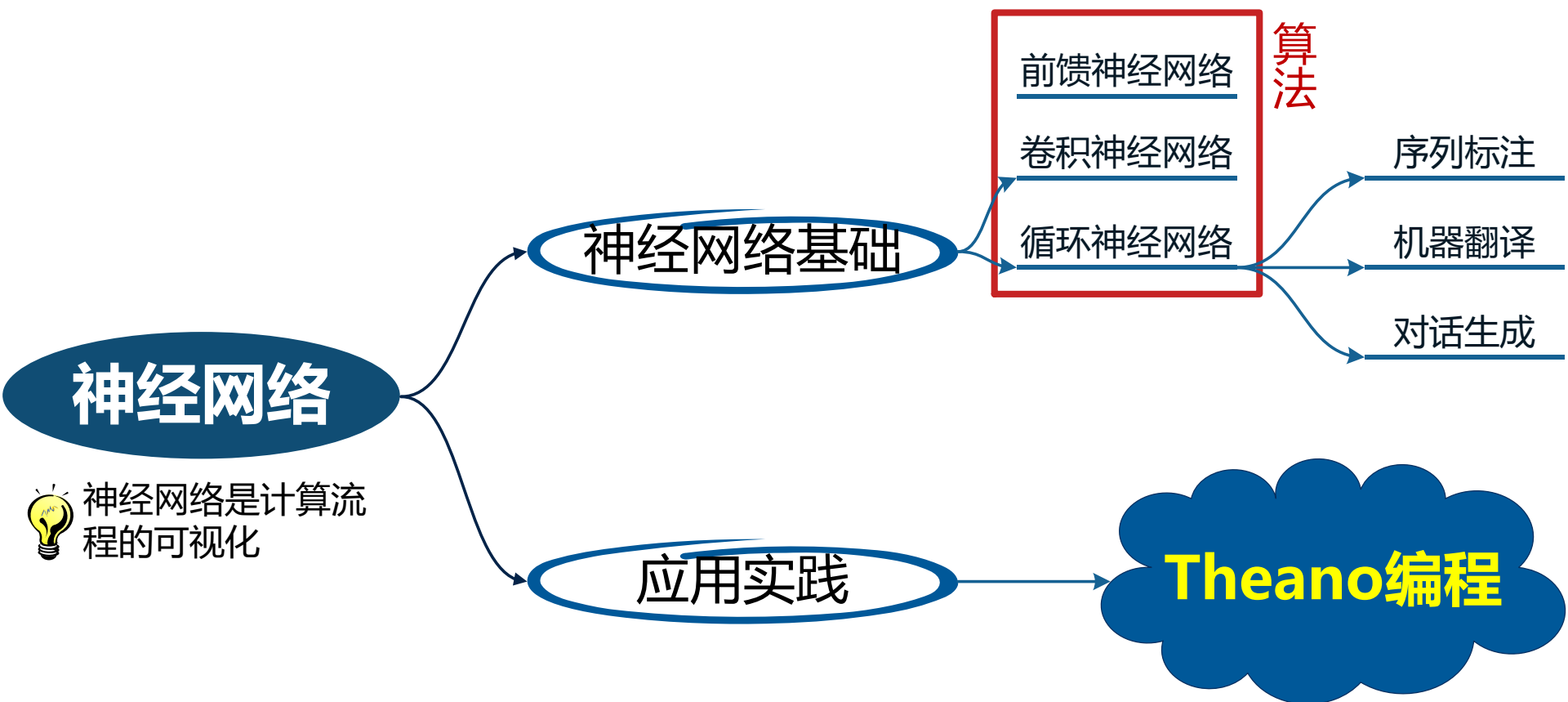


课程专题三：神经网络方法



监督学习的本质

学习从**A**到**B**的映射过程

1. 基于大量标注数据学习
2. 根据问题背景，确定从**A**到**B**的函数形式（怎样设计才合理）

Andrew Ng: 思考时间少于n秒的人类工作都将被取代

自然语言处理任务

- 序列标注

- 分词

从/S哈/B尔/M滨/E到/S广/B州/E的/S机/B票/E

- 词性标注

从/P哈尔滨/NS到/P广州/NS的/U机票/N

- 命名实体识别、关键词抽取...

- 序列到序列

- **机器翻译**

第13届上海国际电影节开幕



The 13th Shanghai International Film Festival is open

- **对话系统**

- 摘要、事件抽取...



北京今天会不会下雨?

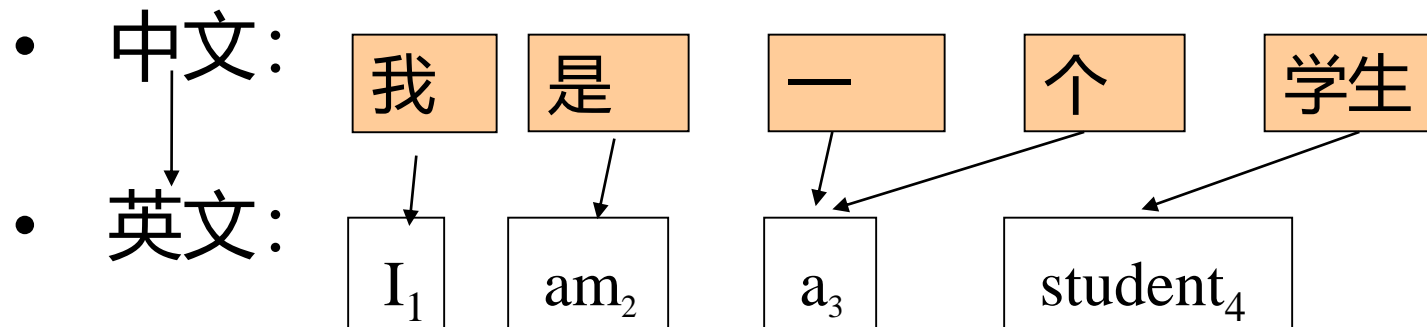
北京今天晴天, 不会下雨。



应用一：翻译系统

- 统计机器翻译 $T^* = \arg \max_T P(S | T)$

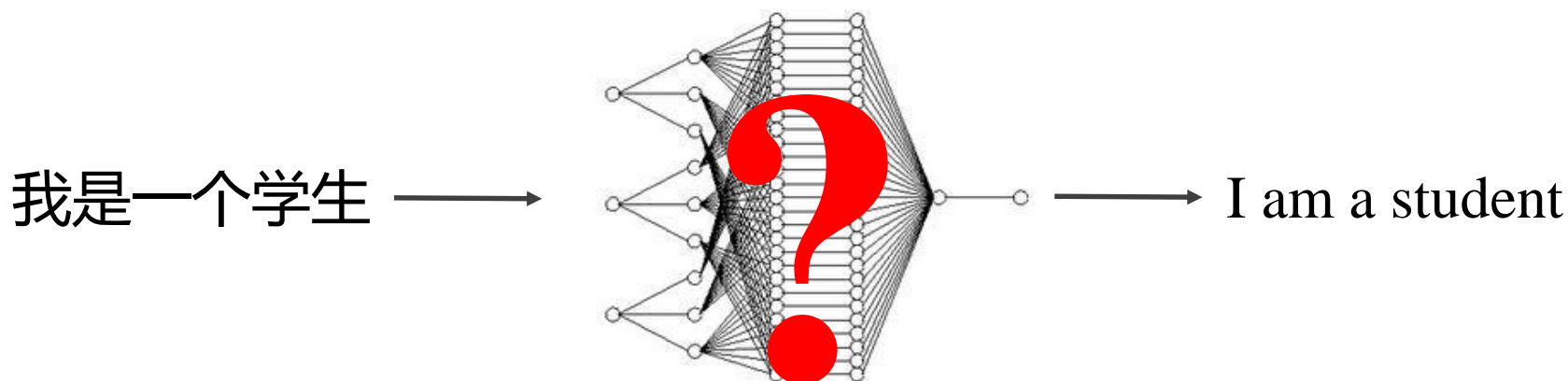
- 其核心技术是短语对齐



- 短语对齐后，再用语言模型串成一句话。

神经翻译模型

- 传统的短语翻译模型，**难以处理短语间的重排序，并且短语之间的衔接不顺畅。**
- 我们能否设计一种翻译算法，从一个完整句子出发，**一次性**翻译出另一个完整句子？

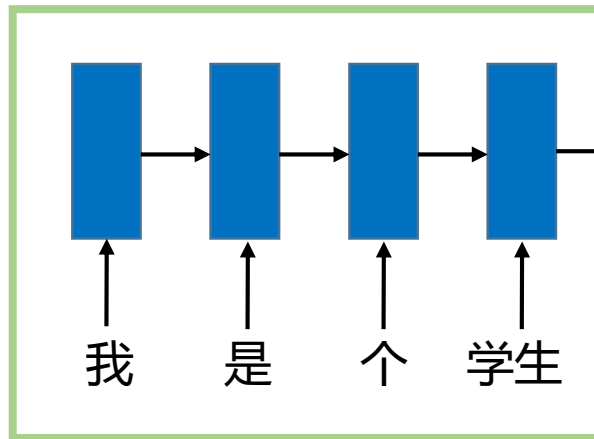


预备知识：词嵌入(word embedding)

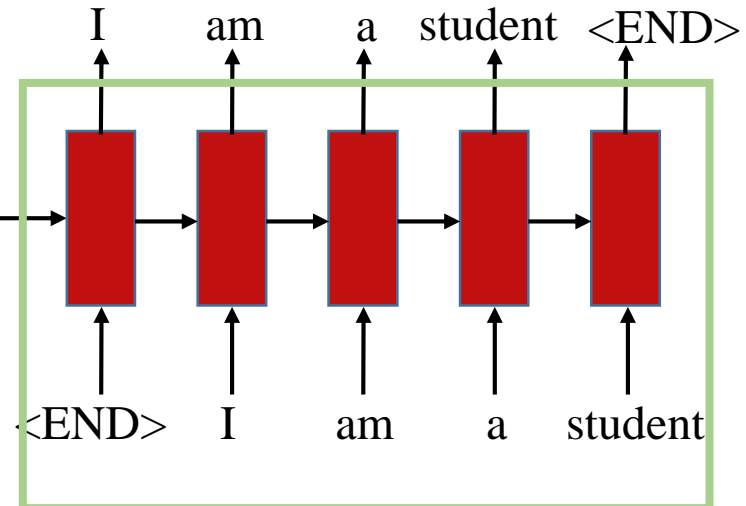
- 词的状态空间非常大(十万量级)而且离散。
- 文本状态下，词语无法直接计算相似度。
- 需要将文字转化为可计算的向量，向量和向量就可以直接计算了。
- 有各种各样的计算词嵌入的方法，如CBOW、Skip-Gram等，这里不详细介绍。

RNN翻译模型

Encoder



Decoder

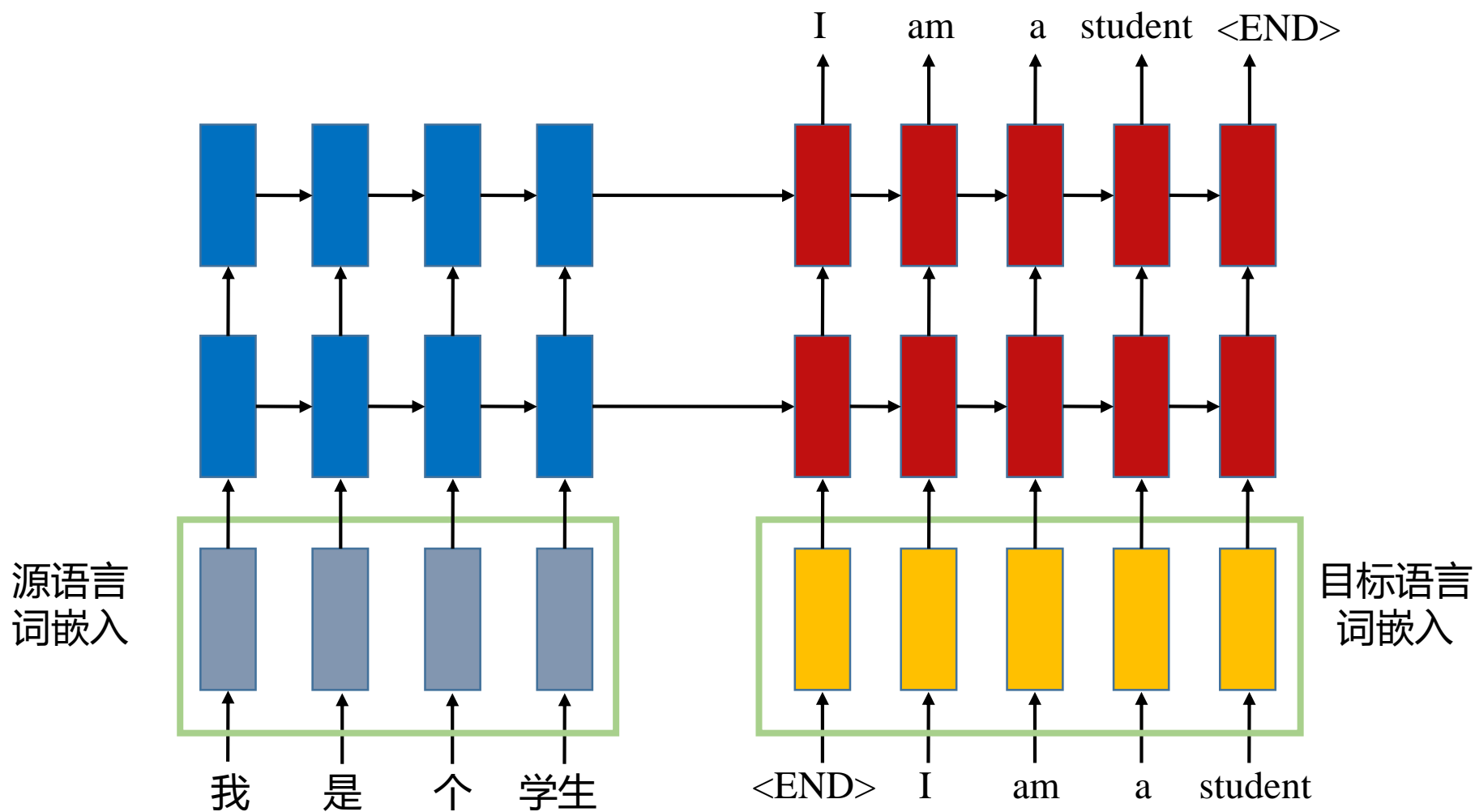


RNN:

