

第7讲 知识图谱的构建

(实体识别与实体消歧)

冯晓骋

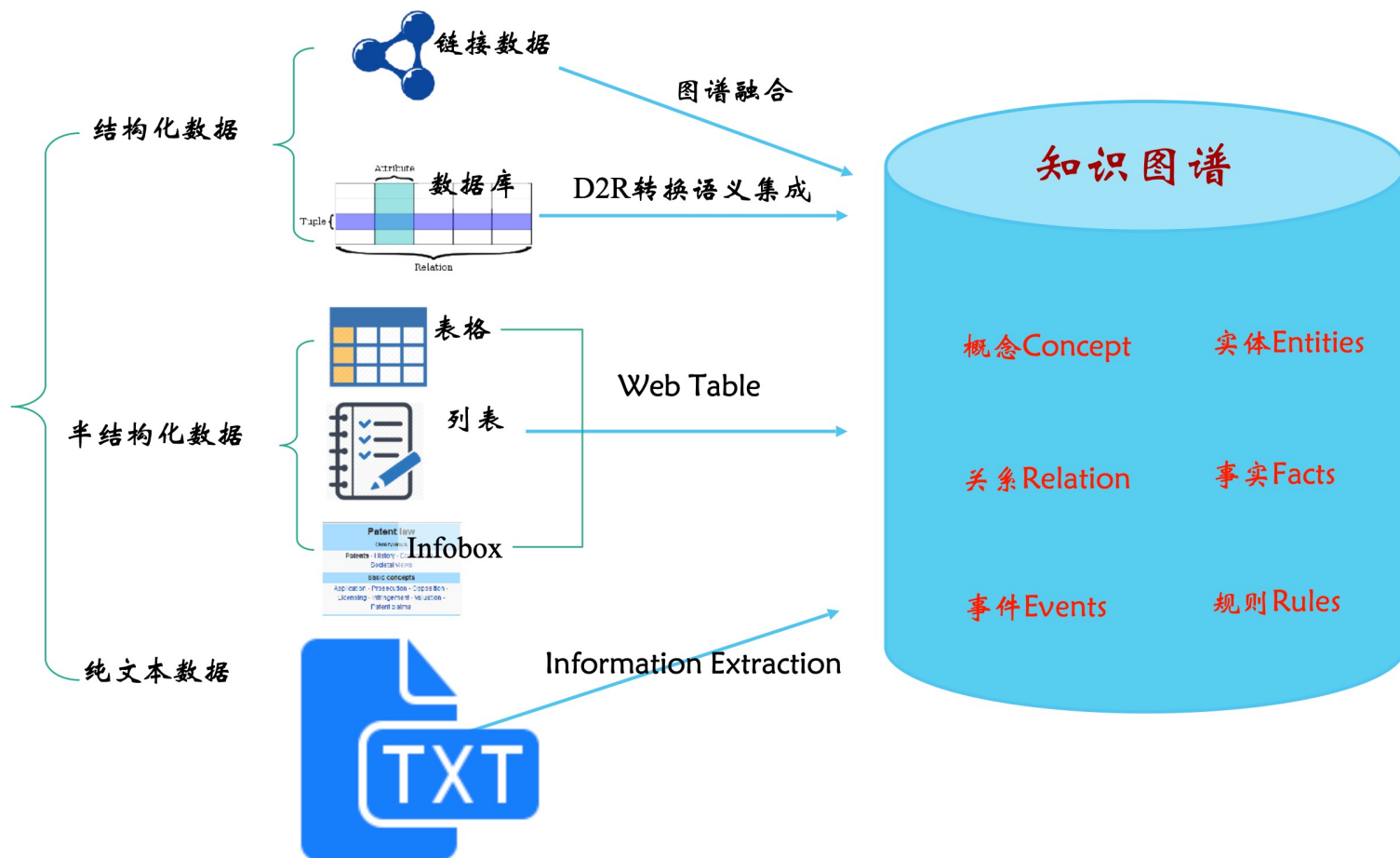
社会计算与信息检索研究中心

哈工大计算学部

- ▶ **知识图谱的构建流程**
- ▶ **实体识别**
 - ▶ 基本定义
 - ▶ 经典模型
 - ▶ 词汇增强
 - ▶ 实体嵌套
 - ▶ 生成式模型
- ▶ **实体消歧**
- ▶ **关系抽取**
- ▶ **事件抽取**
- ▶ **开放域知识抽取**
- ▶ **多模态知识抽取**

知识图谱的构建流程

- ▶ 从多种异构数据源中抽取实体和关系，并形成完整的大规模知识图谱
- ▶ 文本一般不作为知识图谱构建的初始来源，而多用来进行知识图谱补全



基于统计模型的方法

- ▶ 将命名实体识别看成**序列标注**任务，利用**HMM**、**CRF**等统计模型进行序列标注

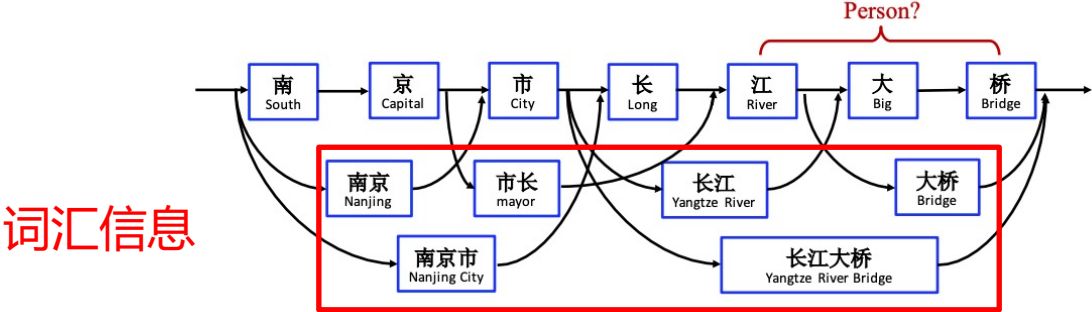
- ▶ **序列标注任务定义**

- ▶ 输入序列 $X = \{x_1, x_2, x_3, \dots, x_n\}$ ，输出标注序列 $Y = \{y_1, y_2, y_3, \dots, y_n\}$ ，其中

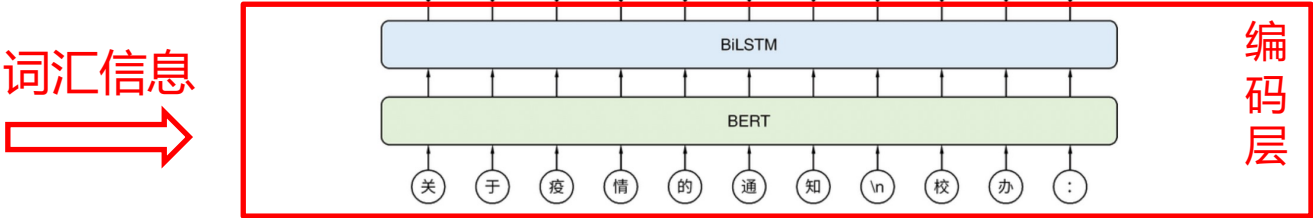
	$x_{1:T}$		$y_{1:T}$
▶ BIC	• Bill		B-PER
▶	• Gates		E-PER
▶	• is		O
▶	• the		O
▶ BIC	• CEO		O
▶	• and		O
▶	• co-founder		O
	• of		O
	• Microsoft		B-ORG
	• Inc.		E-ORG

中文命名实体识别——词汇增强

- 在基于字符的中文NER系统中引入词汇信息
 - 词汇边界信息能很好地辅助实体边界识别



- 在不同的编码器中引入词汇信息
 - LSTM
 - 图神经网络
 - Transformer

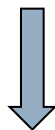


- ▶ 知识图谱的构建流程
- ▶ **实体识别**
 - ▶ 基本定义
 - ▶ 经典模型
 - ▶ 词汇增强
 - ▶ **实体嵌套**
 - ▶ 生成式模型
- ▶ 实体消歧
- ▶ 关系抽取
- ▶ 事件抽取
- ▶ 开放域知识抽取
- ▶ 多模态知识抽取

嵌套实体识别

- ▶ 命名实体的嵌套或重叠现象无处不在
- ▶ 嵌套命名识别时难以通过序列标注模式解决

...哈尔滨工业大学附属哈尔滨市第一医院位于...

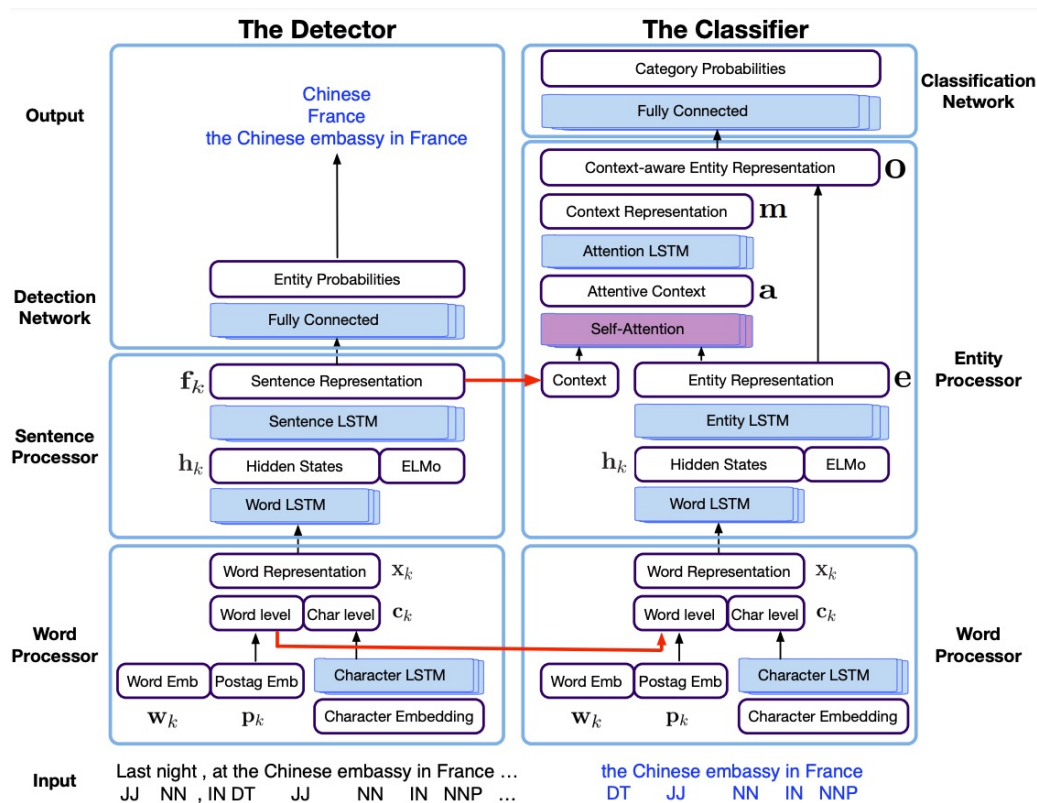


位置：哈尔滨

机构：哈尔滨工业大学、哈尔滨市第一医院、
哈尔滨工业大学附属哈尔滨市第一医院

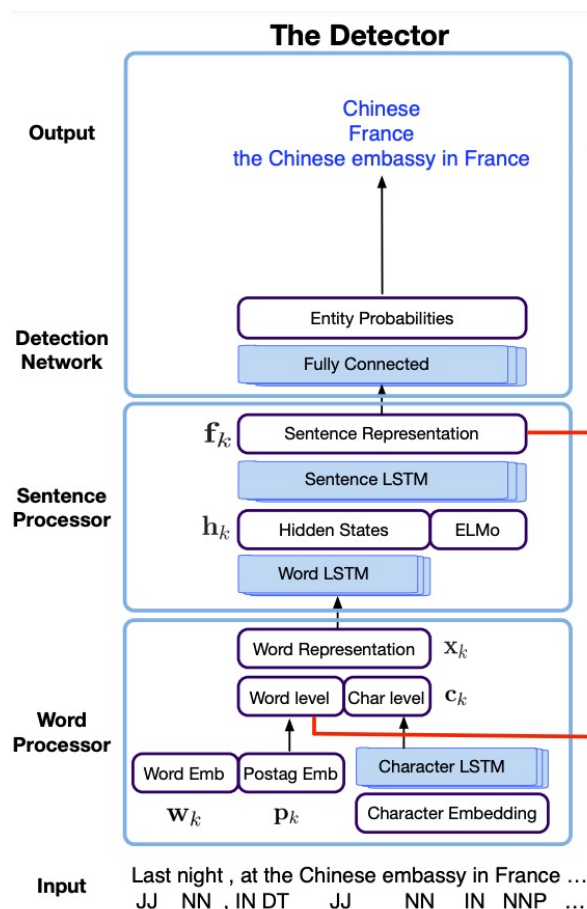
► Multi-Grained Named Entity Recognition(MGNER)

- 先检测所有可能的实体候选，再对所有的实体候选进行分类



[1] Xia C, Zhang C, Yang T, et al. Multi-grained named entity recognition[J]. arXiv preprint arXiv:1906.08449, 2019.

