# TextDream: Conditional Text Generation by Searching in the Semantic Space

Weidi Xu[1], Haoze Sun[2], Chao Deng[1], and Ying Tan[1]

[1]Key Laboratory of Machine Perception (Ministry of Education)
School of Electronics Engineering and Computer Science, Peking University, Beijing, 100871, China
[2]Sogou, Inc.
wead_hsu@pku.edu.cn, pkucissun@foxmail.com, cdspace678@pku.edu.cn, ytan@pku.edu.cn

***Abstract*—Conditional text generation is a fundamental task in natural language generation. Traditional conditional generative models build conditional probability distributions over the given labels. However, categorical label information is usually very abstract, e.g., sentiment, and it is difficult to be disentangled from the content. Therefore, instead of generating text by modeling conditional probability distribution, we propose a novel text generation method *TextDream* through searching in the semantic space. Specifically, in this method, a random text seed is initially given and the new text is generated by local search operation. The generation procedure is guided by a fitness function, typically a classification model. Text with higher fitness will be preserved. This procedure loops until the qualified solution is found. Experimental results show that our method is able to generate more diverse text compared with advanced conditional generative models.**

## I. Introduction

Generative models have drawn much attention over machine learning community. Conditional generative model is a kind of specialized generative model, which aims to generate the output sample conditioned on some given input. Along with the growth of interests in the generative models, many conditional generative models has been put forward, and they are applied successfully in many tasks, e.g., machine translation [1], image caption [9, 23] and conditional image generation [16, 18]. The machine translation takes source sentence as input and output a sentence in the target language. The goal of image caption models aims at describing an image using natural language. The conditional image generation, on the contrary, generates images conditioned on the given text. All of these three tasks have one common property that the input, no matter text or image, contains amounts of information, so that the generation can be done in a straightforward way. In other words, the mapping from the source space to the target space is almost identical and the diversity is not an essential issue.

Different from these problems, this paper considers text generation conditioned on a single categorical label. Although it seems to be much easier than aforementioned tasks (e.g., image caption), it has been shown to be difficult to generate both diverse and meaningful text [7, 22]. Machine translation models can make use of simple sequence-to-sequence framework, but label-based conditional generation requires the diversity in the generation. Otherwise, due to the limitation of label categories, the generative model will collapse to several modes and the generated samples become very similar. Recent works utilize advanced deep generative models, e.g., variational autoencoders (VAEs) [11] and generative adversarial nets (GANs) [5], to introduce the randomness. In these models, the label information and the semantic space are explicitly disentangled. Typically, Xu et al. put forward a conditional generative model based on variational autoencoder. In their work, the encoder is used to extract the feature that is independent on the label information, while the decoder is conditioned on both label and encoded feature. Similarly, Hu et al. also proposed a VAE-based method to generate text, enhanced by the auxiliary classifiers.

We argue that categorical label information can be very abstract, and is hard to be disentangled from the other semantical information. Dieng et al. suggested that instead of representing label information as a discrete feature, it may be better to keep the label information in the unified distributed semantic space. As shown in [3], the authors proposed a method to extract the distributed topic vector, rather than an explicit discrete category, which achieves better performance in both classification and generation tasks. In this way, the label information is not especially represented as a conditional input, it is distributed across the representation space. Therefore, instead of modeling the conditional probability distribution, we turn to construct a unified semantic space.

To sample the data with certain property, we can make use of searching method to find the corresponding text in this semantic space. Searching-based generative model [17] has shown state-of-the-art performance in image generation task, in terms of generation quality and diversity. The model is able to generate various images conditioned on different labels by plugging different classifiers. Briefly, the gradient is applied to the input image to maximize the fitness function, e.g., the probability of being classified as "cat". The method is verified for image data, as the image data is in a continuous space, which is naturally suitable with gradient-based method.

