基于掩码机制的非自回归神经机器翻译

贾浩，王煦，季佰军，段湘煜，张民

苏州大学

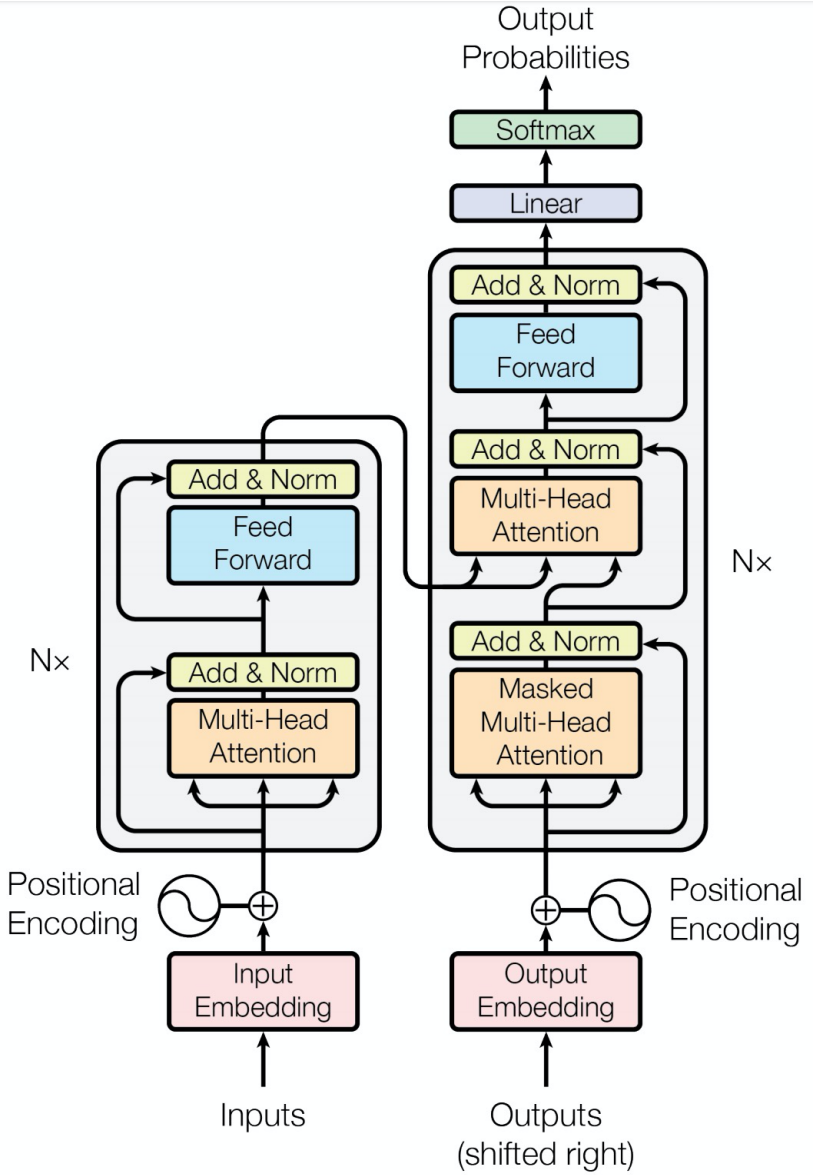
计算机科学与技术学院

提纲

- 背景介绍
- 相关工作
- 模型结构
- 实验设计
- 结果分析
- 总结

背景介绍

- 传统的自回归神经机器翻译（AT）
  - encoder-decoder结构，结构复杂
  - 解码时从左到右，按位置依次解码，不可并行
- 非自回归神经机器翻译（NAT）
  - 并行解码，一次生成所有目标端单词



