

大模型解码

《大语言模型》编写团队：唐天一

- 解码：基于输入（提示）生成输出的过程
- 自回归解码：逐词生成下一个词
- 目前大模型主要采用Transformer 解码器架构

输入： 模型 \mathcal{M} ，输入词元序列 u

输出： 输出词元序列 y

1: **repeat**

2: $P = \mathcal{M}(u)$ # 生成下一个词元的概率分布

3: $u' \sim P$ # 从分布中采样得到下一个词元

4: $u \leftarrow u \oplus [u']$

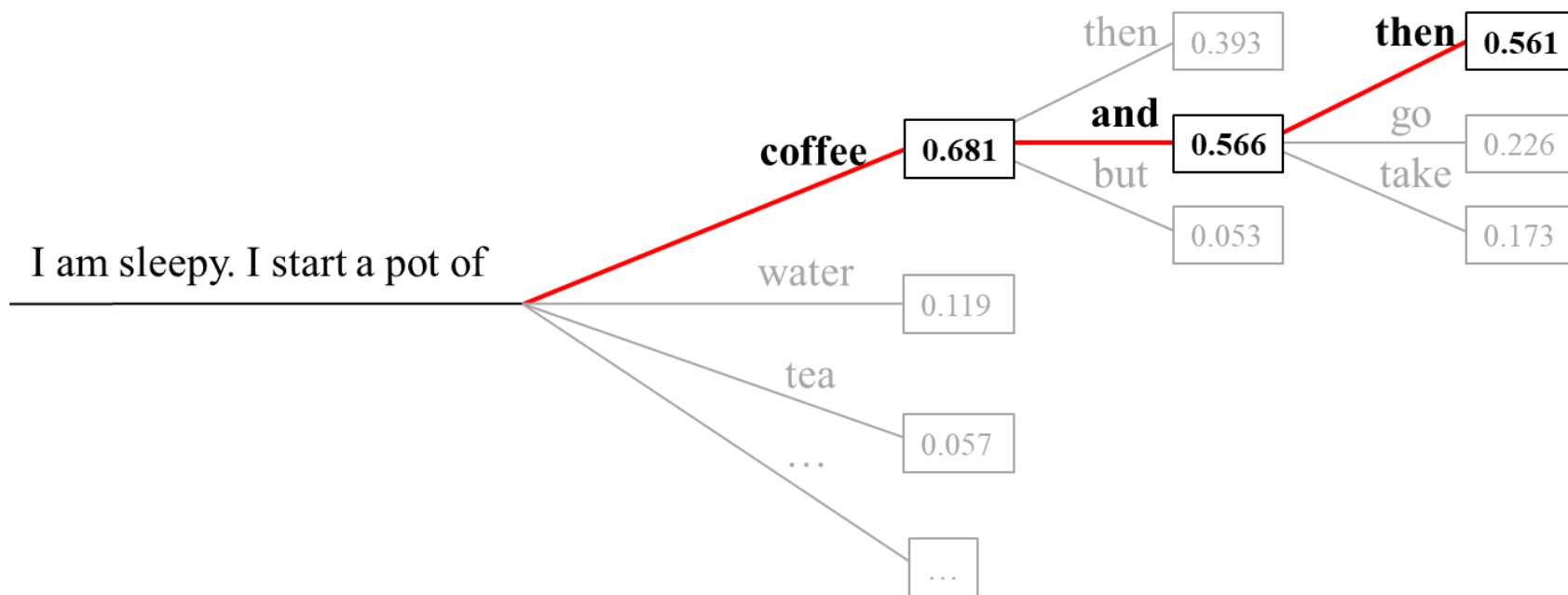
5: **until** u' 是结束词元或者 u 的长度超过预设长度.