Unfortunately, the text data is composed of the discrete symbols and this method [17] is no longer valid. To approach this problem, we propose *TextDream* to search in the continuous semantic space, and make an initial attempt to the searching-based method for text data. In the proposed method, the text is gradually generated towards a certain category, in the principle

978-1-5090-6017-7/18/$31.00 ©2018 IEEE

of evolutionary algorithms. The algorithm briefly goes as follows. In each turn, the candidate samples are produced by local search operations, where an encode-decode approach is used. Then the candidates are evaluated by a fitness function. Samples with higher scores survive. Eventually, a text with high fitness is produced.

Despite of its simplicity, experimental results demonstrate that this approach is capable of generating not only fluent but also diverse sentences given different conditional labels. Searching in text domain is much easier, making it a potential practical generative model. We have also investigated two kinds of autoencoders and verified their performance.

The article is organized as follows. In the next section, the framework and the details of our method are presented. Then, experimental results will be shown in the Sec. IV. In the last, we conclude with a discussion.

## II. Related Work

Actually, there are two lines for generative models that are conditioned on the given label.

The first line directly tries to build a conditional probability distribution $p(x|y)$ or $p(x|y,z)$, where the $x$ is the input data, $y$ is the given label and $z$ is a stochastic variable. These models originate from the recently proposed deep generative models, e.g., VAE and GAN. In this case, the models for both image data and text data follow the same framework, except that the decoder varies. For image data, the decoder is typically a MLP or CNN [12, 15, 19, 25]. While for text data, the decoder is required to consider the sequential nature of text, and RNN is usually adopted [7, 24, 26]. All of these models have shown strong performance on the generation tasks.

The second line is drawn by modeling the distribution $p(x)$ or $p(x|z)$, ignoring the condition $y$. In other words, the $x$s with various $y$s are distributed on the entire space. Hence, if we want to extract a $x$ with given $y$, we have to walk around the space to find one. Nguyen et al. proposed a method to search the image in the raw space. The gradient from the auxiliary classifier is propagate to the input image to maximize the probability $p(y|x)$. Each updating can be regarded as a searching step. After several searching steps, the target sample will be extracted. This method has demonstrated best performance in image generation task. It indicates that modeling $p(x)$ is likely to be more expressive than modeling the conditional distribution $p(x|y)$. However, the method [17] is not applicable for NLP tasks, as the text data is discrete in nature and the gradient is not valid. To remedy this problem and make an attempt to verify whether the searching-based method can achieve better performance in text data, we propose a novel method to search in the semantic space. In the following, we will introduce the algorithm.

## III. TextDream

Generic conditional generative model aims to build a conditional generative distribution, and the data can be generated by given some certain labels. Having observed that the difficulty in modeling such distribution in the latent space, we shift to find the data via searching in the complete latent space. Formally, the goal is to find a text to maximize a certain condition evaluation:

$$\arg\max_{z \in L} f_{eval}(f_{dec}(z)), \tag{1}$$

where $z$ is a point in the latent space $L$, $f_{dec} : L \to S$ is a function to decode $z$ into the original data space $S$, so that the evaluation function $f_{eval}$ can give the scalar fitness score.

Our work is based on the assumption that the text can be encoded in a single semantic space, and the space is smooth enough so that the each point is well defined. By randomly searching in latent semantic space, various samples can be drawn and decoded into different sentences. The samples with high fitness value will be preserved. To increase the diversity, we also proposed an approach to alleviate the selection pressure. The framework is shown in Fig. 1.

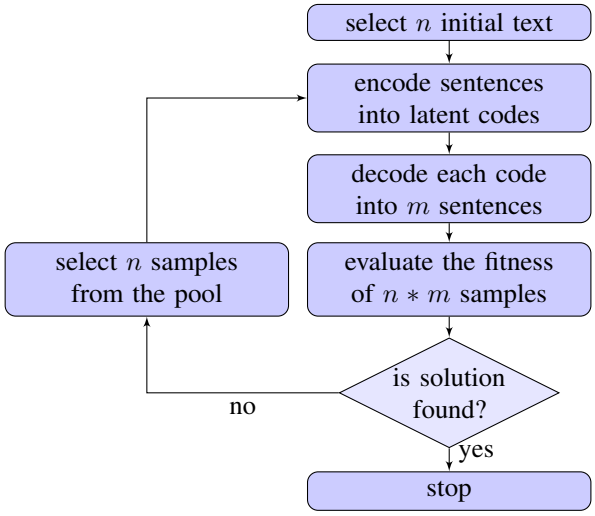


Fig. 1. The flowchart of the proposed method.

