and 2-2, we use the same order of keywords to generate text. It can be seen that *Apex* can generate diverse results with the same input order of keywords. However, the baseline methods sometimes miss some keywords in the generated text. Hence, *Apex* outperforms the baselines in terms of diversity and accuracy.

4.7 Human Evaluation

We ask 25 human graders to evaluate the relevance (accuracy) and fluency of generated sentences by different methods above. Specifically, we randomly sampled 200 item text descriptions from each method identically, and mixed (shuffled) them together without telling graders which method generates which text. Then we provided the graders with the input keywords of items, ground truth and the corresponding generated results. The graders were asked to grade relevance and fluency separately on a three point scale based on the our instructions with demo examples. Score “2” means the generated text is relevant/fluently, and score “1” means it is somewhat relevant/fluently, while score “0” means it is not. To ensure the consistency of graders, we computed the Kendall’s tau coefficient to

measure the correlation among the graders. The computed score is 0.82, which indicates high consistency among graders. In addition, we use the two-tailed t-test to compare the scores of our proposed *Apex-PI* and the baselines. The evaluation results are statistically significant with $p < 0.05$.

Table 3: Scores of fluency and relevance by graders.

Model	Fluency	Relevance	Avg.
Apex-PI-5	1.57	1.67	1.62
Apex-PI-2	1.62	1.72	1.67
Apex-PI-1	1.60	1.72	1.66
Apex-cost-anneal	1.55	1.61	1.58
Apex-cyclical-anneal	1.52	1.62	1.57
Seq2seq	1.55	1.57	1.56
ExpansionNet	1.56	1.50	1.53
DP-GAN	1.28	1.45	1.36

Table 3 illustrates the evaluation results by averaging the scores provided by graders. We can see that *Apex-PI* outperform the baselines in term of fluency and relevance. The fluency and relevance of *Apex-PI-2* are better than those of *Apex-PI-5*. This is because a large KL divergence diversifies the generated text, but it lowers its quality. Besides, DP-GAN does not perform well because it is hard to stabilize the GAN. From this evaluation, we can conclude that KL divergence affects the accuracy of generated text. Thus, it is important to control the KL divergence to achieve the trade off between diversity and accuracy.

⁴The order of keywords may be different in English translation

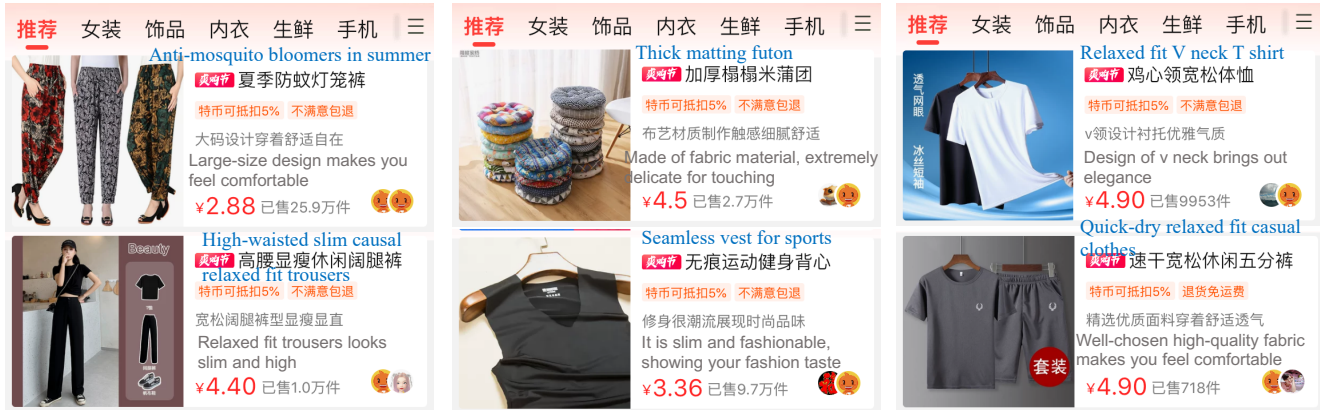


Figure 4: Some examples of generated text for item recommendation in Taobao E-commerce platform. Production description is in blue color while recommendation reason is in darkgray color.