6: $y \leftarrow u$

伪代码：自回归解码流程

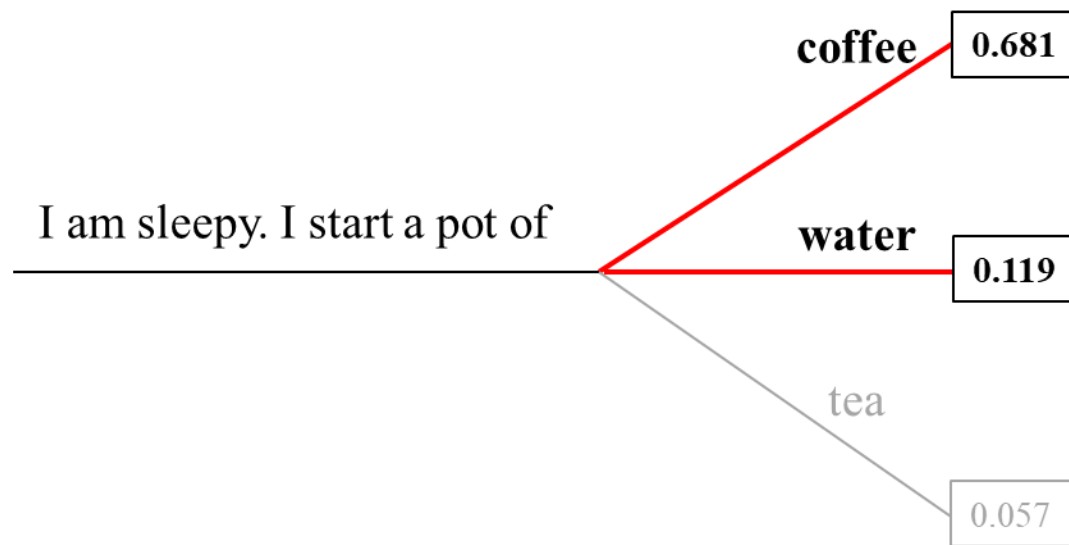
- 贪心搜索：每个生成步骤中都选择概率最高的词元

$$u_i = \arg \max_u P(u \mid \mathbf{u}_{<i})$$



图中红线路径是贪心搜索最后选择的路径

- 束搜索：每步保留前 n 个具有最高概率的句子
- 缓解贪心搜索陷入“局部最优”的问题

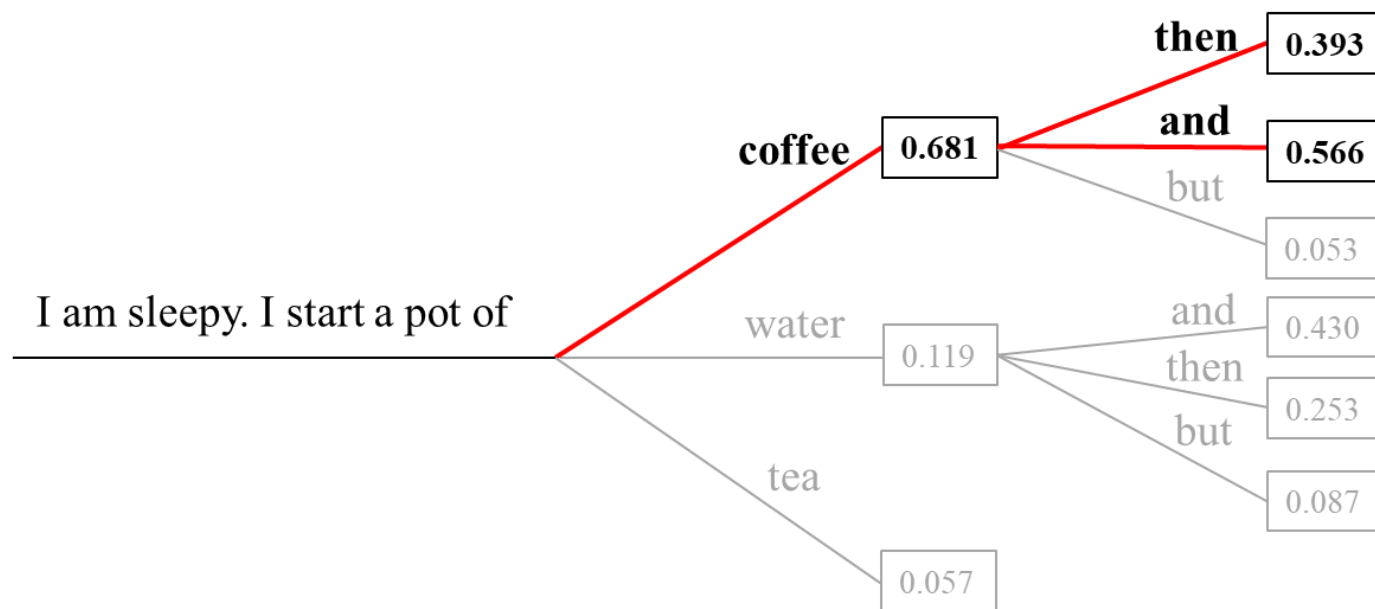


在第一步解码过程中保留了概率最高的两个单词

束搜索第一步 ($n=2$)

贪心搜索的改进

- 束搜索：每步保留前 n 个具有最高概率的句子
- 缓解贪心搜索陷入“局部最优”的问题