### A. Local Search by Autoencoder

In [17] , the data is sampled from the a latent space by a Markov Chain Monte Carlo (MCMC) sampler (*MALA-approx* specifically). Thanks to the continuity of the image data, the sampler can derive new samples by calculating the gradient of latent variable (w.r.t. the joint probability). However, as the text data is represented by discrete symbols, the loss signal from the fitness function cannot be propagated directly to the input sequence. Therefore, we make use of the power of evolutionary algorithm instead, i.e., to find the solution by stochastic searching and selection.

Here we utilize an encode-and-decode way to generate new samples. At each iteration, the text is firstly encoded by a RNN, typically a LSTM network [6]. When the encoding is available, another RNN decodes them into another sequences. Since multiple different decodings can be draw from a single input, there are differences between input and output sentences. This property can be used as a mutation operation to derive new samples. As the autoencoder is trained to output the sample that is exactly same with input, the decoded offsprings are

very similar with the input sequence, which guarantees that the searching process is smooth in the raw space.

This paper studies two typical autoencoders. The first model is SkipThought [13], which demonstrates strong performance in learning sequence representation. The model not only decodes the encoding to the input sequence, but also predicts previous and next sentences in the context. Specifically, the sequence is fed into the encoder to compute a representation vector. Then the vector is used to predict three sentences, i.e., previous sentence, current sentence and next sentence. This implementation is useful, because more information is contained in the latent semantic space.

The second model is variational recurrent neural network [2], to which we refer as VRNN in this work. It is a powerful generative model and has been verified to be effective at extracting global features from sequences (e.g., sentiment, topic and style). This property plays a crucial role in producing meaningful sentences. The model is trained by maximizing the variational lower bound. Instead of encoding the input sequence into a single point, it is encoded into a sharp distribution. Therefore, the latent semantic space of the VAE is much more smooth than the SkipThought.

Note that when an autoencoder is fully trained, the decoded data will be almost exactly same with the input data. Here, the randomness comes from 2 parts, 1) the uncertainty in the autoencoder, 2) the diversity from the beam-search. These two sources endow the model with the exploration ability.

### B. Evaluation

Actually we do not impose any constrains to the evaluation function $f_{eval}$, so long as the function is able to give the fitness evaluation about a certain input text. As the searching is in the latent semantic space, the smoothness of $f_{eval}$ is not required. In the experiment, a simple RNN classifier is adopted. The model predicts whether the input sequence is belong to a certain category.

### C. Regularization

The meaningless sentences can be produced in the evolution steps. These sentences will mislead the entire population into a local optimum and the population will converge to a single meaningless solution. As will be shown in the experiment, the classifier can be easily fooled by the irregular inputs. To void such situation, the abnormal solutions should be filtered. Therefore, a regularization trick is adopted in the generation. We only keep the decoded text with low perplexity. In the experiments the sentences with perplexity less than 8 will be preserved and the others will be discarded.

### D. Selection

Selection is the critical component in balancing both exploration and exploitation. To make sure that the good solution will not be discarded, elite selection is used. This paper investigates two kinds of selection method. The first one is the roulette wheel selection based on the evaluation function $f_{eval}$. The second is motivated by the selection in the firework algorithm [14, 20, 21, 27, 29].

*a) selection based on the fitness:* Selection method based on fitness is widely used over evolutionary algorithms. The method is quite simple. The fitness of each sentence is given by the evaluation function $f_{eval}$. And the probability of being selected is :

$$p(x_i) = f_{eval}(x_i) / \sum_j f_{eval}(x_j)\,. \tag{2}$$

Following this probability, the population in the evolution will quickly converge.

*b) selection based on the diversity:* To increase the diversity among the text population, a selection method is proposed based on the diversity score:

$$D(x_i) = \sum_j d(x_i, x_j), \tag{3}$$

where $d$ is a distance metric, $x_i$ is a sample in the candidate pool. We used *Levenshtein distance* to measure the distance between to sentences. Each sample will be selected with the probability:

$$p(x_i) = D(x_i) / \sum_j D(x_j). \tag{4}$$

Besides the best generation, another $n-1$ different sequences will be selected.