- 建模 $P(\text{目标语句}|\text{源语句})$
- 可以端到端训练

I.Sutskever,O.Vinyals,andQ.VLe,“Sequence to sequence learning with neural networks,”in*Advances in Neural Information Processing Systems(NIPS)*,2014,pp.3104–3112.

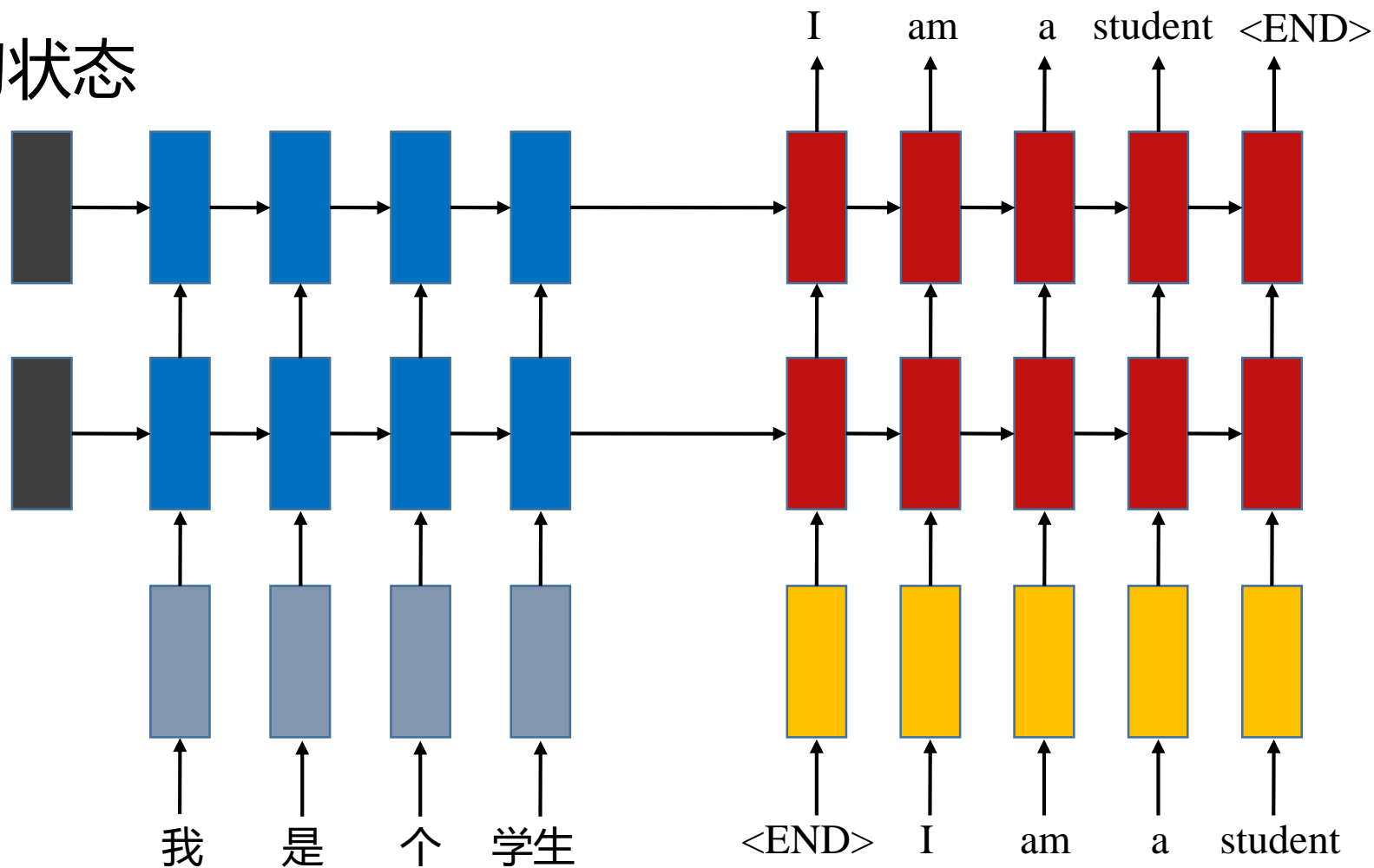
RNN翻译模型的计算过程



- 学习不同语言中的词嵌入。

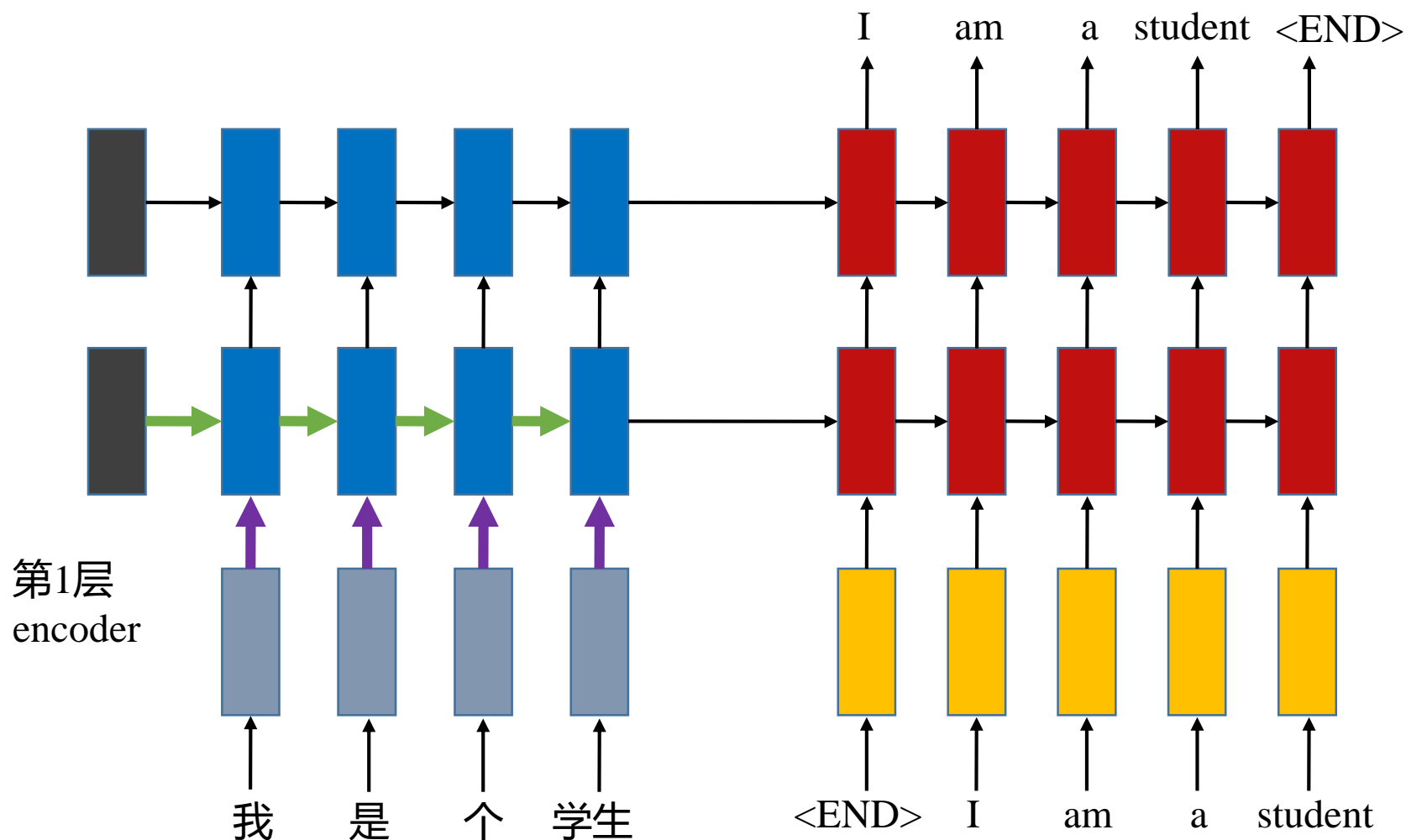
RNN翻译模型的计算过程

初状态



- 初状态习惯上设置为零向量。

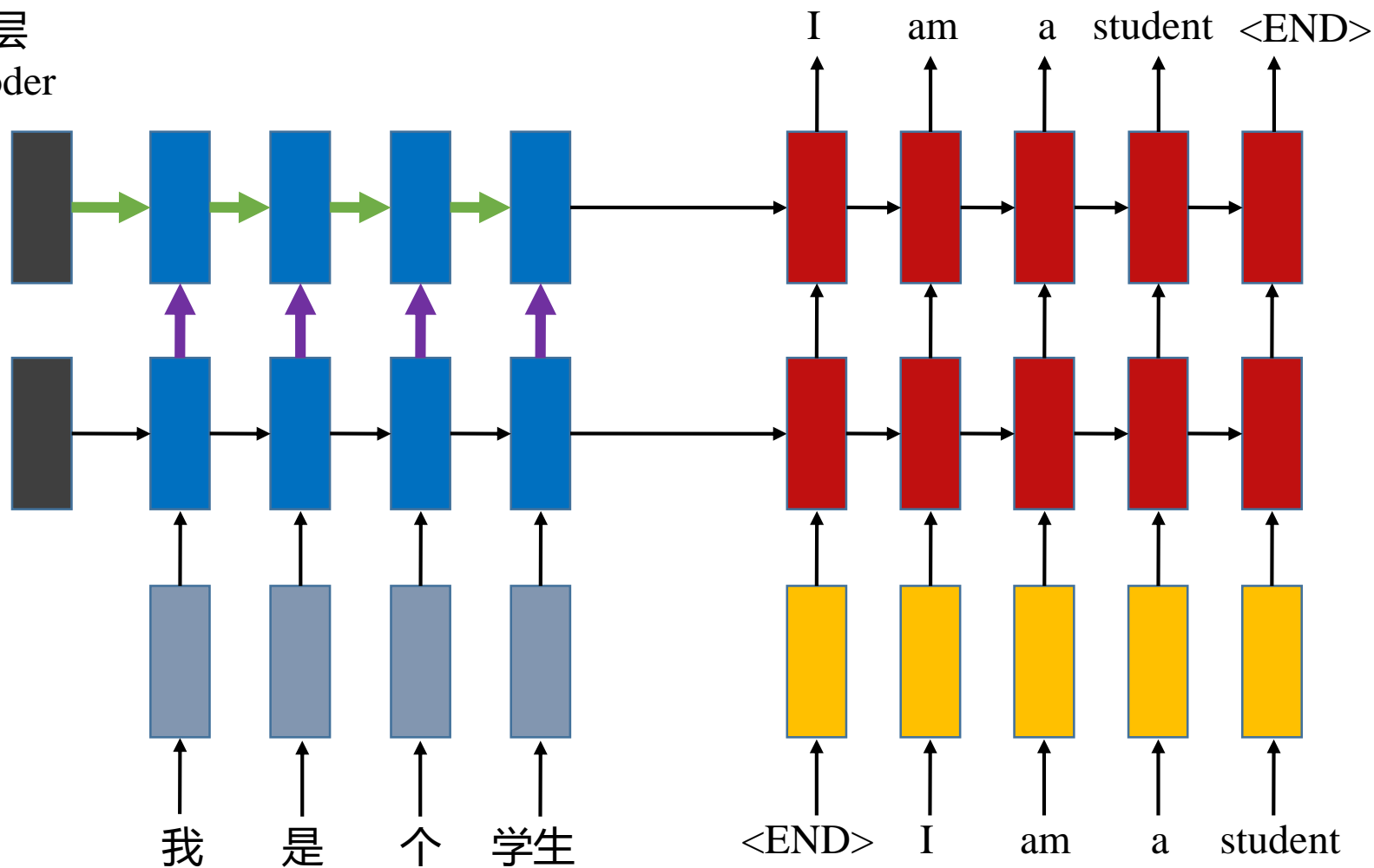
RNN翻译模型的计算过程



- 可以构建多层网络。

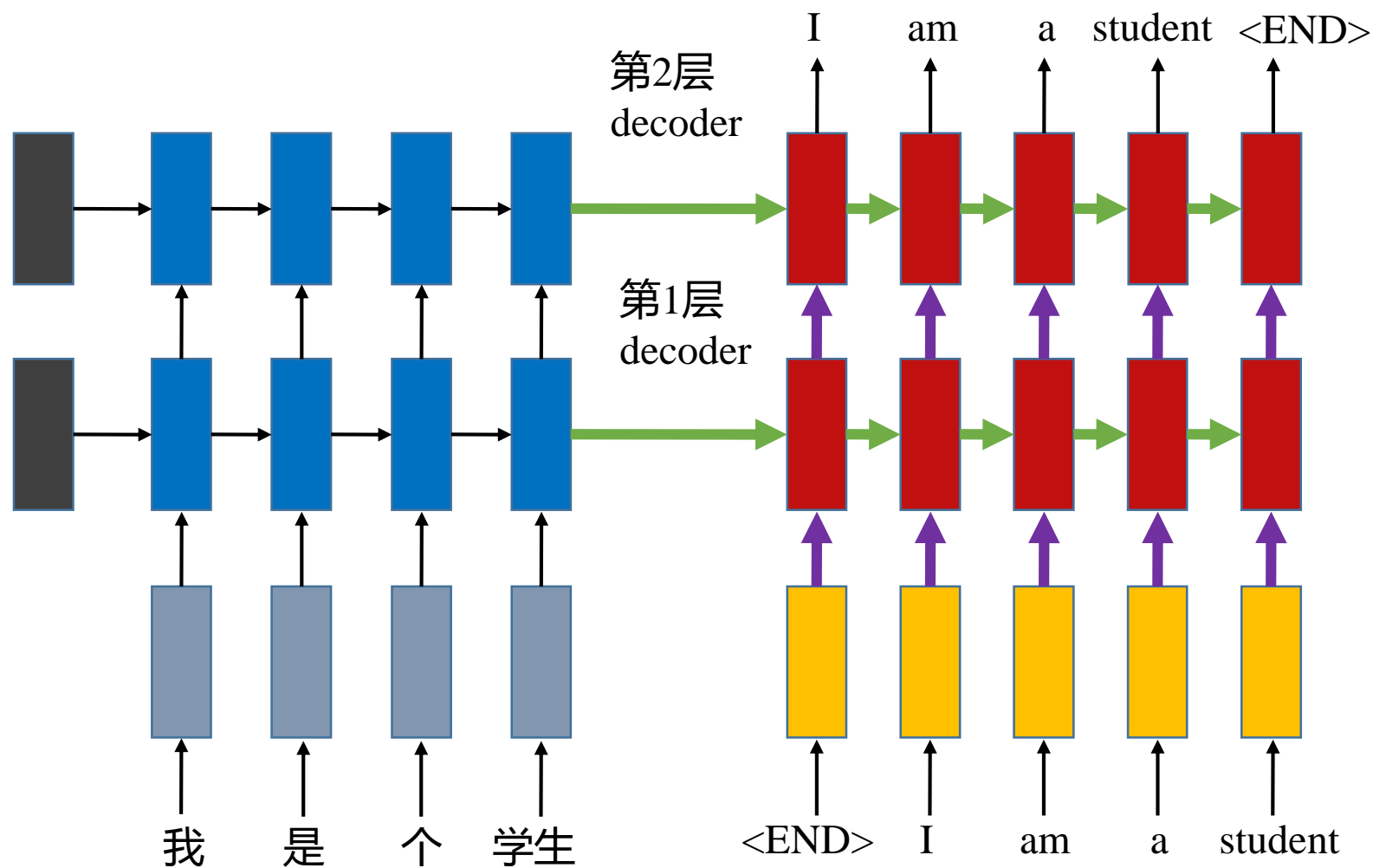
RNN翻译模型的计算过程

第2层
encoder