束搜索第二步 ($n=2$)

在第二步解码过程中进一步拓展了两个概率最高的单词

- 随机采样：基于概率分布采样得到下一个词元
- 增强结果的多样性

$$u_i \sim P(u \mid \mathbf{u}_{<i})$$

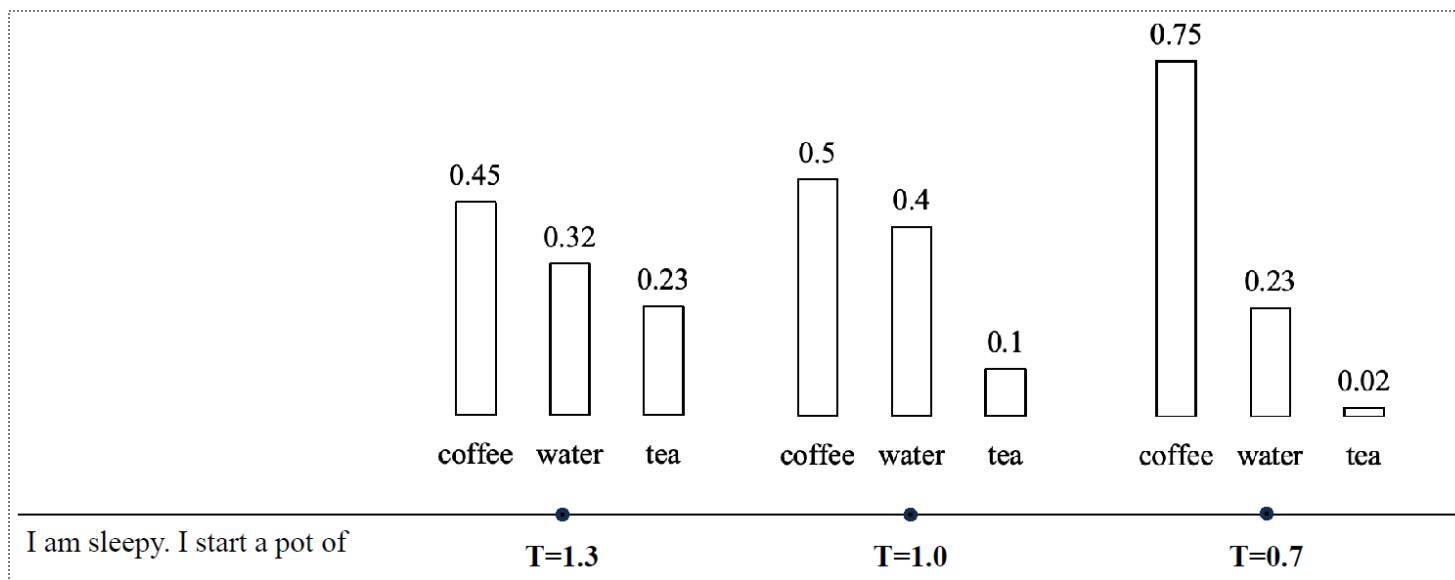
I am sleepy. I start a pot of _____					
coffee	0.681	strong	0.008	soup	0.005
water	0.119	black	0.008
tea	0.057	hot	0.007	happy	4.3e-6
rice	0.017	oat	0.006	Boh	4.3e-6
chai	0.012	beans	0.006