相关工作

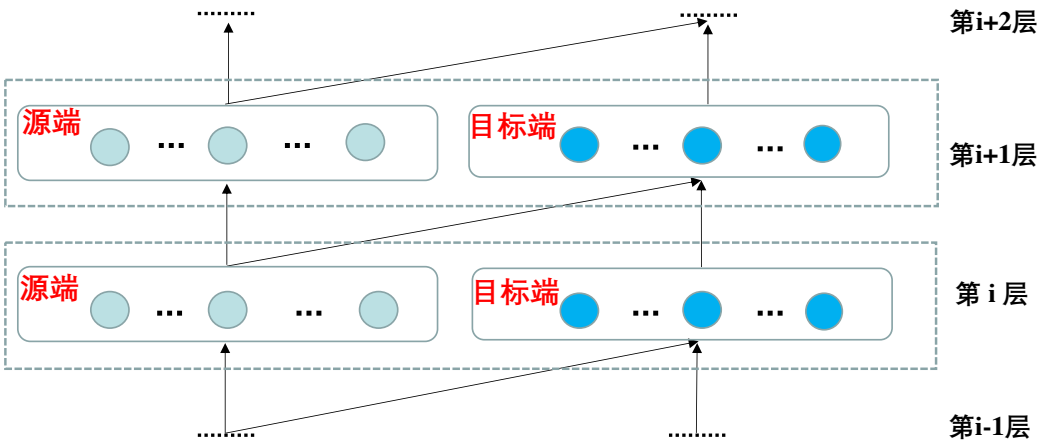
- 提高并行解码的译文合理性
  - Fertility-based (Gu 等人, 2018)：采用Fertility Predictor来决定解码器的输入
  - Extra refinement model (Lee 等人, 2018) 额外的基于潜变量模型和降噪自编码器的迭代优化模型
- 提高并行解码的译文完整性
  - Imitation learning (Wei 等人, 2019)：用一个自回归NMT模型指导非自回归NMT模型
  - Reinforcement learning (Shao 等人, 2019)：在训练过程中给予句子级别的奖赏
  - Generative flow (Ma 等人, 2019)：集成Generative flow隐变量

