

Natural Language Processing

#L8

句义分析

■句子语义：不同的角度

■形式语义：句子的真假值

■与客观所指的关系

■他高高瘦瘦的。 真/假

■句子之间的逻辑关系

■他高高瘦瘦的。 → 他高。 句子间的关系

■概念语义：句子反映的概念内容

■他高高瘦瘦的：对象“他”有“高”和“瘦”两个属性

■他买了一本书：他执行了买的动作，买的对象是一本书

■.....

■情感语义：句子反映的情感内容

■这电影太好了，我反复看了好几遍

■句子语义表示与分析

■基于符号的方法

■基于数值的方法

- 基于符号的方法
 - 基于符号的句子语义表示
 - 基于AI知识表示
 - 基于语言学语义表示
 - 基于符号的句子语义分析
 - 句法驱动的句义分析
 - 基于句法结构的句义分析
 - 基于语义语法的语义分析
 - 语义驱动的句法分析
 - 语义角色标注 (语言学语义表示+语义语法驱动)

■ 基于符号的方法

■ 基于符号的句子语义表示

- 基于AI知识表示

- 基于语言学语义表示

■ 基于符号的句子语义分析

- 句法驱动的句义分析

- 基于句法结构的句义分析

- 基于语义语法的语义分析

- 语义驱动的句法分析

■ 语义角色标注 (语言学语义表示+语义语法驱动)

基于符号的表示：AI知识表示

■谓词

- 一阶谓词、

- He ran fast. RAN(He) or RAN-FAST(He)

■框架

- 格框架(语义角色)

- S(Agent=He, Manner=fast)

■三元组集合

- <S PRED RAN>, <S SUB He>...

■图

- 概念图

基于符号的表示：语言学语义表示理论

■ AMR(Abstract Meaning Representation), 例如:

■ 男孩想去学校

■ x/想-01 (01: 想的第一个义项)

■ : arg0 x1/男孩

■ : arg1 x2/去-01 (01: 去的第一个义项)

■ : arg0 x1

■ : arg1 x3/学校

■ UCCA(Universal Conceptual Cognitive Annotation)

■ UDS(Universal Decompositional Semantics)

■

■ 基于符号的方法

■ 基于符号的句子语义表示

- 基于AI知识表示

- 基于语言学语义表示

■ 基于符号的句子语义分析

- 句法驱动的句义分析

- 基于句法结构的句义分析

- 基于语义语法的语义分析

- 语义驱动的句法分析

■ 语义角色标注 (语言学语义表示+语义语法驱动)

- 基于符号的句子语义分析
 - 句法驱动的句义分析
 - 基于句法结构的句义分析
 - 基于语义语法的语义分析
 - 语义驱动的句法分析
- 使用的句法信息逐渐减少
- 语义逐渐从从属地位变为主导地位

句法驱动的语义分析

■句法驱动

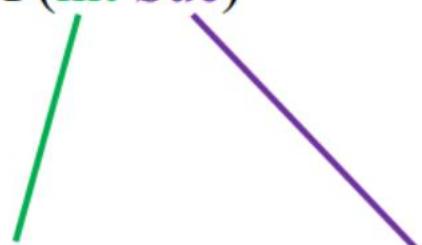
■语义是以句法特征的形式出现，在进行句法分析的同时得到句子的句义。句法规则包含语义规则。

■句义组合

■句法的一个典型特点是组合性，即大的语法单元由较小的语法单元构成，而句义分析要能和句法分析同时进行，也要求句义应该有和句法相似的组合性。

- 句法结构：句法规则表示
- 语义结构：用谓词逻辑表示
- 例子：
- hit Sue.

■句法结构： VP(hit Sue)

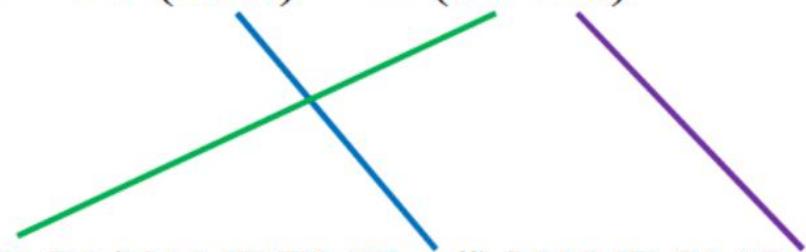


- 语义结构： (HIT1 E1 (NAME S1 “Sue”))
- 一致的组合方式，可以，不过→

■例子：

■Tom hit Sue.

■句法结构： $S \rightarrow NP(Tom) VP(hit\ Sue)$



■语义结构： (HIT1 E1(NAME T1 “Tom”)(NAME S1 “Sue”))

■二者的组合方式不一致，导致不能同时进行。

■问题：语义结构中，谓词的动作(或事件)在最前面，其涉及的参与者均在后，但是句法结构中，谓词的主语在前。

简单运算

- 例. 两个具有不同句法结构的动词短语的连结
 - Sue laughs and opens the door.
- 先分别为laughs和opens the door建立λ-表达式
- 对于laughs，有：
 - $(\lambda x (\text{LAUGHS1 E1 } x))$
- 对于opens the door，有：
 - $(\lambda y (\text{OPENS1 E2 } y (\text{OBJ o1 "the door"})))$