基于文本前缀的下一个词概率分布

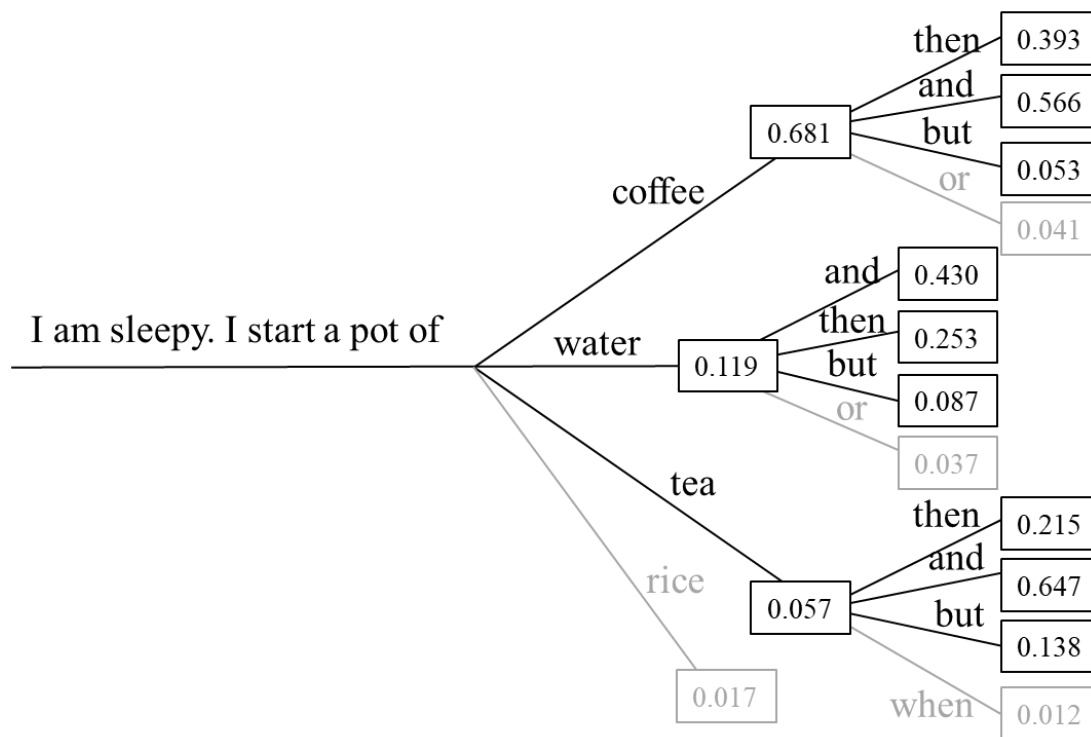
- 温度采样：调整 softmax 函数的温度系数

$$P(u_j | \mathbf{u}_{<i}) = \frac{\exp(l_j/t)}{\sum_{j'} \exp(l_{j'}/t)}$$

- 降低温度系数，增加高概率词元可能性，降低低概率词元可能性

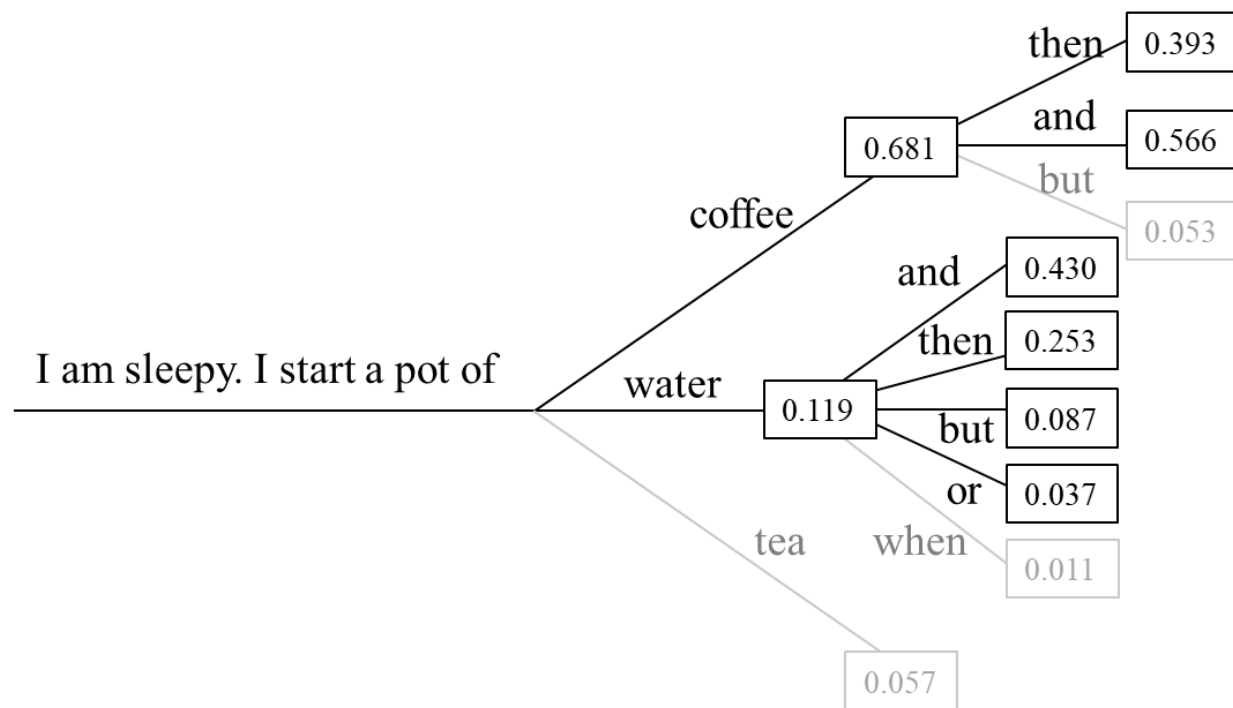


- Top- k 采样：仅从概率前 k 高的词元中采样
 - 直接剔除低概率的词，兼顾生成质量



top- k 采样 ($k=3$)

- Top- p 采样：仅从累积概率和为 p 的高概率词元中采样
- 考虑整体概率分布，适应不同场景



top- p 采样 ($p=0.8$)

- 重复惩罚：缓解生成重复文本
 - n -元惩罚
 - 直接避免生成重复的连续 n 个词元（通常 $n \in \{3,4,5\}$ ）
 - 出现惩罚
 - 词元 t 生成的概率 $P(t) = \text{logit}(t) - \text{是否出现}(t) \times \alpha$
 - 频率惩罚
 - 词元 t 生成的概率 $P(t) = \text{logit}(t) - \text{出现次数}(t) \times \alpha$

➤ 对比解码

- 在大模型和小模型的概率分布差值中采样
- 大模型会为更重要的单词分配更高的概率
- 借助两者之间的差异发掘重要词汇

李时珍是湖北人，他出生于__

小模型下一个词
概率分布

湖北 10%

明朝 0.1%

大模型下一个词
概率分布

湖北 15%

明朝 10%

生成“明朝”概率大幅增加



谢谢