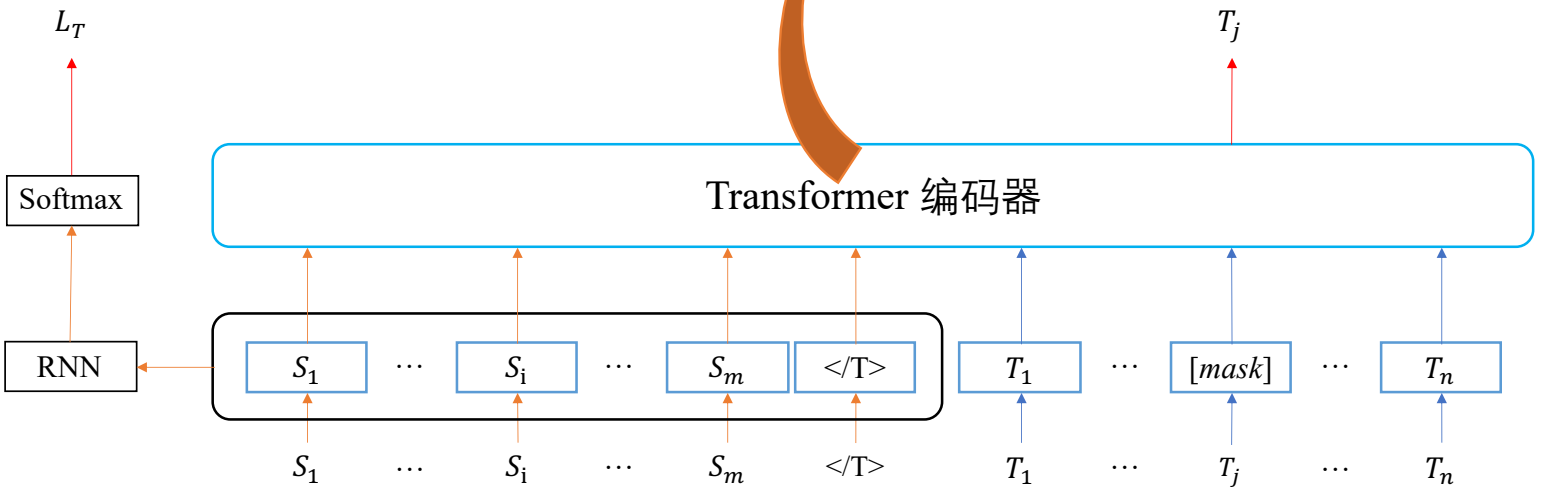
我们的想法

- 提出一个新的NAT模型：
  - ✓简化encoder-decoder的模型结构
  - ✓让源端信息和目标端信息在每一层都能交互

基于掩码机制的NAT

- 长度预测模块
- 掩码机制
- 非自回归生成





基于掩码机制的NAT

- 长度预测

  $H_t = tanh(WH_{t-1} + Us_t), t = 1,2,\dots,I$　　$s_1,\dots,s_i,\dots,s_I$为源端的输入

  $\hat{y_I} = softmax(VH_I)$，将$H_I$映射为$L_{max}$维的向量，$L_{max}$为目标语言句子的最大长度

  $Loss1 = CE(Y, \hat{y_I})$，Y为将目标端句子的真实长度编码成的一个L_{max}维的one-hot向量

- 掩码机制

  $$Loss2 = -\frac{1}{N}\sum_{i=1}^{N} log\, P_\theta(t_i | s_1^I, \tilde{T}_i)$$

  其中，$s_1^I = s_1,\dots,s_i,\dots,s_I$为源端句子，$t_1^J = t_1,\dots,t_j,\dots,t_J$为目标端句子，$\tilde{T}_i = \tilde{T}(t_1^{i-1}, t_{i+1}^J)$为对于目标语言句子$t_1^J$随机遮掩掉单词$t_i$后的其前后文信息

- 损失函数

  $Loss = Loss1 + Loss2$

基于掩码机制的NAT

- 遮掩-预测解码机制

在每一轮（总共N轮）：

- **遮掩：**在第t轮，对于上一轮预测得到的目标端句子，遮掩掉句子中预测概率p_i最低的k个词$T_{mask}^{(t)}$后，得到目标端句子$\widetilde{T^{(t)}}$

$$k = L \cdot \frac{N-t}{N}$$

$$T_{mask}^{(t)} = \operatorname{argmin}_i(p_i, k)$$

- **预测：**预测出被遮掩掉的词$y_i^{(t)}$及其预测概率，并对这部分词的预测概率进行更新，形成新的目标端句子$\widehat{T^{(t)}}$ 。

$$y_i^{(t)} = \operatorname*{argmax}_w P\left(y_i = w \middle| s_1^I, \widetilde{T^{(t)}}\right)$$

$$p_i^{(t)} = \max_w P\left(y_i = w \middle| s_1^I, \widetilde{T^{(t)}}\right)$$

实验设计

- 数据集
  - WMT 2016 Romanian-English
    - 训练集: 613k，开发集: 2k，测试集: 2k
    - MOSES tokenizer
    - 联合BPE
- 掩码规则
  - 在训练时，随机将目标端句子的N个词用[*mask*]代替，其中N从均匀分布$U(1, L)$中取样得到，L为目标端句子的长度。
- 解码的优化迭代次数
  - 解码时，我们将迭代优化的次数设为10
- 评测指标
  - SacreBLEU