- 可以构建多层网络。

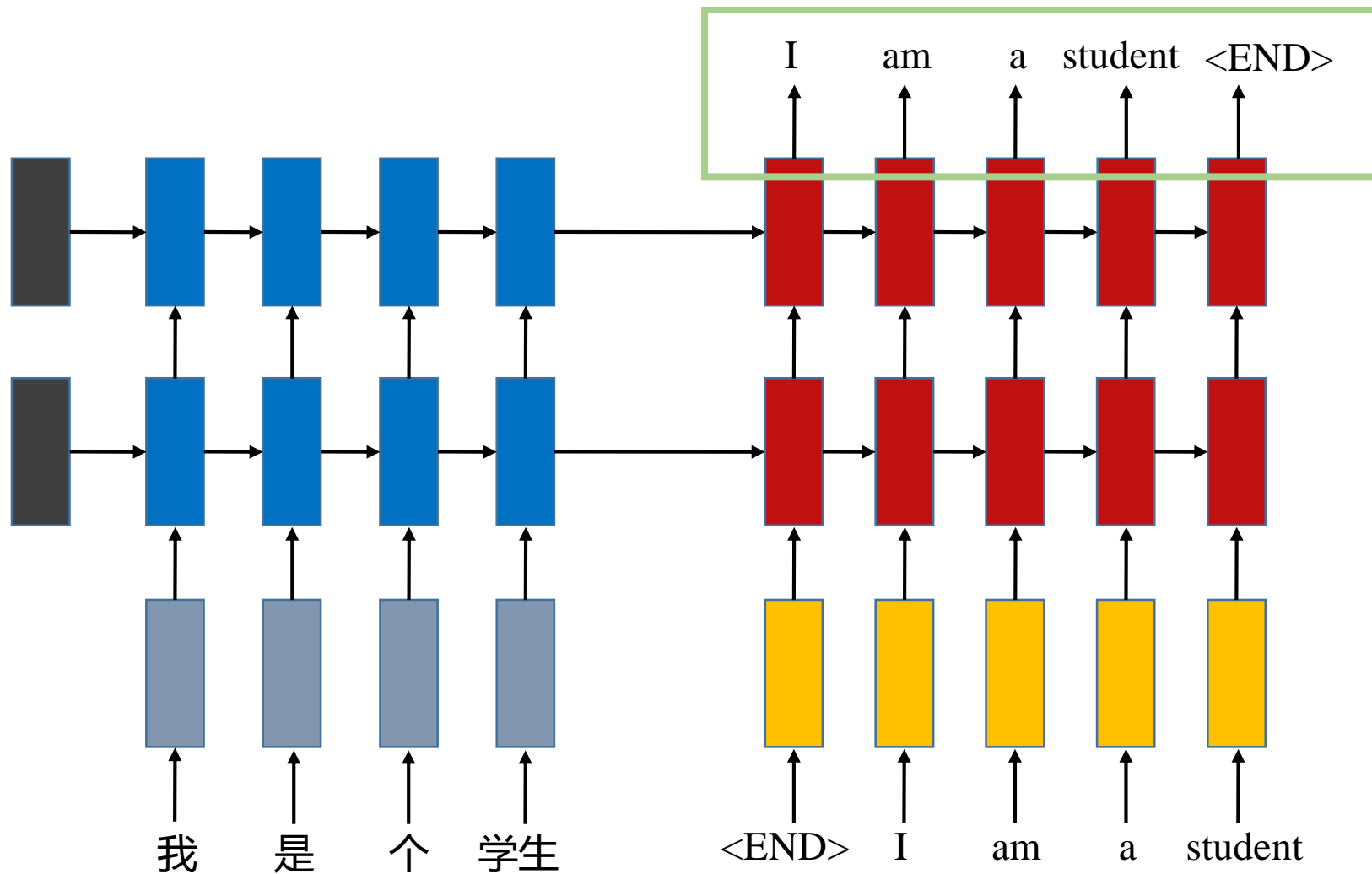
RNN翻译模型的计算过程



- 可以构建多层网络。

RNN翻译模型的计算过程

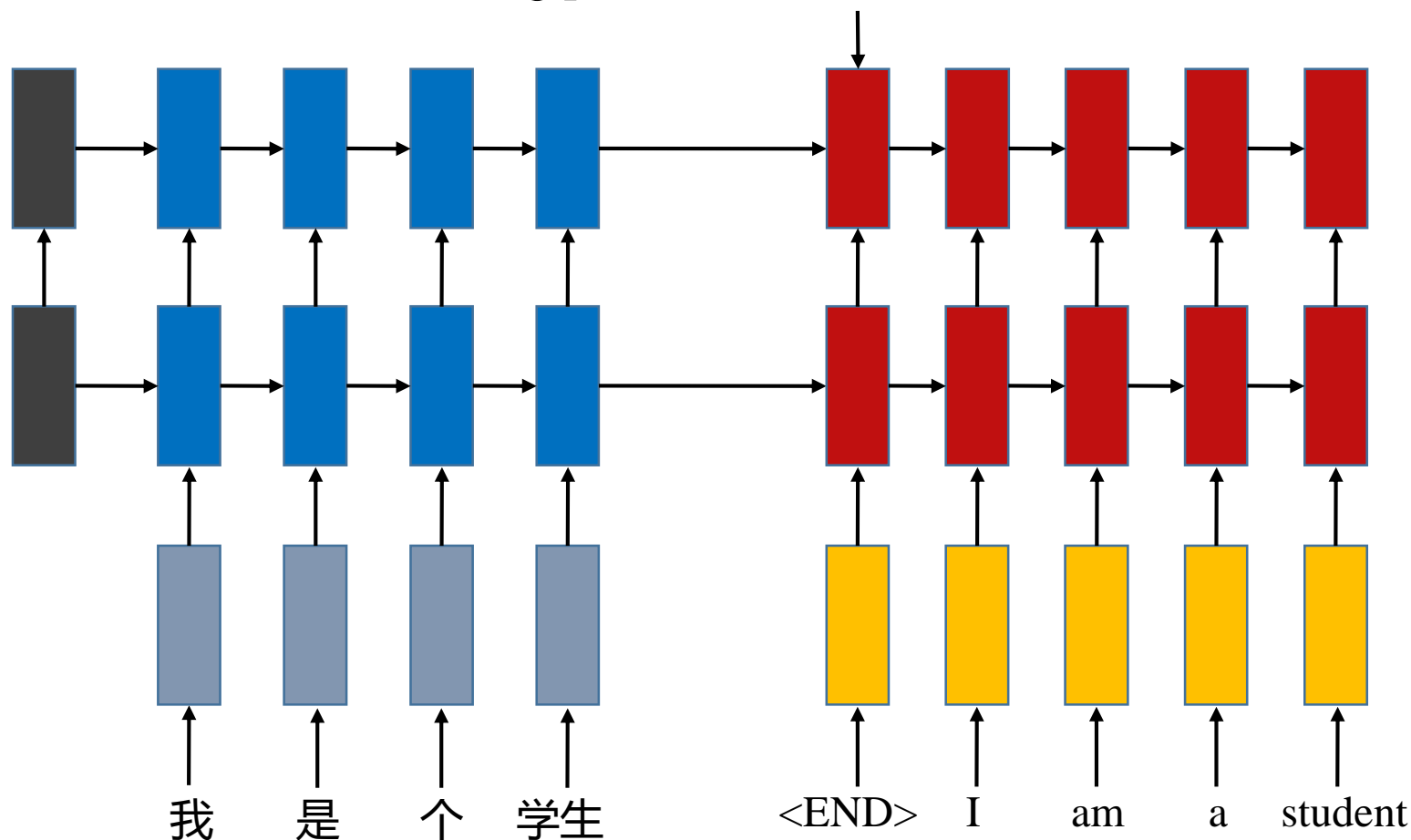
Softmax建模下个词概率分布



RNN翻译模型的训练目标

$$p(Y|X) = \prod_{y_t \in Y} p(y_t | Y_{<t}, X)$$

$-\log p(\text{I}|\text{我是个学生}, \text{<END>})$

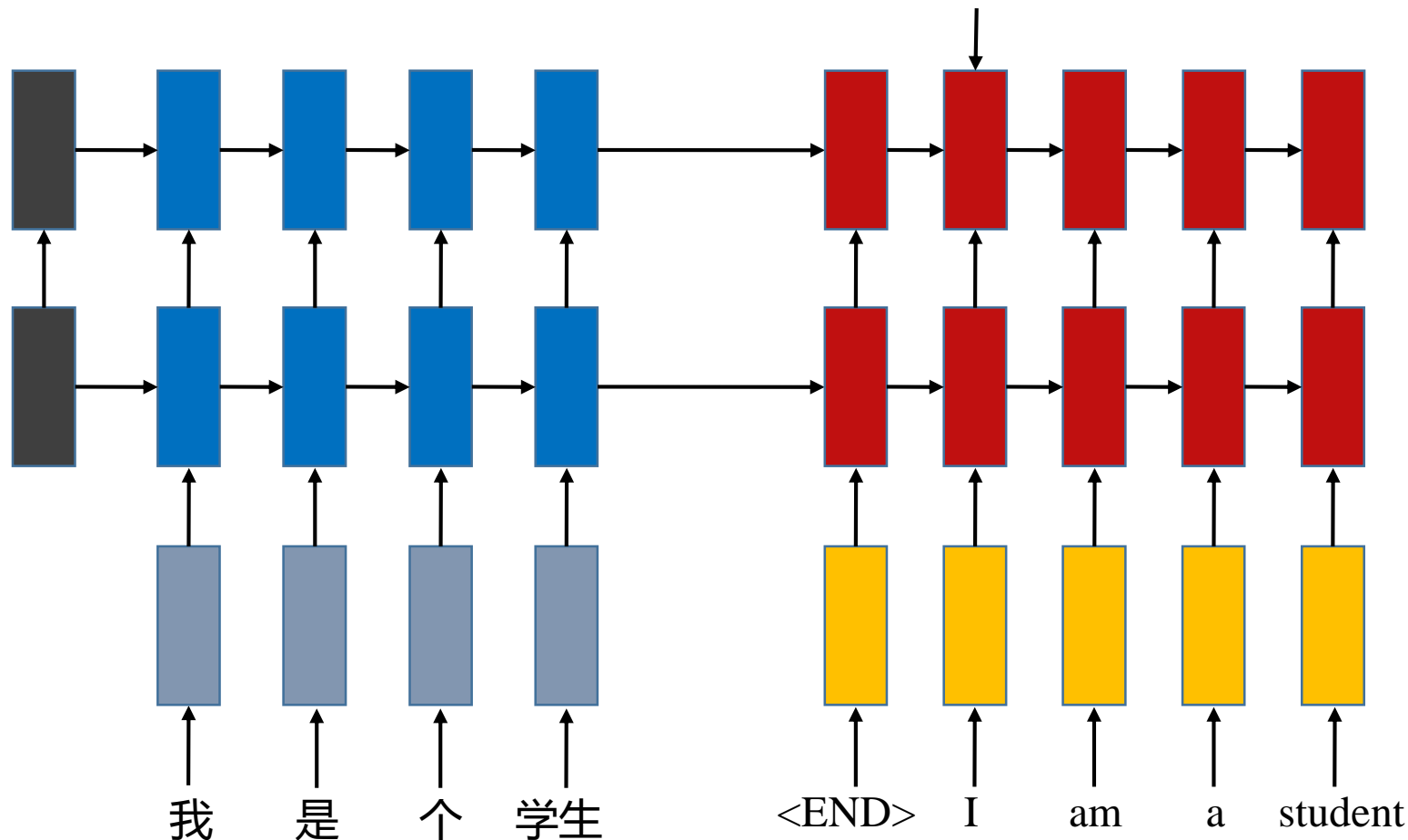


- 每个译句上单词的损失函数求和。

RNN翻译模型的训练目标

$$p(Y|X) = \prod_{y_t \in Y} p(y_t | Y_{<t}, X)$$

$$-\log p(\text{am} | \text{我是个学生}, <\text{END}> \text{I})$$

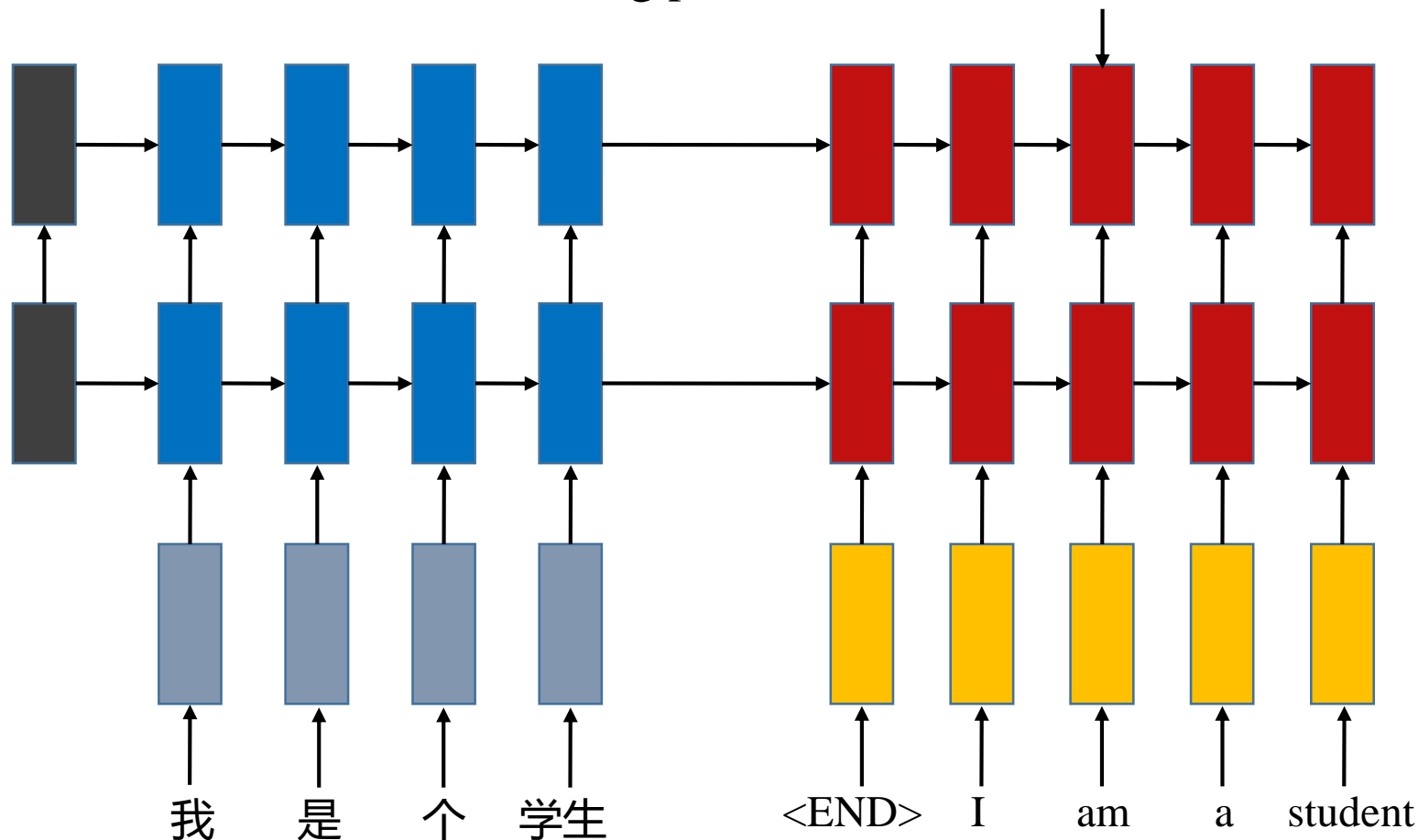