4.8 Online A/B Testing

We also do A/B test experiments to evaluate the performance of the proposed *Apex*. We deploy *Apex* in a real-world product in Taobao E-commerce platform for two applications: product description generation and item recommendation reason. For each item, we randomly choose one or two from some keywords with an average of 4 to generate different product descriptions and recommendation reasons. To get reliable and convincing evaluation results, the A/B test lasted for one week, from Aug.22th to Aug.28th, 2020. Fig. 3 illustrates some examples of generated text for item recommendation in Taobao. In the first A/B test experiment, we use *Apex* to generate short production descriptions for recommended items, and then compare it with the long production descriptions by the existing method. The A/B test results show that *Apex* can improve the Click-Through Rate (CTR) by 13.17%. We also get the statistics about user demographics who are more likely to click, and the results show that our text generator improves CTR by 14.9%, 10.9% for older people and young people, respectively. For the second application, the proposed method generates one sentence to describe the advantages of the recommended items, and then compare it with user reviews and top-K items without reviews. The test result demonstrates that *Apex* can achieve an improvement of about 6.89% and 1.42% in term of CTR over the baseline methods respectively. Therefore, the test experiments further verify the effectiveness of the proposed method.

5 RELATED WORK

In this section, we review the related work on text generation in recent years.

In early work, sequence to sequence (seq2seq) [31] models were widely used for text generation. However, seq2seq often overproduces high-frequency words and lacks diversity. In order to improve diversity, some follow-up work adopted deep generative models, such as VAE [16, 28] and generative adversarial networks (GANs) [11]. One of the most popular models is seqGAN [37], which leverages a discriminator and a generator to play a mini-max game to generate text. Extensions were developed, such as DP-GAN [36], RevGAN [20], RankGAN [21] and LeakGAN [12].

However, these algorithms do not offer means to control the trade-off between diversity and accuracy.

Recent work attempted more controllable text generation. Some techniques control the generated text, conditioned on the attributes of users/items, key phrases, keywords and semantics of interest. For instance, Ni et al. [25] proposed ExpansionNet to generate personalized reviews by controlling user/item attributes and short phrases. In order to control story ending valence or keywords, researchers developed a recurrent neural network (RNN) based generation model [26]. However, they did not consider controlling the order of keywords to diversity generated text.

There are also some works on production description generation in E-commerce [5, 10, 22, 33]. Most of prior studies mainly leverage statistical methods with template to generate product descriptions. Recently, Chen et al. [6?] developed a personalized product description generation model based on knowledge graph. However, none of them is able to finely control the diversity or the order of keywords of generated text.

Different from existing work, we propose a novel controllable text generation model that can manipulate both the order of keywords to fit consumer interest, and the KL-divergence to achieve a controlled trade off between diversity and accuracy.

6 CONCLUSION

This paper developed a novel generative model, called *Apex*, that combines CVAEs with a PI algorithm to controllably generate accurate and diverse text. Specifically, the CVAEs diversify text generation by controlling the order of input keywords. In addition, the designed PI algorithm manipulates the KL divergence to simultaneously achieve good accuracy and diversity, while averting the KL vanishing problem. The evaluation results on a real world dataset demonstrate that our method outperforms the existing generative models in terms of both diversity and relevance. Moreover, it can control the order of keywords in the generated sentences with about 97% accuracy. Finally, we deploy *Apex* in a real-world product in a Chinese E-commerce platform to do A/B testing. Results show that *Apex* significantly improves CTR over the existing methods.

ACKNOWLEDGMENTS

Research reported in this paper was sponsored in part by DARPA award W911NF-17-C-0099, DTRA award HDTRA118-1-0026, the Army Research Laboratory under Cooperative Agreement W911NF-17-20196, NSF CNS 18-15891, NSF CNS 19-32529, and Navy N00014-17-1-2783. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies of the CCDC Army Research Laboratory, DARPA, DTRA, or the US government. The US government is authorized to reproduce and distribute reprints for government purposes notwithstanding any copyright notation hereon.

