2022.5，2022.10

一句话总结

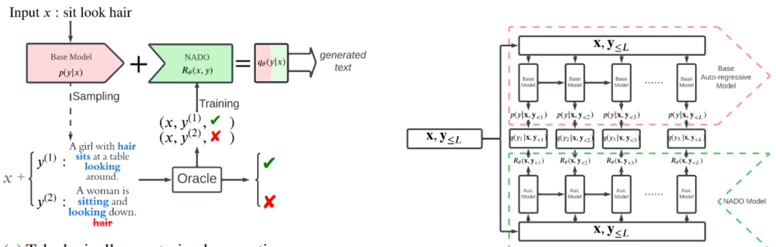
给定一个基本的预训练语言模型和sequence-level oracle function（指示是否满足规则），通过训练辅助模型NADO，把序列级规则分解问token级指导，引导模型进行可控文本生成。

> Oracle Function：A function is a subprogram that is used to return a single value. Site Unreachable理解为一个0-1判别函数，相当于是reward model（基于规则）；

文章主要方法是把一个sequence-level信号分解为token-level的guidance信号；
启发：我们的setting是序列决策，从最终的结果的reward信号，如何分解得到**过程**的reward信号；

核心

- 基于NeurAlly-Decomposed Oracle（NADO）提出了**可控的**自回归生成模型；
- pre-trained base language model + sequence-level boolean oracle function -> **oracle function** into token-level guidance to steer the base model in text generation；
- token-level指导：从一个base model的数据中进行采样，训练了一个辅助模型NADO；
- 把可控生成问题定义为：基于后验正则化的优化问题。得到**解析最优解**，用来在token-level指导模型的可控生成；
- 对于NADO的近似的质量如何影响最终可控生成的结果进行分析，做了2个任务的实验：
  - text generation with lexical constraints 具有词汇约束的文本生成；
  - machine translation with formality control 带有形式控制的机器翻译；



(a) Take lexically constrained generation as an example, where the oracle checks whether all keywords in the input x are incorporated in generated text y. With proper training using samples from the base model p (dashed arrow) labeled by the oracle, we decompose the oracle into token-level guidance and parameterize it by an auxiliary model R_θ (NADO). We use R_θ to provide guidance when generating text with the base model (see details in Fig. 1(b)).

(b) Illustration of the controlled generation process. Both the base model and the auxiliary model (NADO) take input $\mathbf{x}$ and the generated sequence (prefix) $\mathbf{y}_{<L}$ as input. The base model, in each step, outputs a token distribution $p(y_i|\mathbf{x}, \mathbf{y}_{<i})$. Guided by NADO R_θ, we obtain the distribution q (See Sec. 3.2), based on which we generate the output token.

Figure 1: Illustration of pipeline incorporating NADO (left) and model architecture (right).

左图：NADO的训练，从一个base model的生成中进行采样，由一个sequence-level的判别器产生监督信号；base-model在这个过程中不需要fine-tuning；

右图：decompose the sequence-level oracle into token-level guidance, such that when generating the i−th token in the output sequence given the prefix, instead of sampling from the base model, we modify the probability distribution of the output token based on the token-level guidance.

- 把sequence-level规则分解为**token-level的指导**，最终的每个token的生成由base-model + guidance的分布决定；
- base-model + guidance的过程后面再看；

Intro

可控生成

- 要求模型的输出遵循sequence-level的属性：
  - 由一系列规则定义（譬如语法规则）；
  - 由某个抽象概念定义（譬如文风）；
- 现有工作

    - 基于搜索算法的词汇约束的算法，不能应用于风格写作任务；
    - 训练辅助模型（用来微调模型，或者需要外部标记数据），无理论保证，或者成本高；
    - 使用KL-adaptive分布策略来近似一个energy-based model；（粒度太粗？）
- 实验
    - 词汇约束生成 (LCG) 任务：oracle 是一个基于规则的关键字检查器；
    - 形式控制的机器翻译任务：提供了一个形式预言机来预测句子是否正式，目标是引导模型生成形式翻译；
- 后处理
    - 可控文本生成的方法归为三大类：fine-tuning, refactor/retraining and post-processing；
    - 后处理的主要步骤：修改decoding算法（如beam search），通过辅助模型指导生成；
    - 辅助模型：PPLM，GeDi，DEXPERTS，FUDGE，要么需要外部token-level oracle指导，要么需要辅助标记数据集来训练辅助模型；用于训练辅助模型的数据分布与所训练的模型的分布不同，导致生成质量下降；