- 当两个VP有相同动作主体，组合在一起时，有 $x=y$ ，组合后的 λ -表达式为：
 - $(\lambda x (\& (\text{LAUGHS1 E1 } x) (\text{OPENS1 E2 } x (\text{OBJ o1 "the door"}))))$
- 用(NAME s1 “Sue”)替换 λ -表达式中的x，进行 λ -还原后就完整地表示了句义：
 - $(\& (\text{LAUGHS1 E1 } (\text{NAME s1 "Sue"}) (\text{OPENS1 E2 } (\text{NAME s1 "Sue"}) (\text{OBJ o1 "the door"}))))$

- 有了与句法结构对应的语义结构表示，语义分析就可以方便地在句法分析的驱动下进行
- 为此需要一些扩展，主要的扩展是在每一个词条和语法规则中增加一个特征，这个特征就是反映语义的，通常用SEM来表示这个特征。

■例：词条Tom，在没有语义特征时，是：

$$Tom = [POS \quad N]$$

■增加语义特征后为：

$$Tom = \begin{bmatrix} POS & & & N \\ SEM & NAME & t1 & "Tom" \end{bmatrix}$$

■句法规则

■ $S \rightarrow NP \ VP \rightarrow$

■ $[POS \ S] = [POS \ NP] [POS \ VP]$

■在增加语义特征后的形式为：

■ $\begin{bmatrix} POS & S \\ SEM & ?semS \end{bmatrix} = \begin{bmatrix} POS & NP \\ SEM & ?semNP \end{bmatrix} \begin{bmatrix} POS & VP \\ SEM & ?semVP \end{bmatrix}$

■ $S(?semS) \rightarrow NP(?semNP) VP(?semVP)$ 1)

■其中的语义特征就是对句法规则在语义上的约束

设几条其他带语义特征的规则如下：

- $\text{VP}(\text{SEM}(\lambda \text{ a1}(\text{?semv?v a1?semnp})))$
 $\rightarrow (\text{V SEM?semv})(\text{NP SEM?semnp})$ 2)
- $\text{NP}(\text{SEM}(\text{NAME?semname}))$
 $\rightarrow (\text{NAME SEM?semname})$ 3)
- $\text{NP}(\text{SEM}(\text{?semart ?semcnp}))$
 $\rightarrow (\text{ART SEM?semart})(\text{CNP SEM?semcnp})$ 4)
- $\text{CNP}(\text{SEM?semn}) \rightarrow (\text{N SEM?semn})$ 5)

■ Tom saw the dog.

■ 首先读入Tom，由规则(3)得到一个NP，其语义特征：

■ NP1(SEM(NAME t1 "Tom"))

■ 随后读入saw，没有可应用的规则

■ 继续the;无可用规则

■ 继续 dog, 此时, 由规则(5), 得到一个CNP, 其语义特征：

■ CNP(SEM DOG1)

■ 该CNP与前面的the应用规则(4)，得到另一个NP：

■ NP2(SEM(The Dog1))

■ 习语的意义很难用组合性来处理，例：

■ Tom kicked the bucket.

■ 意为：

■ Tom died

■ 句子的意义不可能由其组成成分来组合而成

基于句法结构的语义分析

- 建立在句法分析的结果之上，以句法分析的结果作为语义分析的输入。
- 在这一类方法中，依据对句法分析的结构的处理不同，又有各种变化。

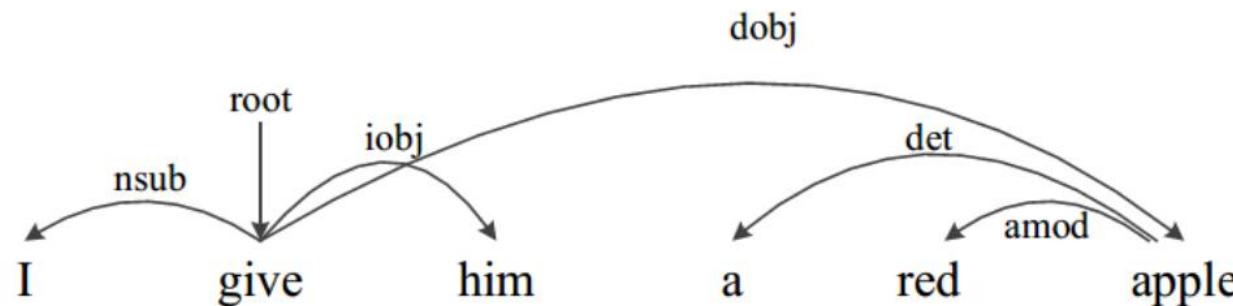
一种基于句法结构的语义分析的示例

■首先：

■利用依存语法，得到句子的依存结构，例如，对于句子：

■I give him a red apple.

■可以得到其句法结构为：



■其次：

■把句法结构转化成一系列三元组：

■<give nsubj I >

■<give dobj apple>

■<give iobj him>

■<apple amod red>

■<apple det the >

■.....

■最后：

- 把三元组映射到逻辑形式, 即实现将语法关系到逻辑形式的映射

语法关系	逻辑形式
<give nsub I >	$\lambda x \lambda y (\text{GIVE1 E1 (PERSON p1 "I") (x) (y)})$
<give dobj apple>	$\lambda x \lambda y (\text{GIVE1 E1 (x) (y) (OBJECT o1 "apple")})$
<give iobj him>	$\lambda x \lambda y (\text{GIVE1 E1 (x) (PERSON p2 "him") (y)})$
...	

