Switch-GPT: an effective method for constrained text generation under few-shot settings

Chang Ma* [*]
Peking University
changma@pku.edu.cn

Song Zhang*
Bytedance
songz@pku.edu.cn

Gehui Shen
Peking University
juelianggukc@pku.edu.cn

Zhihong Deng
Peking University
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 on FewShotAmazon dataset show that the proposed Switch-GPT model is effective and remarkably outperforms the baselines.

[1]

1 Introduction

Constrained text generation (CTG) is a vital research problem for various applications, including neural machine translation (Bahdanau et al., 2014; Luong et al., 2015), task-oriented dialogues (Liu et al., 2018; Budzianowski et al., 2018), and abstractive text summarization (See et al., 2017).

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. Previous works broadly falls into two categories, sampling or searching-based methods (Berglund et al., 2015; Hokamp and Liu, 2017; Miao et al., 2019; Sha, 2020) and insertion-based models (Zhang et al., 2020). Hokamp and Liu (2017) incorporates constraints by performing Grid Beam Search in the sentence space. However, searching based methods have a high time complexity, as generating texts involve a large sentence space. The Metropolis-Hastings sampling framework (Miao et al., 2019) models local transitions (e.g., deletion, insertion) to achieve better fluency, but is slow in convergence. Recently, fine-tuning on large-scale pre-trained language models(e.g. BERT (Devlin et al., 2018) and OpenAI GPT (Radford et al., 2018)) provide new opportunities to CTG (Song et al., 2019; Chen et al., 2019a; Ghazvininejad et al., 2019; Budzianowski et al., 2018; Yang et al., 2020; Zhu et al., 2020). Chen et al. (2019b) 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 on both human and automatic evaluation.

Although previous models generate reasonable results, performance relies on large training datsets, e.g., 5M training samples for CGMH, and 160K fine-tuning samples for POINTER in a single domain. 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?*

This work proposes the task of few-shot CTG, which aims to make the best of few training samples. We revisit the benchmarks for CTG, and notices that current datasets are monotonous in domains and lack suitable metrics. To simulate few-shot learning on various domains, we have de-

[1]* denotes equal contribution

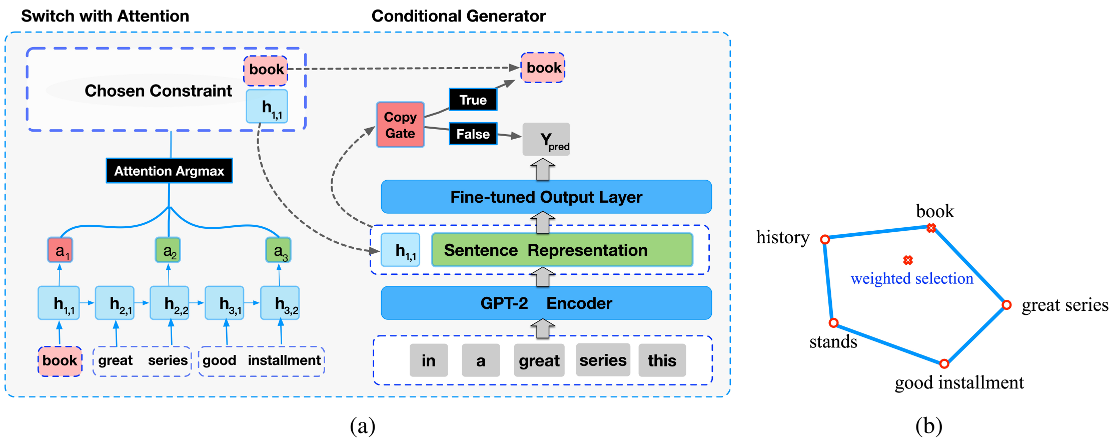


Figure 1: a) An illustration of Switch-GPT. $h_{i,j}$ is the hidden state of LSTM. a_i is the attention of the i^{th} constraint. Y_{pred} is the generation output of the GPT decoder. b) A comparison between soft attention and hard attention as choice of constraint.

veloped a new benchmark FewShotAmazon. We also propose a new metrics ΔBLEU to capture the model's ability to connect keywords smoothly. We believe that the FewShotAmazon benchmark can inspire future research to address CTG realistically.

To deal with the challenges of few-shot learning, 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. In short, the contributions are summarized as the following:

- A new benchmark FewShotAmazon is introduced to simulate the few-shot learning setting on multiple domains.

- We propose a new model Switch-GPT and creates a new SOTA on the FewShotAmazon benchmark.

2 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., $|D| = 100$.

3 Switch-GPT

The architecture of the framework of Switch-GPT is depicted in Figure 1(a). 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.

Switch with Attention Inspired by Grid Beam Search (GBS) (Hokamp and Liu, 2017), we model the generation by 3 policies, which are start copying, continue copying and generating. In our work, these policies are applied as training objectives instead of beam search algorithm.

At timestep t, given generated tokens $y_1, ..., y_{t-1}$, the encoder of GPT returns the representation s_{t-1} for the generated context. The model selects the constraint x_k most important for this context with an attention module, and then decides whether to copy x_k right now or generate several tokens for transition.

