

## Lee) Seq-to-seq

\* 神经网络方法

\* RNN, Att., Transformer

Encoder → Decoder

△ RNN

编码器: 每一步输入隐藏状态 → context vec

解码器: 每一步输入: context vec 上一词

每一步 → context vec 固定不变

△ Att.

解码器: 每一步两个输出

k.v 在 encoder 中: 隐藏状态

通过 Att. 分布 / 分成 decoder: 单步隐藏状态

辅助 Δ t

\* 作用: 简化问题 ✓

{ 梯度消失 ✓ (远距也可注意)

human-like ✓

可解释性

△ Transformer

Encoder: No mask - Transformer. &

→ 去除了  $\text{Softmax}$  分类 (不再输出分布).

Decoder: 基于 Transformer:

第一层为 2 层 = mask self-attn.  $\xrightarrow{\text{mul-head}}$

第二层多加一个逐层 mul-head attn  
Cross

经过 mask-self-attn

→ 未来 decoder. k.v 来自 encoder:

一层双层 (N 层)

\* Learning

最大化对数条件似然

→ 最小化  $-\log P(\vec{x}_1 \dots \vec{x}_T | \vec{z})$

训练方法: Teacher forcing → 每个时间步

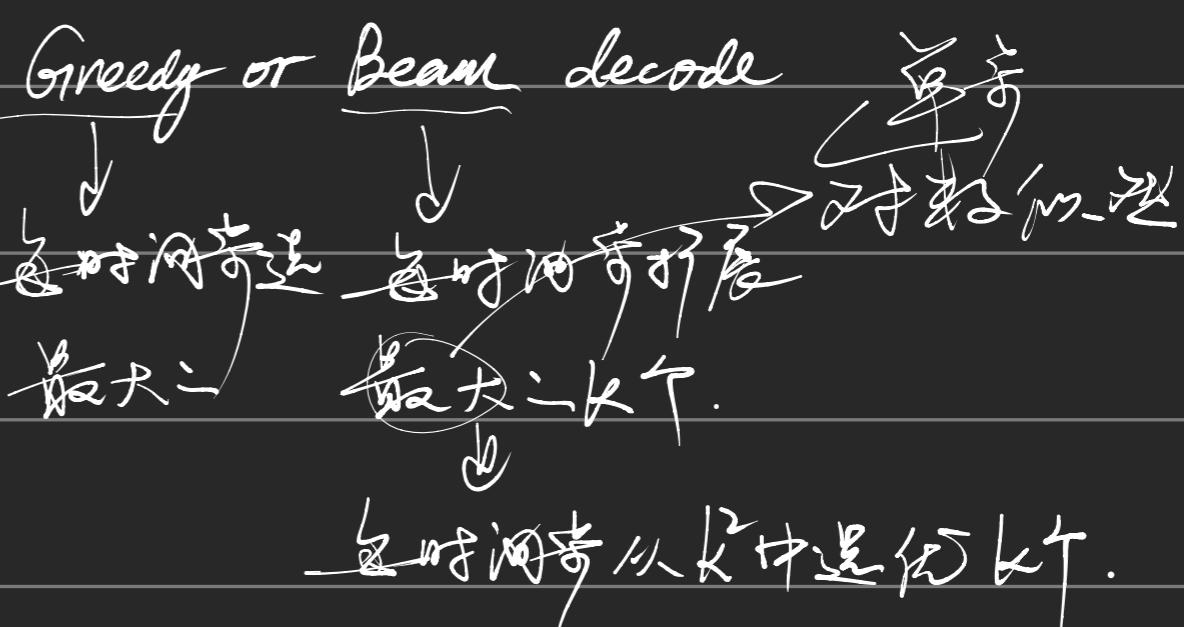
都输入一个真实目标词, 让模型填上一个预测出来之词

问题: exposure bias (never seen its own errors)

方法: scheduled sampling (一定概率输入自己预测)

训练: end to end: Encoder, Decoder 一起训练

\* Decoding



终止判断:

greedy: 当些预测出  $\langle \text{END} \rangle$

Beam: { 通过预判, 通过分支遍历

到达最大 T / 包含 n 个候选句子

$$\text{Beam 解法: } \frac{1}{t} \sum \log P_{\text{LM}}(y_t | y_{t-1}, \dots, y_1, x)$$

$\Rightarrow$  每步考虑对数似然最高

△ Seq-to-Set

输入序列输出集合

△ Problems: 多样性

$\Rightarrow$  Solution: 以下问题解决多样性

1. 取样 Top-k 或采样 - k

2. 核心部分 > p-采样 (nucleus)

其他: 重复性  $\Rightarrow$  已生成词惩罚

$\Rightarrow$  hallucination (幻觉)

\* (Non-) Autoregressive

Auto: one by one

Non-Auto: 并行生成  $\Rightarrow$  Transformer

△ Non-Auto:

Encoder 方式:

1. 独立向量  $\rightarrow$  2. 生成 k MASK tokens

3. 各位置并行编码 (对未元数据)

Problems:  $\begin{cases} \text{tokens independently.} \\ \text{多个可生成位置的 fail} \end{cases}$

$\Rightarrow$  Directed acyclic Transformer. iterative NFT

\* Extensions □ 抽取/插入 / Copy mechanism

$\Rightarrow$  多种生成分布: 生成 or 抽取