REFERENCES

- [1] Karl Johan Åström, Tore Hägglund, and Karl J Astrom. 2006. *Advanced PID control*. Vol. 461. ISA-The Instrumentation, Systems, and Automation Society Research Triangle.
- [2] Karl Johan Åström, Tore Hägglund, Chang C Hang, and Weng K Ho. 1993. Automatic tuning and adaptation for PID controllers—a survey. *Control Engineering Practice* 1, 4 (1993), 699–714.
- [3] Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*. 65–72.
- [4] Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. 2015. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349* (2015).
- [5] Zhangming Chan, Xiuying Chen, Yongliang Wang, Juntao Li, Zhiqiang Zhang, Kun Gai, Dongyan Zhao, and Rui Yan. 2019. Stick to the Facts: Learning towards a Fidelity-oriented E-Commerce Product Description Generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 4960–4969.
- [6] Qibin Chen, Junyang Lin, Yichang Zhang, Hongxia Yang, Jingren Zhou, and Jie Tang. 2019. Towards knowledge-based personalized product description generation in e-commerce. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 3040–3050.
- [7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 4171–4186.
- [8] Ondřej Dušek and Filip Jurčiček. 2016. Sequence-to-sequence generation for spoken dialogue via deep syntax trees and strings. *arXiv preprint arXiv:1606.05491* (2016).
- [9] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 1243–1252.
- [10] Shima Gerani, Yashar Mehdad, Giuseppe Carenini, Raymond Ng, and Bitu Nejat. 2014. Abstractive summarization of product reviews using discourse structure. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1602–1613.
- [11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680.
- [12] Jiaxian Guo, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, and Jun Wang. 2018. Long Text Generation via Adversarial Training with Leaked Information. In *AAAI*.
- [13] Joseph L Hellerstein, Yixin Diao, Sujay Parekh, and Dawn M Tilbury. 2004. *Feedback control of computing systems*. John Wiley & Sons.
- [14] Eduard H Hovy, Chin-Yew Lin, Liang Zhou, and Junichi Fukumoto. 2006. Automated Summarization Evaluation with Basic Elements. In *LREC*, Vol. 6. Citeseer, 604–611.
- [15] Zhiting Hu, Haoran Shi, Zichao Yang, Bowen Tan, Tiancheng Zhao, Junxian He, Wentao Wang, Xingjiang Yu, Lianhui Qin, Di Wang, et al. 2018. Texar: A modularized, versatile, and extensible toolkit for text generation. *arXiv preprint arXiv:1809.00794* (2018).
- [16] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P King. 2017. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 1587–1596.
- [17] Oleg Ivanov, Michael Figurnov, and Dmitry Vetrov. 2018. Variational Autoencoder with Arbitrary Conditioning. *arXiv preprint arXiv:1806.02382* (2018).
- [18] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).
- [19] Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017. Adversarial learning for neural dialogue generation. *arXiv preprint arXiv:1701.06547* (2017).
- [20] Pan Li and Alexander Tuzhilin. 2019. Towards Controllable and Personalized Review Generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 3228–3236.
- [21] Kevin Lin, Dianqi Li, Xiaodong He, Zhengyou Zhang, and Ming-Ting Sun. 2017. Adversarial ranking for language generation. In *Advances in Neural Information Processing Systems*. 3155–3165.
- [22] Zachary C Lipton, Sharad Vikram, and Julian McAuley. 2015. Capturing meaning in product reviews with character-level generative text models. *arXiv preprint arXiv:1511.03683* (2015).