- 语法形式之后的逻辑形式是否可以得到原始的语义?

基于语义语法的语义分析

- 在为特定的应用在特定领域中构造自然语言应用系统时，通常可以利用一些很强的约束技术来提高句法、语义分析的性能。
- 在特定应用中，人们可能只会用到自然语言中非常小的一部分结构，而且句子结构中的每个成分可能都有十分明确的语义约束。

- 例如，对于飞机航班数据库的查询中，通常会有语言结构：
 - TIME-QUERY → When does FLIGHT-NP FLIGHT-VP
 - 例:when does the flight to Chicago leave?
 - FLIGHT-NP → FLIGHT-N NUMB
 - 例:flight 457
- 这是一种以领域内语义范畴为结构单元的语法：语义语法
 - TIME-QUERY、FLIGHT-NP、FLIGHT-VP、FLIGHT-NP、FLIGHT-N、NUMB 都是该领域特定的语义范畴，其组合方式也是该领域特定的。

- 利用语义语法可以快速地开发在特定领域针对特定任务的应用系统，
- 但是其构造的语义语法基本上不能推广到其他的领域，新的领域需要构造新的语义语法。而通常的句法分析就不会受到如此强烈的领域约束。

语义驱动的句法分析

- 前面介绍的几种语义分析都是要建立在句法分析(或句法组合思想)的基础上,
- 一个极端是语义作为一个特征发生在句法分析的同时, 无需单独存在, 有的在句法分析完成之后, 而基于语义语法的分析是用语义信息改造语法规则。
- 另一个极端, 即完全抛开语法(除了基本的词的形态分析), 直接对句子进行语义分析。当这种语义分析结束时, 也同时就获得了句子的结构, 因此称之为语义驱动的句法分析。

■语义驱动的句法分析

■主要特定：

■语义信息主要储存在词典中，包括静态词的不同可能意义，动词、形容词的格框架信息，还包括一些过程性的知识，如：消除词汇歧义的操作，语义结构组合的操作等等。

■例：词典中的语义信息以模式-行为规则的形式存储，在读入一个词时，就进行模式匹配，一旦匹配上一条规则，就按规定的操作做相应的操作。

■例如，在词典中book一词可能包含如下信息：

■BOOK .1

■<ANIMATE> “book” \rightarrow (RESERVING * [AGENT 1])

■ BOOK .2

■<RESERVING><TRANSPORT> \rightarrow .1 \wedge (RESERVING * [THEME 2])

■BOOK .3

■<RESERVING><ANIMATE> \rightarrow .1 \wedge (RESERVING * [BENEFICIARY 2])

- 基于语义驱动的句法分析是依赖词典中对每个词构造的规则，不同的词有不同的规则，因此难以抓住语言现象的共性。
- 对于复杂句子结构也难以仅仅利用对词的构造的规则进行分析，因为词的规则通常只在小范围发生作用。

■ 基于符号的方法

■ 基于符号的句子语义表示

- 基于AI知识表示

- 基于语言学语义表示

■ 基于符号的句子语义分析

- 句法驱动的句义分析

- 基于句法结构的句义分析

- 基于语义语法的语义分析

- 语义驱动的句法分析

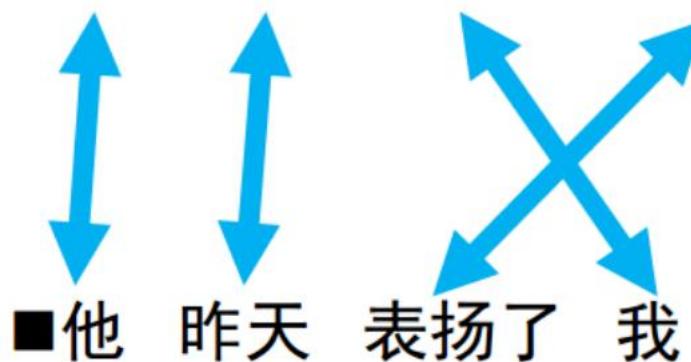
■ 语义角色标注 (语言学语义表示+语义语法驱动)

基于语义角色的语义分析

- 格语法：语义语法

- 句子表达事件

- 谁 何时 对谁 做了何事



- 表扬了：谓词，表达事件

- 他，我，昨天：谓词的论元，事件的参与者(参数)

- 语义角色：论元在谓词所描述的事件中所扮演角色的抽象模型
- 例如句子： Jim gave the book to Tom
 - 谓词give描述了一个事件GIVE
 - Jim、 the book、 Tom三个论元是该事件的参与者
 - Jim扮演了GIVE事件的发起者
 - the book 是GIVE事件的内容
 - Tom是GIVE事件的内容接受者

■这种抽象语义描述的优势：

- Jim gave the book to Tom
- Tom was give the book by Jim
- Jim gave Tom the book
- The book was given to Tom by Jim
- ...