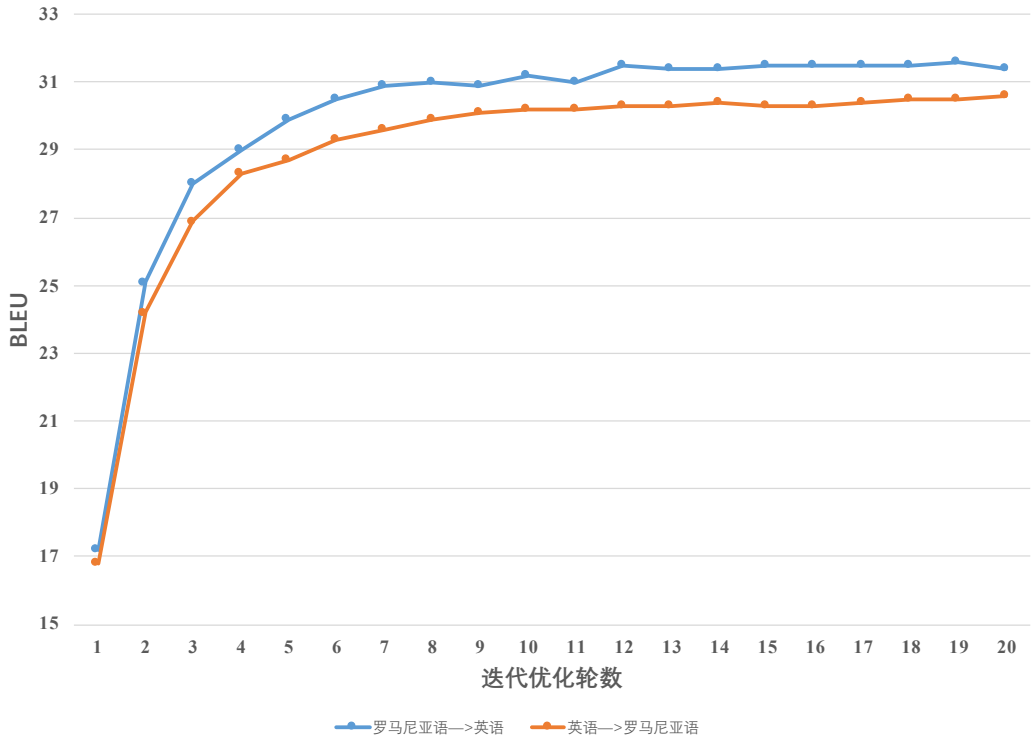
结果分析

- 主要结果

系统	BLEU		解码复杂度
	英语→罗马尼亚语	罗马尼亚语→英语	
Fertility-based (Gu等人, 2018)	27.3	29.1	$O(1)$
CTC (Libovicky和Helcl, 2018)	19.5	24.7	
Imitation learning (Wei等人, 2019)	28.6	28.9	
Reinforcement learning (Shao等人, 2019)	27.1	27.9	
Generative flow (Ma等人, 2019)	29.3	30.2	
Extra refinement model (Lee等人, 2018)	29.3	30.2	$O(T)$
我们的方法	**30.2**	**31.2**	

结果分析

- 不同迭代优化轮数的影响
  - 在迭代优化轮数*N*小于12时，迭代的轮数越多，模型解码的效果越好，而在N大于12时，模型解码的效果趋于平稳，不会因为迭代的次数增多而产生较大的影响。



结果分析

- 模型不同初始化参数的影响
  - 用预训练的英语-罗马尼亚语MLM参数初始化我们的Transformer编码器
  - 英语→罗马尼亚语和罗马尼亚语→英语两个语向上的BLEU都取得了很大的提升
  - 同时与Lample等人使用预训练的MLM参数初始化Transformer编码器、解码器得到的自回归神经机器翻译模型性能相当

	BLEU		参数量
	英语→罗马尼亚语	罗马尼亚语→英语	
我们的方法（参数随机初始化）	30.2	31.2	约100M
我们的方法（使用预训练的MLM参数初始化）	33.8	34.5	约100M
Transformer（使用预训练的MLM参数初始化编码器、解码器）	-	35.3	约200M

结果分析

- 给定目标端长度的表现
  - 解码效果并没有很大的提升
  - 说明我们的长度预测模块在整个模型中起到了不错的效果

	BLEU	
	英语→罗马尼亚语	罗马尼亚语→英语
我们的方法	30.2	31.2
我们的方法 +给定目标端长度	30.4	31.5

结果分析

- 不同源端信息的影响
  - 直接用Transformer编码器对源语言句子进行独立编码，然后将编码器最终输出的隐状态作为源端信息，参与每一层目标端注意力的计算。
  - 效果不及我们的方法。因为每一层的源端隐状态都关注到了不同的信息，参与该层目标端注意力计算能让目标端注意到该层的信息，让源端信息和目标端信息在每一层都能交互。

	BLEU	
	英语→罗马尼亚语	罗马尼亚语→英语
我们的方法	30.2	31.2
独立编码源端注意力信息，参与每一层目标端注意力计算	29.6	30.5

总结

- 我们提出了一个新的NAT模型：
  - ✓简化了encoder-decoder的模型结构
  - ✓长度预测模块、掩码机制
  - ✓让源端信息和目标端信息在每一层都能交互

Thanks!