Switch-GPT: An Effective Method for Constrained Text Generation under Few-Shot Settings (Student Abstract)

Chang Ma[*1], **Song Zhang**[*12], **Gehui Shen**[1], **Zhihong Deng**[1]
[1]Peking University [2]ByteDance
{changma, songz, jueliangguke, zhdeng}@pku.edu.cn

Abstract

In real-world applications of natural language generation, target sentences are often required to satisfy some lexical constraints. However, the success of most neural-based models relies heavily on data, which is infeasible for data-scarce new domains. In this work, we present FewShotAmazon, the first benchmark for the task of Constrained Text Generation under few-shot settings on multiple domains. Further, we propose the Switch-GPT model, in which we utilize the strong language modeling capacity of GPT-2 to generate fluent and well-formulated sentences, while using a light attention module to decide which constraint to attend to at each step. Experiments show that the proposed Switch-GPT model is effective and remarkably outperforms the baselines. Codes will be available at https://github.com/chang-github-00/Switch-GPT.

Introduction

Constrained text generation (CTG) is a vital research problem for various applications, including neural machine translation, task-oriented dialogues, and abstractive text summarization.

Prior tasks can be classified into two categories: (1)hard-constrained generation, where the inclusion of certain keywords are mandatory in generated results; and, (2)soft-constrained generation, where the generated sentence is only required to be semantically related to a given sentence. While Soft-constrained generation models are easier to design and tend to generate more coherent sentences, missing keywords lead to the loss of pivotal facts. Hard-constrained generation, however, involves intricate design of network architectures and time-consuming sampling. Recently, fine-tuning on large-scale pre-trained language models provide new opportunities to CTG. Chen et al. (2019) used a GPT model alongside attention mechanism to tackle few-shot learning on table-to-text. POINTER (Zhang et al. 2020) incorporates pre-trained language models on an insertion-based scheme and achieves state-of-the-art (SOTA) results.

Although previous models generate reasonable results, performance relies on large training datsets, e.g., 160K fine-tuning samples for POINTER. Such data-hungry nature makes it difficult for models to be adopted into real-world scenarios, especially on new domains where data is scarce. This leads us to study this problem : *Can we use the prior knowledge from pre-trained models efficiently, and learn to generate constrained text from only a handful of samples?*

Inspired by this hardship, we propose the task of few-shot CTG, and try to make the best of few training samples. We have developed a new benchmark FewShotAmazon and also propose a new metrics ΔBLEU to capture the ability of models to connect keywords smoothly. We believe that the FewShotAmazon benchmark can inspire future research to address CTG realistically. Also, we develop the Switch-GPT model, which satisfies the constraints in an autoregressive generative manner, while controlling ‘copy’ and ‘generate’ actions with a switch based on hard-attention. Experiments show that our model surpasses prior works on the FewShotAmazon benchmark.

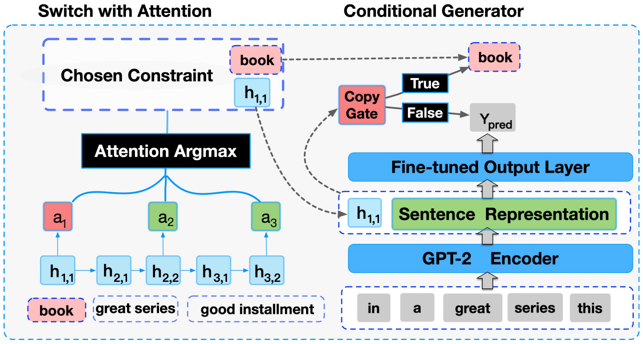


Figure 1: An illustration of Switch-GPT. $h_{i,j}$ is the hidden state of LSTM. a_i is the attention of the i^{th} constraint. Y_{pred} is the generation output of the GPT decoder.

Task Formulation

The CTG task is formulated as follows: given several disordered constraints $X = \{x_i\}_{i=1}^{n}$, each of which can either be a word or a phrase, the target is to generate a fluent sentence of nature language that contains all these constraints, e.g., $Y = [y_1, y_2, ..., y_m]$. Furthermore, training is conducted in few-shot settings, which means that the provided training set $D = \{X_d, Y_d\}_{d=1}^{|D|}$ contains limited samples, i.e., $|D| = 100$.

[*] These authors contributed equally.

Domain	Books			Clothing			Music			Movies			Electronics		
	Cov ↑	BLEU↑	ΔB↑	Cov↑	BLEU↑	ΔB↑	Cov↑	BLEU↑	ΔB↑	Cov↑	BLEU↑	ΔB↑	Cov↑	BLEU↑	ΔB↑
Copy	100	8.38	-	100	8.57	-	100	8.67	-	100	8.78	-	100	8.85	-
Pointer	48.74	9.44	3.74	49.89	8.90	3.2	44.20	7.95	3.83	48.30	8.42	3.47	47.67	9.12	3.57
Switch	63.69	10.94	4.01	60.56	9.60	2.99	63.12	11.89	4.26	55.97	10.45	3.23	51.75	8.46	2.80
POINTER	92.02	8.68	3.64	91.92	7.47	2.90	91.86	8.29	3.43	91.54	8.72	3.57	92.18	8.00	3.25
Switch-GPT	**100**	**24.15**	**4.85**	**100**	**22.25**	**4.68**	**100**	**23.76**	**4.72**	**100**	**22.73**	**4.79**	**100**	**23.71**	**3.75**

Table 1: Results on different domains. Cov and ΔB are the abbreviations for Coverage and ΔBLEU score.