方法

- 文章的主要方法就是提出NADO，这是一个近似的辅助模型，得到token-level的指导；
- we discuss 1) the formulation to decompose the sequence-level oracle function into token-level guidance; 2) the formulation to incorporate the token-level guidance into the base model to achieve control; 3) the approximation of the token-level guidance using NADO; 4) a theoretical analysis of the impact of NADO approximation to the controllable generation results; and 5) the training of NADO.
- 非常重要的部分！

概念

- base-model p；
- 指示函数C；
- 要得到一个token-level distribution $q^*(y_i|\mathbf{x},\mathbf{y}_{<i})$，满足：

1. $q^*(\mathbf{y}|\mathbf{x}) = \prod_i q^*(y_i|\mathbf{x},\mathbf{y}_{<i})$, i.e., q^* can be treated as an auto-regressive model.
2. $q^*(\mathbf{y}|\mathbf{x}) = 0$ if $C(\mathbf{x},\mathbf{y}) = 0$, i.e., q^* only generates sequences satisfying the oracle C.
3. Given an input $\mathbf{x}$, $KL(p(\mathbf{y}|\mathbf{x})\|q^*(\mathbf{y}|\mathbf{x}))$ is minimized, i.e., q^* should be as similar to the base model as possible.

辅助模型也是自回归model；辅助模型的结果和指示函数的结果一致；辅助模型与base model尽可能接近；

Before we compute the solution for q^*, given the base model p and oracle C, we first define the token-level guidance as a success rate prediction function $R_p^C(\mathbf{x})$, which defines the probability of the sequence generated by p satisfies the oracle C given the input $\mathbf{x}$. We similarly define $R_p^C(\mathbf{x},\mathbf{y}_{\leq i})$ as the probability of success given input $\mathbf{x}$ and prefix $\mathbf{y}_{<i}$. By definition, we have

$$
\begin{aligned}
R_p^C(\mathbf{x}) &= \Pr_{\mathbf{y}\sim p(\mathbf{y}|\mathbf{x})}[C(\mathbf{x},\mathbf{y})=1] = \sum_{\mathbf{y}\in\mathcal{Y}} p(\mathbf{y}|\mathbf{x})C(\mathbf{x},\mathbf{y}) \\
R_p^C(\mathbf{x},\mathbf{y}_{\leq i}) &= \Pr_{\mathbf{y}\sim p(\mathbf{y}|\mathbf{x})}[C(\mathbf{x},\mathbf{y})=1|\mathbf{y}_{<i}] = \sum_{\mathbf{y}\in\mathcal{Y}} p(\mathbf{y}|\mathbf{x},\mathbf{y}_{<i})C(\mathbf{x},\mathbf{y}).
\end{aligned} \tag{1}
$$

定义两个后验概率（待定）：

- 输入成功率：对于给定输入x，通过base model p得到的结果y最终满足指示函数的概率$R_p^C(x)$；
- 序列成功率：对于给定输入x以及部分序列$y_{<i}$，通过base model p得到结果y最终最终满足指示函数的概率$R_p^C(x, y_{<i})$；

解析解的给出

对于给定的x，定义一个sequence-level的分布Q满足指示函数引导的分布，所以得到q^*的解析解为：

With the function R_p^C, we now derive the closed-form solution of q^* considering conditions 2 and 3 defined in Sec. 3.1. Given input $\mathbf{x}$, we define the feasible sequence-level distribution set Q as

$$Q := \{q | \sum_{\mathbf{y}:\, C(\mathbf{x},\mathbf{y})=0} q(\mathbf{y}|\mathbf{x}) = 0\}, \tag{2}$$

then the sequence-level closed-form solution for q^* is given by

$$q^*(\mathbf{y}|\mathbf{x}) = \arg\min_{q\in Q} KL(p(\mathbf{y}|\mathbf{x})\|q(\mathbf{y}|\mathbf{x})) = \frac{p(\mathbf{y}|\mathbf{x})C(\mathbf{x},\mathbf{y})}{R_p^C(\mathbf{x})}. \tag{3}$$