■这些不同描述后面有一个相同的语义结构：

- 事件GIVE(GIVE事件的发起者=Jim, GIVE事件的内容=the book, GIVE事件的内容接受者=Tom)

■语义角色抽象模型的具体实施方式

■论旨角色(Thematic roles)

■为所有事件定义一个有限的角色列表

■以下在分析论旨角色问题之后，介绍后续的变种

■更具体：为每一个动词定义一个语义角色列表

■PropBank

■更一般：提供一些高层角色proto-agent/proto-patient

■PropBank

■折衷：逐框架地定义语义角色

■FrameNet：具有相同格框架

■ 论旨角色列表(部分)

论旨角色	描述	例子
施事(Agent)	执行动作者	<i>He</i> ran
主题(Theme)	承担动作者	He found <i>the dog</i>
目标(Goal)	动作的指向	Put the cat on <i>the porch</i>
位置(Location)	动作发生地	It rains <i>in Spain</i>
工具(Instrument)	动作执行的方式	He cuts hair <i>with a razor</i>
原因(Causative)	导致改变的力量	<i>The wind</i> damaged a roof
来源(Source)	动作发源地	He flew <i>from Iowa</i> to here
体验者(Experiencer)	感受者	<i>He</i> heard her playing the piano
拥有者(Possessor)	某物所属者	The tail of <i>the dog</i> wagged
受益者(Beneficiary)	事件受益者	He makes reservations for <i>her</i>
结果(Result)	事件产出物	They built <i>a house.</i>

■ 基于论旨角色的句子语义描述：

- 1) Jim gave the book to Tom.
 - <=> Event=GIVE, Agent=Jim, Theme=the book, Goal=Tom
- 2) Jim gave Tom the book.
 - <=> Event=GIVE, Agent=Jim, Theme=the book, Goal=Tom
- 3) Tom was given the book by Jim.
 - <=> Event=GIVE, Agent=Jim, Theme=the book, Goal=Tom
-

■ 可以作为句子的浅层语义表示：

- 不同表层形式的句子的共性浅层语义
- GIVE(Agent=Jim; theme=the book; Goal=Tom)

- 以动词为核心的句子表示
- 论旨角色是动词对其论元的要求
 - Give: (Agent=oblige; theme=oblige; Goal=oblige;)
 - I give him a book.
 - Find: (Agent= oblige; theme=oblige; Loc=optional)
 - I found the book (in next room).
 - See: (Agent= oblige; theme=oblige; Time=optional)
 - I saw him (yesterday).

■格框架(case frame, thematic grid, Θ -grid)

■1) 动词需要的论旨角色组合称为该动词的~

■Jim gave the book to Tom yesterday.

■Give: (Agent=oblige; theme=oblige; Goal=oblige;
Time=optional)

■2) 一个动词可能有多个格框架

■You're going to have to give a little.

■Give : (Agent=oblige)

■ 基于格框架的动词分类

■ 具有类似的格框架的动词构成一类

■ Give、Offer等： (Agent=oblige; theme=oblige; Goal=oblige;
Time=optional)

■ See、hear等： (Agent=oblige; theme=oblige; Time=optional)

■

■ Levin(1993)基于此将3100个英语动词分为47大类，193小类

■ Kipper et al. (2000)基于此构建VerbNet.

■论旨角色的问题

■定义清楚一个角色很难

■什么是Agent？充要条件是什么？

■规定一个角色集很难，经常需要细分

■例如：同为INSTRUMENT，也存在差异，如下，一种工具可做主语，另一种不可，所以，是否需要进一步细分？

可做主语	厨师用开瓶器打开罐头盖。	√
	这种新型开瓶器能开各种酒瓶盖。	√
不可做主语	中国人用筷子吃饭。	√
	*筷子吃饭	✗



■解决方法：

■1)更一般化的语义角色：所有词

- 例如：PROTO-AGENT and PROTO-PATIENT

- 没有充要定义，原型相似性，满足AGENT的条件越多就越可能作为

■2)更具体的语义角色：每个词/一组词

- 分别为不同的动词或不同类动词定义语义角色

■相关资源：

- PropBank同时使用 proto-roles以及动词特定的语义角色

- FrameNet使用frame.

- PropBank(Proposition Bank): 标注了语义角色的句子资源
- 每个动词的每一个义项有一个角色集合，但是只给出编号，没有名字： Arg0, Arg1, Arg2, ...
- Arg0: PROTO-AGENT
- Arg1: PROTO-PATIENT
- Arg2: often the benefactive, instrument, attribute, or end state
- Arg3: start point, benefactive, instrument, or attribute
- Arg4: the end point.

■例：agree.01

- Arg0: Agree
- Arg1: Proposition
- Arg2: Other entity agreeing
- Ex1: [Arg0 The group] agreed [Arg1 it wouldn't make an offer].
- Ex2: [ArgM-TMP Usually] [Arg0 John] agrees [Arg2 with Mary] [Arg1 on everything].

■例：fall.01

- Arg1: Logical subject, patient, thing falling
- Arg2: Extent, amount fallen
- Arg3: start point
- Arg4: end point, end state of arg1
- Ex1: [Arg1 Sales] fell [Arg4 to \$25 million] [Arg3 from \$27 million].
- Ex2: [Arg1 The average junk bond] fell [Arg2 by 4.2%].

■NomBank

■名词性谓词

■例：

■agreement in Apple's agreement with IBM

■Apple: Arg0

■IBM: Arg2

- FrameNet：一组词
 - Frame：一套语义角色、使用该语义集的词集
- 例如 frame: change_position_on_a_scale
 - 语义角色集
 - 核心角色：ATTRIBUTE、DIFFERENCE...
 - 非核心角色：DURATION、SPEED...
 - 词集
 - 动词：dwindle、move、soar、advance、climb、edge...
 - 名词：escalation、explosion、fall...
 - 副词：increasingly

■例句

- [ITEM Oil] rose [ATTRIBUTE in price] [DIFFERENCE by 2%].
- [ITEM It] has increased [FINAL STATE to having them 1 day a month].
- [ITEM Microsoft shares] fell [FINAL VALUE to 7 5/8].
- [ITEM Colon cancer incidence] fell [DIFFERENCE by 50%] [GROUP among men].
- a steady increase [INITIAL VALUE from 9.5] [FINAL VALUE to 14.3] [ITEM in dividends]
- a [DIFFERENCE 5%] [ITEM dividend] increase...