### E. Algorithm

Combining all aforementioned components, we propose the TextDream algorithm in Algorithm 1. The models, i.e., the encoder, the decoder and the evaluation model, are trained in advance. The datasets used to train these models are not necessarily same, and hence various combinations between the autoencoder and the classifier are valid. The framework shares the similar pipelines with the evolutionary algorithms (EAs), e.g., Genetic Algorithm [4]. The differences between our algorithm and typical EAs are: 1) The solutions in the group are not numerical, but sentences which is composed of many discrete symbols. 2) To mutate the sentence individuals, we first convert the sentence into a continuous numerical space, and then create offsprings using the encoding vector. 3) The evaluation model can be replaced by various other models, which endows us with the ability to generate different kinds of data samples. Actually, the framework is applicable for many other tasks. By converting the raw input into a semantic space, the new samples can be drawn by searching in this space. As the latent space is semantically smooth, the searching can be done in a steady speed.

## IV. Experiments

The conducted experiments aim at answering following questions? 1) Whether the proposed method is able to generate sentences, conditioned on the given label? 2) How effective is the proposed method? 3) Can it compete with other advanced generative model? 4) What is the difference between the different configurations?

**Algorithm 1** Framework of TextDream.

**Input:** Number of sentences in the population $n$; Number of samplings for each sentence $m$; The encoder $f_{enc}$ of the trained autoencoder; The decoder $f_{dec}$ of the trained autoencoder; The fitness function $f_{eval}$; The fitness threshold $t$;

**Output:** The best sentence $s_b$ after searching process;

Initialize $n$ sentences $S = \{s_1, s_2, ..., s_n\}$ as the population;
$s_b \leftarrow \arg\max_{s_i}(f_{eval}(s_i))$;
**while** $f_{eval}(s_b) <$ t **do**
  **for** $s_i$ in the population **do**
    $z_i \leftarrow f_{enc}(s_i)$;
    $o_i^0, o_i^1, ...o_i^m \sim f_{dec}(z_i)$;
  **end for**
  $O \leftarrow \{o_i^j : 1 \leq i \leq n, 1 \leq j \leq m\} \cap S$;
  Select $n$ sentences from the offspring group $O$ as $S$;
  $s_b \leftarrow \arg\max_i(f_{eval}(s_i))$;
**end while**
**return** $s_b$;

### A. Experimental Setting

Before generating sentences using TextDream, three components should be trained in advance. For the classifier, we used a simple LSTM classifier. The number of units in the LSTM cell is 512 and the state at final step is fed to a fully connected layer for the prediction. For the autoencoder, we investigated 2 powerful models, i.e., SkipThought and VRNN. The SkipThought is implemented using GRU network with 1024 cell units. The encoder network is trained by minimizing the log likelihood of sentences around the context. There are 3 decoders available after training. The decoder used to reconstruct the input sentence is used as the decoder $f_{dec}$. The VRNN is implemented using GRU network with 1024 cell units as well. The dimension of the latent variable is 100. According to the [2], the higher dimension of the latent space does not bring additional improvement.

For the classifier, the AG's News [28] dataset is used for training. AG's News is a large topic classification dataset, which consists of four new topics. The classifier is trained to predict whether the given sentence is about the sports. For the autoencoder, the AG's News and BookCorpus [30] datasets are used for training. The BookCorpus dataset is a large text corpus. It is usually used to train unsupervised models. We use two different datasets here to verify whether the proposed model is valid in domain adaption setting. When the autoencoder is trained on the BookCorpus dataset, domain adaption ability of our method is the essential issue that affects the generation performance.

All these models are optimized end-to-end using the ADAM [10] optimizer with learning rate of 1e-3. For the VRNN, the cost annealing trick [2, 8] was adopted to smooth the training by gradually increasing the weight of KL cost from zero to one. Gradient clip is set to be 5 and the word embeddings are initialized randomly.