- 每个译句上单词的损失函数求和。

RNN翻译模型的训练目标

$$p(Y|X) = \prod_{y_t \in Y} p(y_t | Y_{<t}, X)$$

$$-\log p(\mathbf{a} | \text{我是个学生}, \text{<END>I am})$$

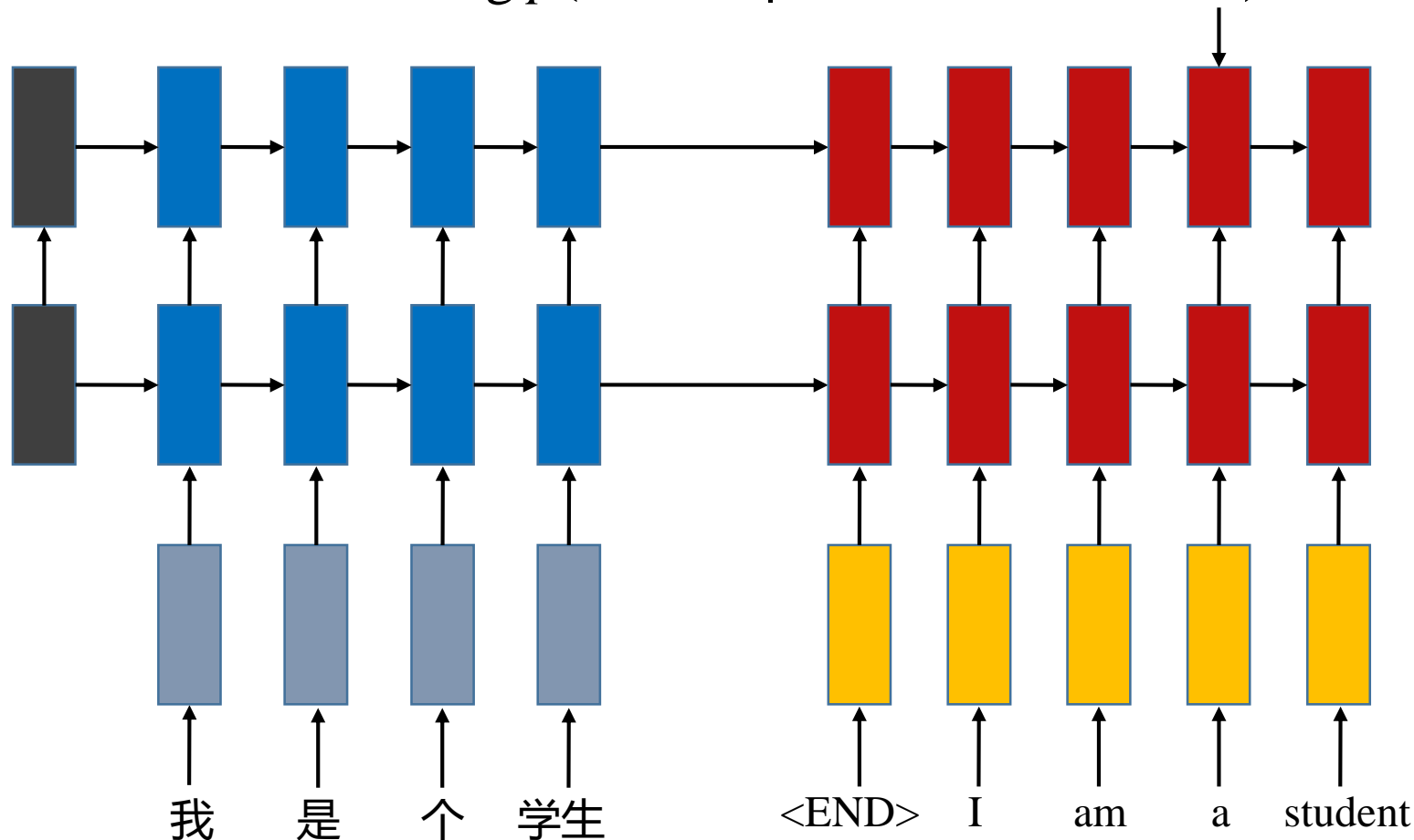


- 每个译句上单词的损失函数求和。

RNN翻译模型的训练目标

$$p(Y|X) = \prod_{y_t \in Y} p(y_t | Y_{<t}, X)$$

$-\log p(\text{student}|\text{我是个学生}, \text{<END>I am a})$

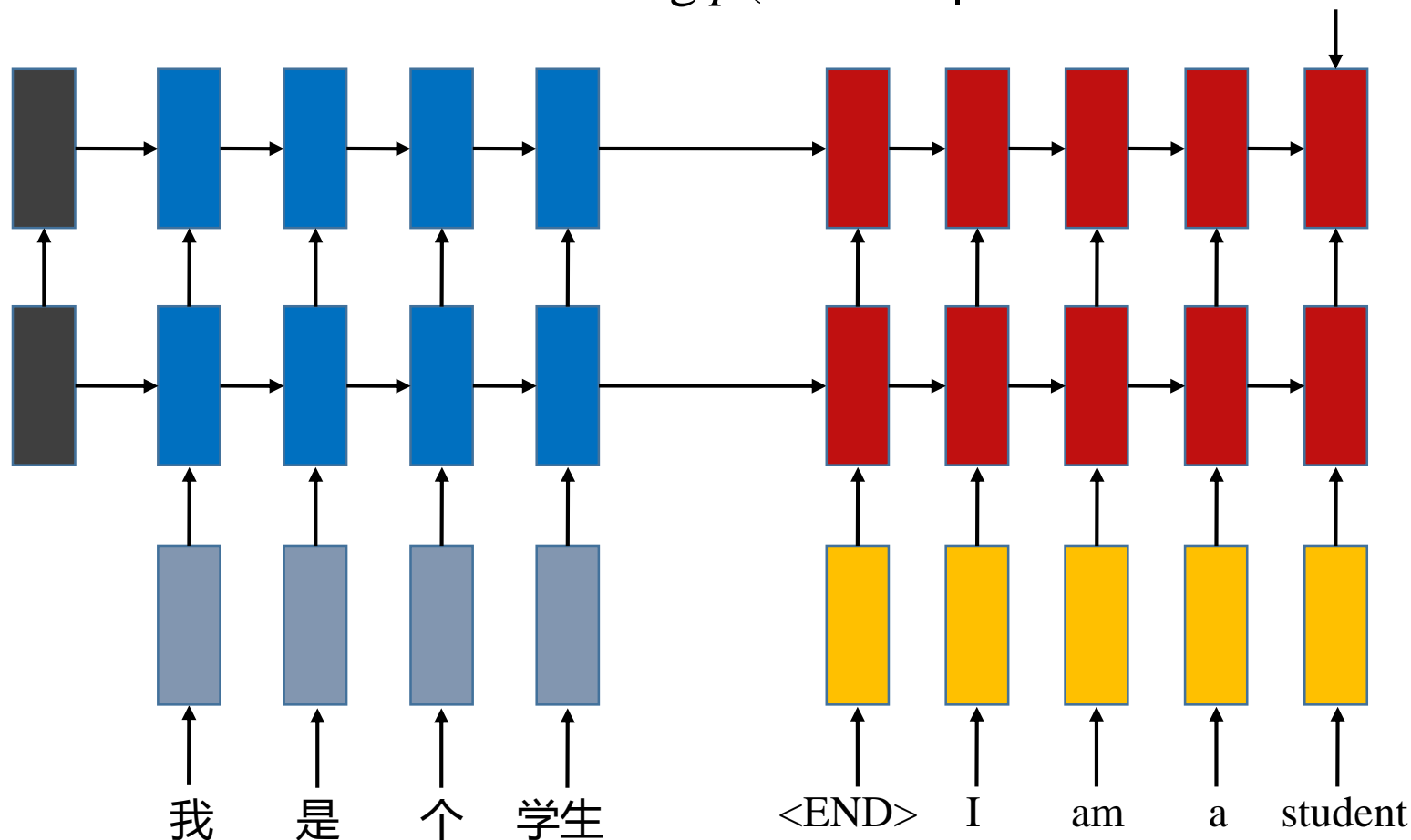


- 每个译句上单词的损失函数求和。

RNN翻译模型的训练目标

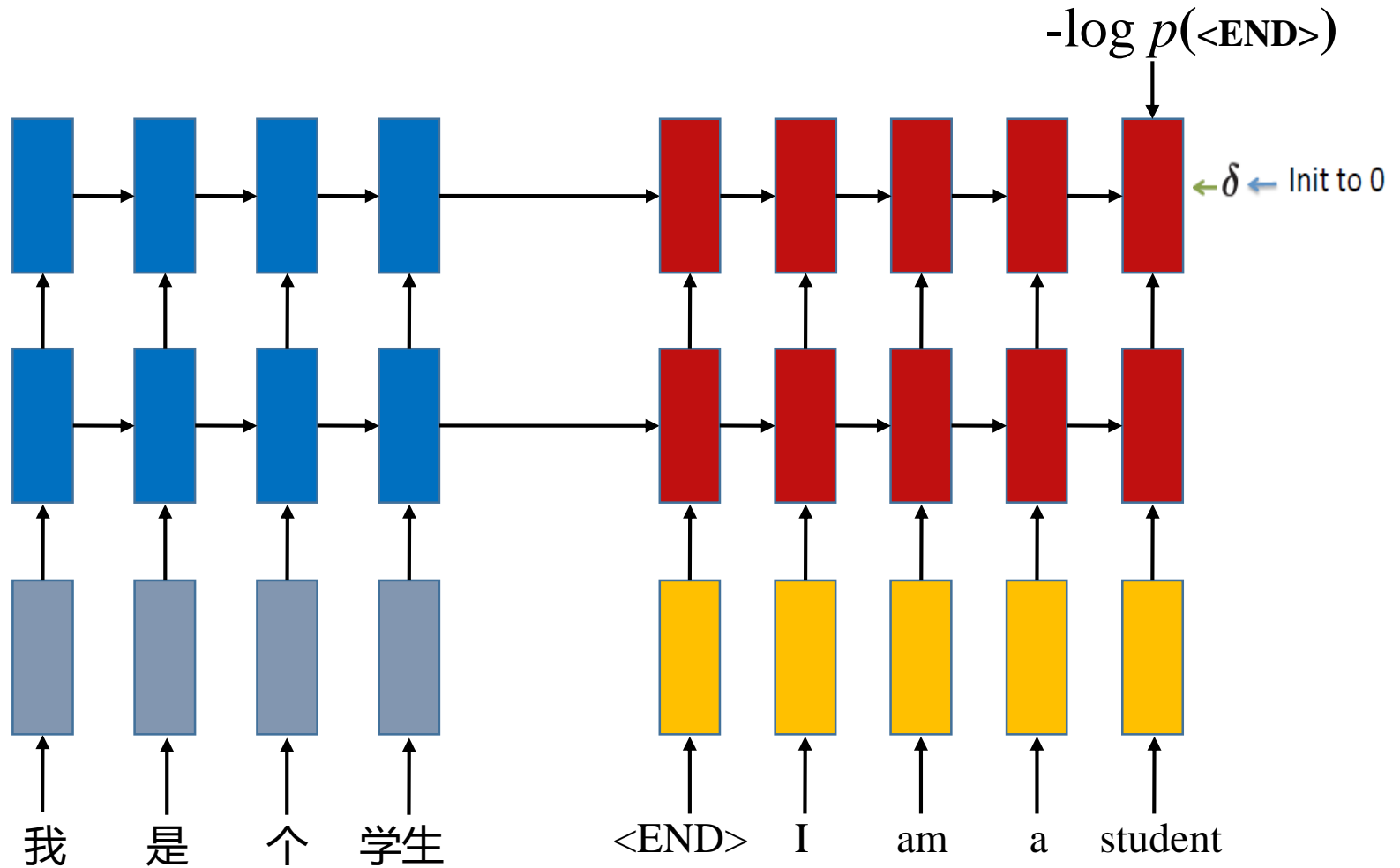
$$p(Y|X) = \prod_{y_t \in Y} p(y_t | Y_{<t}, X)$$