- SRL(Semantic role labeling)
 - 自动发现动词论元的语义角色
- 角色集：FrameNet and PropBank或其他
- 标注方法：分类、序标

■ 分类方法

- 1) 对句子进行句法分析，得到谓词及其论元
- 2) 对每个论元进行分类，指派语义角色

■ 问题

- 1) 依赖句法分析的结果
- 2) 论元分类性能依赖设计的特征
 - predicate、phrase type、headword、...
- 3) 论元各自独立分类，可能存在冲突
 - 每个分类取多个类标后再进行全局优化

■序列标注方法

■标注集：{B-ARG0、I-ARG0...}

■标注模型：HMM\CRF\LSTM\...

■分析越深入，标注数据越困难

■真实问题：小规模标注数据下的求解

■NLP任务共性问题

■语义角色的应用

■句子语义的等价性判定

- 形式不同的句子具有相同的浅层语义表示

- Jim gave the book to Tom = The book was given to Tom by Jim

■句子生成：复述生成

- 作为句子的中间表达以帮助产生不同的句子

- GIVE(Agent=Jim; theme=the book; Goal=Tom)

■句子推理以回答问题

- Jim gave the book to Tom.

- Was Jim given the book?

■信息抽取：事件挖掘…

■句子语义表示与分析

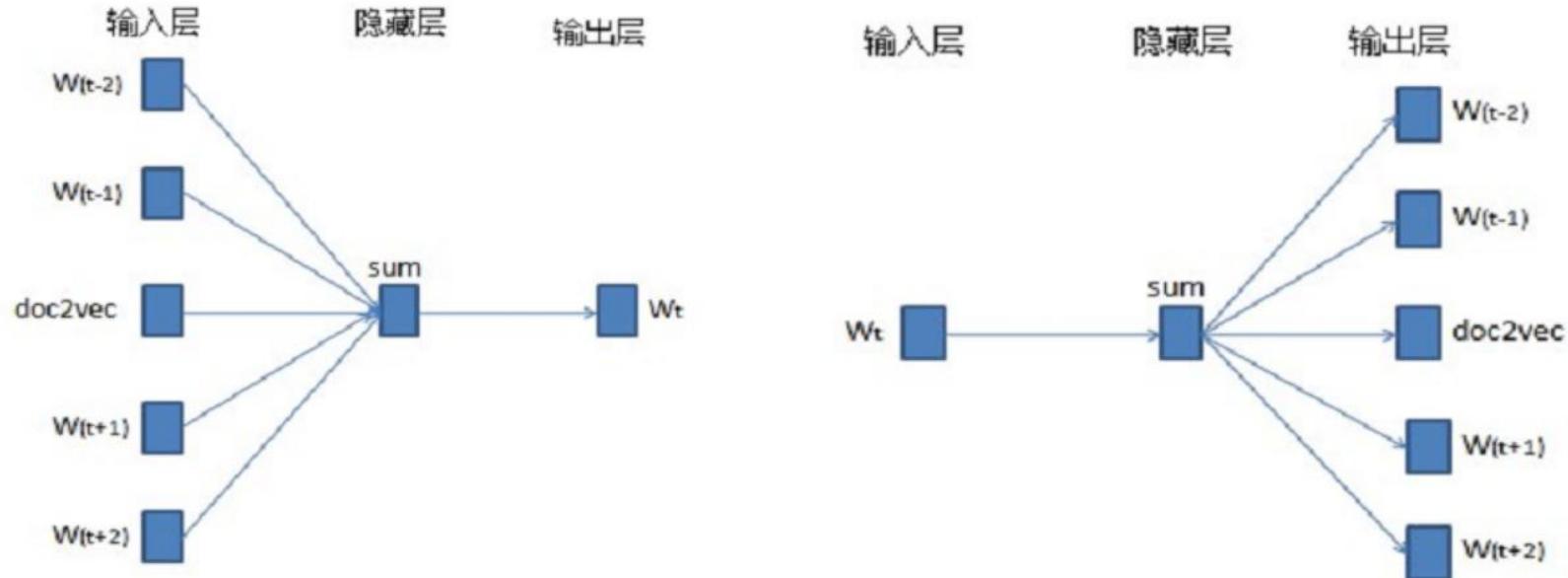
- 基于符号的方法

- 基于数值的方法

- 基于数值的句子语义表示与分析方法
 - 句子语义表示和语义分析的融合
 - 直接基于词表示构建
- 基于循环神经网络
- 基于递归神经网络
- 基于注意力网络

■ 在词向量学习模型中加入句子向量表示

■ Doc2vec (Paragraph2vec)



■直接基于词向量的方案

- 在学习到词表示后，简单地对句子中的所有词语取平均、求和或进一步的变换操作。
- DAN(Deep Averaging Networks): 在对所有词语取平均后，在上面加上几层神经网络进行再编码。
- 优点：计算速度较快
- 缺点：忽略了句子中的词序：他打我=我打他
- 利用能建模词序的RNN模型来进行句子编码