嵌套实体识别——MGNER



► Word Processor

- 通过拼接Glove词嵌入、词性embedding和字母级别单词表示作为每个词汇的表征向量

► Sentence Processor

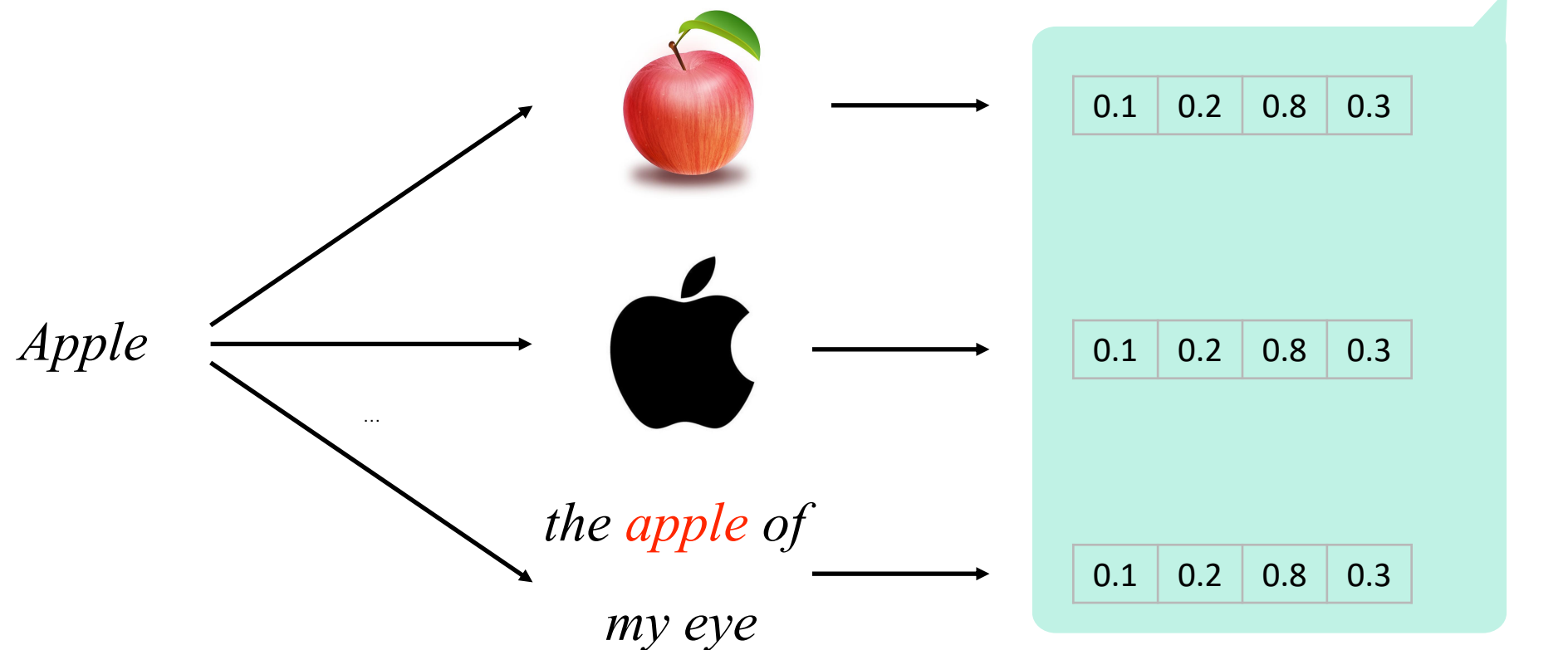
- 通过ELMo和Bi-LSTM获得词汇的上下文增强表征向量

动态词向量预训练模型

词向量——从静态到动态

▸ 静态词向量的问题

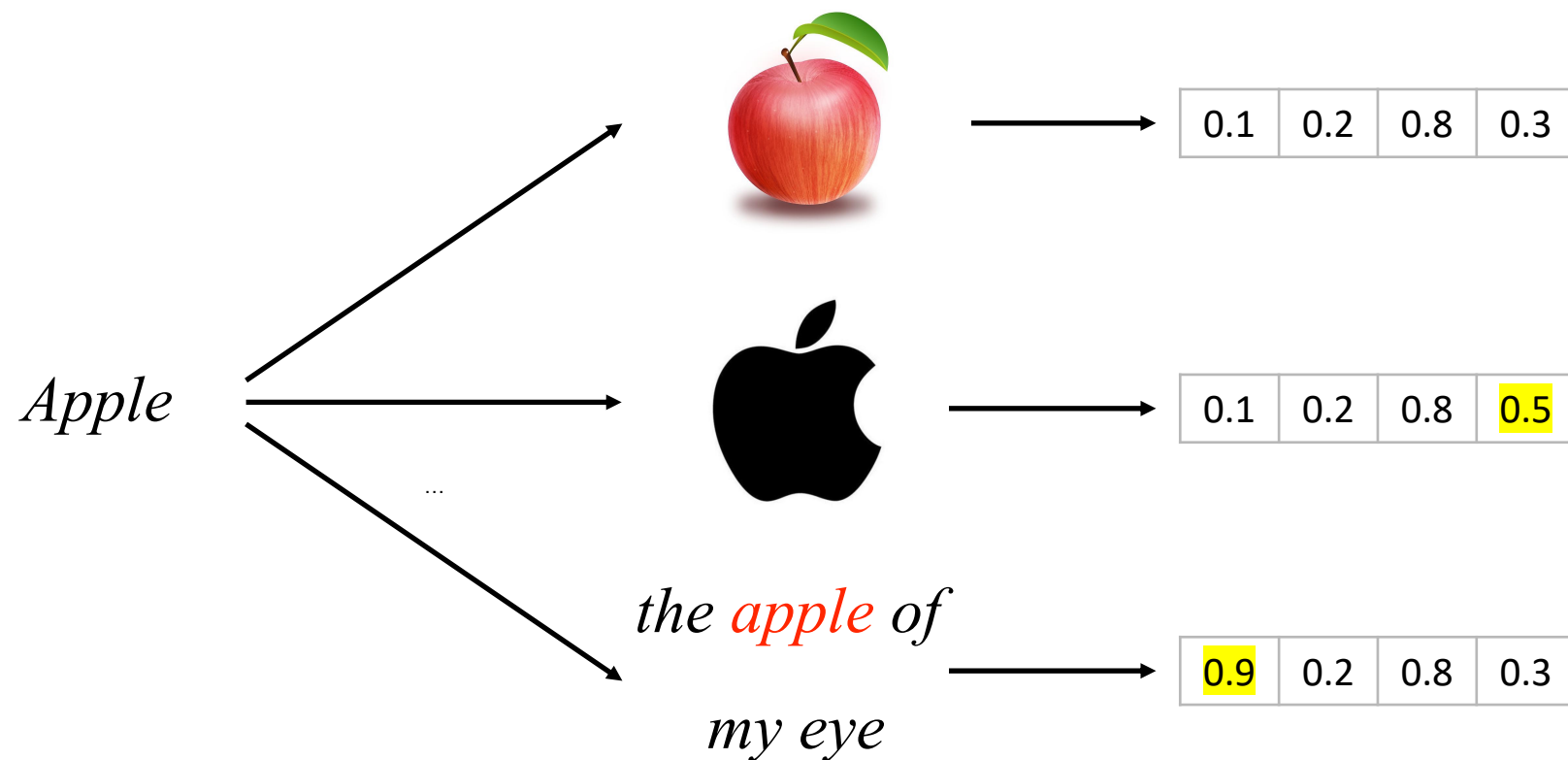
- 很多词包含多种语义信息，静态词向量无法解决“一词多义” 的表示问题



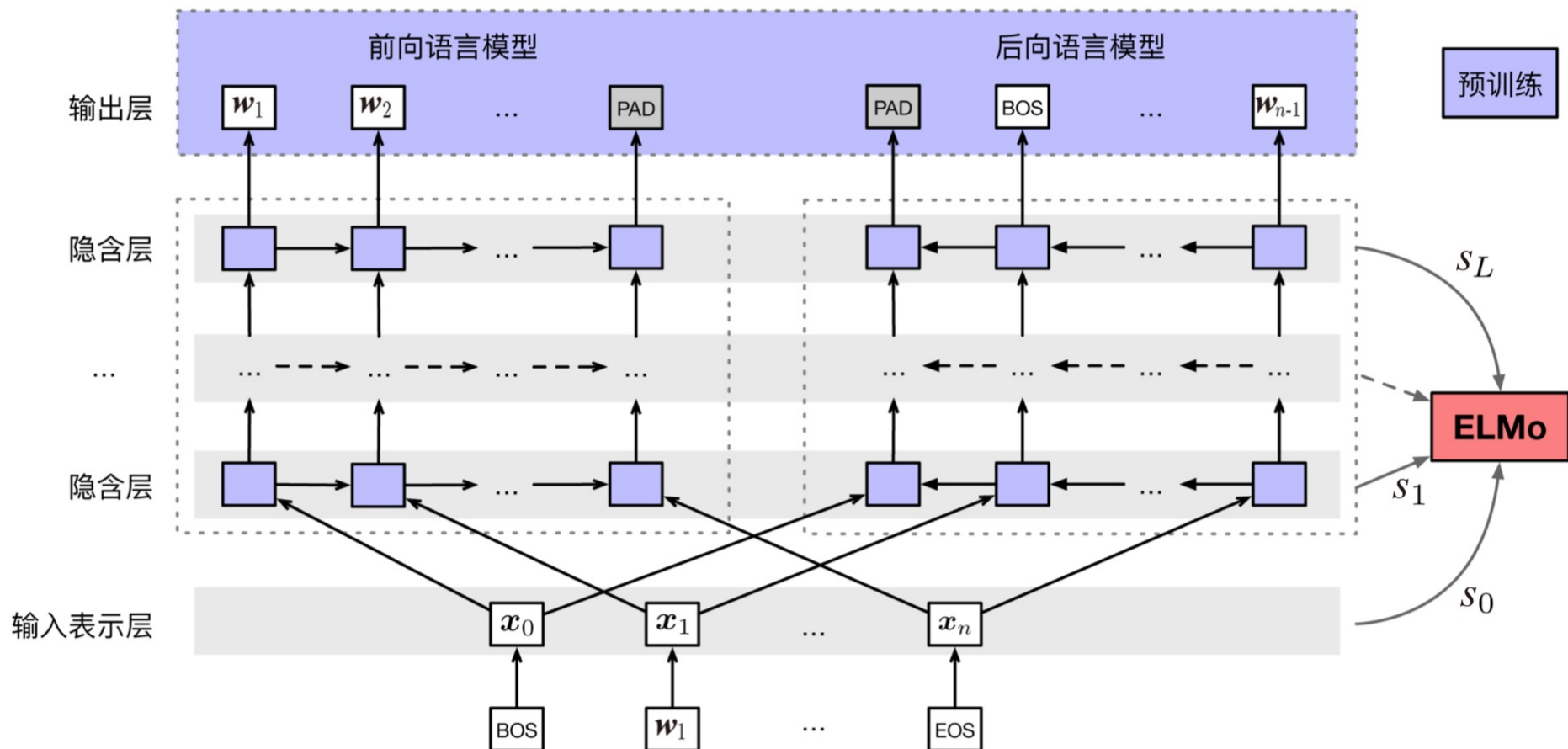
词向量——从静态到动态

▸ 静态词向量的问题

- 词向量应根据其所处的上下文的不同而发生改变



基于语言模型的动态词向量预训练



► 双向语言模型BiLM

► 输入表示层

$$\mathbf{v}_{c_i} = \mathbf{E}^{\text{char}} \mathbf{e}_{c_i}$$

$$\mathbf{x}_t = \mathbf{g} \odot \mathbf{f}_t + (\mathbf{1} - \mathbf{g}) \odot \text{ReLU}(\mathbf{W} \mathbf{f}_t + \mathbf{b})$$

$$\mathbf{g} = \sigma(\mathbf{W}^g \mathbf{f}_t + \mathbf{b}^g)$$

► 前向语言模型

$$P(w_1 w_2 \cdots w_n) = \prod_{t=1}^n P(w_t | \mathbf{x}_{1:t-1}; \overrightarrow{\boldsymbol{\theta}}^{\text{lstn}}, \boldsymbol{\theta}^{\text{out}})$$

► 后向语言模型

$$P(w_1 w_2 \cdots w_n) = \prod_{t=1}^n P(w_t | \mathbf{x}_{t+1:n}; \overleftarrow{\boldsymbol{\theta}}^{\text{lstn}}, \boldsymbol{\theta}^{\text{out}})$$

► ELMo词向量

- ELMo采取对不同层次的向量表示进行加权平均的机制，为不同的下游任务提供更多的组合自由度

$$\mathbb{R}_t = \{\mathbf{x}_t, \mathbf{h}_{t,j} | j = 1, \dots, L\}$$

$$\text{ELMo}_t = f(\mathbb{R}_t, \Psi) = \gamma^{\text{task}} \sum_{j=0}^L s_j^{\text{task}} \mathbf{h}_{t,j}$$

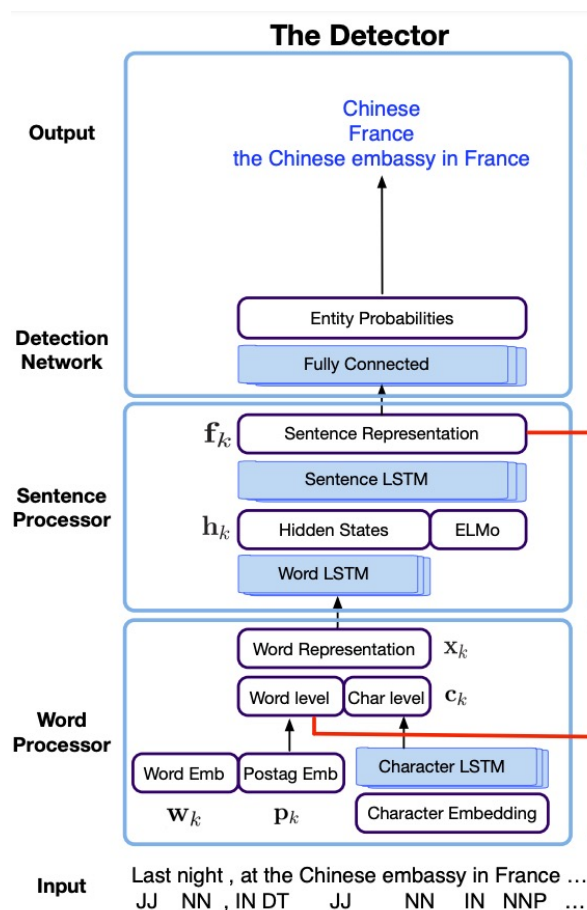
► ELMo特点

- **动态（上下文相关）**：词的ELMo向量表示由其当前上下文决定；
- **鲁棒（Robust）**：ELMo向量表示使用字符级输入，对于未登录词具有强鲁棒性；
- **层次**：ELMo词向量由深度预训练模型中各个层次的向量表示进行组合，为下游任务提供了较大的使用自由度。

- ▶ 上下文相关的词义相似性检索
 - ▶ ELMo相比GloVe（静态词向量）在词义消歧和近邻分析任务上都有比较好的表现

模型	词	近邻
GloVe	play	playing, game, games, played, players, plays, player, Play, football, multiplayer
ELMo	Chico Ruiz made a spectacular <u>play</u> on Alusik’s grounder . . .	Kieffer , the only junior in the group , was com-mended for his ability to hit in the clutch , as well as his all-round excellent <u>play</u>
	Olivia De Havilland signed to do a Broadway <u>play</u> for Garson they were actors who had been handed fat roles in a successful <u>play</u> , and had talent enough to fill the roles competently , with nice understatement

嵌套实体识别——MGNER



► Word Processor

- 通过拼接Glove词嵌入、词性embedding和字母级别单词表示作为每个词汇的表征向量

► Sentence Processor

- 通过ELMo和Bi-LSTM获得词汇的上下文增强表征向量

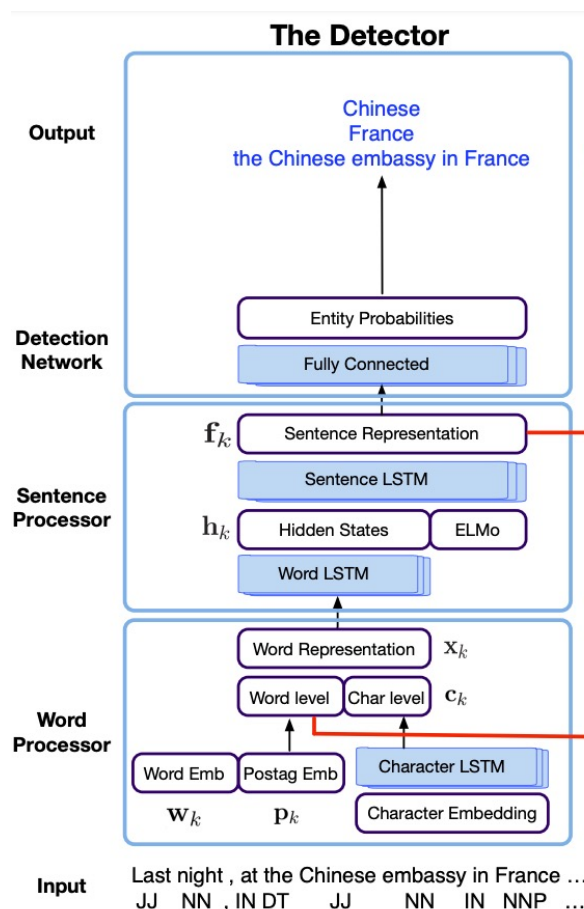
动态词向量预训练模型

嵌套实体识别——MGNER

► Detection Network

- 基于每个单词的词向量 f_k 生成R组(R=6) 分数，每组分数代表每个proposal是否为一个实体

$$s_k = \text{softmax}(f_k W_p + b_p)$$

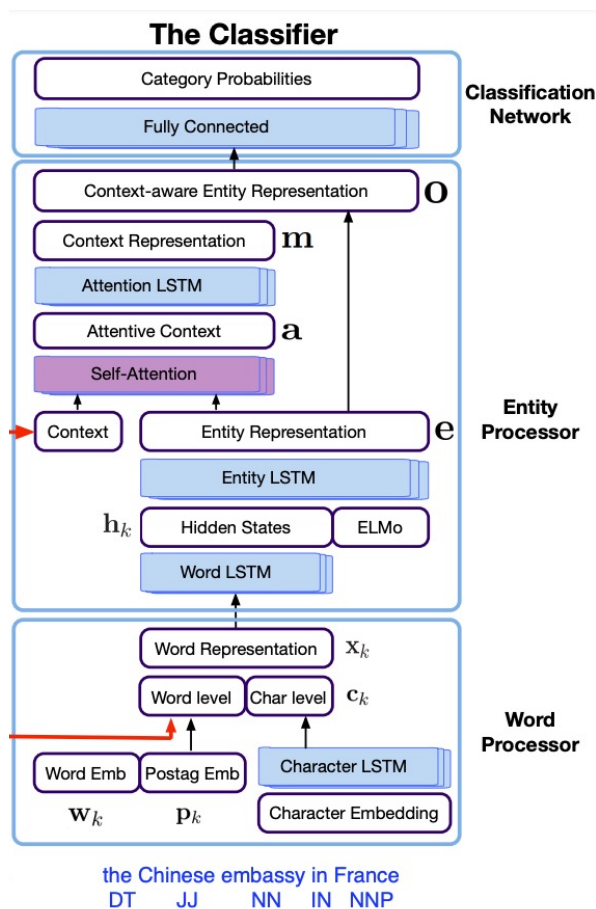


Proposal 1:	t1	t2	t3	t4	t5	t6
Proposal 2:	t1	t2	t3	t4	t5	t6
Proposal 3:	t1	t2	t3	t4	t5	t6
Proposal 4:	t1	t2	t3	t4	t5	t6
Proposal 5:	t1	t2	t3	t4	t5	t6
Proposal 6:	t1	t2	t3	t4	t5	t6