$$-\log p(\text{<END>} | \text{我是个学生}, \text{<END>} \text{I am a student})$$



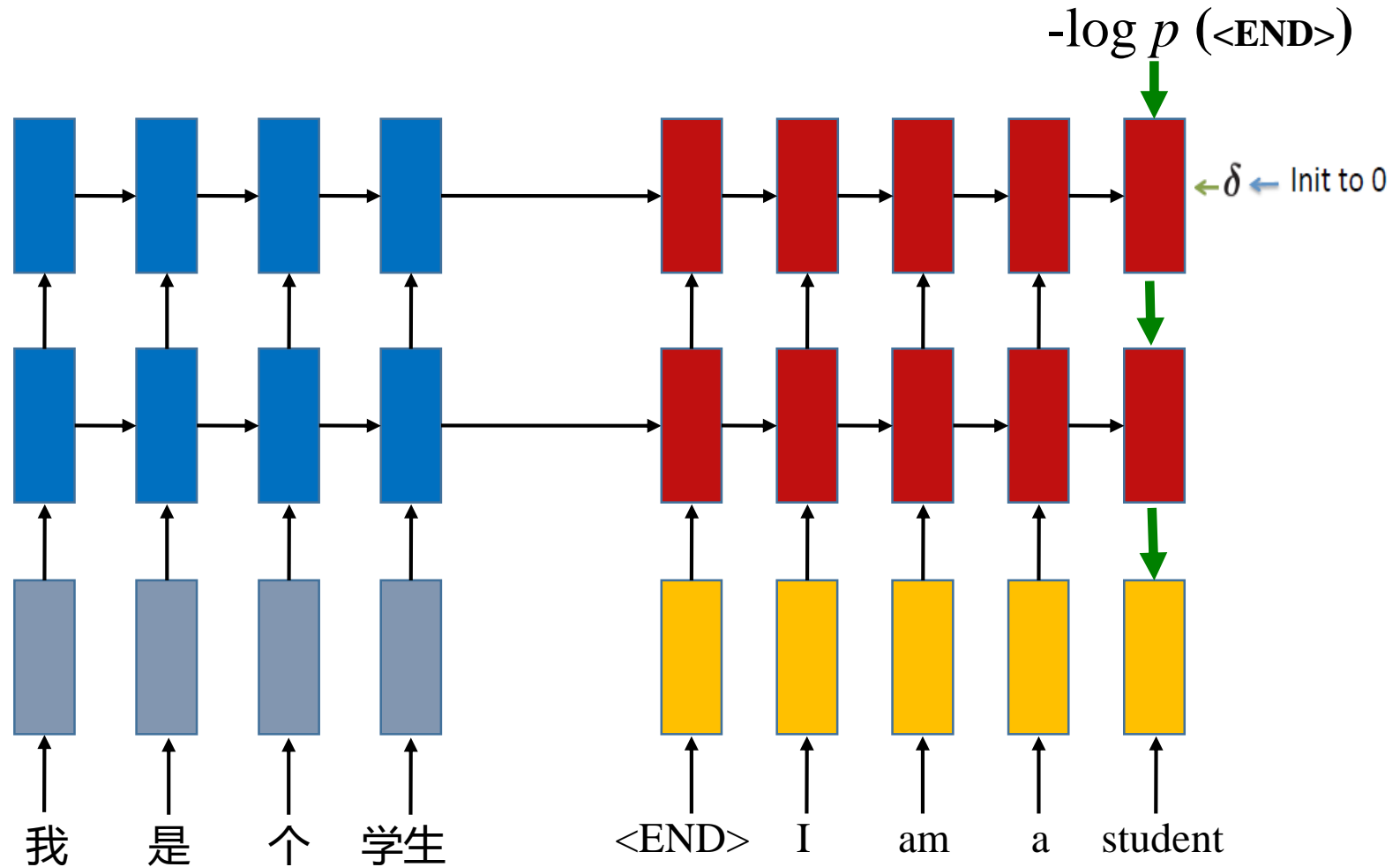
- 每个译句上单词的损失函数求和。

Back propagation Through Time



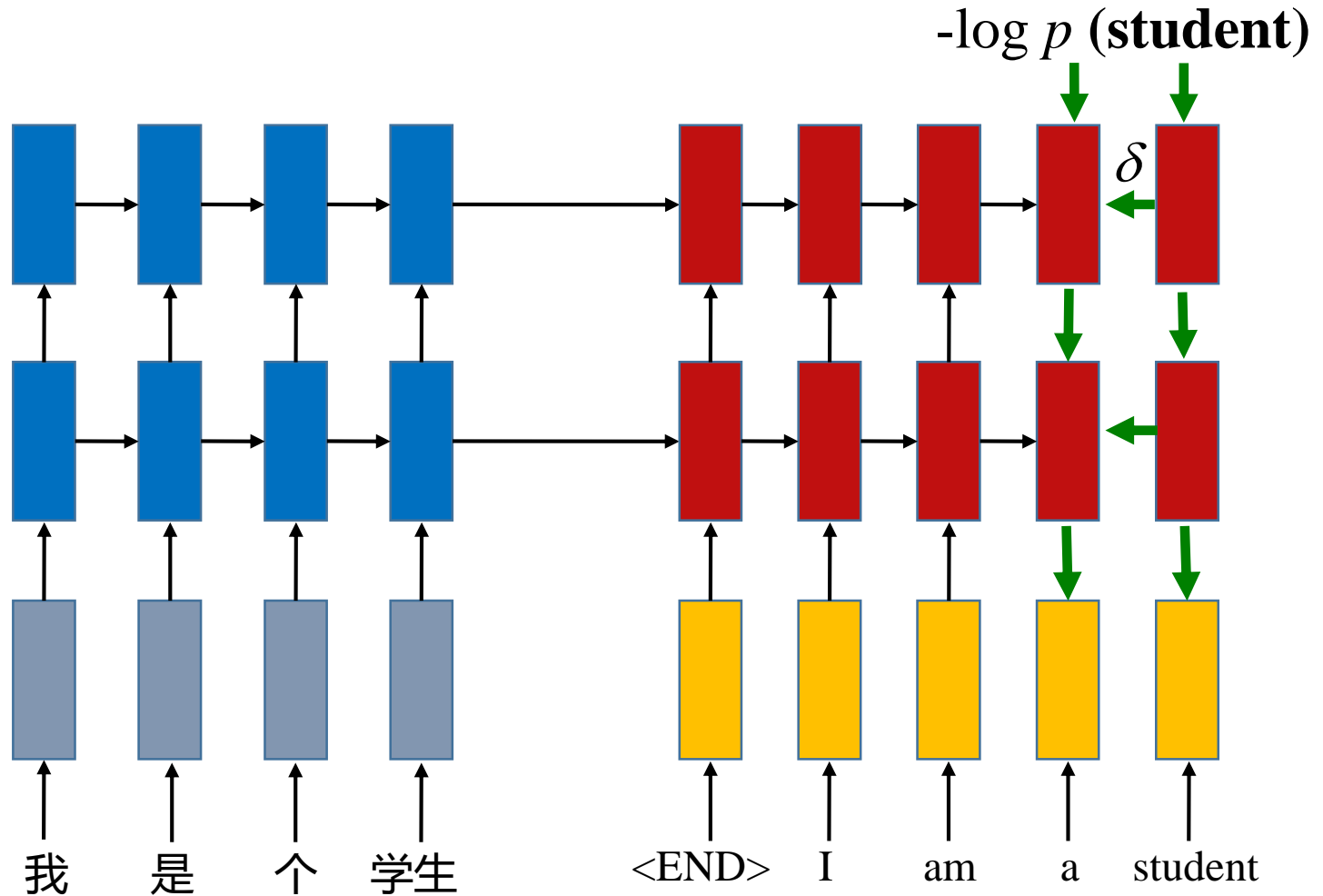
- 把误差层层传递到第一个隐藏层。

Back propagation Through Time



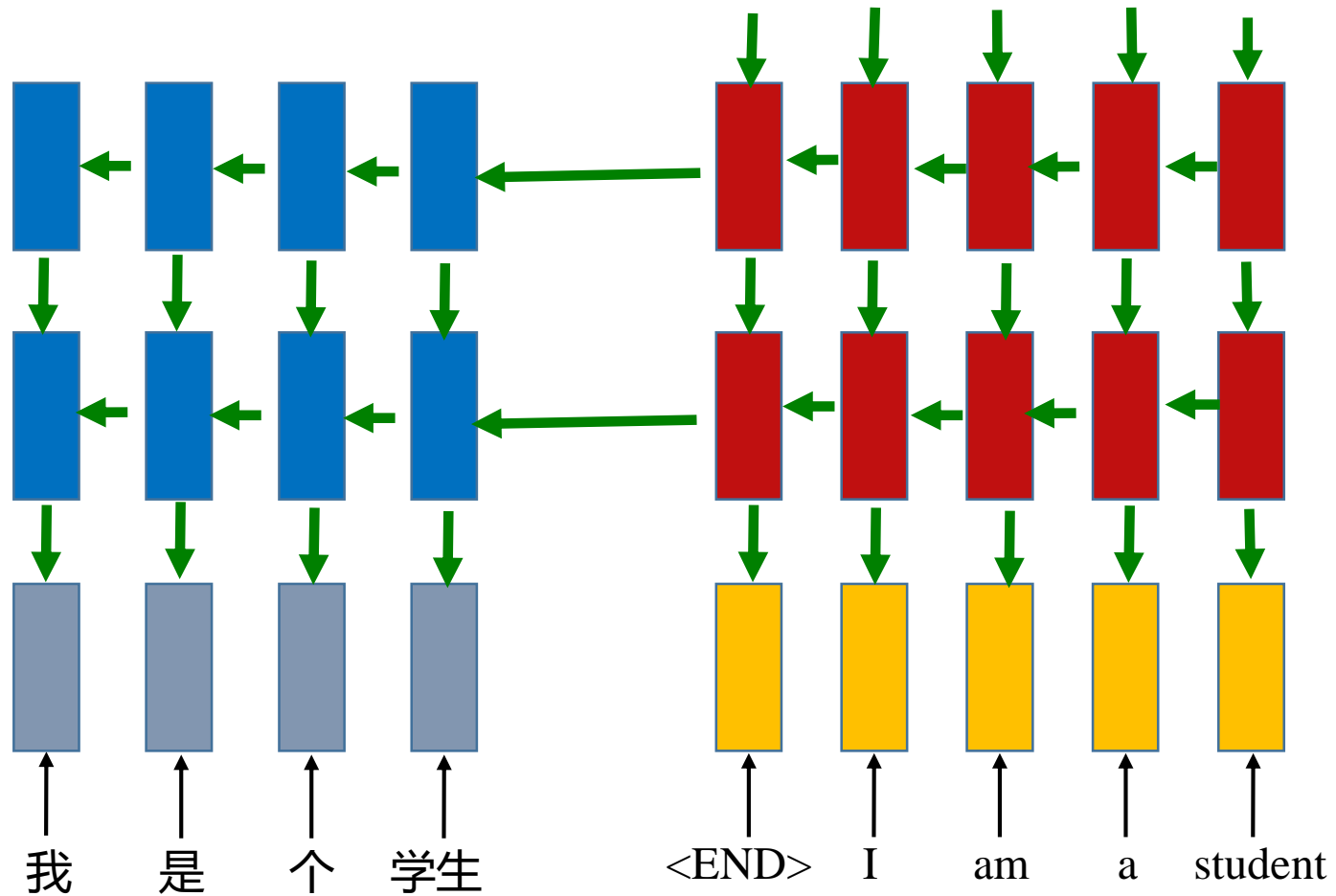
- 把误差层层传递到第一个隐藏层。

Back propagation Through Time



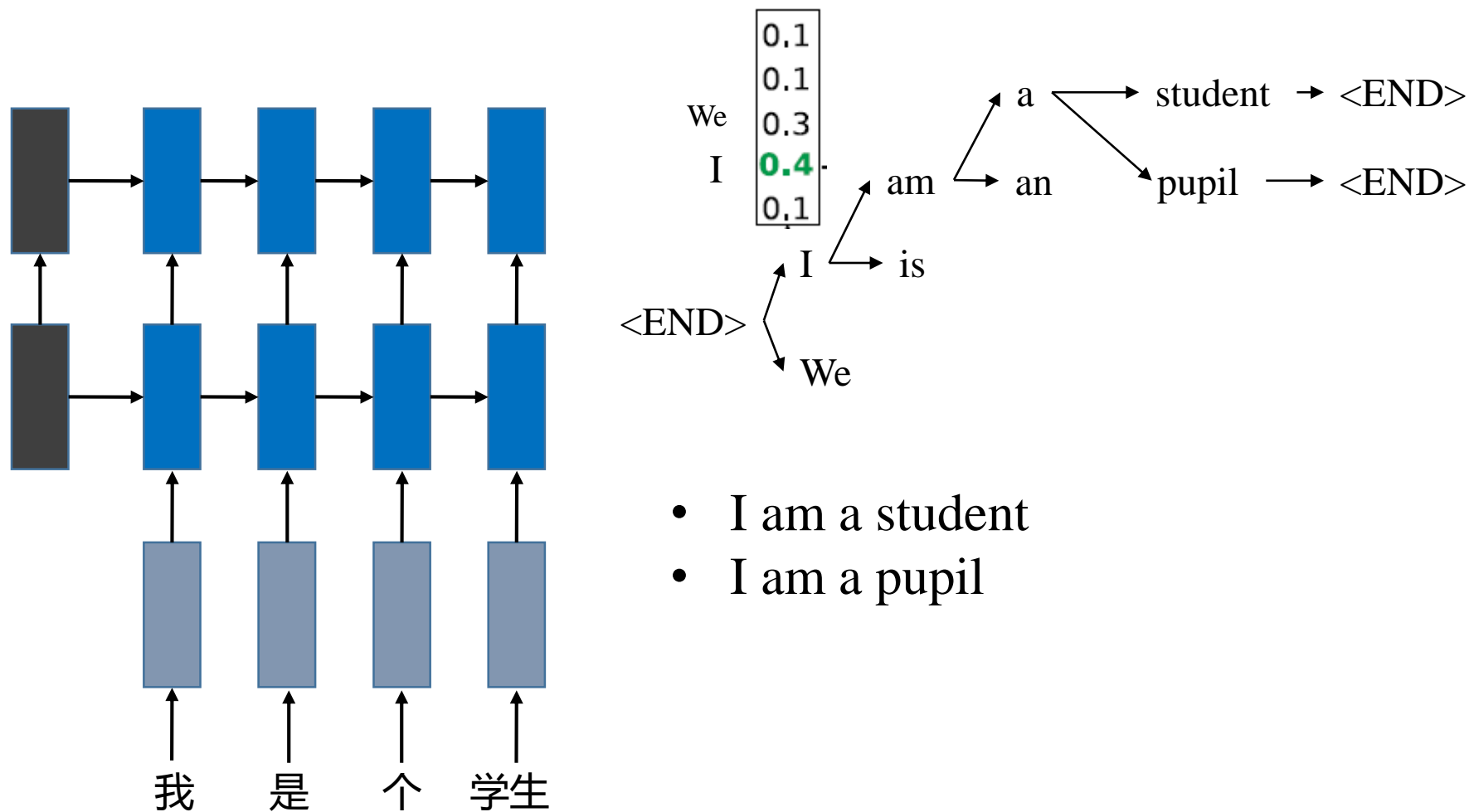
- 把误差层层传递到第一个隐藏层。