We employ a LSTM encoder to learn representations of constraints with variable lengths. Each constraint is represented by a hidden state h_i corresponding to its last token. Then we obtained weights of attention $\alpha^t = \{\alpha_i{}^t\}_{i=1}^n$ as in Bahdanau et al. (2014) and modelled the chosen constraint as the one with the largest attention.

$$\alpha_i^t = \frac{\exp(\mathrm{e}_\mathrm{i}^\mathrm{t})}{\sum_{k=1}^n \exp(\mathrm{e}_\mathrm{k}^\mathrm{t})} \quad (1)$$

where

$$e_i^t = a(s_{t-1}, h_i) \quad (2)$$

is an alignment model that scores the similarity between the two vectors.

During training, the token selected at each timestep is available by pairing the constraints and the output, e.g., y_t is copied from x_k or generated before next copy action of x_k, then we set k as the label of attention at this step. We trained the attention module with cross entropy:

$$L_{att}^t = Cross\_Entropy(\alpha^t, k) \tag{3}$$

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, $c_t = \sum \alpha_i h_i$. In this way, the hard attention mechanism is implemented such that it is different from the soft attention approach of (Zhang et al., 2019). 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, as illustrated in Figure 1(b).

Then, following the approach of (See et al., 2017), a switch p_{copy} is maintained to explicitly decide whether to copy x_k. Once decided to copy, the output of generator is not used.

$$p_{copy} = \mathrm{sigmoid}(\mathrm{W_c c_t + W_s s_t + W_i i_t + b}) \tag{4}$$

$$o_t = \begin{cases} x_k, & p_{copy} > 0.5 \\ \arg\max\ logits, & \text{else} \end{cases} \tag{5}$$

where o_t, i_t, s_t, $logits$ are the final output, decoder input, hidden state and the output of the generator respectively. The switch p_{copy} is also trained with cross entropy, supervised by the copy action *copy* in target text :

$$L_{copy}^t = Cross\_Entropy(p_{copy}^t, copy) \tag{6}$$

Conditional Generator We use the pre-trained language model GPT-2 as the generator to produce $p(y_t|y_1, ..., y_{t-1})$. In this task, we expect the current step of generation could be a smooth transition to select constraint x_k. Therefore, we need to model $p(y_t|y_1, ..., y_t - 1, x_k)$. To condition the generated result on x_k, we define $s'_t = f_{\text{MLP}}(s_{t-1}, x_k)$, where a new context representation is generated via the fully connected layer, then we feed s'_t to the output layer to obtain the eventual outputs. The overall loss function is as follows:

$$L = \sum_{t=1}^{m} \alpha * L_{att}^t + \beta * L_{copy}^t + L_{output}^t \tag{7}$$

where L_{output}^t is the cross entropy between the outputs and the targets, while α, β are hyper-parameters. The generator is fine-tuned from pre-trained parameters, while the parameters of the encoder (word embedding layer) are fixed. The LSTM and the attention modules are learned from scratch. During inference, the chosen constraints are masked to avoid repetition. The generation process does not end until all constraints are copied.

4 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* (Honnibal et al., 2020) 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.

5 Experiments

5.1 Experimental Setup

Switch-GPT and other baselines are evaluated on FewShotAmazon Benchmark. A general description of the experiment settings is shown below. Details and generated samples can be found in the appendices.

Model Implementation We adopt Byte Pair Encoding (BPE) (Sennrich et al., 2015) to deal with words out of vocabulary. The labels of attention and copy are generated by matching inputs and outputs during training. For pre-trained language model, we use the open-source implementation of the GPT architecture that provides GPT-2 finetunable checkpoints with 124M parameters (Radford et al., 2019).

Baselines Since our model is based on switch mechanism, the following models are selected as baselines: Pointer (See et al., 2017), and its variant Switch (Chen et al., 2019b) designed for few-shot generation. These models are also fine-tuned on

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.

the same checkpoints of GPT-2. In addition, we also compared with the SOTA model POINTER (Zhang et al., 2020). To illustrate the influence of the overlap between the inputs and outputs, we also list the results of simply copying all constraints.

5.2 Main Results

We adopt both automatic and human evaluation metrics to evaluate generation results, as shown in Table 1 and Figure2.

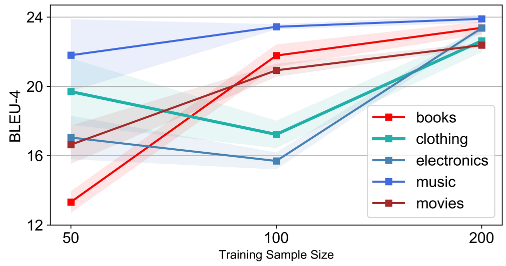
Coverage The task aims to generate a fluent sentence that contains all constraints. Coverage score is used to show the percentage of constraint tokens included in the generated results. From the results of Pointer and Switch, we can observe that it is difficult to train a model with soft copy mechanism to satisfy all the constraints, which is in line with the previous analysis. In contrast, our model provides 100% coverage.