- 对所有词汇位置和所有proposal的组合进行训练

$$L_p = - \sum_{k=1}^K \sum_{r=1}^R y_k^r \log s_k^r$$

嵌套实体识别——MGNER



► The Classifier

- 对所有由Detector网络生成的实体候选进行分类

► Word Processor

- 和Detector使用参数共享的word embedding和pos embedding，非参数共享的character embedding

► Entity Processor

- 通过注意力机制汇聚Detector中实体周围的上下文信息

- ▶ 知识图谱的构建流程
- ▶ **实体识别**
 - ▶ 基本定义
 - ▶ 经典模型
 - ▶ 词汇增强
 - ▶ 实体嵌套
 - ▶ **统一模型**
- ▶ 实体消歧
- ▶ 关系抽取
- ▶ 事件抽取
- ▶ 开放域知识抽取
- ▶ 多模态知识抽取

多种不同类型的命名实体识别

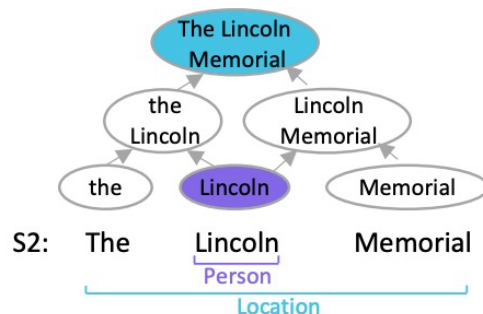
- 平滑命名实体识别、嵌套命名实体识别、不连续命名实体识别

平滑命名实体识别

S1: Barack Obama was born in the US
B-Per I-Per O O O O B-Loc
Person Location

(a) Sequence labelling for flat NER

嵌套命名实体识别



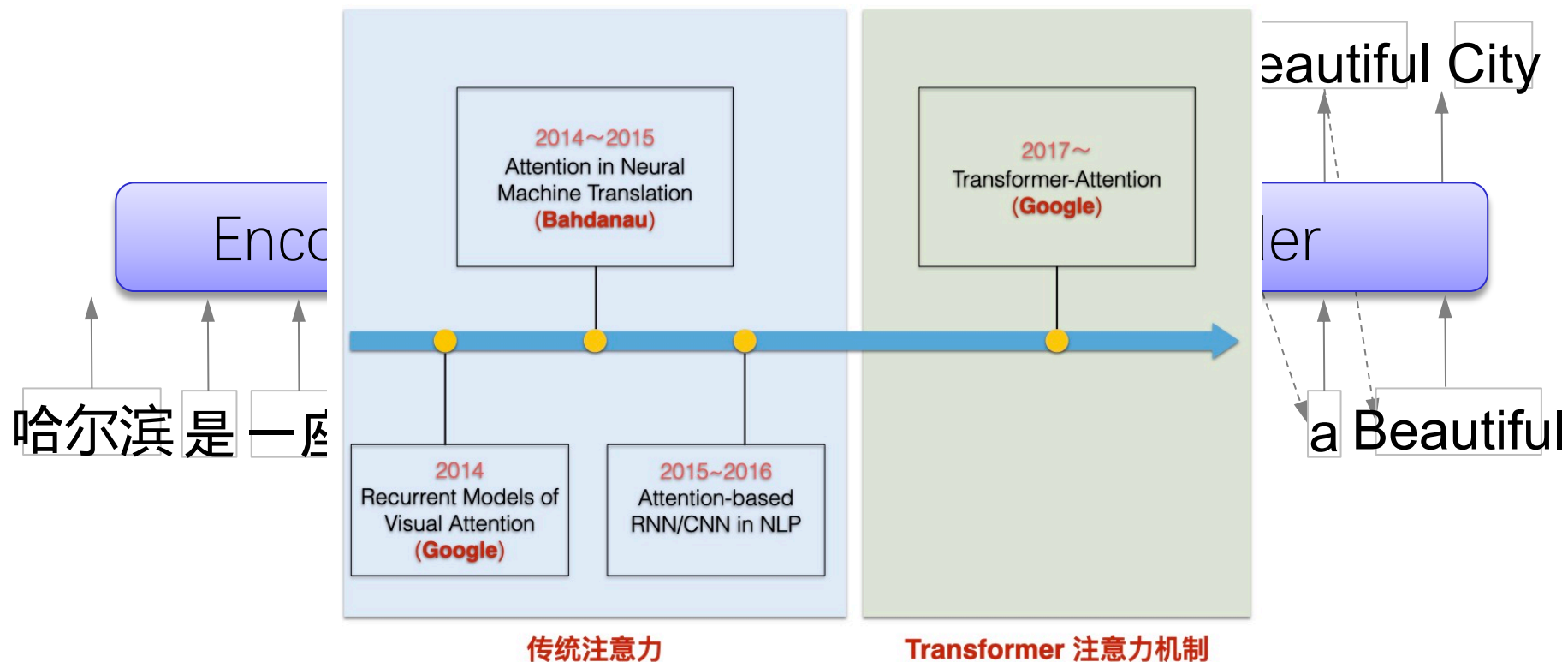
(b) Span-based classification for nested NER

不连续命名实体识别

Actions: OUT OUT SHIFT SHIFT LEFT-REDUCE COMPLETE ...
S3: have much muscle pain and fatigue
Disorder Disorder

(c) Transition-based method for discontinuous NER

Attention is all your need

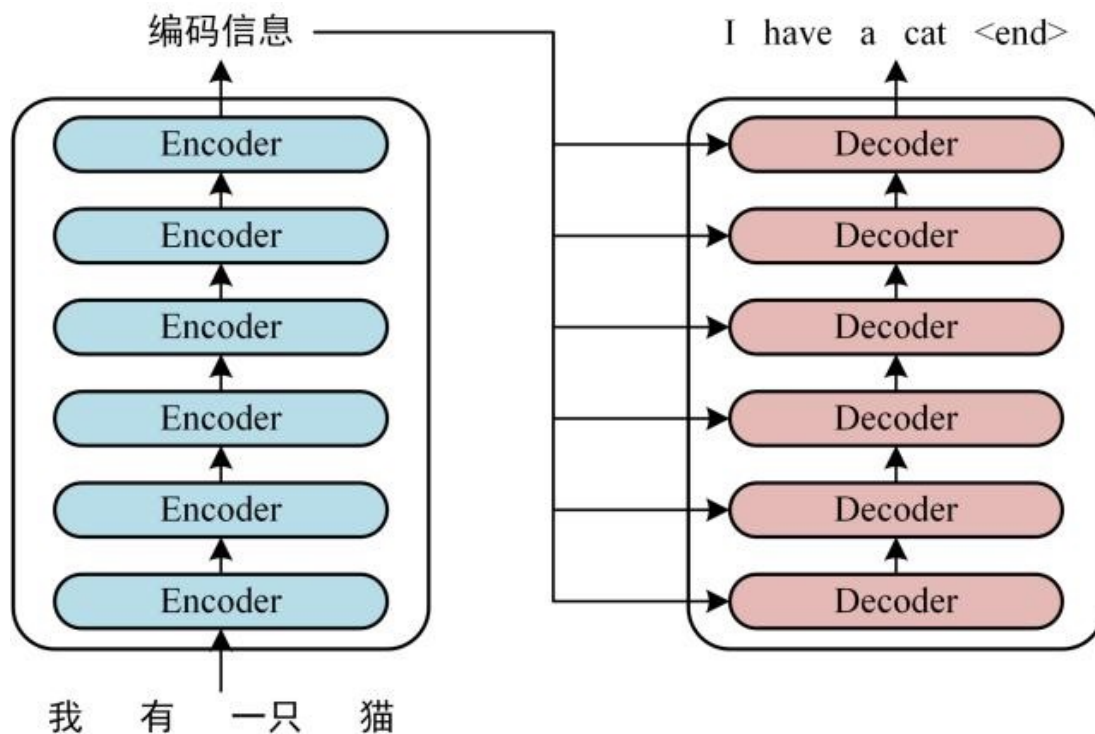


注意力机制 ——Transformer

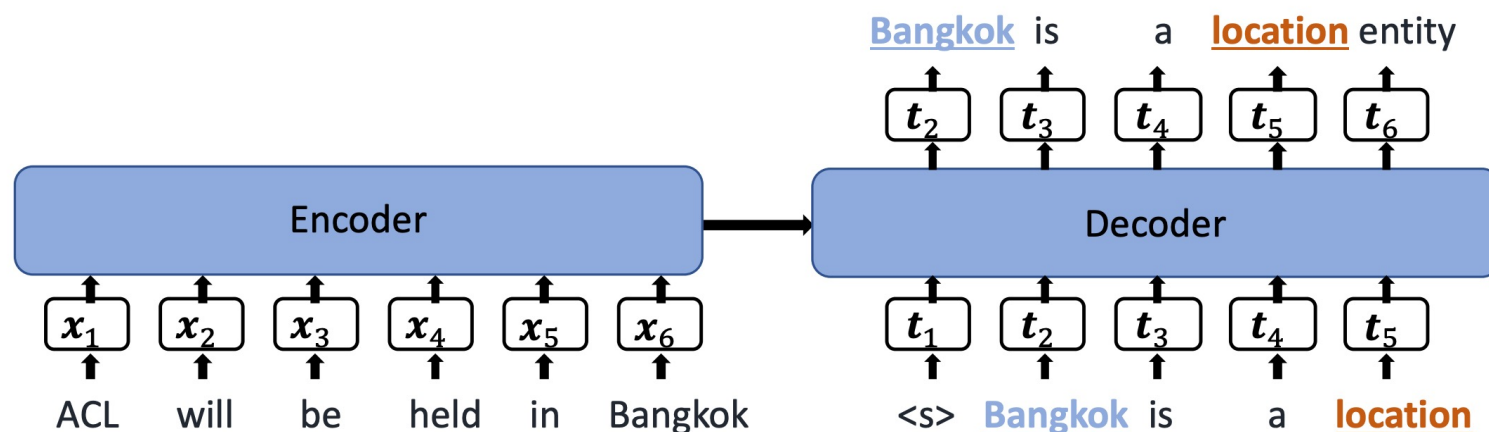
- ▶ Transformer

- ▶ 编码器 (Encoder) + 解码器 (Decoder)

- ▶ Transformer 用于中英文翻译的整体结构：



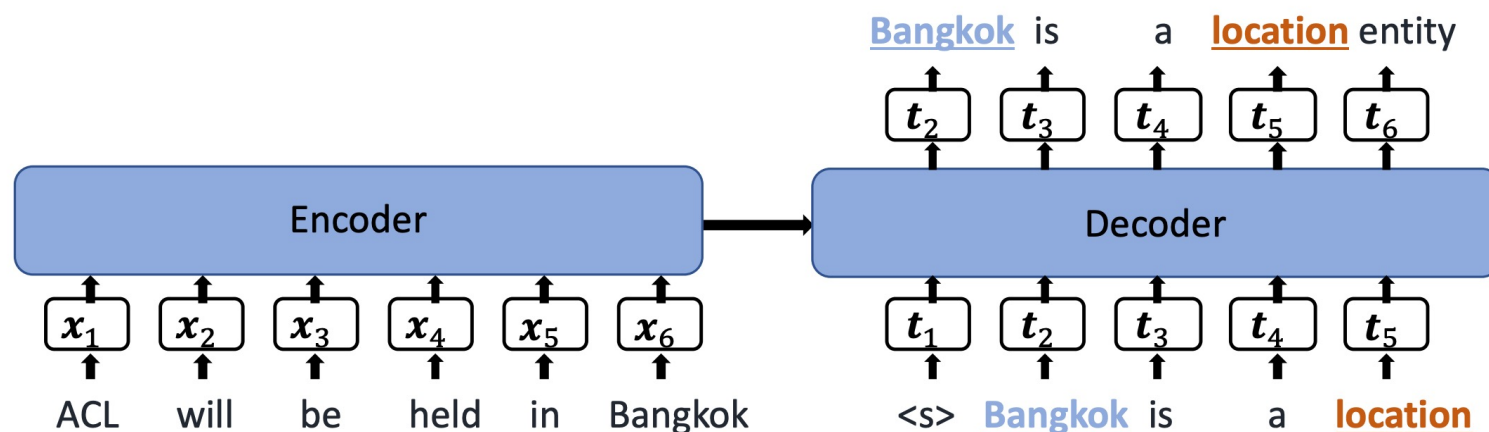
基于模版的生成式命名实体识别



- ▶ 根据模板构建正负训练样例
 - ▶ $\langle \text{candidate_span}, \text{entity type} \rangle \longrightarrow \langle \text{candidate_span} \rangle \text{ is a } \langle \text{entity_type} \rangle \text{ entity}$
 - ▶ $\langle \text{candidate_span}, \text{none} \rangle \longrightarrow \langle \text{candidate_span} \rangle \text{ is not a named entity}$
- ▶ 例如:
 - ▶ 正样例 : Bangkok is a location entity.
 - ▶ 负样例 : Will be is not a named entity. // Held in is not a named entity.

[1] Cui L, Wu Y, Liu J, et al. Template-based named entity recognition using BART[J]. arXiv preprint arXiv:2106.01760, 2021.