Back propagation Through Time



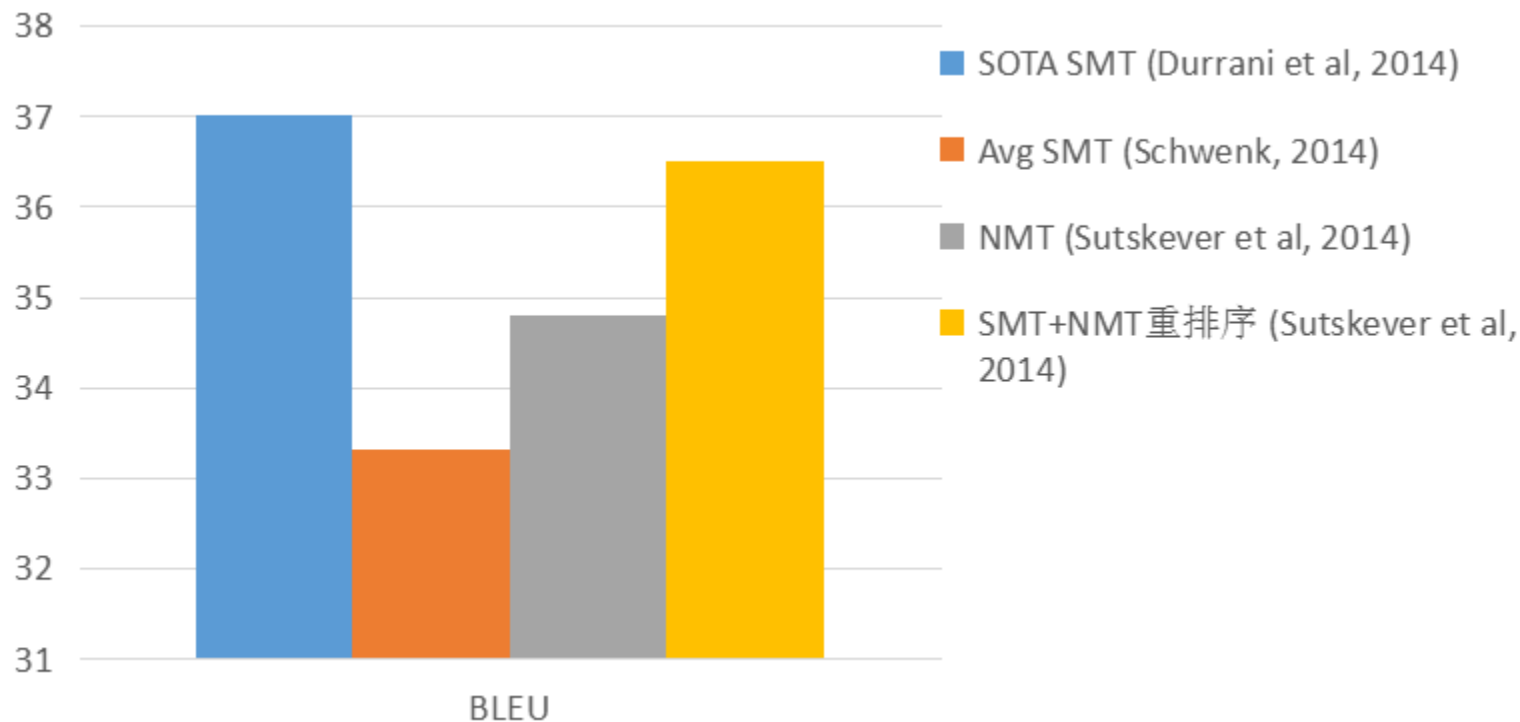
- 把误差重新层层传递到第一个隐藏层。

RNN翻译模型的生成过程



- 给定源句，利用BeamSearch从左到右生成。

神经机器翻译NMTVS统计机器翻译SMT

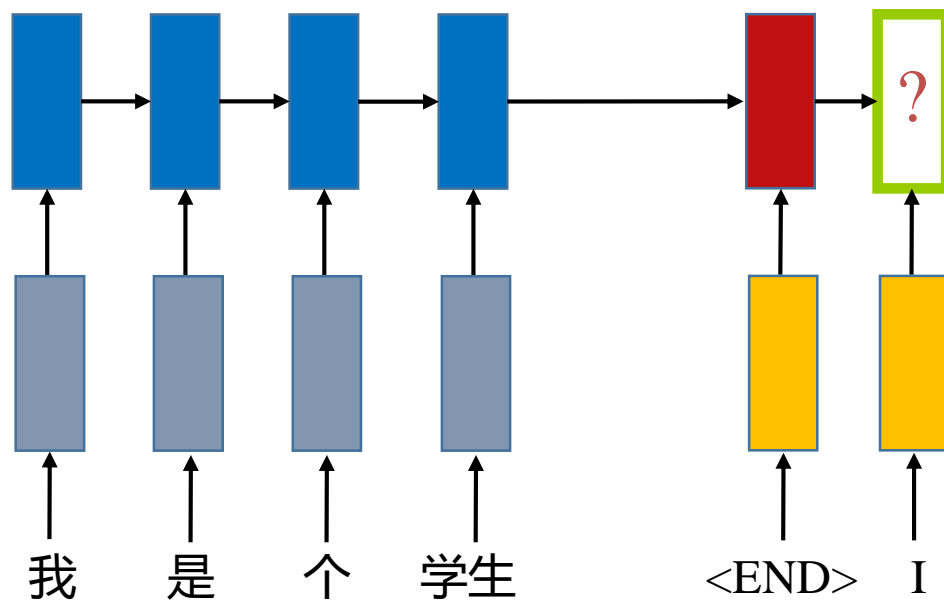


NMT从提出用了1年多时间，几乎打败了SMT20多年的研究成果

基本RNN翻译模型遇到的问题

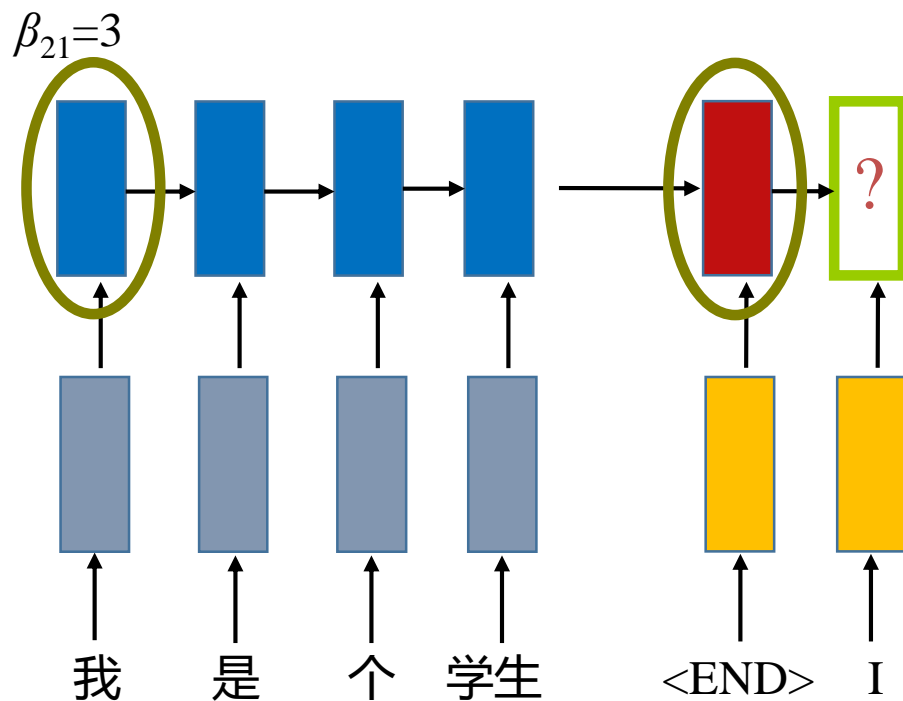
- 朴素RNN模型只能保留最后处理的单词的信息
- 对于长句子，会遗忘掉源句早期处理单词的信息
 - “你是不是不爱我了” 翻译成 “Don't love me”
- 引入注意力机制 (attention mechanism)