- [23] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. 2017. Unsupervised image-to-image translation networks. In *Advances in neural information processing systems*. 700–708.
- [24] Xiaodong Liu, Jianfeng Gao, Asli Celikyilmaz, Lawrence Carin, et al. 2019. Cyclical Annealing Schedule: A Simple Approach to Mitigating KL Vanishing. *arXiv preprint arXiv:1903.10145* (2019).
- [25] Jianmo Ni and Julian McAuley. 2018. Personalized review generation by expanding phrases and attending on aspect-aware representations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 706–711.
- [26] Nanyun Peng, Marjan Ghazvininejad, Jonathan May, and Kevin Knight. 2018. Towards controllable story generation. In *Proceedings of the First Workshop on Storytelling*. 43–49.
- [27] Youbin Peng, Damir Vrancic, and Raymond Hanus. 1996. Anti-windup, bumpless, and conditioned transfer techniques for PID controllers. *IEEE Control systems magazine* 16, 4 (1996), 48–57.
- [28] Stanislau Semeniuta, Aliaksei Severyn, and Erhardt Barth. 2017. A hybrid convolutional variational autoencoder for text generation. *arXiv preprint arXiv:1702.02390* (2017).
- [29] Xiaoyu Shen, Hui Su, Shuzi Niu, and Vera Demberg. 2018. Improving variational encoder-decoders in dialogue generation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
- [30] Guillermo J Silva, Aniruddha Datta, and Shankar P Bhattacharyya. 2003. On the stability and controller robustness of some popular PID tuning rules. *IEEE Trans. Automat. Control* 48, 9 (2003), 1638–1641.
- [31] I Sutskever, O Vinyals, and QV Le. 2014. Sequence to sequence learning with neural networks. *Advances in NIPS* (2014).
- [32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.
- [33] Jinpeng Wang, Yutai Hou, Jing Liu, Yunbo Cao, and Chin-Yew Lin. 2017. A statistical framework for product description generation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. 187–192.
- [34] Qingyun Wang, Qi Zeng, Lifu Huang, Kevin Knight, Heng Ji, and Nazneen Fatema Rajani. 2020. ReviewRobot: Explainable Paper Review Generation based on Knowledge Synthesis. *arXiv preprint arXiv:2010.06119* (2020).
- [35] Wenlin Wang, Zhe Gan, Hongteng Xu, Ruiyi Zhang, Guoyin Wang, Dinghan Shen, Changyue Chen, and Lawrence Carin. 2019. Topic-Guided Variational Autoencoders for Text Generation. *arXiv preprint arXiv:1903.07137* (2019).
- [36] Jingjing Xu, Xuancheng Ren, Junyang Lin, and Xu Sun. 2018. Diversity-promoting gan: A cross-entropy based generative adversarial network for diversified text generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 3940–3949.
- [37] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Thirty-First AAAI Conference on Artificial Intelligence*.
- [38] Hongyu Zang and Xiaojun Wan. 2017. Towards automatic generation of product reviews from aspect-sentiment scores. In *Proceedings of the 10th International Conference on Natural Language Generation*. 168–177.
- [39] Yizhe Zhang, Michel Galley, Jianfeng Gao, Zhe Gan, Xijun Li, Chris Brockett, and Bill Dolan. 2018. Generating informative and diverse conversational responses via adversarial information maximization. In *Advances in Neural Information Processing Systems*. 1810–1820.
- [40] Yizhe Zhang, Zhe Gan, and Lawrence Carin. 2016. Generating text via adversarial training. In *NIPS workshop on Adversarial Training*, Vol. 21.
- [41] Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. *arXiv preprint arXiv:1703.10960* (2017).
- [42] Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Tegygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 1097–1100.