基于模版的生成式命名实体识别



- **推理过程**：给定输入句子，将给定模板下句子的生成概率作为分数

Bangkok is a **person** entity. $P = 0.01$

Bangkok is a **location** entity. $P = 0.5$ ✓

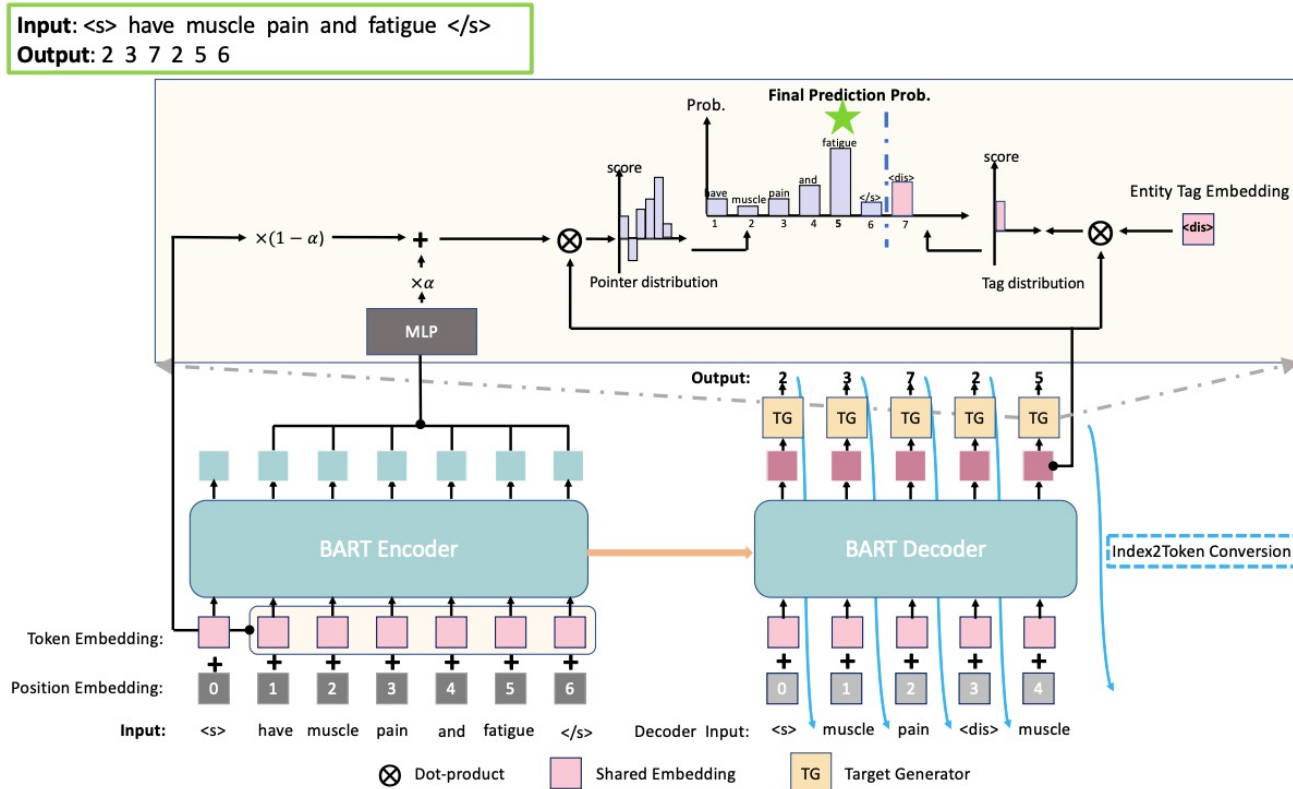
Bangkok is a **organization** entity. $P = 0.1$

Bangkok is a **miscellaneous** entity. $P = 0.2$

Bangkok is not a named entity. $P = 0.1$

Q: 这个模型有什么优势和劣势?

基于拷贝机制的生成式命名实体识别



- ▶ 使用拷贝 (copy) 机制生成实体以及实体类别
 - ▶ 解码时复制输入序列中的字符作为输出
- ▶ 使用预训练生成模型BART

[1] Yan H, Gui T, Dai J, et al. A unified generative framework for various NER subtasks[J]. arXiv preprint arXiv:2106.01223, 2021.

基于拷贝机制的生成式命名实体识别——Pointer Network

► 传统的注意力计算

$$\begin{aligned}u_j^i &= v^T \tanh(W_1 e_j + W_2 d_i) & j \in (1, \dots, n) \\a_j^i &= \text{softmax}(u_j^i) & j \in (1, \dots, n) \\d_i' &= \sum_{j=1}^n a_j^i e_j\end{aligned}$$

e_j : encoder的第j个词汇的隐层状态

d_i : decoder在解码生成第i个词汇时的隐层状态

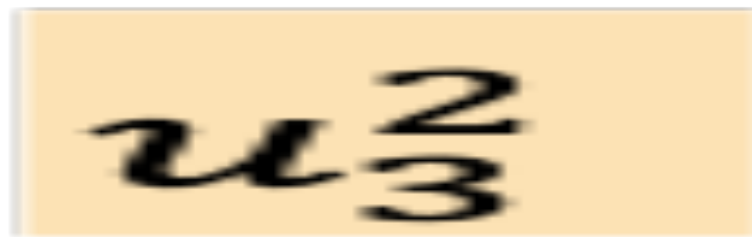
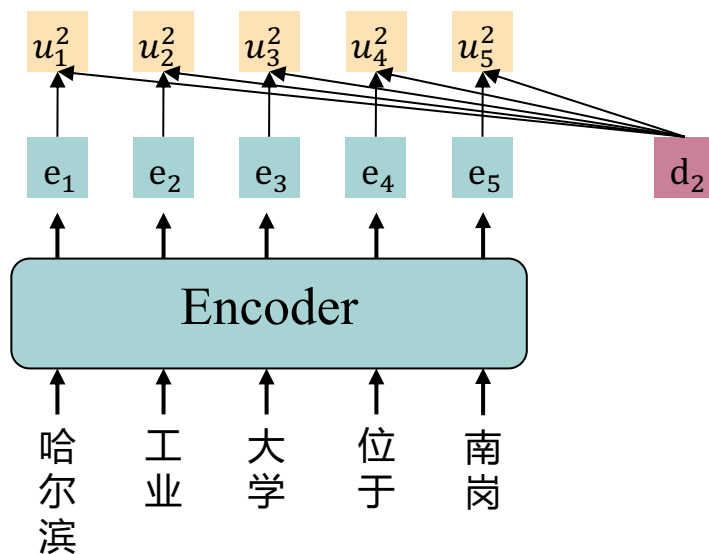
生成式命名实体识别——Pointer Network

► 传统的注意力计算

$$u_j^i = v^T \tanh(W_1 e_j + W_2 d_i) \quad j \in (1, \dots, n)$$

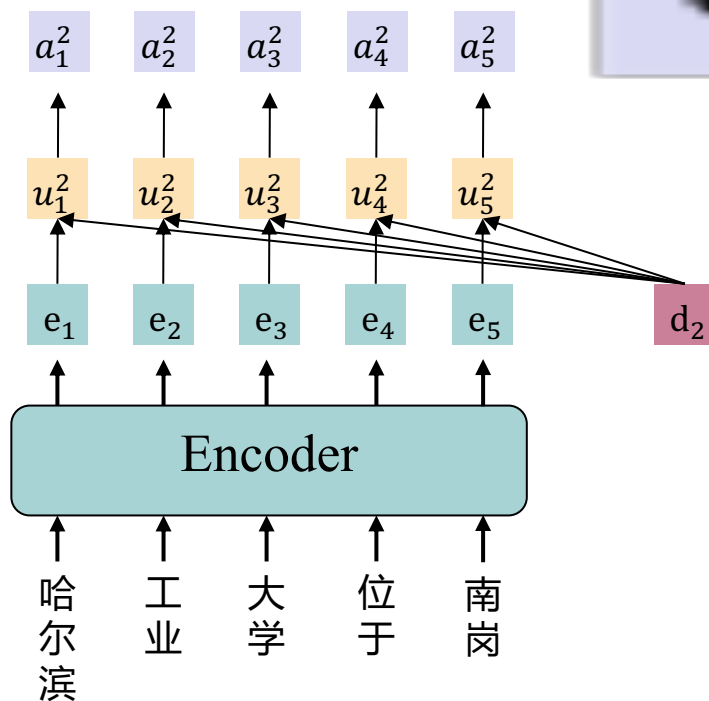
$$a_j^i = \text{softmax}(u_j^i) \quad j \in (1, \dots, n)$$

$$d'_i = \sum_{j=1}^n a_j^i e_j$$



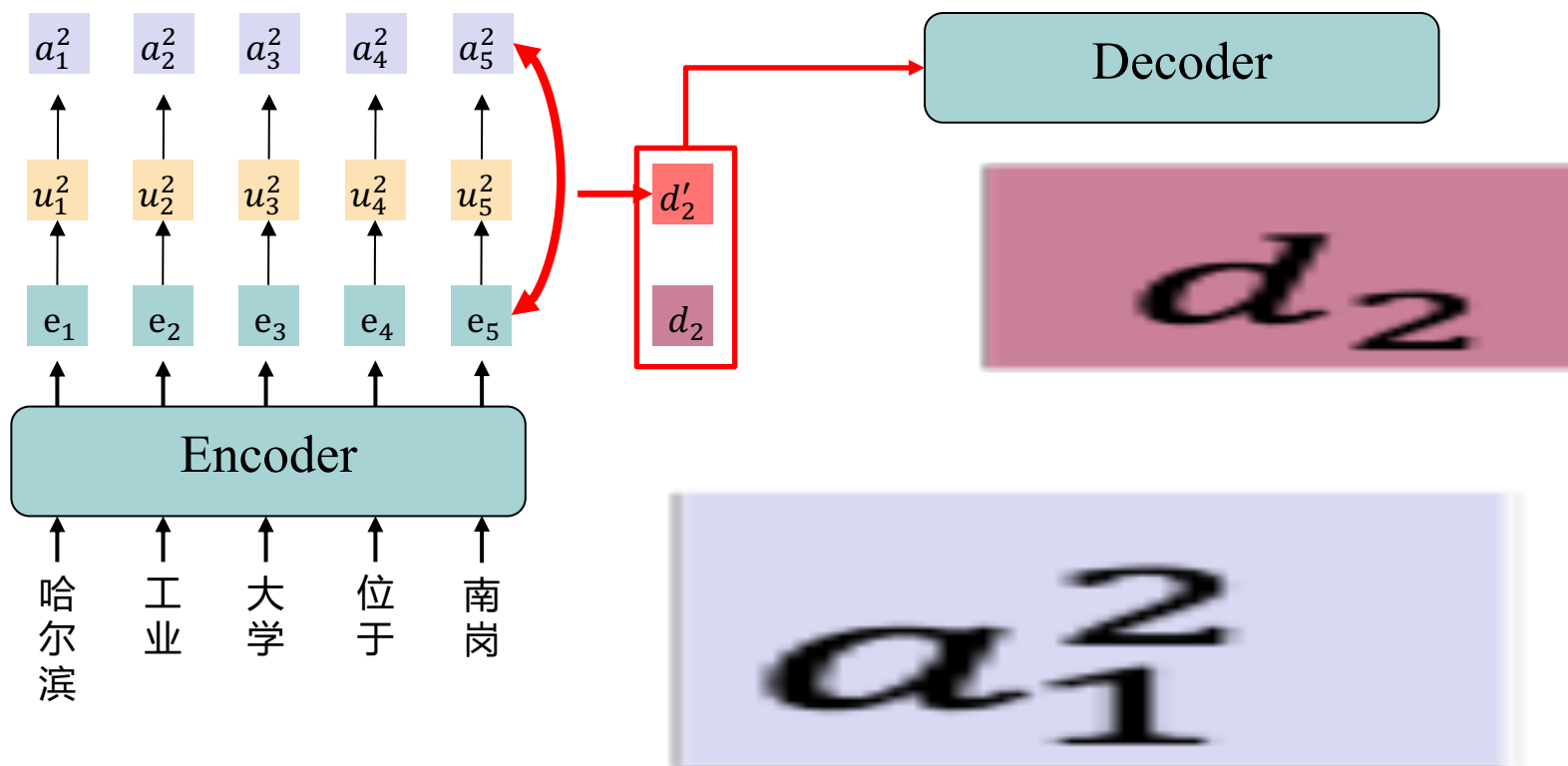
生成式命名实体识别——Pointer Network

► 传统的注意力计算



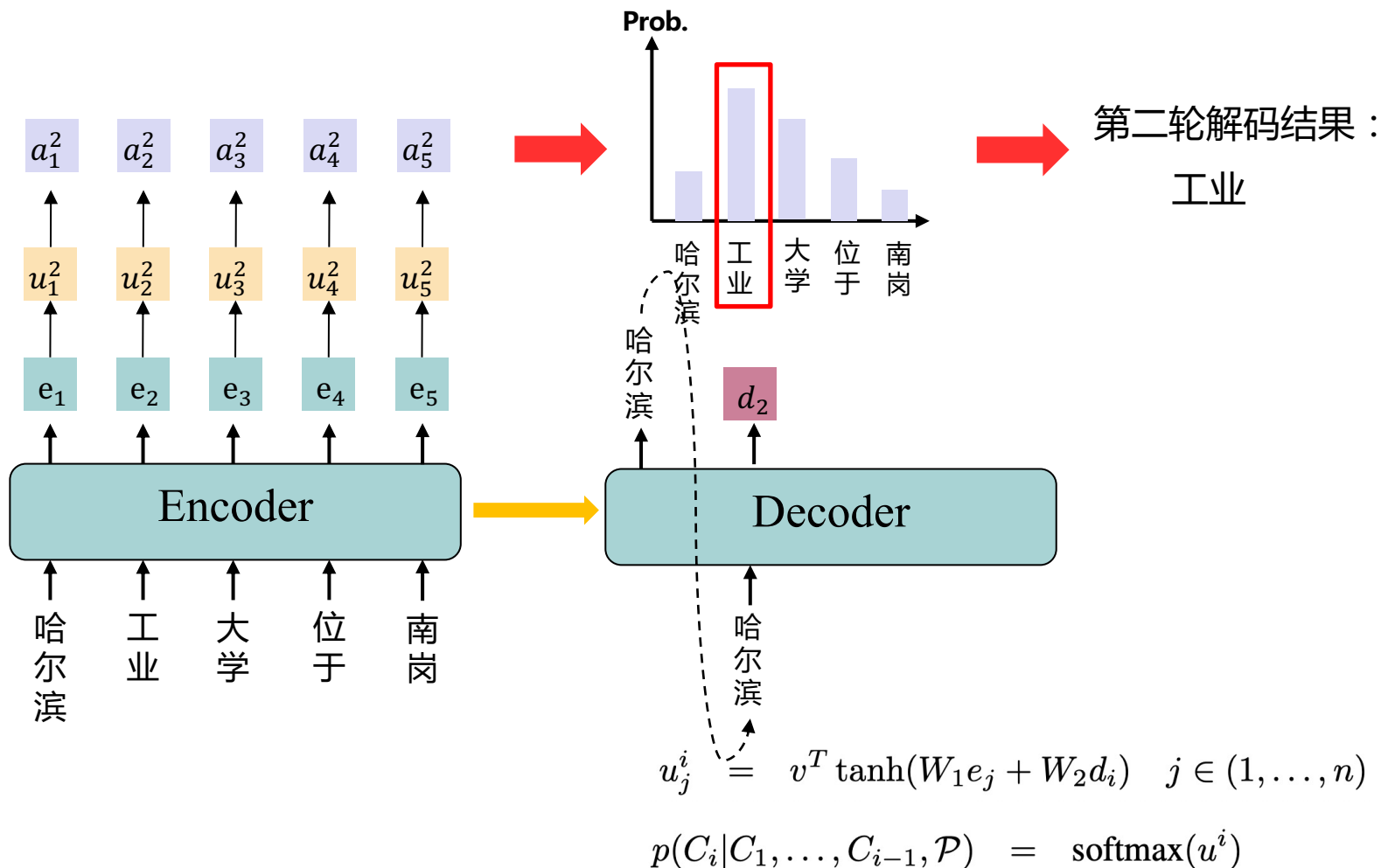
生成式命名实体识别——Pointer Network

► 传统的注意力计算



基于拷贝机制的生成式命名实体识别——Pointer Network

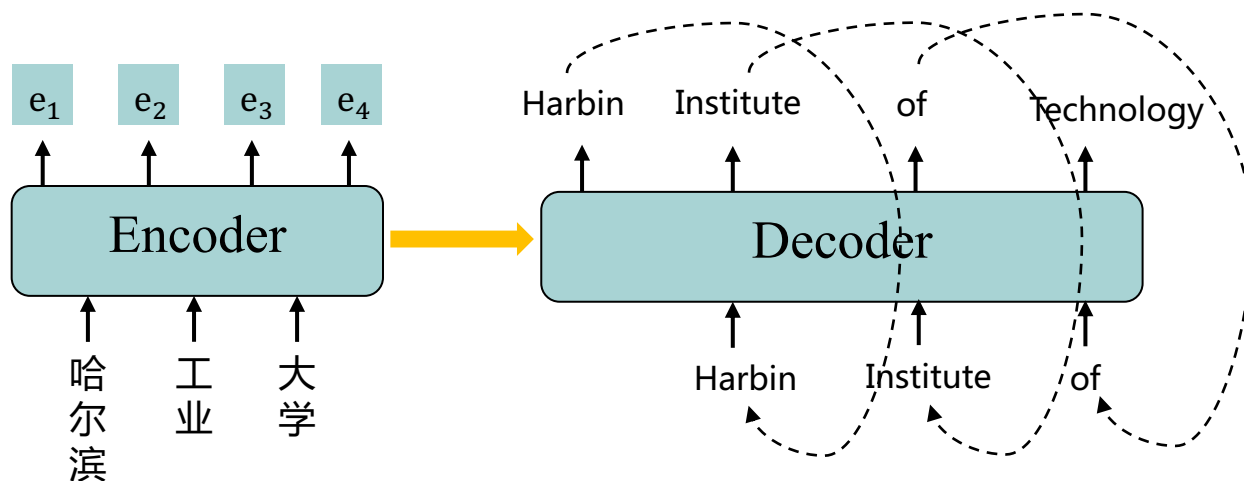
- Pointer Network: 使用注意力机制中计算出的权值作为生成



生成式命名实体识别—Beam Search(集束搜索)