The hyper-parameters of the TextDream are chosen without careful design. The stop threshold $t$ is set to be 0.98 across all the experiments. The number of individuals in the population $n$ is 10 and each individual produces 10 offsprings, i.e., $m = 10$. During the generation steps, the generated samples with log perplexity less than 8 will be removed.

### B. Conditional Generation

The first question to answer is whether the proposed method is able to generate the target samples. To verify this, we evaluated the performance by the success rate. The generation is regarded as "success" if the sample can be generated within 50 steps:

$$SR = \frac{\sum_{i=1}^{N_s} 1\{f_{eval}(s_i) > t\}}{N_s}, \tag{5}$$

where the $N_s$ is the number of generated samples. Although the classifier is not perfect and there may be deviation when comparing these models, the SR is a valuable criterion to measure the performance.

Table I demonstrates the SR of the our methods. ST/VRNN denotes that TextDream method is using SkipThought/VRNN as the autoencoder. SF/SD denotes that the TextDream is using the selection method based on fitness/diversity. For example, TextDream (ST, SF) means that the model is configured with SkipThought autoencoder and fitness-based selection method. All the experiments here is based on the components trained on the AG's News. We compare our proposed method with the conditional VRNN (CVRNN). Different from our method, the CVRNN directly models the conditional probability, i.e., $p(x|y)$. The CVRNN is implemented by ourselves following the hyper-parameter setting in [24]. All the models are trained until the performance on the validation dataset does not improve anymore. As illustrated in Table I, our methods outperform CVRNN remarkably. No matter what kind of configuration is adopted by the TextDream, the model almost guaranteed to generate the sample successfully. In contrast, the CVRNN will possibly fail to generate the sample that satisfies the requirement.

TABLE I
THE SUCCESS RATE OF THE MODELS.

| Model | SR | Round | Dis1 | Dis2 |
|---|---|---|---|---|
| CVRNN | 0.91 | 1* | 0.345 | 0.726 |
| TextDream (ST, SF) | 0.97 | 9.40 | 0.392 | 0.863 |
| TextDream (ST, SD) | 0.99 | 77.92 | 0.317 | 0.756 |
| TextDream (VRNN, SF) | **1.00** | 2.15 | **0.436** | **0.878** |
| TextDream (VRNN, SD) | **1.00** | 2.05 | 0.400 | 0.772 |

Even though the model is able to achieve the goal of conditional generation, the efficiency is another important issue in practice. Table I shows the average rounds that the method used to produce the desired samples, i.e., $p(y|x) > t$. The experimental results illustrate that the TextDream can generate within 10 rounds, which is comparable with the CVRNN (the generation in the CVRNN only requires one feedforward step).

The iterations used in the TextDream is acceptable. Fig. 2 shows the curve of the fitness w.r.t. the number of iterations. The VRNN-based TextDream can produce the target sample within 2 rounds, while SkipThought-based model takes much more iterations to maximize the fitness. And the VRNN has a lower variation comparing to SkipThought in this case.

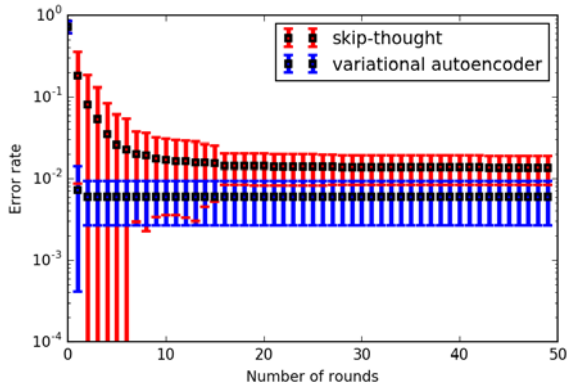


Fig. 2. Error rate w.r.t. the number of iterations. The vertical axes indicates the probability of being misclassified. The horizontal axes indicates the number of iterations. The selection method used here is based on the diversity (SD).