注意力机制



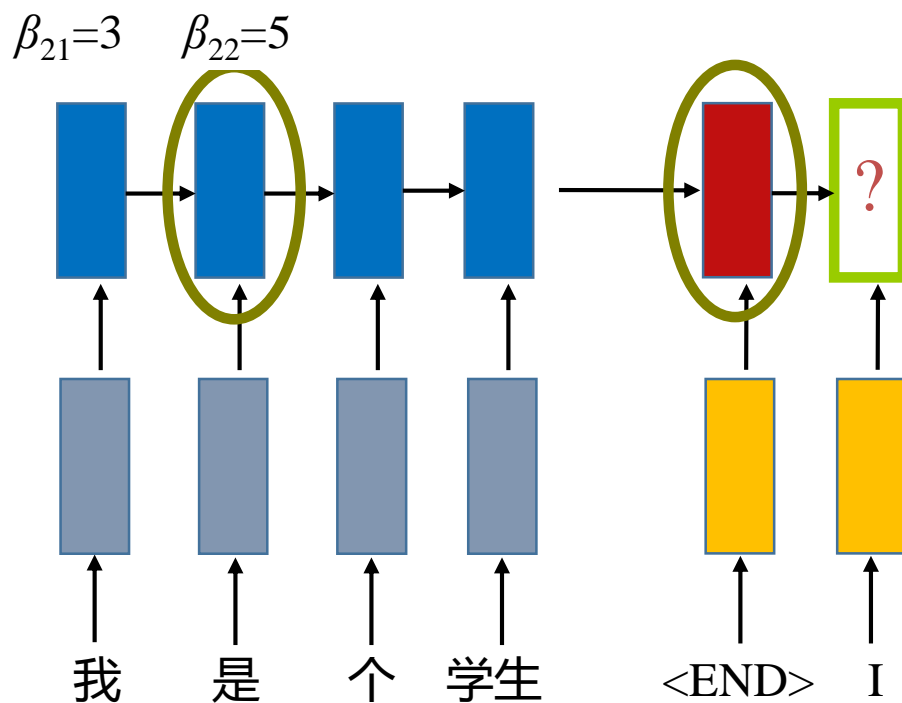
注意力机制–Scoring

$$\beta_{21} = q(s_1, h_1)$$



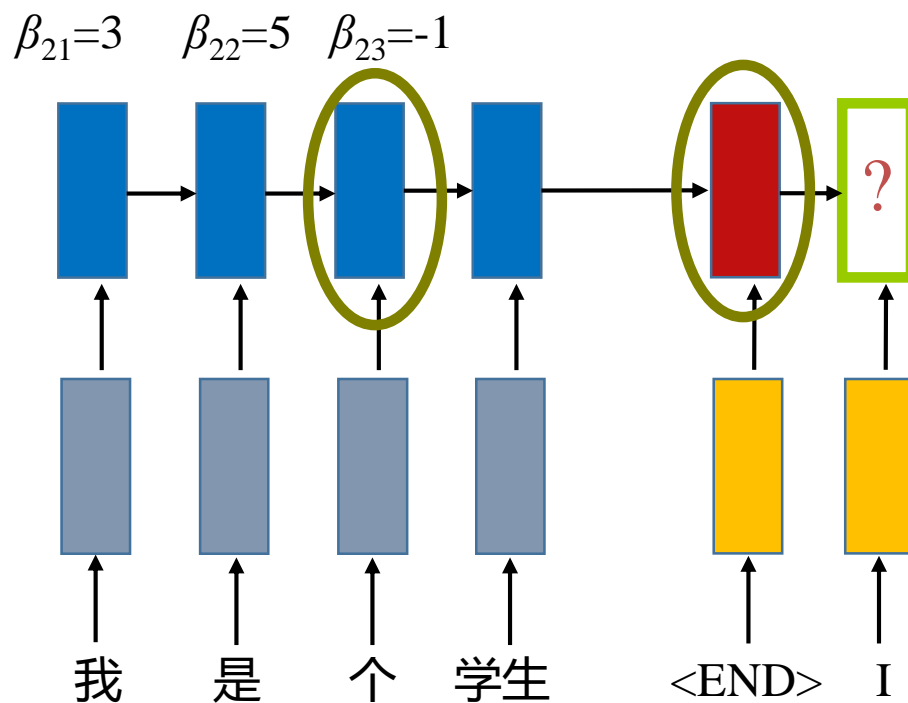
注意力机制–Scoring

$$\beta_{22} = q(s_1, h_2)$$



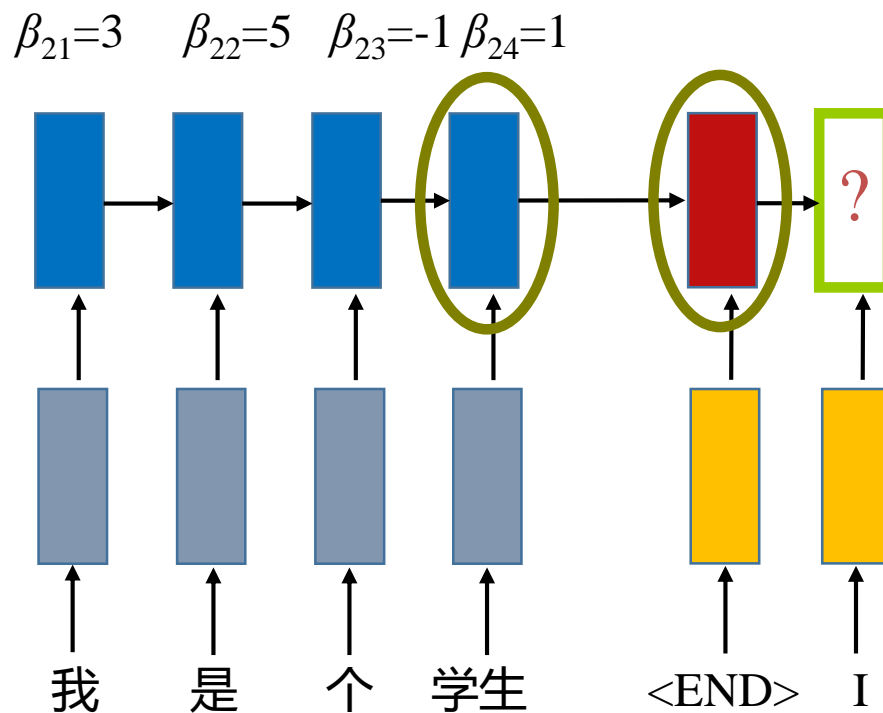
注意力机制–Scoring

$$\beta_{23} = q(s_1, h_3)$$

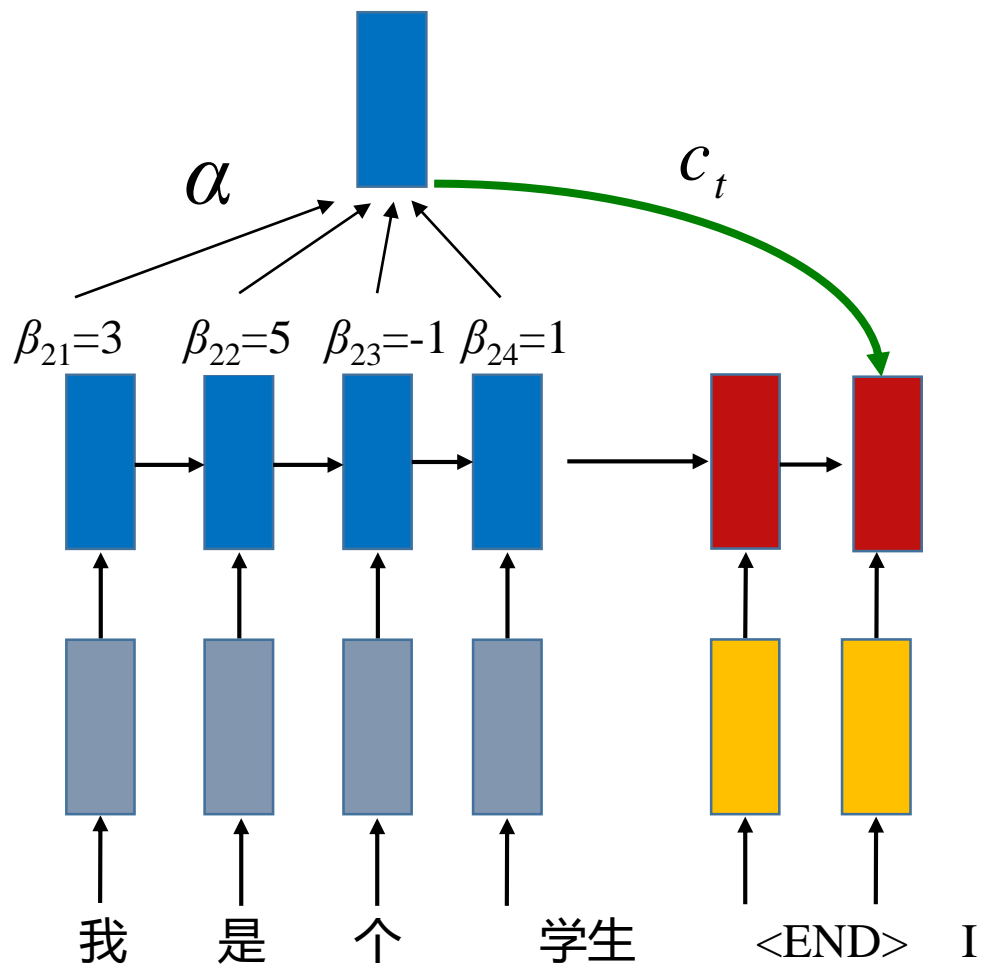


注意力机制–Scoring

$$\beta_{24} = q(s_1, h_4)$$



注意力机制–Hidden State



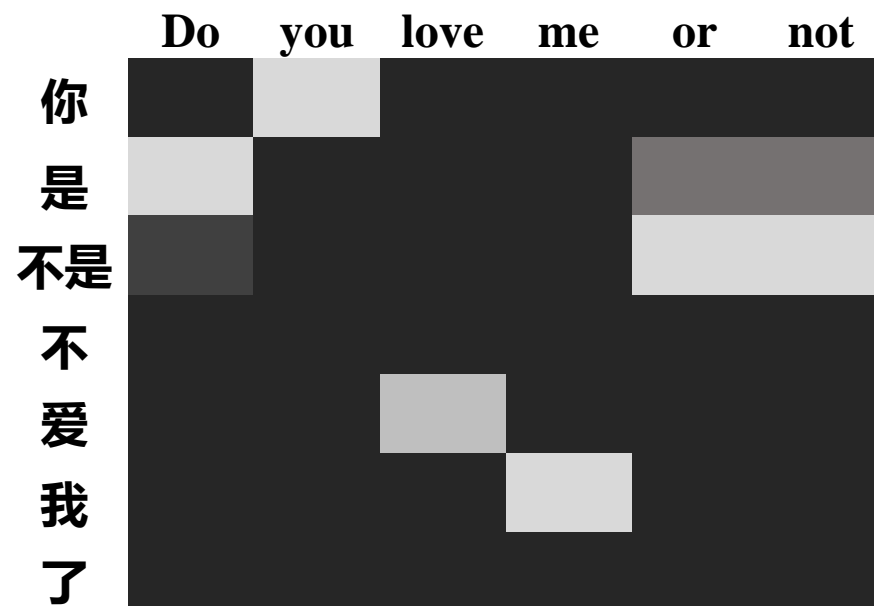
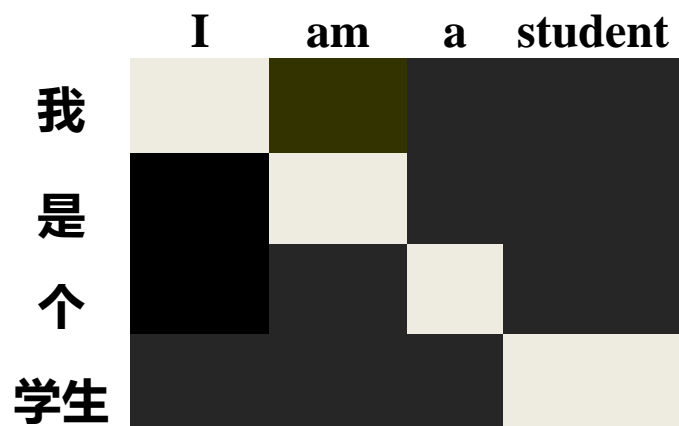
$$\beta_{tj} = q(s_{t-1}, h_j) = h_j^T W s_{t-1}$$

$$a_{tj} = \frac{\exp \beta_{tj}}{\sum_i \exp \beta_{ti}}$$

$$c_t = \sum_{j=1}^T a_{tj} h_j$$

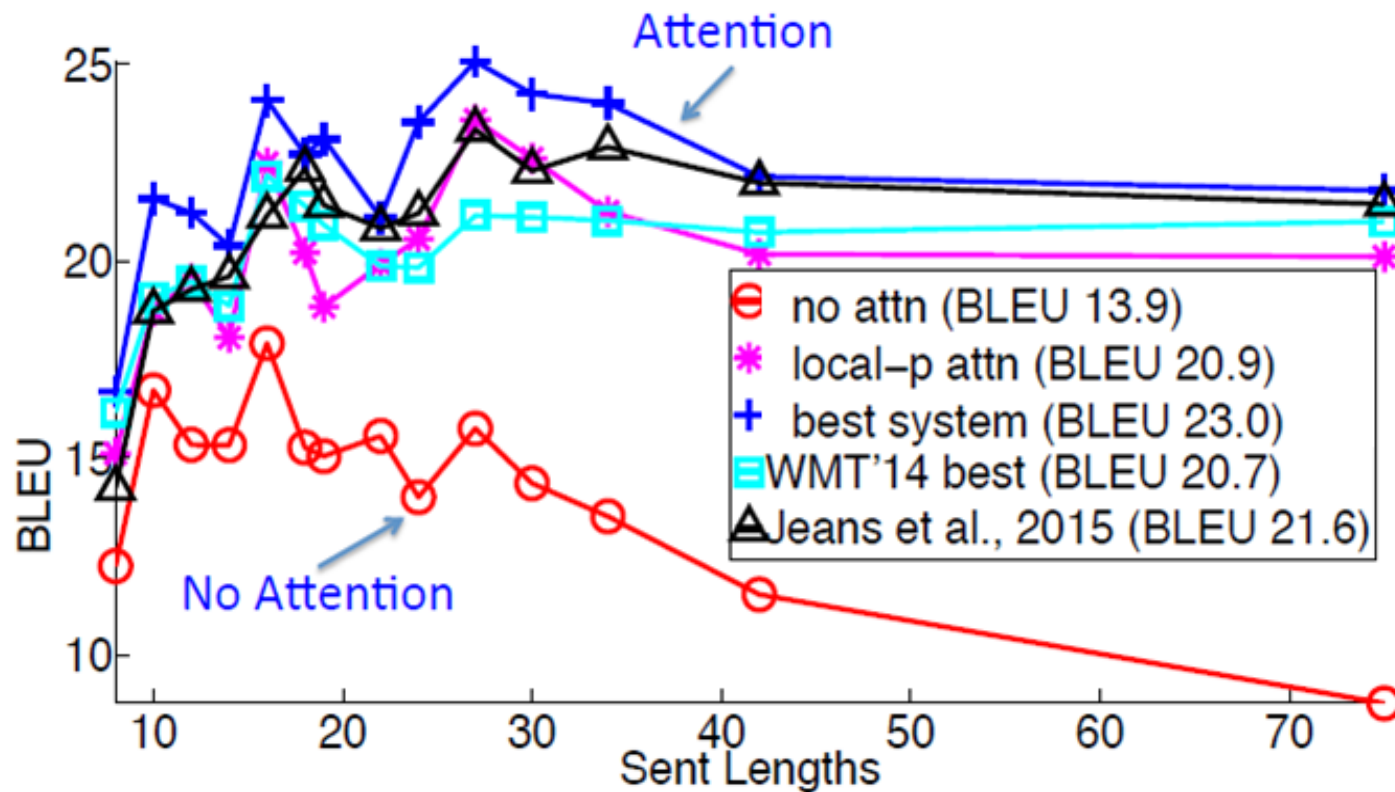
$$s_t = f(s_{t-1}, c_t, y_t)$$

注意力机制



- 颜色越白代表注意力越强

注意力机制



应用二：对话系统

- 用户通过对话交互，来获取所需要的信息或者服务
- 应用场景来分
 - 闲聊类：小冰、贤二机器僧
 - 咨询类：客服、各种轻咨询（公积金、签证等）
 - 任务类：订票、订餐、打车、订酒店等
 - 助理类：SIRI、微软小娜、Google Assistant、小度

对话任务与翻译的不同

- “你好笨”有六十多种不同回答
 - 不笨、你才笨、为什么笨、别这样说...
- 对于同一个问题，对话语料往往对应大量不同答案

我现在没工作	那就努力工作
猜我有没有在谈恋爱	缘分未到
我要睡了	那就睡吧
你在烦恼什么	不告诉你
那感情受挫呢	我也没办法
我只想有个真心的朋友	真心找找
苍老师照片	我也想看看
三国演义谁写的	去网上搜一下
有什么好看的喜剧	没什么好看的
地球为何是圆的	地球上的
世界上最高的山是什么	赵本山
云南在哪里	在云南