- 给定序列输入，解码时如何得到最优序列？

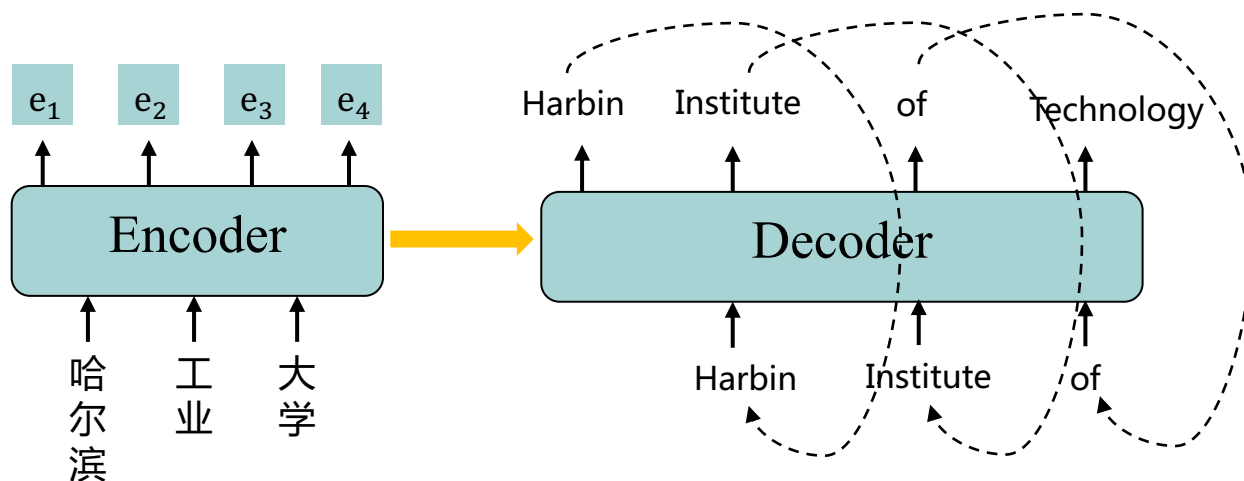
$$\operatorname{argmax}_Y p(Y|X) = \prod_{t=1}^T p(y_t|\{y_1, \dots, y_{t-1}\}, X)$$



假设给定输入序列“哈尔滨、工业、大学”，输出词表仅有“Harbin、Institute、of、Technology”四个词以及结束符“<eos>”，如何得到概率最大的解码序列？

生成式命名实体识别—Beam Search(集束搜索)

- ▶ 给定序列输入，解码时如何得到最优序列？
 - ▶ 穷举搜索: 计算所有输出序列的概率，取全局概率最大的序列作为解码结果

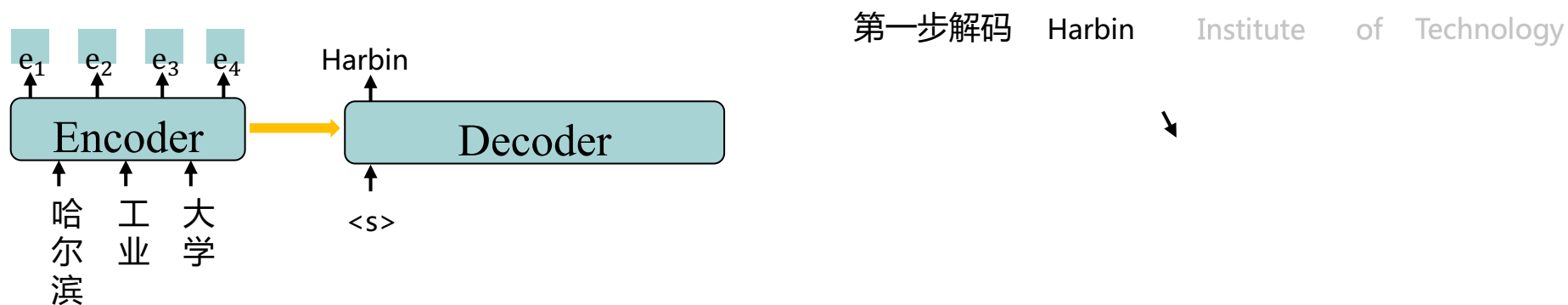


- ▶ 例：计算输出序列为" *Harbin Institute of Techonology*" 的概率
 $p(\text{Harbin Institute of Techonology} | \text{哈尔滨工业大学})$

生成式命名实体识别—Beam Search(集束搜索)

▸ 穷举搜索

- 例：计算输出序列为" *Harbin Institute of Techonology*" 的概率

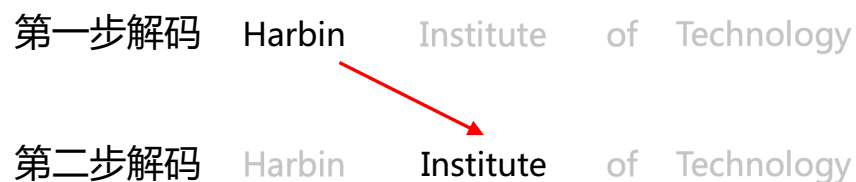
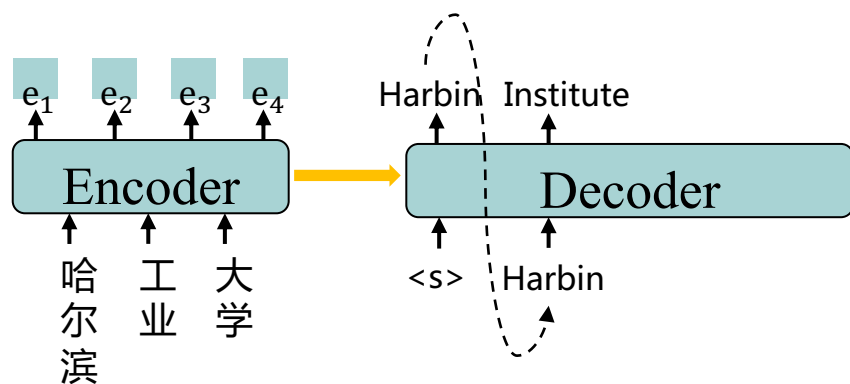


$$\begin{aligned} & p(\text{Harbin Institute of Techonology} | \text{哈尔滨工业大学}) \\ &= p(\text{Harbin} | X) * p(\text{Institute} | \text{Harbin}, X) * p(\text{of} | \text{Harbin Institute}, X) \\ & \quad * p(\text{Technology} | \text{Harbin Institute of Technology}, X) \end{aligned}$$

生成式命名实体识别—Beam Search(集束搜索)

► 穷举搜索

- 例：计算输出序列为" *Harbin Institute of Techonology*" 的概率

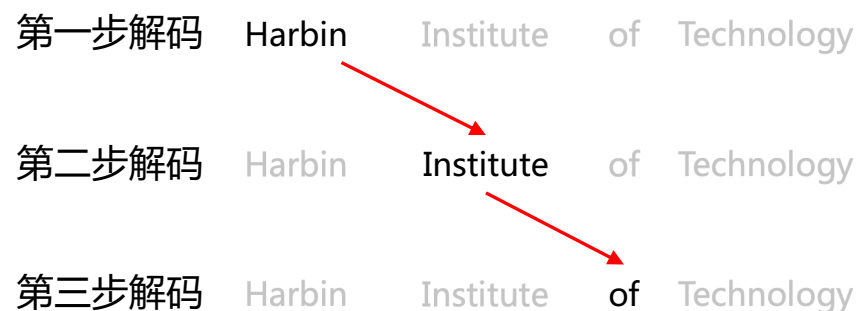
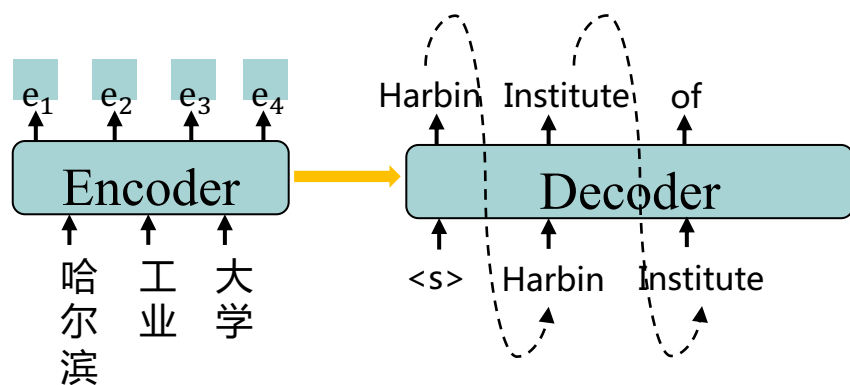


$$\begin{aligned} & p(\text{Harbin Institute of Techonology} | \text{哈尔滨工业大学}) \\ &= p(\text{Harbin} | X) * p(\text{Institute} | \text{Harbin}, X) * p(\text{of} | \text{Harbin Institute}, X) \\ & \quad * p(\text{Technology} | \text{Harbin Institute of Technology}, X) \end{aligned}$$

生成式命名实体识别—Beam Search(集束搜索)

穷举搜索

- 例：计算输出序列为" *Harbin Institute of Techonology*" 的概率

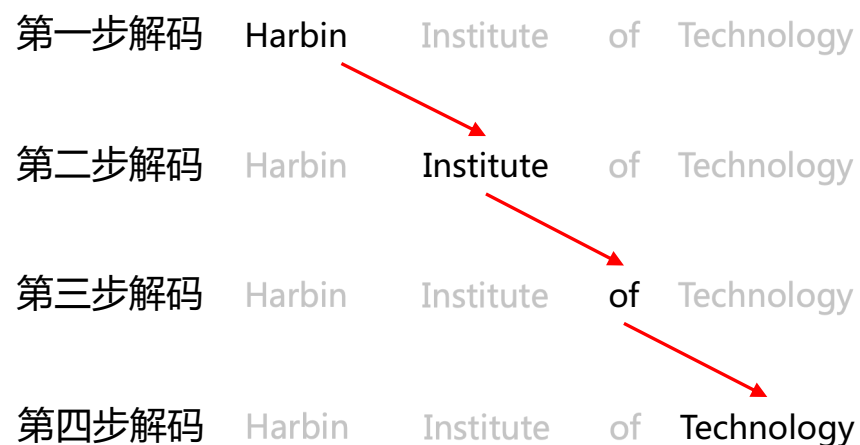
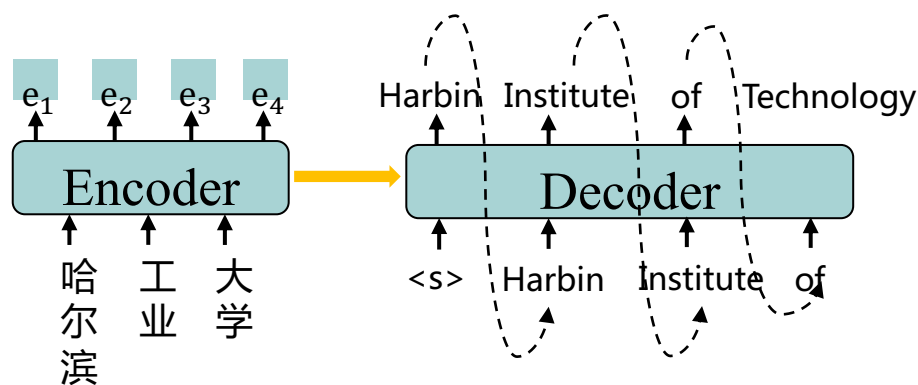


$$\begin{aligned} & p(\text{Harbin Institute of Techonology} | \text{哈尔滨工业大学}) \\ &= p(\text{Harbin} | X) * p(\text{Institute} | \text{Harbin}, X) * p(\text{of} | \text{Harbin Institute}, X) \\ & \quad * p(\text{Technology} | \text{Harbin Institute of Technology}, X) \end{aligned}$$

生成式命名实体识别—Beam Search(集束搜索)

► 穷举搜索

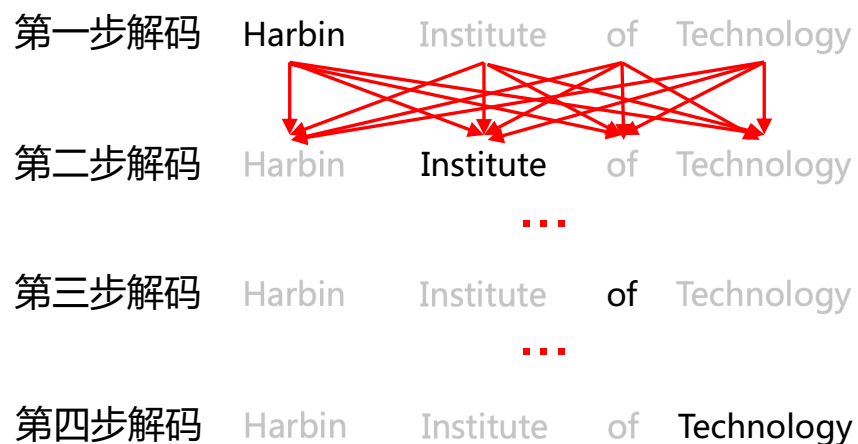
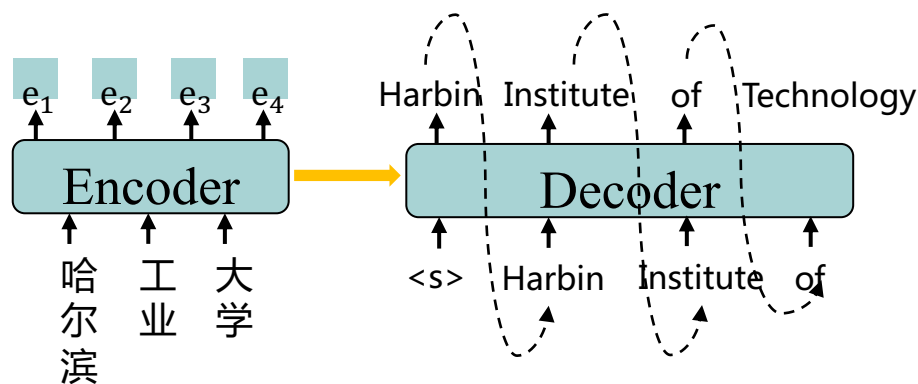
- 例：计算输出序列为" *Harbin Institute of Techonology*" 的概率



$$\begin{aligned} & p(\text{Harbin Institute of Techonology} | \text{哈尔滨工业大学}) \\ &= p(\text{Harbin} | X) * p(\text{Institute} | \text{Harbin}, X) * p(\text{of} | \text{Harbin Institute}, X) \\ & \quad * p(\text{Technology} | \text{Harbin Institute of Technology}, X) \end{aligned}$$

生成式命名实体识别—Beam Search(集束搜索)