We also investigated the model's ability of generating diverse samples. Diversity is an essential index to evaluate the performance of a generation model. The models that modeling the conditional probability, e.g., CVRNN, tend to converge into several modes and therefore, these models tend to produce limited sentences. Here the DISTINCT-1 and DISTINCT-2 are used to evaluate the diversity level of generated samples. DISTINCT-$n$ denotes the percentage of unique $n$-gram among all the sentences. The results are shown in the Table I. In contrast with our assumption, the fitness-based selection method has a more strong performance over the diversity-based selection. We suggest that the selection method based on the diversity score will probably miss many feasible solutions so that the diversity is deteriorated. The further investigation remains as a future work. When comparing to the CVRNN, all of the proposed models demonstrate better performances. These results indicate that our method can produce more diverse sentences against traditional methods and verify that our method can be regarded as an alternative method in the conditional generation.

### C. Analysis of Different Autoencoders

According to Table I, the autoencoder plays an important role in the TextDream. The performance varies a lot between two kinds of autoencoders, i.e., SkipThought and VRNN. In most evaluations, the model using SkipThought is outperformed by that using VRNN. VRNN-based model is more computational efficient as the average rounds used is much fewer than SkipThought-based model. This is also illustrated in 2. VRNN-based model has a higher success rate. It can certainly produce the qualified sample within 50 iterations. And VRNN-base model demonstrates more strong ability in generating diverse sentences. Overall, the VRNN is consistently better than SkipThought as a component in the TextDream. Therefore, the VRNN is suggested as a standard component.

The intuitive explanation can be given as follows. The VRNN benefits from its definition over the latent space. The input data is encoded into a sharp distribution that locates in a small region in the latent space, rather than a single point in the SkipThought. Hence the latent space of VRNN is trained to be much smoother and the ill-defined points can be neglected. During the process of TextDream, various sentences will be sampled on the fly and many of them are not seen in the training. These samples will lay in the points in the latent space that are far from the well-defined region. VRNN-based model can alleviate this problem by making the latent space smoother.

### D. Domain Adaptation

It is also interesting to study if the proposed method is able to adapt between different domains. The method is naturally suitable for domain adaption setting as the generative model can cope with any kind of evaluation function ideally. When the generative model is trained using different dataset from the classifier, it can be regarded as a domain adaption experiment. In this paper, we conduct experiments using two datasets. The generative model, i.e., autoencoder, is trained using two datasets respectively while the classifier remains fixed. Table II shows the experimental results about domain adaption. The experiment with '*' is conducted with domain adaption. Experimental results indicate that it is more difficult to produce the qualified sentence when the generative model and the classifier are not compatible. All the scores are lower in the domain adaption setting. However, it is reasonable as the different domains bring the discrepancy and the discrepancy will mislead the generation process. And note that when considering domain adaption, the VRNN is outperformed by the SkipThought due to the strong regularization of the VRNN. The converge curve of the model using SkipThought and the VRNN are shown in Fig. 3 and Fig. 4 respectively.

TABLE II
THE SUCCESS RATE OF THE MODELS.

| Model | SR | Round | Dis1 | Dis2 |
|---|---|---|---|---|
| TextDream (ST, SF) | 0.97 | 9.40 | 0.392 | 0.863 |
| TextDream (ST, SF) * | 0.77 | 31.6 | 0.376 | 0.860 |
| TextDream (ST, SD) | 0.99 | 77.92 | 0.317 | 0.756 |
| TextDream (ST, SD) * | 0.91 | 21.49 | 0.261 | 0.617 |
| TextDream (VRNN, SF) | 1.00 | 2.15 | 0.436 | 0.878 |
| TextDream (VRNN, SF) * | 0.02 | 50.43 | 0.325 | 0.817 |
| TextDream (VRNN, SD) | 1.00 | 2.05 | 0.400 | 0.772 |
| TextDream (VRNN, SD) * | 0.43 | 42.26 | 0.268 | 0.775 |

### E. Case Study

Another explorative evaluation of the model's ability to comprehend the method is to dive into the generation process. We illustrate several cases in Table III. We save the population at every iteration and trace back to draw the generation process.