Proposed Benchmark: FewShotAmazon

The benchmark is based on Amazon Product Reviews (He and McAuley 2016), which contains reviews from 5 different domains: books, music, movies, electronics and clothing. For each domain, 200 samples are selected for training, 1000 samples for validation, and 2000 samples for test. Preprocessing is conducted as follows: sentences are parsed using *spaCy* and keywords are extracted through keeping only the parent nodes of the dependency structure. The keywords include entities for the specific domain, which are more similar to real-life scenarios and increase the difficulty for models.

We use the following metrics to analyze performance: **Coverage** score measures the percentage of constraints convered in generation results; **BLEU** score measures the generalization capability of models; and we propose the **ΔBLEU** score which is the difference in BLEU score of generated sentences to the original sentences. We use it to illustrate how models provide transitions for constraints.

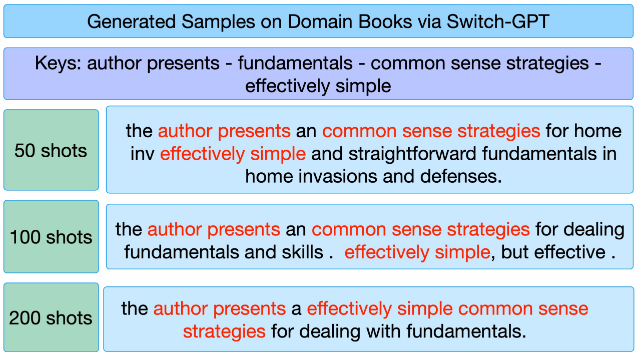
Switch-GPT Architecture and Performance

The architecture of the framework of Switch-GPT is depicted in Figure 1. The framework can be divided into two components : a Switch module to choose the constraint and to decide whether to copy the constraint; and a GPT-2 language model with an encoder to generate context embeddings and a conditional decoder to generate sentences.

Different from previous approaches(Zhang, Xu, and Wang 2019), we introduce the hard attention copy technique: after selecting the next constraint, the result of the attention module is set as the corresponding hidden state of the chosen constraint $c_t = h_k$ instead of a weighted summation. The intuition is that the weighted summation can be perceived as choosing a point within a convex hull constructed by the candidate constraints, i.e., $H(c) = \{\sum_{j=1}^{n} \alpha_i h_i | \sum_{j=1}^{n} \alpha_i = 1, \alpha_i \geq 0\}$. Due to the sparsity of hidden space, the weighted sum typically fails to represent meaningful constraint. Therefore, to guarantee choosing a meaningful constraint, hard attention is used to enforce the choice of a vertex rather than a point inside the convex hull.

We evaluate Switch-GPT, POINTER (Zhang et al. 2020), Pointer (See, Liu, and Manning 2017) and Switch (Chen et al. 2019) on our proposed benchmark FewShotAmazon, as shown in Table1. Our model scores higher than SOTA on all metrics in all domains. Also, Switch-GPT performs better when trained with fewer shots. The BLEU score of Switch-GPT trained on 50 samples(average = 17.42) outperforms Pointer trained on 20000 samples(average = 10.74).

The quality of generation improves with the increase in training samples. When trained on only 50 samples, the model still gives irrelevant details, which possibly comes from the pre-trained language model. After fine-tuning on 200 samples, the model gives more relevant generation.



Generated Samples on Domain Books via Switch-GPT

Keys: author presents - fundamentals - common sense strategies - effectively simple

50 shots: the author presents an common sense strategies for home inv effectively simple and straightforward fundamentals in home invasions and defenses.

100 shots: the author presents an common sense strategies for dealing fundamentals and skills . effectively simple, but effective .

200 shots: the author presents a effectively simple common sense strategies for dealing with fundamentals.

Figure 2: Generated samples w.r.t. different shots.

Future Work

Our current work enlightens the potential of pre-trained models on few-shot learning by combining a large pre-trained model with a light-weighted network well-designed for the downstream task. This paradigm is applicable to many NLP few-shot learning tasks. In future work, we will focus on generalizing to various few-shot learning tasks.

References

Chen, Z.; Eavani, H.; Chen, W.; Liu, Y.; and Wang, W. Y. 2019. Few-shot NLG with pre-trained language model. *arXiv preprint arXiv:1904.09521*.

He, R.; and McAuley, J. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*, 507–517.

See, A.; Liu, P. J.; and Manning, C. D. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Zhang, H.; Xu, J.; and Wang, J. 2019. Pretraining-based natural language generation for text summarization. *arXiv preprint arXiv:1902.09243*.

Zhang, Y.; Wang, G.; Li, C.; Gan, Z.; Brockett, C.; and Dolan, B. 2020. Pointer: Constrained text generation via insertion-based generative pre-training. *arXiv preprint arXiv:2005.00558*.