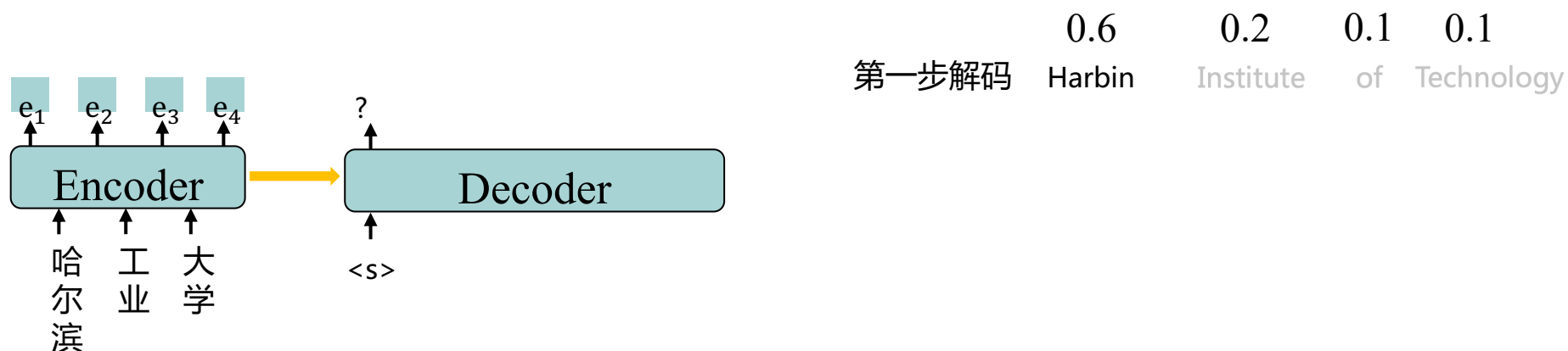
- ▶ 给定序列输入，解码时如何得到最优序列？
 - ▶ 穷举搜索：计算所有输出序列的概率，取全局概率最大的序列作为解码结果



共 $4*4*4*4=256$ 种可能的解码结果

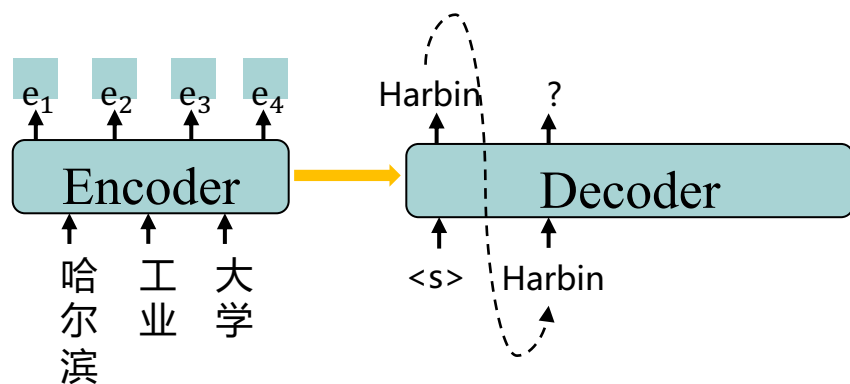
生成式命名实体识别—Beam Search(集束搜索)

- ▶ 给定序列输入，解码时如何得到最优序列？
 - ▶ 贪心搜索：每一步解码时选择局部概率最大的词汇作为输出



生成式命名实体识别—Beam Search(集束搜索)

- ▶ 给定序列输入，解码时如何得到最优序列？
 - ▶ 贪心搜索：每一步解码时选择局部概率最大的词汇作为输出

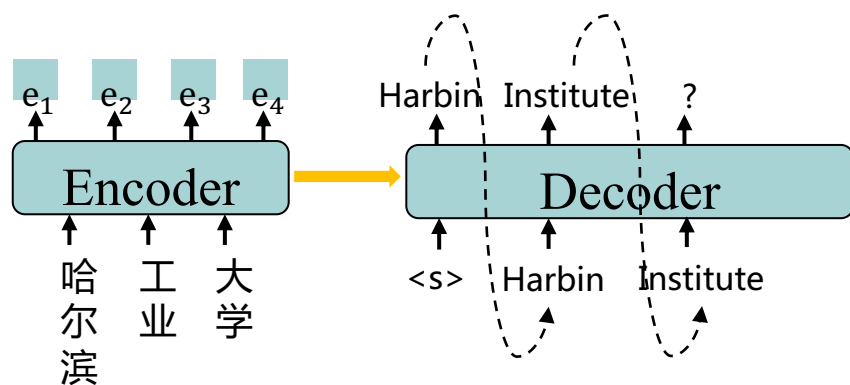


第一步解码	0.6 Harbin	0.2 Institute	0.1 of	0.1 Technology
第二步解码	0.2 Harbin	0.3 Institute	0.4 of	0.1 Technology

A red arrow points from the "Harbin" cell in the first row to the "of" cell in the second row, indicating the selection of the most probable word at each step.

生成式命名实体识别—Beam Search(集束搜索)

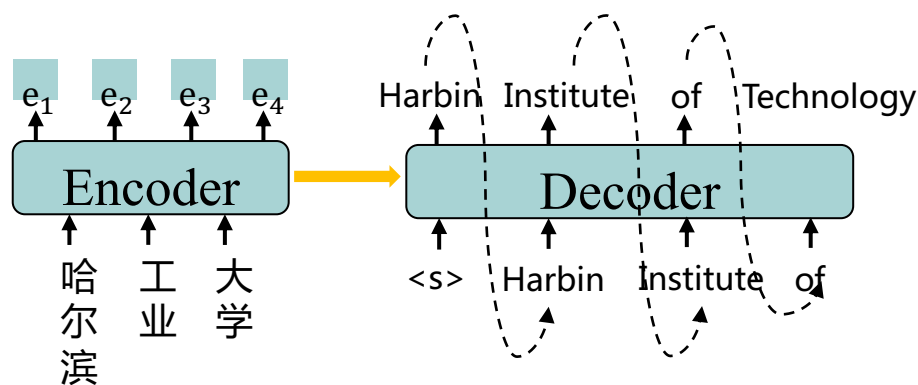
- ▶ 给定序列输入，解码时如何得到最优序列？
 - ▶ 贪心搜索：每一步解码时选择局部概率最大的词汇作为输出



第一步解码	0.6 Harbin	0.2 Institute	0.1 of	0.1 Technology
第二步解码	0.2 Harbin	0.3 Institute	0.4 of	0.1 Technology
第三步解码	0.2 Harbin	0.6 Institute	0.1 of	0.1 Technology

生成式命名实体识别—Beam Search(集束搜索)

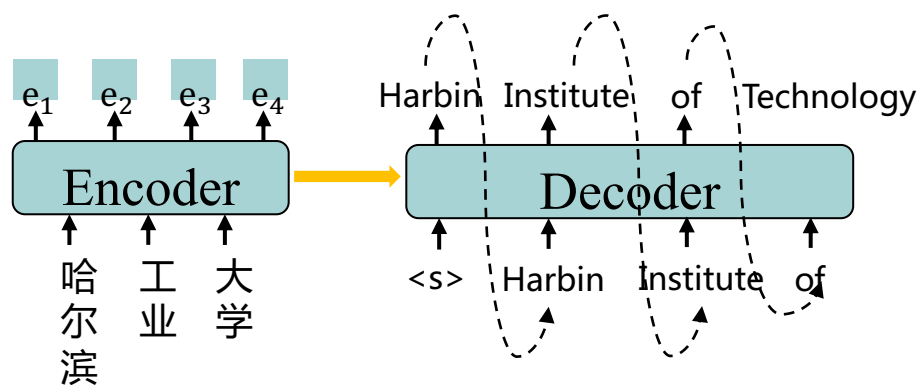
- ▶ 给定序列输入，解码时如何得到最优序列？
 - ▶ 贪心搜索：每一步解码时选择局部概率最大的词汇作为输出



第一步解码	0.6 Harbin	0.2 Institute	0.1 of	0.1 Technology
第二步解码	0.2 Harbin	0.3 Institute	0.4 of	0.1 Technology
第三步解码	0.2 Harbin	0.6 Institute	0.1 of	0.1 Technology
第三步解码	0.1 Harbin	0.1 Institute	0.3 of	0.5 Technology

生成式命名实体识别—Beam Search(集束搜索)

- ▶ 给定序列输入，解码时如何得到最优序列？
 - ▶ 贪心搜索：每一步解码时选择局部概率最大的词汇作为输出

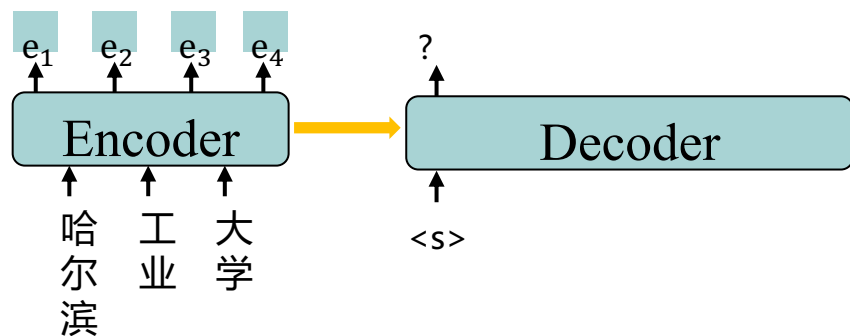


第一步解码	0.6 Harbin	0.2 Institute	0.1 of	0.1 Technology
第二步解码	0.2 Harbin	0.3 Institute	0.4 of	0.1 Technology
第三步解码	0.2 Harbin	0.6 Institute	0.1 of	0.1 Technology
第三步解码	0.1 Harbin	0.1 Institute	0.3 of	0.5 Technology

局部最优解不一定为全局最优解！

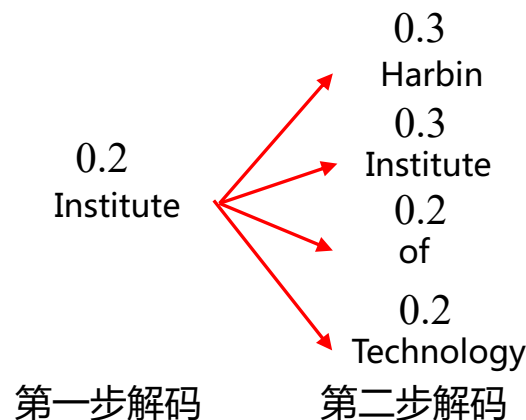
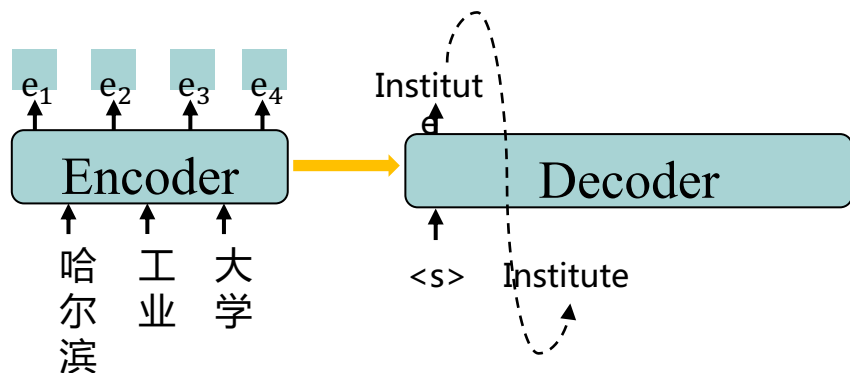
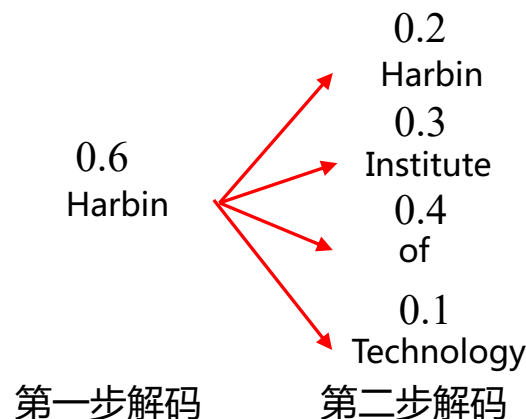
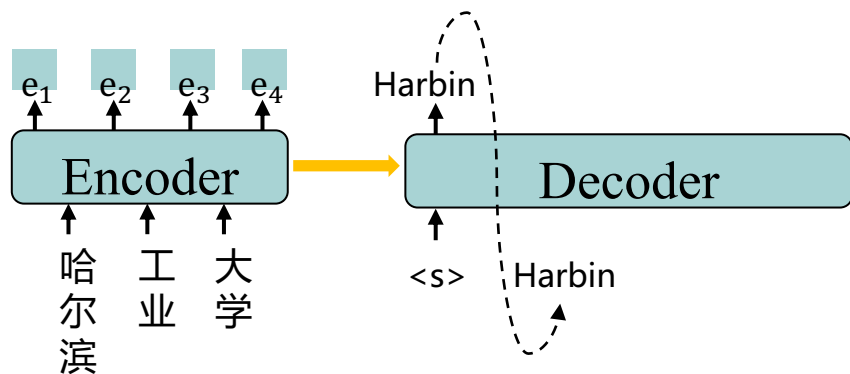
生成式命名实体识别—Beam Search(集束搜索)

- ▶ 给定序列输入，解码时如何得到最优序列？
 - ▶ 集束搜索：同时保留beam size(e.g., 2)个使得当前解码概率最大化的输出序列



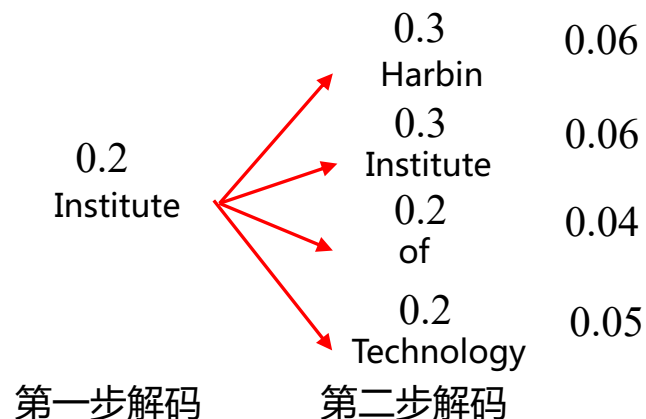
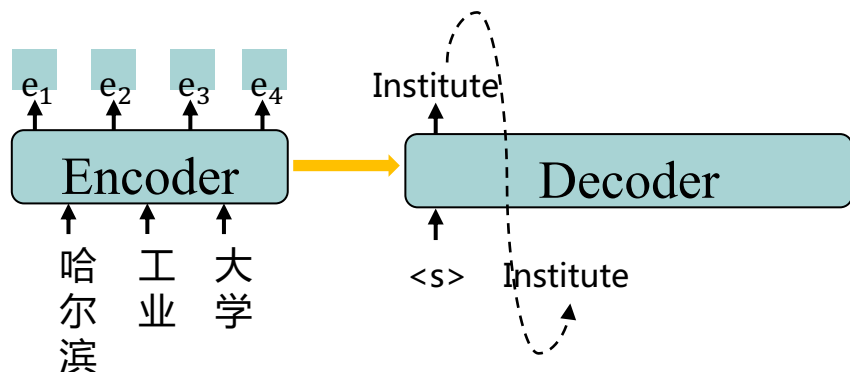
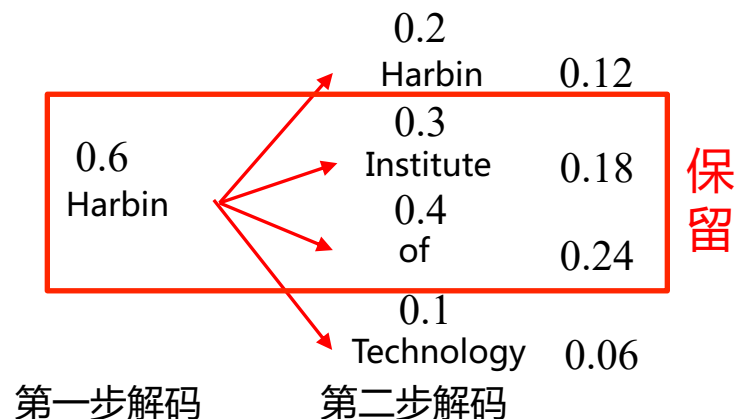
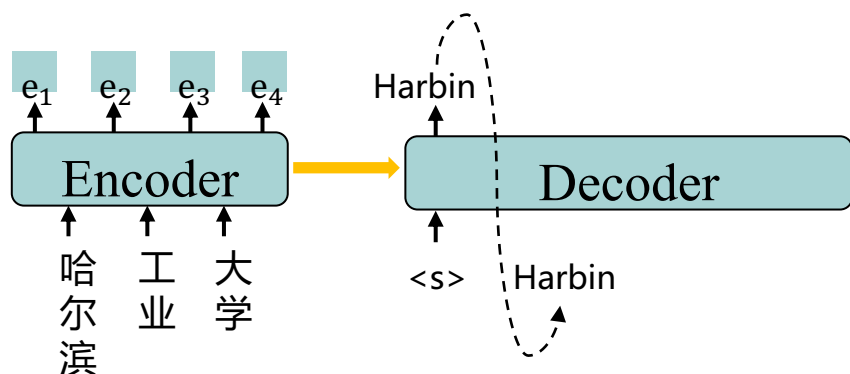
生成式命名实体识别—Beam Search(集束搜索)