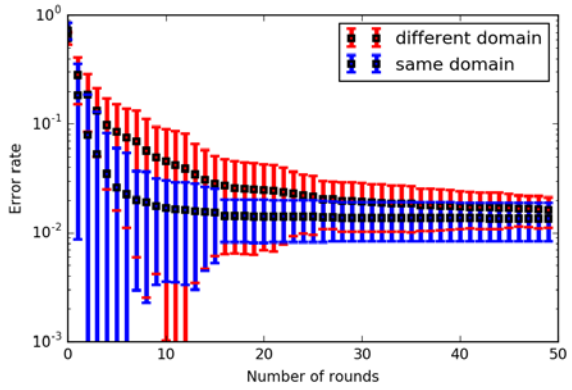


Fig. 3. Error rate w.r.t. the number of iterations using SkipThought.

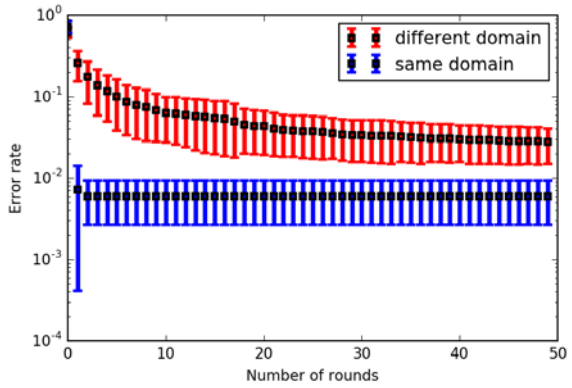


Fig. 4. Error rate w.r.t. the number of iterations using VRNN.

As shown in the table, the sentences about the sports (recall that the sport category is used in our experiments) are successfully drawn within few steps. The sentences in the successive steps are semantically similar. By gradually searching in the semantic space, we eventually get what we want.

TABLE III
SEVERAL CASES OF THE GENERATION PROCESS.

| |
|---|
| → seattle center robbie UNK once took a swing at buffalo quarterback drew bledsoe .<br>→ maryland still has a chance , but improbable , was missing last season .<br>→ basketball players use video games to hone their skills. |
| → while the card will cost nearer 30 , rather than the 12 .<br>→ but drexler never dreamed he would be inducted into the basketball hall of fame . |
| → under the terms of the agreement , cisco will pay approximately $ 200 million in cash and options .<br>→ under the terms of his contract , wagner , 33 , will be paid $ UNK next season . |
| → major us airlines yesterday reported heavy third quarter losses , under pressure from record fuel prices and fierce competition with budget carriers .<br>→ major league baseball has yet to reach an agreement with peter angelos on on a deal that would financially protect the orioles . |

## V. CONCLUSION

A simple yet efficient method is proposed to generate text conditioned on a single label. In contrast with the traditional methods that modeling the conditional probability, the proposed method turns to searching in the semantic space. Massive experimental results show that our method can be an alternative method.

## ACKNOWLEDGMENT

This work was supported by the Natural Science Foundation of China (NSFC) under grant no. 61673025 and 61375119 and Supported by Beijing Natural Science Foundation (4162029), and partially supported by National Key Basic Research Development Plan (973 Plan) Project of China under grant no. 2015CB352302.

## REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[2] Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *The International Conference on Learning Representations (ICLR)*, Caribe Hilton, San Juan, Puerto Rico, 2016.

[3] Adji B Dieng, Chong Wang, Jianfeng Gao, and John Paisley. Topicrnn: A recurrent neural network with long-range semantic dependency. *arXiv preprint arXiv:1611.01702*, 2016.

[4] David E. Goldberg. Dynamic system control using rule learning and genetic algorithms. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence. Los Angeles, CA, USA, August 1985*, pages 588–592, 1985. URL http://ijcai.org/Proceedings/85-1/Papers/112.pdf.

[5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

[6] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735.

[7] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. Controllable text generation. *arXiv preprint arXiv:1703.00955*, 2017.

[8] Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. How to train deep variational autoencoders and probabilistic ladder networks. *arXiv preprint arXiv:1602.02282*, 2016.

[9] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. *arXiv preprint arXiv:1412.2306*, 2014.

[10] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *The International Conference on Learning Representations (ICLR)*, San Diego, USA, 2015.