为了理解这个式子，论文中没有给出过程或者解释，我这里给一个非常直观的例子：假设输入x可以通过p均匀分布得到y1, y2, .., y5，其中C(x, y1)=1, C(x, y2)=1, 其他都为0；那么q相当于是将概率分布聚焦于正例之上，而确保负例的输出概率为0（这是一个硬约束）；或者说是一个缩放

$$q^*(y|x) = \frac{p(y|x)}{R_p^C(x)}$$

接下来，把这个概率分布q唯一分解为token-level：通过第i步相关的后验概率，影响于p；

$$q^*(y_i|\mathbf{x}, \mathbf{y}_{<i}) = \frac{R_p^C(\mathbf{x}, \mathbf{y}_{\le i})}{R_p^C(\mathbf{x}, \mathbf{y}_{\le i-1})} p(y_i|\mathbf{x}, \mathbf{y}_{<i}). \tag{4}$$

The sequence-level solution q^* is given by

$$q^*(\mathbf{y}|\mathbf{x}) = \frac{p(\mathbf{y}|\mathbf{x})C(\mathbf{x}, \mathbf{y})}{R_p^C(\mathbf{x})}.$$

Now we prove that

$$q^*(y_i|\mathbf{x}, \mathbf{y}_{<i}) = \frac{R_p^C(\mathbf{x}, \mathbf{y}_{\le i})}{R_p^C(\mathbf{x}, \mathbf{y}_{\le i-1})} p(y_i|\mathbf{x}, \mathbf{y}_{<i}),$$

is the unique token-level decomposition. On one hand, we verify q^* is a valid decomposition, which can be demonstrated by

$$\begin{aligned}
\prod_{i=0}^{L} q^*(y_i|\mathbf{x}, \mathbf{y}_{<i}) &= \prod_{i=1}^{L} \frac{R_p^C(\mathbf{x}, \mathbf{y}_{\le i})}{R_p^C(\mathbf{x}, \mathbf{y}_{\le i-1})} p(y_i|\mathbf{x}, \mathbf{y}_{<i}) \\
&= \frac{R_p^C(\mathbf{x}, \mathbf{y}_{\le L})}{R_p^C(\mathbf{x}, \mathbf{y}_{\le 0})} \prod_{i=0}^{L} p(y_i|\mathbf{x}, \mathbf{y}_{<i}) \\
&= \frac{C(\mathbf{x}, \mathbf{y})}{R_p^C(\mathbf{x})} p(\mathbf{y}|\mathbf{x}) \\
&= q^*(\mathbf{y}|\mathbf{x}),
\end{aligned} \tag{10}$$

together with

$$\sum_{y_i} q^*(y_i|\mathbf{x}, \mathbf{y}_{<i}) = \frac{\sum_{y_i} R_p^C(\mathbf{x}, \mathbf{y}_{\le i}) p(y_i|\mathbf{x}, \mathbf{y}_{<i})}{R_p^C(\mathbf{x}, \mathbf{y}_{\le i-1})} = 1 \tag{11}$$

On the other hand, we demonstrate that the decomposition is unique. We generally prove that

软约束的情况

- 2式的约束过于强硬，可能缺乏多样性；改成一个软约束：用一个比率r来调整准确性；

about sports with probability $r = 0.8$. Our framework also supports controlling the generation with soft constraints. To achieve this, with a pre-defined ratio $r \in [0, 1]$, we alternatively define a general feasible set Q as

$$Q := \{q | \sum_{\mathbf{y}:\, C(\mathbf{x},\mathbf{y})=1} q(\mathbf{y}|\mathbf{x}) = r\},$$

where Eq. (2) is the special case when $r = 1$. The general token-level closed-form solution is

$$q^*(y_i|\mathbf{x}, \mathbf{y}_{<i}) = \frac{\alpha R_p^C(\mathbf{x}, \mathbf{y}_{\le i}) + \beta(1 - R_p^C(\mathbf{x}, \mathbf{y}_{\le i}))}{\alpha R_p^C(\mathbf{x}, \mathbf{y}_{\le i-1}) + \beta(1 - R_p^C(\mathbf{x}, \mathbf{y}_{\le i-1}))} p(y_i|\mathbf{x}, \mathbf{y}_{<i}),$$

where $\alpha = \frac{r}{R_p^C(\mathbf{x})}, \beta = \frac{1-r}{1-R_p^C(\mathbf{x})}$.