- ▶ 给定序列输入，解码时如何得到最优序列？
 - ▶ 集束搜索：同时保留beam size(e.g., 2)个使得当前解码概率最大化的输出序列



生成式命名实体识别—Beam Search(集束搜索)

- ▶ 给定序列输入，解码时如何得到最优序列？
 - ▶ 集束搜索：同时保留beam size(e.g., 2)个使得当前解码概率最大化的输出序列



生成式命名实体识别—Beam Search(集束搜索)

- ▶ 给定序列输入，解码时如何得到最优序列？
 - ▶ 穷举搜索：计算量和解码时间随词表大小和序列长度呈指数级递增
 - ▶ 贪心搜索：实现最为简单，解码性能较差
 - ▶ 集束搜索：实现较为复杂，解码性能较好 ✓
 - ▶ 搜索空间与解码开销的trade-off
 - ▶ 长序列和短序列的长度正则化
 - ▶ 多种变体：Diverse Beam Search，lexically constrained decoding



<https://github.com/pytorch/fairseq>

Seq2seq

<https://github.com/bentrevett/pytorch-seq2seq>

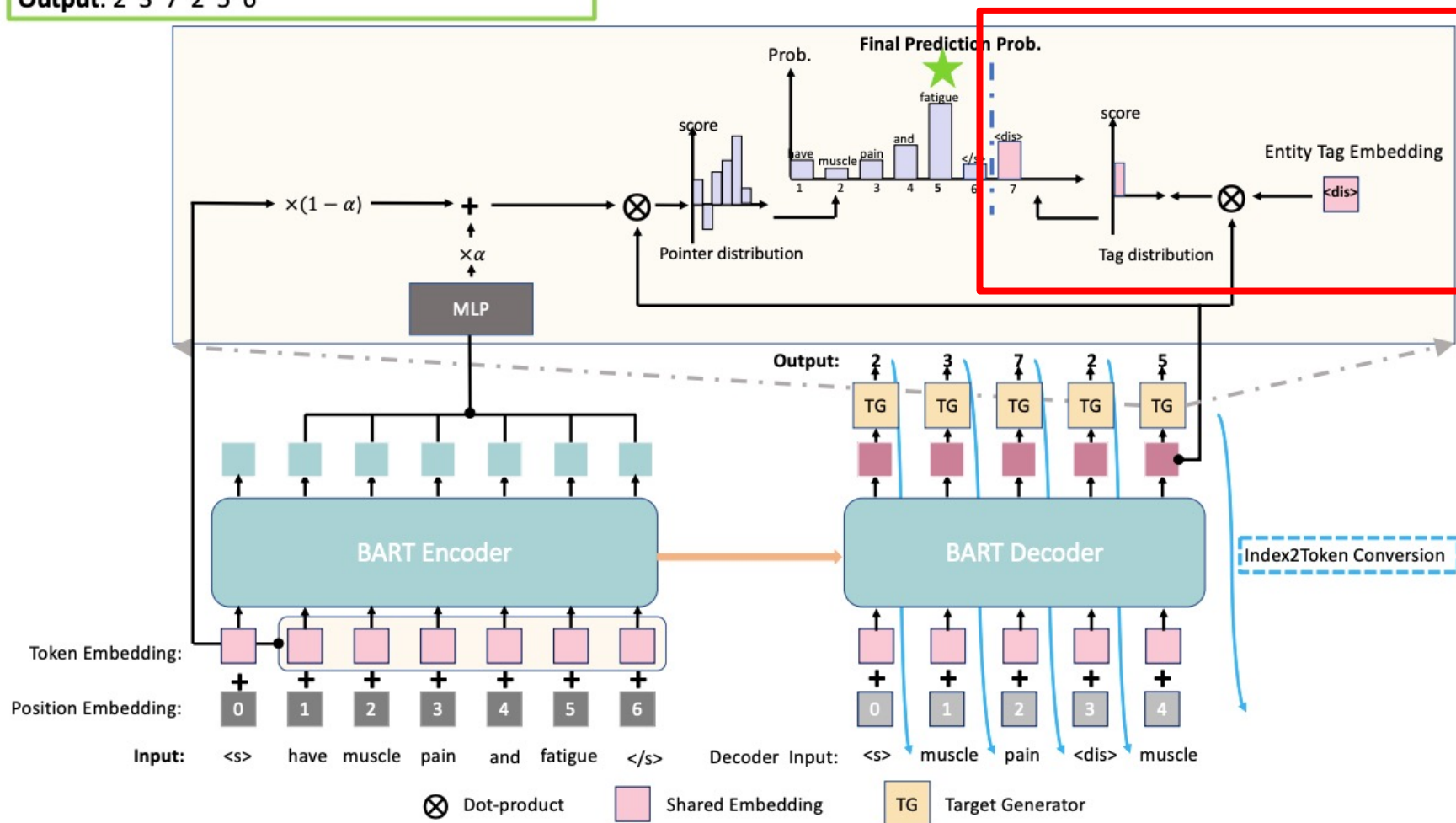
Beam
Search

<https://github.com/budzianowski/PyTorch-Beam-Search-Decoding>

基于拷贝机制的生成式命名实体识别

Input: <s> have muscle pain and fatigue </s>
Output: 2 3 7 2 5 6

增加了实体类别的生成



基于拷贝机制的生成式命名实体识别

- 针对平滑命名实体识别、嵌套命名实体识别、不连续命名实体识别等，基于**序列到序列**的生成模型统一建模

平滑命名实体识别

S1: B-Per Barack I-Per Obama O was O born O in O the B-Loc US
Person Location

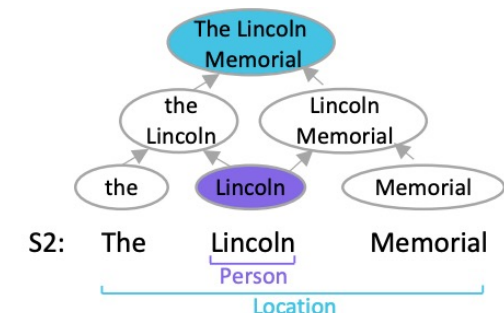
(a) Sequence labelling for flat NER

不连续命名实体识别

Actions: OUT OUT SHIFT SHIFT LEFT-REDUCE COMPLETE ...
S3: have much muscle pain and fatigue
Disorder Disorder

(c) Transition-based method for discontinuous NER

嵌套命名实体识别



(b) Span-based classification for nested NER

生成式统一建模

S1: Barack Obama <Person> US <Location>
S2: The Lincoln Memorial <Location> Lincoln <Person>
S2: muscle pain <Disorder> muscle fatigue <Disorder>

(d) A unified generative solution for all NER tasks

命名实体小结

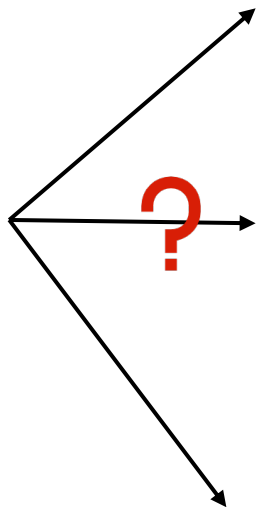
建模方式	连续	嵌套	非连续
序列标注	Y	N	N
检测与分类	Y	Y	N
生成式（模版）	Y	Y	N
生成式（拷贝）	Y	Y	Y

- ▶ 知识图谱的构建流程
- ▶ 实体识别
- ▶ **实体消歧**
 - ▶ 基本定义
 - ▶ 基于聚类的实体消歧
 - ▶ 基于实体链接的实体消歧
- ▶ 关系抽取
- ▶ 事件抽取
- ▶ 开放域知识抽取
- ▶ 多模态知识抽取

实体消歧基本定义

- ▶ 给定待消歧实体（实体指称）及上下文，确定其与目标实体列表的**对应关系**

张杰曾经担任过上海交通大学的校长。



[张杰\(中国科学院院士、物理学家\) - 百度百科](#)



职业：教育科研工作者
生日：1958年1月
主要成就：2003年当选为中国科学院院士，2006年至2017年担任上海...
简介：张杰，1958年1月出生于山西太原，籍贯河北邢台，物...
[人物经历](#) [主要成就](#) [社会任职](#) [人物评价](#)

百度百科

张杰

39周岁 歌手

[百度百科](#) >



职业：歌手
生日：1982年12月20日
个人信息：180 cm/66 kg/射手座/O型
代表作品：天下、勿忘心安、一路之下、这，就是爱、最美的太阳、他...
简介：张杰 (Jason Zhang)，1982年12月20日出生于四川省...
[早年经历](#) [演艺经历](#) [个人生活](#) [主要作品](#) [社会活动](#) [更多 >](#)

[张杰\(中国工程院院士、哈尔滨工业大学教授\) - 百度百科](#)



职业：教师
生日：1938年02月27日
主要成就：1997年当选为**中国工程院院士**
简介：张杰，1938年2月27日出生于辽宁本溪，给排水科学与...
[人物经历](#) [主要成就](#) [社会任职](#) [人物评价](#)

百度百科

实体消歧基本定义

- ▶ 为什么需要实体消歧？
 - ▶ 指称的多样性

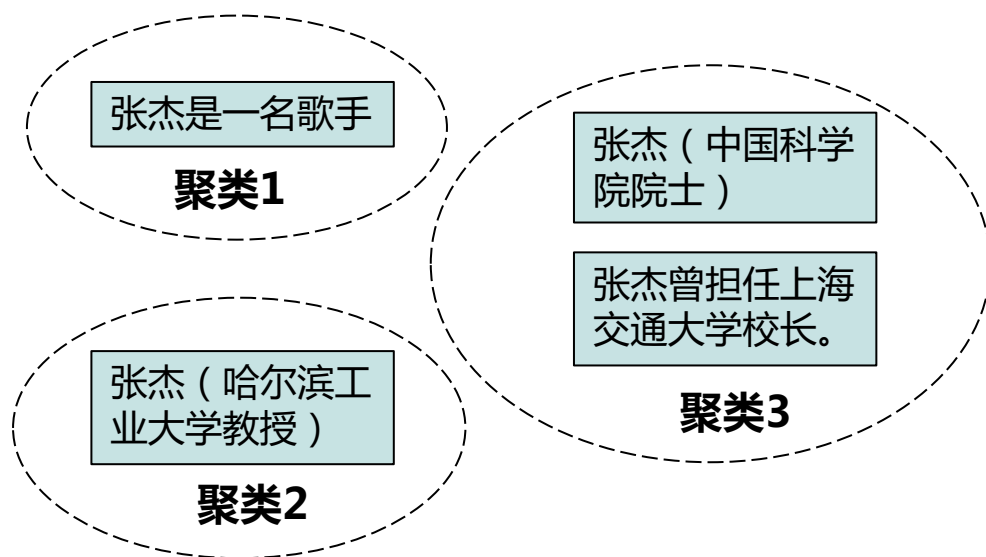
完全实体名	哈尔滨工业大学位于哈尔滨市
部分实体名	哈工大创建于1921年；工大计算机系
缩写	HIT是哈尔滨工业大学的简称
别名	哈尔滨马家沟技工学校；

- ▶ 指称的歧义性

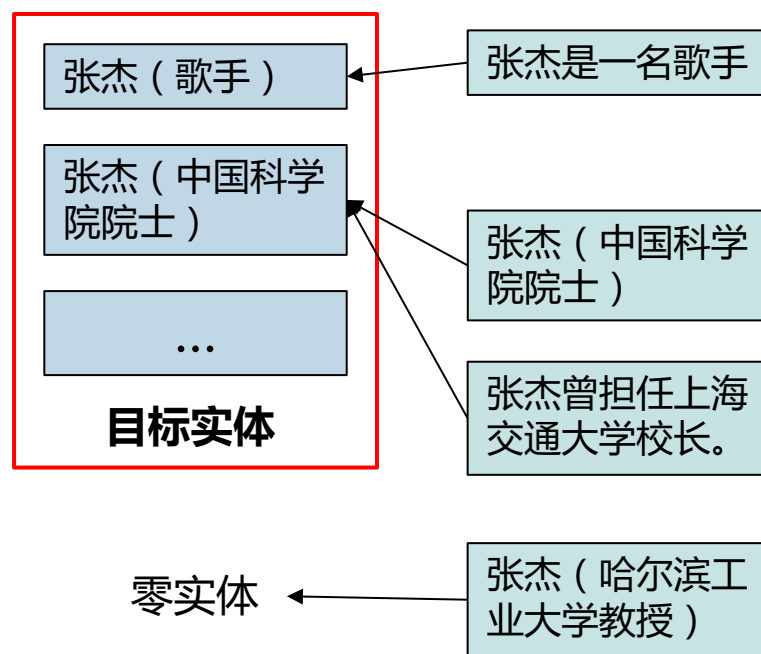
工业大学	哈尔滨工业大学	北京工业大学
张杰	张杰（歌手）	张杰（中国科学院院士）
李娜	李娜（网球运动员）	李娜（跳水运动员）

实体消歧任务分类

- 根据是否给定目标实体列表，实体消歧可分为：



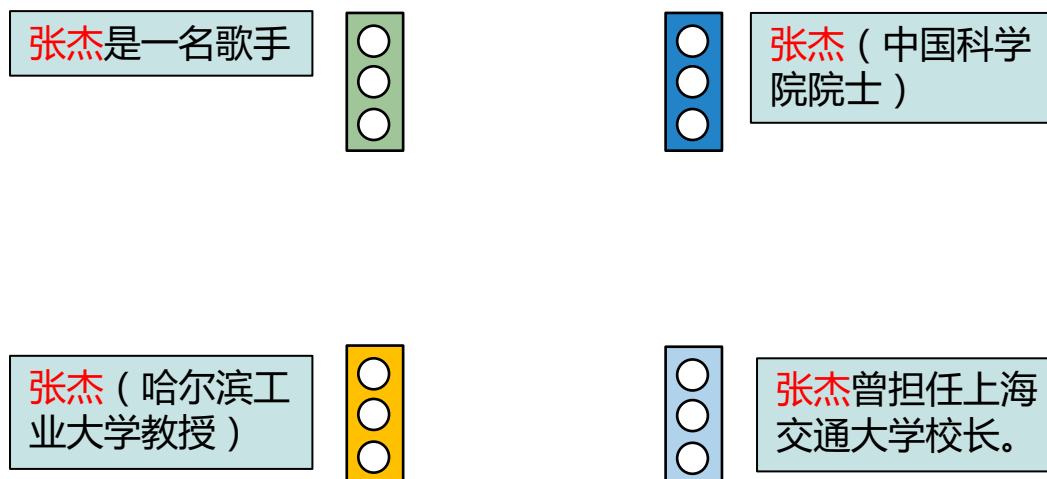
基于**聚类**的实体消歧



基于**实体链接**的实体消歧

基于聚类的实体消歧

- ▶ 基于聚类的实体消歧可分为以下步骤：
 - ▶ 对于每一个实体指称，构建对应的表征向量





基于聚类的实体消歧

▶ 表征向量计算方法

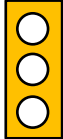
- ▶ 基于检索方法（e.g. TF-IDF）的向量特征
- ▶ 基于扩展特征（e.g. 实体的属性）的向量特征
- ▶ 基于社会化网络（实体的社会化关系）的向量特征
- ▶ 基于神经网络（e.g. BERT）的向量特征