[11] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *The International Conference on Learning Representations (ICLR)*, Banff, Canada, 2014.

[12] Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014.

[13] Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302, 2015.

[14] Jianhua Liu, Shaoqiu Zheng, and Ying Tan. The improvement on controlling exploration and exploitation of firework algorithm. In *Advances in Swarm Intelligence, 4th International Conference, ICSI 2013, Harbin, China, June 12-15, 2013, Proceedings, Part I*, pages 11–23, 2013. doi: 10.1007/978-3-642-38703-6_2. URL https://doi.org/10.1007/978-3-642-38703-6_2.

[15] Lars Maaløe, Casper Kaae Sønderby, Søren Kaae Sønderby, and Ole Winther. Auxiliary deep generative models. *arXiv preprint arXiv:1602.05473*, 2016.

[16] Elman Mansimov, Emilio Parisotto, Jimmy Lei Ba, and Ruslan Salakhutdinov. Generating images from captions with attention. *arXiv preprint arXiv:1511.02793*, 2015.

[17] Anh Nguyen, Jason Yosinski, Yoshua Bengio, Alexey Dosovitskiy, and Jeff Clune. Plug & play generative networks: Conditional iterative generation of images in latent space. *arXiv preprint arXiv:1612.00005*, 2016.

[18] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 3, 2016.

[19] Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 2226–2234, 2016. URL http://papers.nips.cc/paper/6125-improved-techniques-for-training-gans.

[20] Ying Tan and Yuanchun Zhu. Fireworks algorithm for optimization. In *Advances in Swarm Intelligence, First International Conference, ICSI 2010, Beijing, China, June 12-15, 2010, Proceedings, Part I*, pages 355–364, 2010. doi: 10.1007/978-3-642-13495-1_44. URL https://doi.org/10.1007/978-3-642-13495-1_44.

[21] Ying Tan, Chao Yu, Shaoqiu Zheng, and Ke Ding. Introduction to fireworks algorithm. *IJSIR*, 4(4):39–70, 2013. doi: 10.4018/ijsir.2013100103. URL https://doi.org/10.4018/ijsir.2013100103.

[22] Ankit Vani and Vighnesh Birodkar. Challenges with variational autoencoders for text.

[23] Kelvin Xu, Jimmy Ba, Ryan Kiros, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. *arXiv preprint arXiv:1502.03044*, 2015.

[24] Weidi Xu, Haoze Sun, Chao Deng, and Ying Tan. Variational autoencoder for semi-supervised text classification. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 3358–3364, 2017. URL http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14299.

[25] Xinchen Yan, Jimei Yang, Kihyuk Sohn, and Honglak Lee. Attribute2image: Conditional image generation from visual attributes. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV*, pages 776–791, 2016. doi: 10.1007/978-3-319-46493-0_47. URL http://dx.doi.org/10.1007/978-3-319-46493-0_47.

[26] Zichao Yang, Zhiting Hu, Ruslan Salakhutdinov, and Taylor Berg-Kirkpatrick. Improved variational autoencoders for text modeling using dilated convolutions. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 3881–3890, 2017. URL http://proceedings.mlr.press/v70/yang17d.html.

[27] Chao Yu, Lingchen Kelley, Shaoqiu Zheng, and Ying Tan. Fireworks algorithm with differential mutation for solving the CEC 2014 competition problems. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2014, Beijing, China, July 6-11, 2014*, pages 3238–3245, 2014. doi: 10.1109/CEC.2014.6900590. URL https://doi.org/10.1109/CEC.2014.6900590.

[28] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, pages 649–657, 2015.

[29] Shaoqiu Zheng, Junzhi Li, Andreas Janecek, and Ying Tan. A cooperative framework for fireworks algorithm. *IEEE/ACM Trans. Comput. Biology Bioinform.*, 14(1):27–41, 2017. doi: 10.1109/TCBB.2015.2497227. URL https://doi.org/10.1109/TCBB.2015.2497227.

[30] Yukun Zhu, Ryan Kiros, Richard Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *arXiv preprint arXiv:1506.06724*, 2015.