本文中只考虑硬约束的情况，虽然NADO的方法也能适用于软约束的情况；

这里我们不免会提出一个问题，***如何获得***细粒度的后验概率？这便是NADO的核心；

R_p^C的近似计算

- **训练一个model来给出 R_p^C 的近似值；用 R_θ^C 表示**；
- 下面还计算了理论的sequence-level的分布误差的上界；

Lemma 1 We define distribution

$$q(y_i|\mathbf{x},\mathbf{y}_{<i}) \propto \frac{R_\theta^C(\mathbf{x},\mathbf{y}_{\le i})}{R_\theta^C(\mathbf{x},\mathbf{y}_{\le i-1})} p(y_i|\mathbf{x},\mathbf{y}_{<i}). \tag{5}$$

If there exists $\delta > 1$ such that given input $\mathbf{x}$, $\forall \mathbf{y}_{<i}$, $\frac{1}{\delta} < \frac{R_\theta^C(\mathbf{x},\mathbf{y}_{\le i})}{R_p^C(\mathbf{x},\mathbf{y}_{\le i})} < \delta$, we have

$$KL(q^*(\mathbf{y}|\mathbf{x})\|q(\mathbf{y}|\mathbf{x})) < (2L+2)\ln\delta,$$

where L is the length of the sequence $\mathbf{y}$.

We also notice that by definition, R_p^C satisfies the following equation:

$$\sum_{y_i} R_p^C(\mathbf{x},\mathbf{y}_{\le i}) p(y_i|\mathbf{x},\mathbf{y}_{<i}) = R_p^C(\mathbf{x},\mathbf{y}_{\le i-1}). \tag{6}$$

If R also satisfies Eq. (6), we can tighten this bound. Formally,

Lemma 2 Given the condition in Lemma 1, if q is naturally a valid distribution without normalization (i.e., $\sum_{y_i} \frac{R_\theta^C(\mathbf{x},\mathbf{y}_{\le i})}{R_\theta^C(\mathbf{x},\mathbf{y}_{\le i-1})} p(y_i|\mathbf{x},\mathbf{y}_{<i}) = 1$), we have

$$\forall x, KL(q^*(\mathbf{y}|\mathbf{x})\|q(\mathbf{y}|\mathbf{x})) < 2\ln\delta.$$

This lemma shows that with the auto-regressive property, the error does not accumulate along with the sequence. The proof is in the appendix. These two bounds indicate that when training the model R_θ^C, we should push it to satisfy Eq. (6) while approximating R_p^C.

这两个lemma说明，当model足够逼近（用delta）描述最大误差，那么model同最优值的误差存在上限且与长度无关；

训练NADO

- 采样x，y；
- C(x,y)给出label；
- 使用交叉熵损失函数:

$$L_{CE}(x,y,R_\theta^C) = \sum_{i=0}^{T} CE(R_\theta^C(x,y_{<i}),C(x,y))$$

Now we discuss the training objective. In training, with some predefined input distribution $\mathcal{X}$, we sample $\mathbf{x} \sim \mathcal{X}$, $\mathbf{y} \sim p(\mathbf{y}|\mathbf{x})$. We take these sampled $(\mathbf{x},\mathbf{y})$ pairs as training examples, and use the boolean value $C(\mathbf{x},\mathbf{y})$ as their labels for all steps. We use cross entropy (denoted as $CE(\cdot,\cdot)$) as the loss function, formally, $L_{CE}(\mathbf{x},\mathbf{y},R_\theta^C) = \sum_{i=0}^{T} CE(R_\theta^C(\mathbf{x},\mathbf{y}_{\le i}),C(\mathbf{x},\mathbf{y}))$. Given a particular input $\mathbf{x}$, in expectation, we have