BLEU Score We evaluate BLEU-4 (Papineni et al., 2002) score between generated samples and golden example of the test set to measure the generalization capability of the models. It is shown that our model outperforms all baselines by a large margin on this metric. Also, the quality of generation improves with the increase of samples, demonstrating steady generalization, as shown in Figure 2(a). Moreover, the BLEU score of Switch-GPT trained on 50 samples still outperforms the BLEU score of Pointer trained on 20000 samples(average BLEU = 10.74), which shows that Switch-GPT is very effective under few-shot settings.

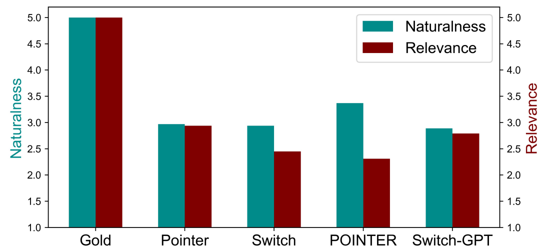
ΔBLEU(Proposed) We introduce a new metric, ΔBLEU, to illustrate how well models provide transitions for constraints. It is calculated by subtracting the BLEU score between generated sentences and input constraints from the overall BLEU score.

$$\Delta\mathrm{BLEU} = \mathrm{f}(\mathrm{y}_{\mathrm{gen}}, \mathrm{y}_{\mathrm{gold}}) - \mathrm{f}(\mathrm{y}_{\mathrm{gen}}, \mathrm{c}) \quad (8)$$

where $f, y_{\mathrm{gen}}, y_{\mathrm{gold}}, c$ stand for the BLEU Score, the generated sentence, the golden reference and the constraints. From results in Table 1, our model connects constraints coherently.



(a) Few-shot learning



(b) Human evaluation

Figure 2: (a) The results of Switch-GPT with different numbers of training samples. Pointer needs at least 100 times more training samples to generalize well. (b) Comparison of 'Naturalness' and 'Relevance' of generated sentences. The first column depicts the scores for golden references.

Human Evaluation We randomly extract 150 sentences on each domain and hire fifteen turk workers to rate each of them according to its 'Naturalness', and 'Relevance' on a scale from 1 to 5. 'Naturalness' indicates the semantic consistency and the grammatical correctness of sentences, and 'Relevance' indicates the information relevance of the generated sentence to the golden example. As shown in Figure 2(b), our model provides matching results on 'Naturalness', which is non-trivial as incoporating 100% constraints harms fluency. The performance of our model on 'Relevance' is competitive among baselines.

6 Conclusion

In this paper, we propose the task of few-shot constrained text generation, which aims at making use of the powerful pre-trained learning models and obtain a constrained generator rapidly. Our method based on autoregressive generation achieves this goal with a hard switch policy, which also provides a new direction for constrained text generation.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate.

Mathias Berglund, Tapani Raiko, Mikko Honkala, Leo Kärkkäinen, Akos Vetek, and Juha T Karhunen. 2015. Bidirectional recurrent neural networks as generative models. In *Advances in Neural Information Processing Systems*, pages 856–864.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

Yen-Chun Chen, Zhe Gan, Yu Cheng, Jingzhou Liu, and Jingjing Liu. 2019a. Distilling the knowledge of bert for text generation.

Zhiyu Chen, Harini Eavani, Wenhu Chen, Yinyin Liu, and William Yang Wang. 2019b. Few-shot nlg with pre-trained language model. *arXiv preprint arXiv:1904.09521*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-predict: Parallel decoding of conditional masked language models. *arXiv preprint arXiv:1904.09324*.

Ruining He and Julian McAuley. 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*, pages 507–517.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. *arXiv preprint arXiv:1704.07138*.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Bing Liu, Gokhan Tür, Dilek Hakkani-Tür, Pararth Shah, and Larry Heck. 2018. Dialogue learning with human teaching and feedback in end-to-end trainable task-oriented dialogue systems. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2060–2069, New Orleans, Louisiana. Association for Computational Linguistics.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation.

Ning Miao, Hao Zhou, Lili Mou, Rui Yan, and Lei Li. 2019. Cgmh: Constrained sentence generation by metropolis-hastings sampling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6834–6842.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Lei Sha. 2020. Gradient-guided unsupervised lexically constrained text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8692–8703.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. *arXiv preprint arXiv:1905.02450*.

Jiacheng Yang, Mingxuan Wang, Hao Zhou, Chengqi Zhao, Weinan Zhang, Yong Yu, and Lei Li. 2020. Towards making the most of bert in neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9378–9385.

Haoyu Zhang, Jianjun Xu, and Ji Wang. 2019. Pretraining-based natural language generation for text summarization. *arXiv preprint arXiv:1902.09243*.

Yizhe Zhang, Guoyin Wang, Chunyuan Li, Zhe Gan, Chris Brockett, and Bill Dolan. 2020. Pointer: Constrained text generation via insertion-based generative pre-training. *arXiv preprint arXiv:2005.00558*.

Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tie-Yan Liu. 2020. Incorporating bert into neural machine translation. *arXiv preprint arXiv:2002.06823*.