■ 基于数值的句子语义表示与分析方法

- 句子语义表示和语义分析的融合

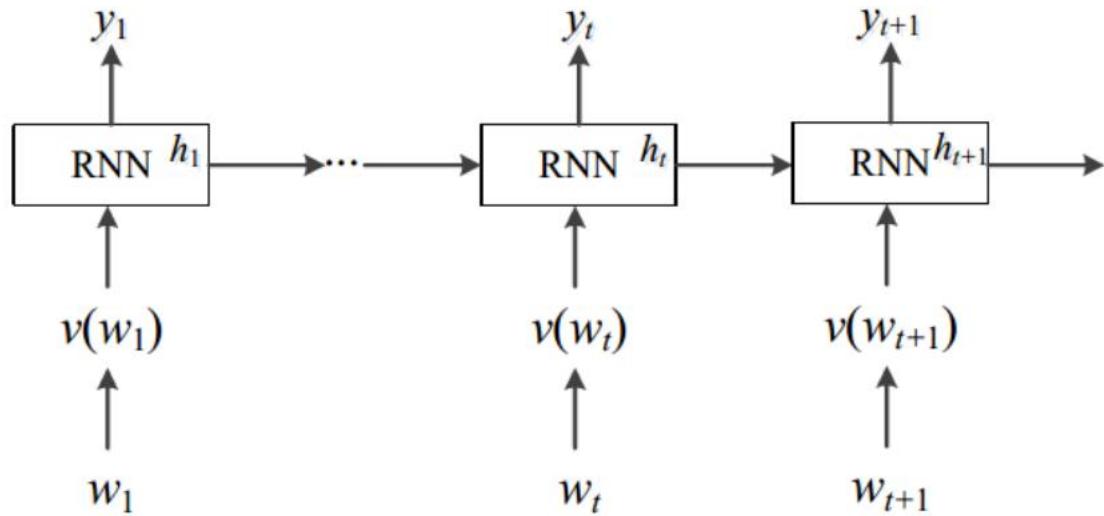
- 直接基于词表示构建

- 基于循环神经网络

- 基于递归神经网络

- 基于注意力网络

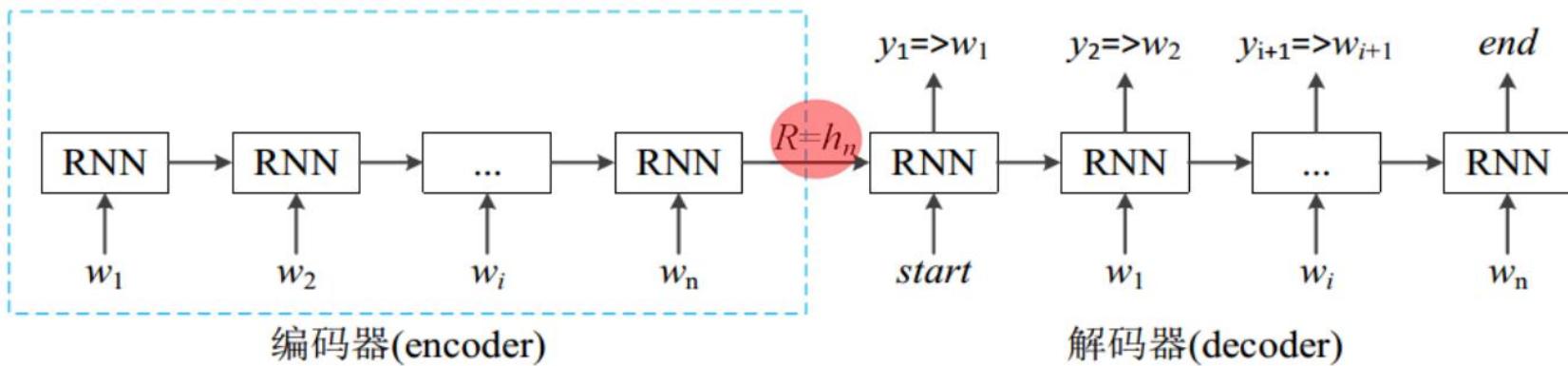
■ 基于RNN的句子表示



- 以 h_{t+1} 或 y_{t+1} 表示句子 (w_1, \dots, w_{t+1})
- 问题：如何获得合适的RNN的参数？

■ 基于自编码器的方法，基本想法：

- 如果输入句子的编码可以较好地恢复输入，则该编码是一个好的编码
- 实现：基于RNN的Seq2Seq自编码器(AutoEncoder)



■ 如果R能较好地恢复输入，则R是一个好的编码

■ 自恢复的要求提供了天然的训练模型的监督信息

■RNN编码器：

■ $R = F(w_1, \dots, w_n)$ 同前， $R = h_n$ ， RNN编码器的最后隐层输出

■RNN解码器：

■ $h_1 = \tanh(UR + Wstart + b)$, $y_1 = softmax(Vh_1)$

■.....

■ $h_i = \tanh(Uh_{i-1} + Wx_i + b)$, $y_i = softmax(Vh_i)$

■.....