张杰是一名歌手



张杰（中国科学院院士）

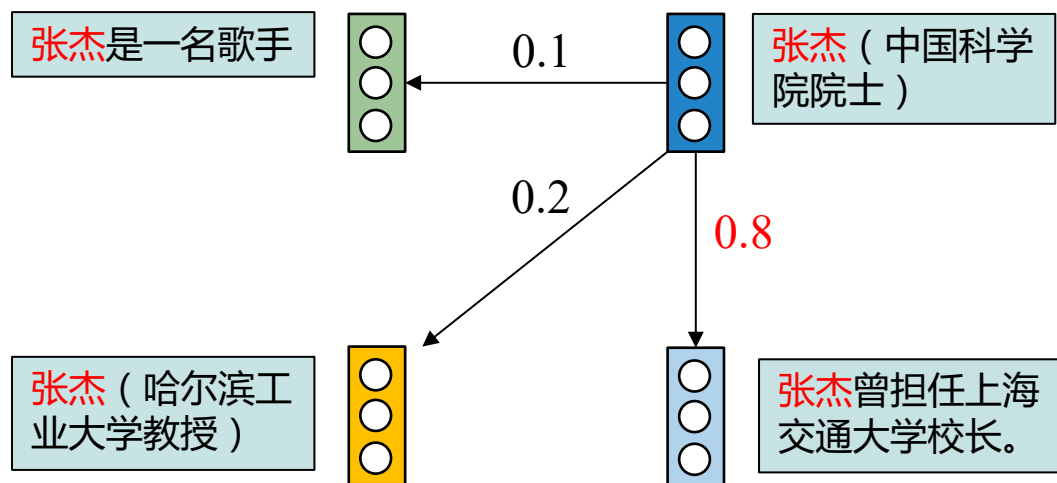
张杰（哈尔滨工业大学教授）



张杰曾担任上海交通大学校长。

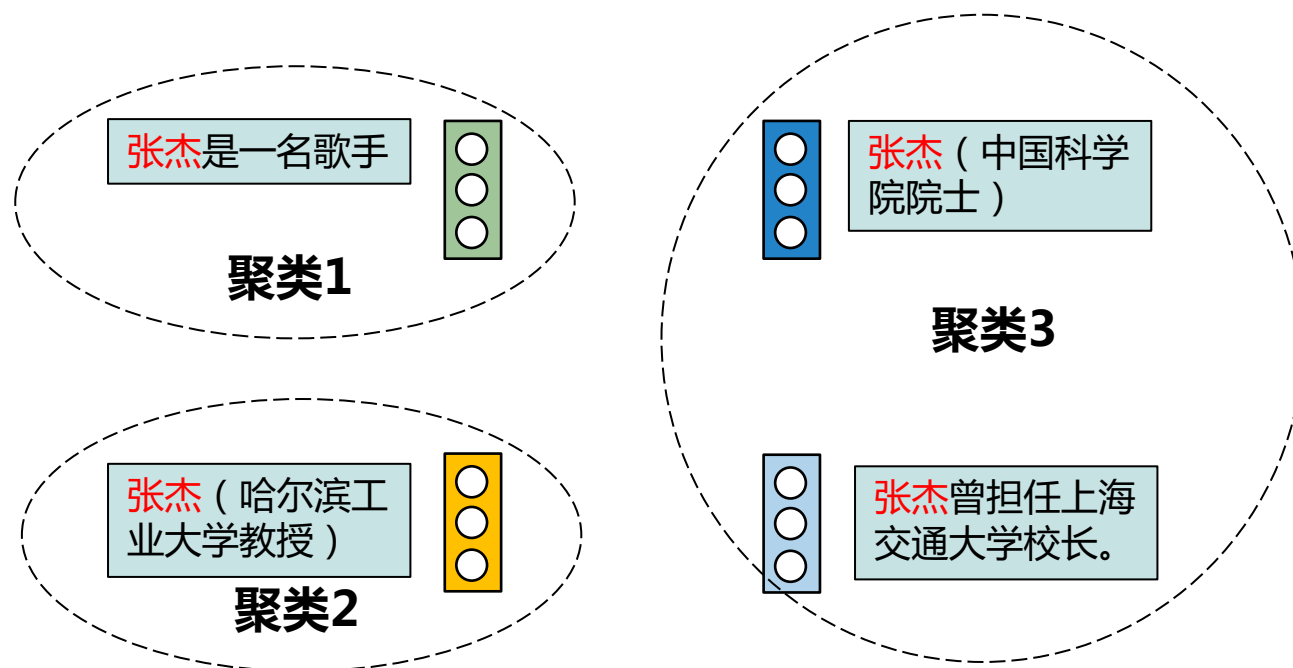
基于聚类的实体消歧

- ▶ 基于聚类的实体消歧可分为以下步骤：
 - ▶ 对于每一个实体指称，构建对应的表征向量
 - ▶ 计算实体指称项之间的相似度



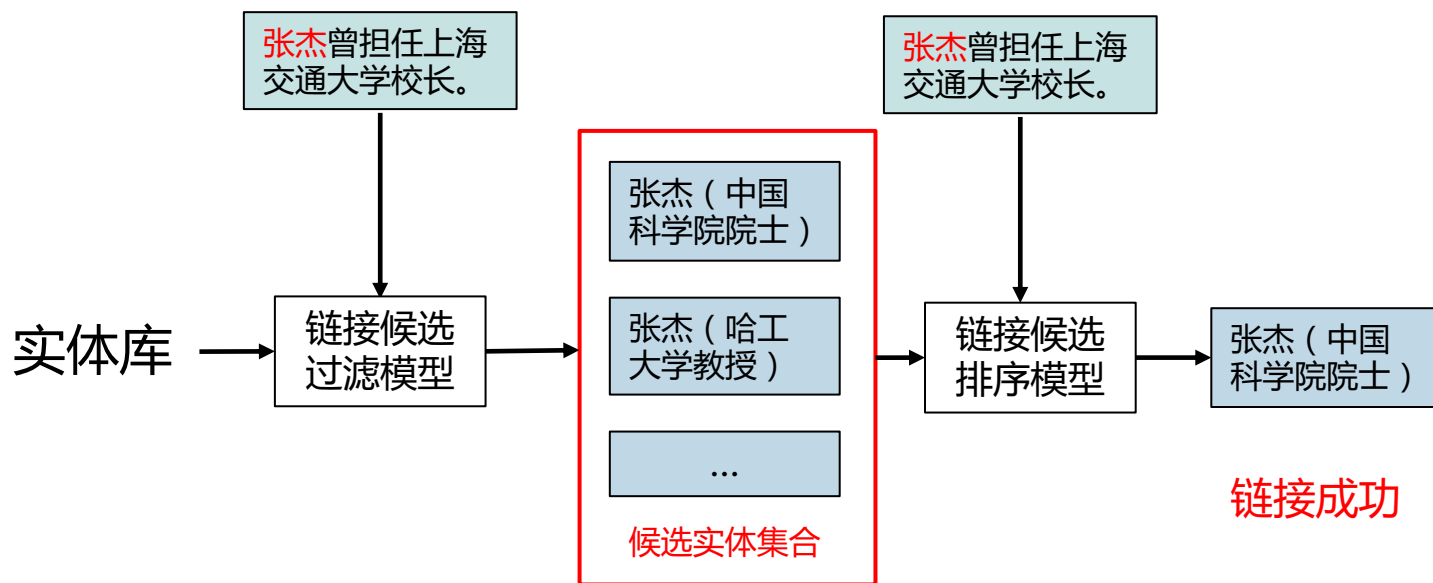
基于聚类的实体消歧

- ▶ 基于聚类的实体消歧可分为以下步骤：
 - ▶ 对于每一个实体指称，构建对应的表征向量
 - ▶ 计算实体指称项之间的相似度
 - ▶ 采用某种聚类算法对实体指称项聚类



基于实体链接的实体消歧

- ▶ 基于实体链接的实体消歧
 - ▶ 将某个实体指称链接到知识库中特定的实体
 - ▶ 可分为链接候选过滤和实体链接两步



链接候选过滤

- ▶ 链接候选过滤
 - ▶ 根据规则或知识过滤掉大部分指称项不可能指向的实体
 - ▶ 常基于**实体指称项词典**进行过滤
 - ▶ 可根据Wikipedia等知识资源来构建实体指称项词典

实体名	目标实体
工大	哈尔滨工业大学 北京工业大学 合肥工业大学 武汉工业大学 哈尔滨工程大学 哈尔滨理工大学
张杰	张杰(男歌手) 张杰(中国科学院院士) 张杰(哈尔滨工业大学教授) 张杰(安徽省政府副秘书长) ...

▶ 实体链接

- ▶ 将某个实体指称链接到候选实体列表中的某个实体
- ▶ 分为向量空间模型、主题一致性模型、神经网络模型等方法



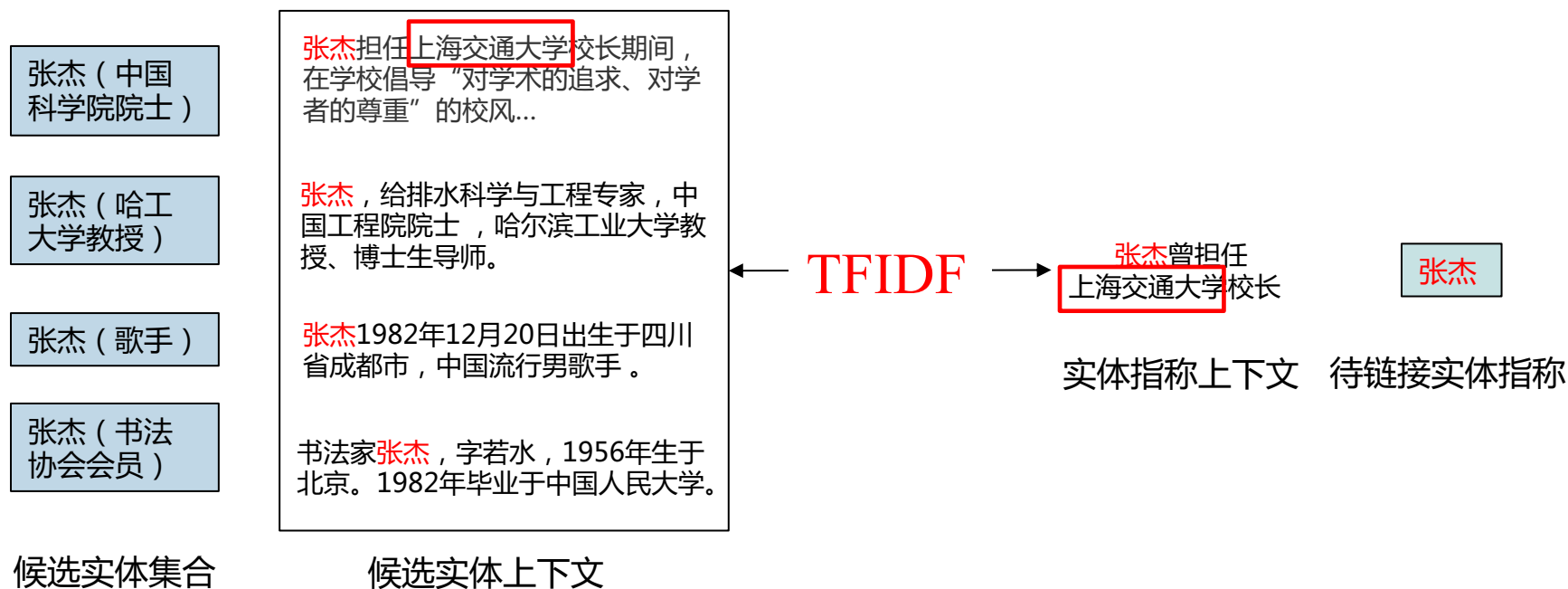
张杰曾担任上海
交通大学校长。

待链接实体指称

候选实体列表

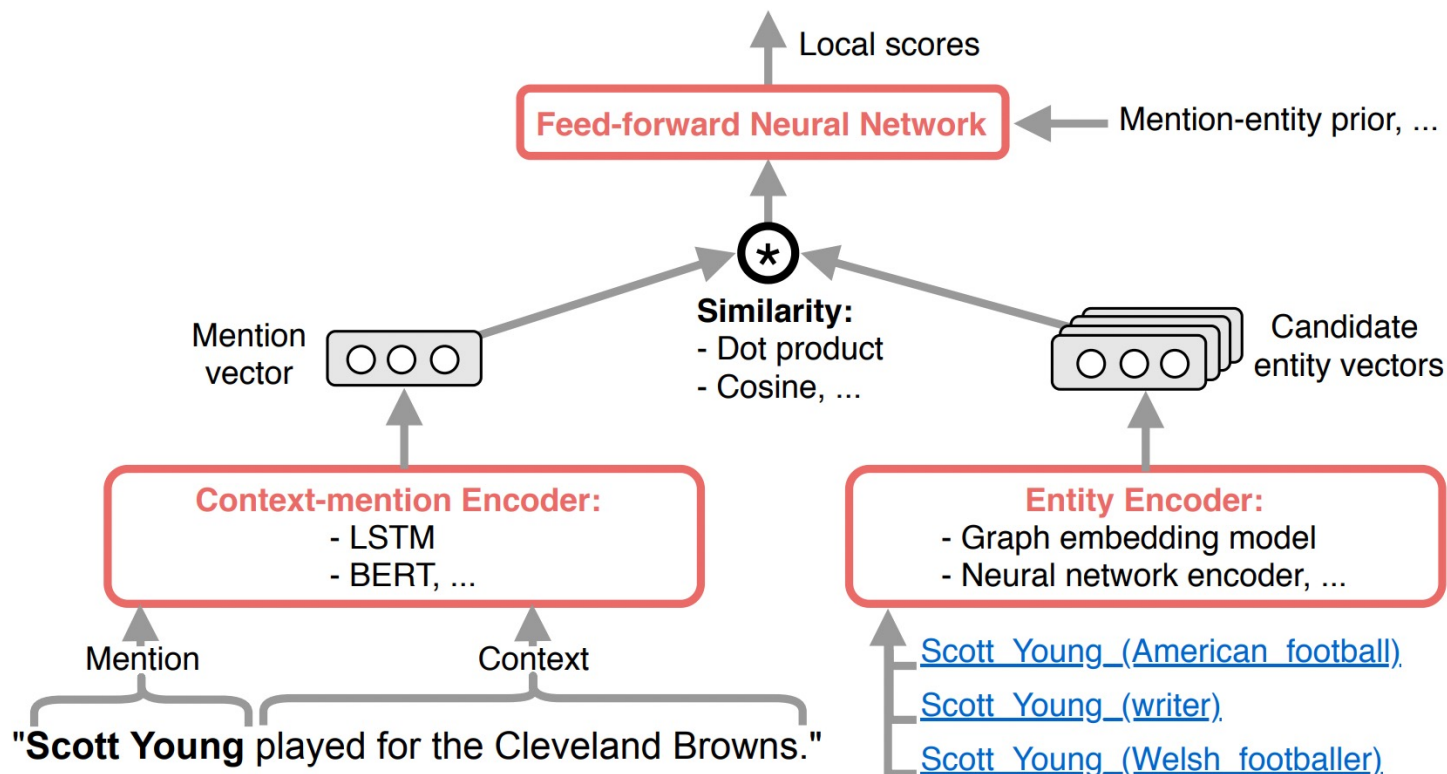
▶ 向量空间模型

- ▶ 将实体指称项上下文与候选实体上下文中特征的共现信息作为两者的一致性分数（TF-IDF算法）



神经网络模型

- 使用神经网络编码器获得候选实体和待链接实体指称的特征表示并进行匹配打分



- ▶ 知识图谱的构建流程
- ▶ 实体识别
- ▶ 实体消歧
- ▶ 关系抽取
- ▶ 事件抽取
- ▶ 开放域知识抽取
- ▶ 多模态知识抽取