一加一等于几	等于二
一加二等于几	等于二
我明天去理发	那就去吧
我要娶个美国妞	那就去吧
打印机坏了	那就打电话吧
感冒了怎么办	去医院看看吧
轮胎坏了怎么办	去医院看看吧

多机制建模

- 人们在正常会话时，对同一个问题，可以有各种各样不同的回答风格、侧重点等，这便是人类对话固有的多样性
- 直接利用机器翻译模型做对话系统，忽视了对话本身的多样性，使得系统返回的语句，多是所谓的高频语句，例如“好的”、“可以”和“没事儿”

多机制建模

- 一种思路是用人工将训练语料中不同风格的问答对剥离开来，分开训练不同的模型。
- 这种方法有两个问题：
 - 在大数据中人工剥离成本很高。
 - 不同标注人员对语言风格的理解不同，很难做的一份数据一个标准。
- 可否采用无监督训练来得到不同的语言风格呢？

多机制建模

- 我们提出了一个多机制学习的方法，先判断回答一个问题要用哪个机制，再利用选出的机制生成答案。
- 整个过程是端到端、无监督训练的。
- 实验发现，用训练得到的不同机制来回答同一个问题，能得到不同语言风格的答案。

多机制学习:

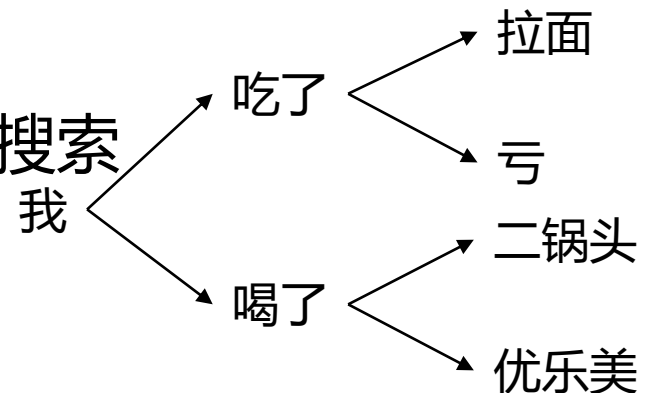
Cho et al.	4 Mechanisms
我现在没工作 那就努力工作 那就好好工作 那就努力的工作 那你就努力工作 那就要努力工作	我现在没工作 那就去找工作 那就好好学习 你想做什么 你在做什么 会有工作的
猫和狗不好 狗狗可是很可爱的 狗狗也很可爱 猫狗可是很可爱的 狗狗是很可爱的 狗狗可是很可爱	猫和狗不好 他们是很好的朋友 猫狗可是很可爱的 哪有， 狗狗很可爱 我觉得挺可爱的 我觉得狗狗很可爱

实际产品： Google Smart Reply

- 机器人的回答
 - 既要可控
 - “如何偷渡？”
 - “冰毒怎么制作？”
 - 等等，绝对不可以从开放语料里检索答案！
 - 统一回答“我不知道”，必要时报告警察蜀黍。
 - 又要流畅
 - 还要相关

控制回答内容

- 首先设计分类器，判断是否有必要回答某个问题
 - 敏感内容统一不回答
 - 不适合生成的内容不回答
- 对于回答内容，从预先设计回答集合中搜索答案
 - 答案库中答案是明确的
 - 建立答案前缀树，在前缀树上搜索



“年轻的” RNN模型

- RNN模型是时下较前沿的模型。
- 它的优化方法、目标函数和网络结构等等，学术界也在不断推陈出新，没有定论。
- 我们讲的研究成果，希望启发同学们创造性思维，希望各位用更独到的思维，**活学活用，设计更优美的网络结构。**

Q&A