■自(恢复)监督信息

■ $y_i = softmax(Vh_i) \rightarrow w_i$ (GT)

■交叉熵损失： y_i 与 $(0, \dots, 0, 1, 0, \dots, 0)$ 的交叉熵 (第*i*个词为1)

■自编码是面向自恢复任务的编码

- 自恢复无需额外的监督信息：自监督，与其他任务无关
- Reminding：Skip-Gram是面向上下文词预测当前词学习词编码，
无需额外的监督信息(自监督)，与其他任务无关
- 后续自监督预训练任务

■还可以面向其他更特定高层任务进行句子编码，即获得取得该任务最好性能的句子编码，不同任务可能导致不同的句子编码

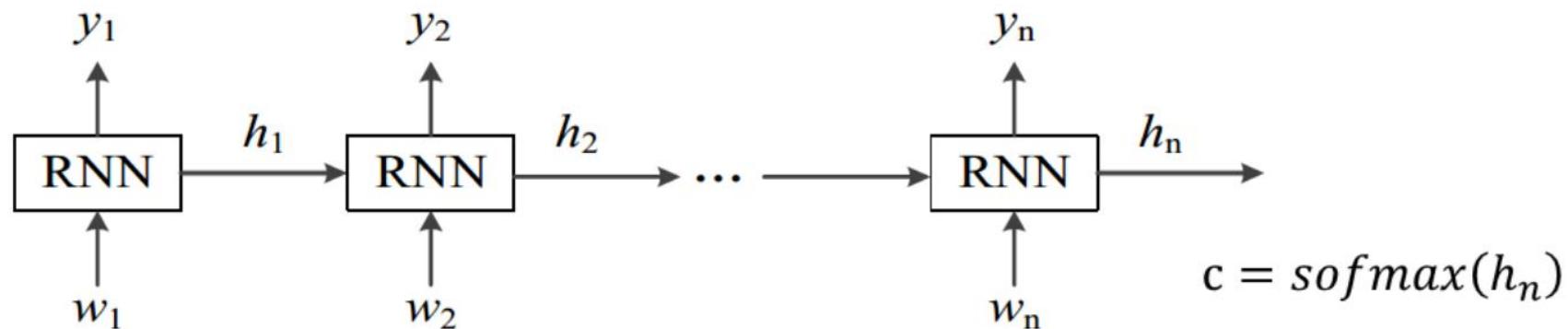
- 分类任务
- 变换任务
- ...

■面向分类任务的句子编码：

- 用RNN获得的句子编码来进行分类，基于极小化分类误差来优化RNN参数，获得更好的面向任务的句子编码

■用RNN的什么作为句子编码？

- 1、最后的隐层状态 h_n 作为编码做分类



■面向分类任务的句子编码：

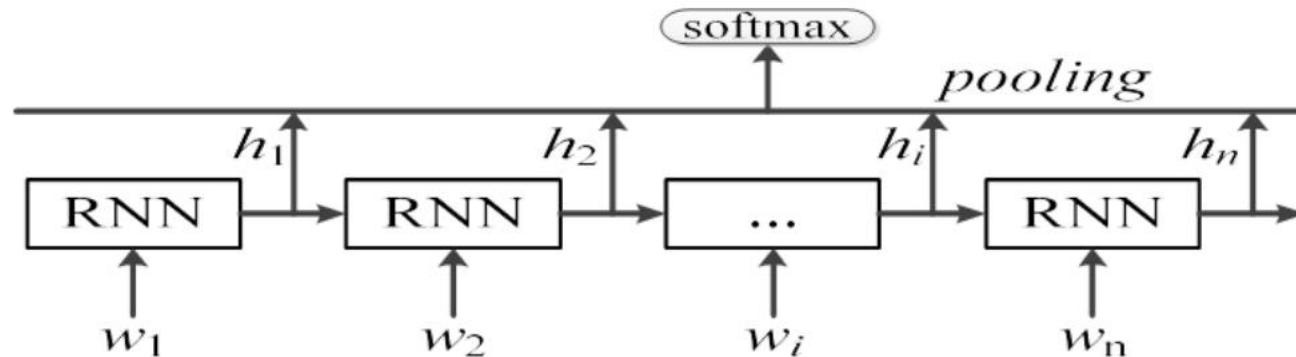
■用RNN获得的句子编码来进行分类，基于极小化分类误差来优化RNN参数，获得更好的面向任务的句子编码