7 EXAMPLES OF GENERATED TEXT BY APEX

Table 4: Generated text by Apex based on the order of input keywords.

Keywords in order	Generated Sentences
磨砂, 系带, 迷人 scrubs, lace, charming	这款皮衣外套采用了 磨砂 质感的面料,手感柔软,穿着舒适; 系带 的设计,勾勒出 迷人 的身材曲线,更显女性的柔美气质,口袋的装饰,丰富了整体的造型,增添了实用性。 This leather jacket is made of scrubs fabric, which feels soft and is comfortable to wear. The design of lace outlines a charming figure, more feminine temperament. The decoration of pocket enriches the overall shape and makes it more practical.
毛衣, 高领, 保暖, 编织 sweater, high-collar warm, weaving	这款 毛衣 采用 高领 设计, 保暖 舒适,精致的 编织 工艺,优雅大方,整体非常的时尚,简约百搭,尽显大气风范。 This sweater is designed with a high collar , warm and comfortable, and its weaving technology is elegant and fashionable, simple and versatile, which can show your grace.
红色, 七分袖, 搭配 red, three quarter sleeve matching.	红色 的连衣裙,尽显女性的知性与优雅。 七分袖 的设计,适合各种不同的身形, 搭配 一双高跟鞋,尽显女性的优雅。 The red dress shows women's grace and elegance. The three quarter sleeve is suitable to different body shapes. It can show women's elegance when matching it with a pair of high heels.
连体, 条纹, 图案 one-piece, stripe, pattern	这款 连体 连衣裙,简约大方,经典 条纹图案 ,时尚大方,时尚潮流,尽显女性的优雅气质,宽松的款式不挑身材,穿着舒适自在。 This one-piece dress is simple and generous, and the classic striped pattern is stylish and fashionable, full of women's elegant temperament. The loose style does not pick your figure, and it is comfortable to wear.
镂空, 破洞, 宽松 hollow, hole, loose	镂空 的 破洞 设计,让这款毛衣看起来更加的有个性,带有小小的性感, 宽松 的款式,让你穿着舒适不受负重感,很好的凹造型。 The hollow hole design makes this sweater look more unique, with a little sexy, loose style, making you comfortable to wear without a sense of weight, very good for posing.
撞色, 宽松, 图案 contrasting, loose, pattern	撞色 格纹的设计,经典大方,带有文艺气息, 宽松 的版型,不挑身材,轻松驾驭;牛角扣的设计,丰富了整体的层次感,显得十分的复古, 图案 的设计,显得精美别致。 The design of the contrasting checkered pattern is classic and elegant, with a good styling, loose feeling, not picking the figure, easy to control, and the design of the horn buckle enriches the overall layering, giving a retro feeling, and the design of the pattern is exquisite and chic.
中长款, 立体, 纯色 long style, stereoscopic, pure color	简约的 中长款 , 立体 挺括,提升穿着品味.宽松的廓形,不挑身材,非常好驾驭。 纯色 的设计,简约大方,营造出沉静秀气的少女感,简约的欧美风格,时尚百搭。 Simple long style , stereoscopic and crisp, enhance the wearing taste. Loose silhouette, not picking the figure, is very easy to control. The pure color design, simple and elegant, creating a calm and delicate girlish feeling, minimalist European style, fashionable and versatile.
条纹, 下摆 stripe, hem	经典的西装领设计,简约大方,将女性的干练与利落气质完美的展现出来.经典的 条纹 拼接,个性独特,丰富了视觉效果,显瘦又显高挑。 下摆 的A字设计,优化身材比例,展现出高挑的身材。 The classic suit collar design, simple and elegant, perfectly showing women's competence and neat temperament. The classic stripe stitching shows uniqueness, enriches the visual effect, and make you look thin and tall. The A-shaped design of the hem optimizes the proportion of the figure and shows a tall figure.
粉色, 圆领, 迷人 pink, round neckline charming	这款连衣裙采用了 粉色 的色调,看起来清新甜美,带来了甜美的少女气息,经典的 圆领 设计,修饰脸型,露出 迷人 的锁骨,包臀的款式,凸显性感的女人味。 This dress uses pink tones, looks fresh and sweet, and brings a sweet girlish style. The classic round neckline design modifies the face and reveals a charming look clavicle, the sheath style highlighting sexy femininity.
大牌, 连帽 big-name, hooded design	这款卫衣版型宽松,穿着舒适,且立体挺括,彰显 大牌 风范. 胸前的Logo刺绣,极具个性,也体现了品牌的魅力。 连帽 的设计,休闲时尚,更能凸显休闲的气质。 This hoodie is loose-fitting, comfortable to wear, and stereoscopic, showing its big-name style. The logo embroidery on the chest is very personalized and also reflects the charm of this brand. Hooded design , which is casual and fashionable, highlights the casual temperament.
毛呢, 格纹, 翻领 wool, plaid, lapel	毛呢 大衣,是冬季的不二之选,经典的 格纹 ,给人一种复古的优雅感。 翻领 设计,撞色线条点缀,给人一种优雅的感觉.衣身两侧的口袋装饰,方便实用。 Wool coat is the only choice in the winter. Classic plaid gives a retro elegance. The lapel design is embellished with contrasting color, giving an elegant feeling. Moreover, the pocket on both sides of the body is convenient and practical.
优质, 系带, 超大 high-quality, lace, large	这款风衣外套采用了 优质 面料,手感柔软舒适,穿着体验非常好.领口采用 系带 设计,增加了亮点,穿着时髦。 超大 口袋,实用方便。 This windbreaker jacket is made of high-quality fabrics and feels soft and comfortable, giving pleasant wearing experience. The design of lace in neckline is fashionable. Plus, its large pocket is practical and convenient.