$$\begin{aligned}\mathbb{E}_{\mathbf{y}\sim p(\mathbf{y}|\mathbf{x})} L_{CE}(\mathbf{x},\mathbf{y},R_\theta^C) &= \sum_{\mathbf{y}\in\mathcal{Y}} p(\mathbf{y}|\mathbf{x}) L_{CE}(\mathbf{x},\mathbf{y},R_\theta^C) \\ &= \sum_{i=0}^{T} R_p^C(\mathbf{x},\mathbf{y}_{\le i})\log R_\theta^C(\mathbf{x},\mathbf{y}_{\le i}) + (1-R_\theta^C(\mathbf{x},\mathbf{y}_{\le i}))\log(1-R_\theta^C(\mathbf{x},\mathbf{y}_{\le i})) \\ &= \sum_{i=0}^{T} CE(R_p^C(\mathbf{x},\mathbf{y}_{\le i}),R_\theta^C(\mathbf{x},\mathbf{y}_{\le i}))\end{aligned} \tag{7}$$

在加上一个正则化项（避免p和q偏离太远），得到完整的损失函数:

As we analyze above, we also regularize R_θ^C for satisfying Eq. (6) based on KL-divergence:

$$L_{reg}(\mathbf{x},\mathbf{y},R_\theta^C) = f_{KL}\left(\sum_{y_i} R_\theta^C(\mathbf{x},\mathbf{y}_{\le i}) p(y_i|\mathbf{x},\mathbf{y}_{<i}), R_\theta^C(\mathbf{x},\mathbf{y}_{\le i-1})\right).$$

$f_{KL}(p,q) = p\log\frac{p}{q} + (1-p)\log\frac{1-p}{1-q}$ is KL-divergence regarding p and q as two Bernoulli distributions. We use a hyper-parameter $\lambda > 0$ to balance these losses. The final training loss is

$$L(\mathbf{x},\mathbf{y},R_\theta^C) = L_{CE}(\mathbf{x},\mathbf{y},R_\theta^C) + \lambda L_{reg}(\mathbf{x},\mathbf{y},R_\theta^C). \tag{8}$$

采样技巧:

- 引入温度，改变对于原始p的相关性；
- 重要性采样，可能数据集并不均匀，C(x,y)=0情况居多；需要平衡正例负例的数量；

具体的训练过程（Text Generation with Lexical Constraints任务）

数据

- 原始样本中没有负例；
- 分为无监督和有监督两类任务（上面提到的图为有监督的情况）；

模型

- seq2seq基础模型: p(y|x),将词汇约束视为条件序列输入；

- (DA base model) A language model that is only domain-adapted to p(y) but unconditioned on anything. 这个更难，因为只用NADO来实现词汇约束；更能验证有效性；
- 从GPT-2-Large进行微调，训练NADO，输入关键词汇，输出token-level guidance；

During training, NADO is trained as a Seq2seq-like model[5], which takes in the keys (for unsupervised LCGs, they are generated by randomly sampling a specific number of natural words in the original sentence) and generates the token-level guidance $R_\theta^C(\mathbf{x}, \mathbf{y}_{\leq i})$. For each pseudo key, we sample 32 target text with top-p ($p = 0.8$) random sampling from base model p. We conduct experiments to test different training setups for NADO:

- (NADO training) The proposed training process described in Sec. 3.4.
- (Warmup) We warm up NADO by maximizing the likelihood of positive samples, but only backpropagating the gradient to the parameters of R_θ. The warm-up R_θ^C is used for importance sampling described in Sec. 3.5. With DA base models, however, the warmup process is always incorporated for practical success of training (see the results for DA pretrained w/o warmup).

对于具体训练过程感觉没有交代的很清楚；不过我大概了解了；

转换难点

- 对于序列决策任务，（2）式的转换并不好直接进行，因为难以判断到底是序列中的哪一个步骤决定了最终结果的错误；对于数学问题而言，可能大部分的action都是正确的，某个token出现了计算错误导致结果错误，那么我们要对后面的token都给一个较低的q？这是否合理？
- 如何与RL相结合？将 $(x, y_{<i})$ 视为state，将 $R_p^C(x, y_i)$ 可以自然地视为 state-value，$q^*(y_i|x, y_{<i})$ 可以视为action-value？如何使用RL方法对于policy进行优化？这真的好训吗，我怎么觉得还是一个稀疏的奖励的……