■用RNN的什么作为句子编码？

■2、用所有的隐层输出 $[h_1, h_2, \dots, h_n]$ 进行pooling操作

■Ave-pooling: $o = \text{Average}(h_1, h_2, \dots, h_n)$

■Max-pooling: $o = \text{maximum}(h_1, h_2, \dots, h_n)$

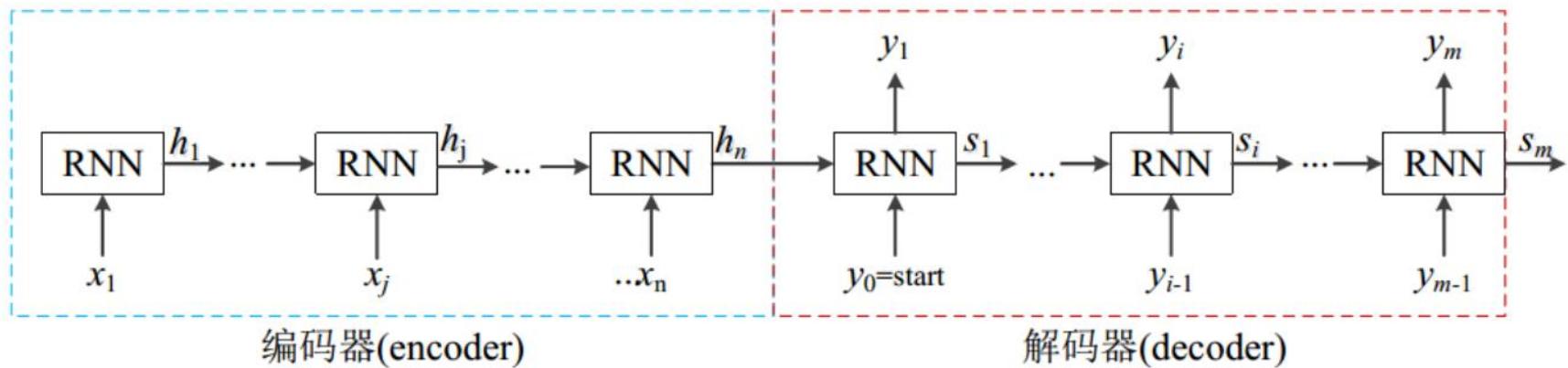


■面向变换任务的句子编码

■自编码: $x_1, \dots, x_n \rightarrow x_1, \dots, x_n$

■扩 展: $x_1, \dots, x_n \rightarrow y_1, \dots, y_m$ 把一个串转换为另一个串

■即得到更一般的(自回归)seq2seq编码-解码模型(输入n, 输出m)



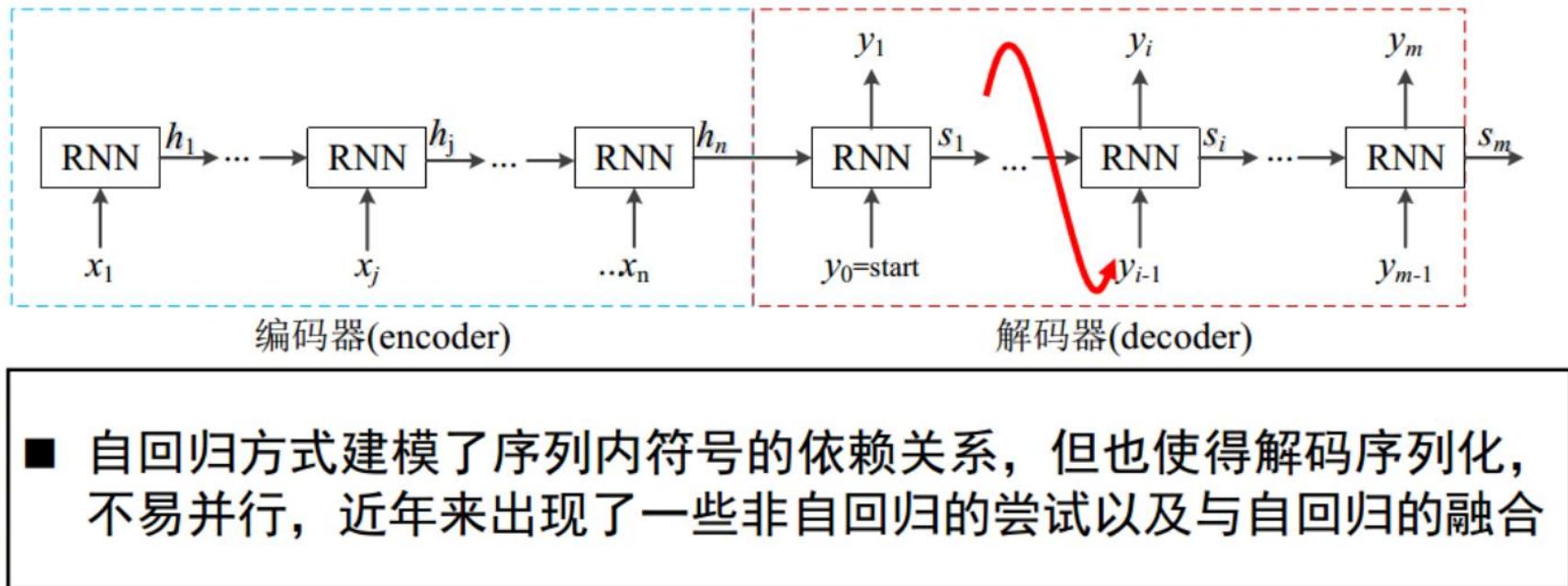
■ $h_n = F(x_1, \dots, x_n)$: $h_j = \tan(W_e x_j + U_e h_{j-1})$,

■ $y_i = \text{softmax}(V_d s_i)$, $s_i = \tan(W_d y_i + U_d s_{i-1})$, $s_{i-1} = \tan(W_d y_{i-1} + U_d s_{i-2}) \dots$,

■ $\rightarrow y_i = G(s_i, y_{i-1}, s_{i-1}, y_{i-2}, \dots, s_1, y_0, h_n)$

■面向变换任务的句子编码

- 自编码: $x_1, \dots, x_n \rightarrow x_1, \dots, x_n$
- 扩 展: $x_1, \dots, x_n \rightarrow y_1, \dots, y_m$ 把一个串转换为另一个串
- 即得到更一般的(自回归)seq2seq编码-解码模型(输入n, 输出m)



- 自回归方式建模了序列内符号的依赖关系，但也使得解码序列化，不易并行，近年来出现了一些非自回归的尝试以及与自回归的融合

■ 基于数值的句子语义表示与分析方法

- 句子语义表示和语义分析的融合

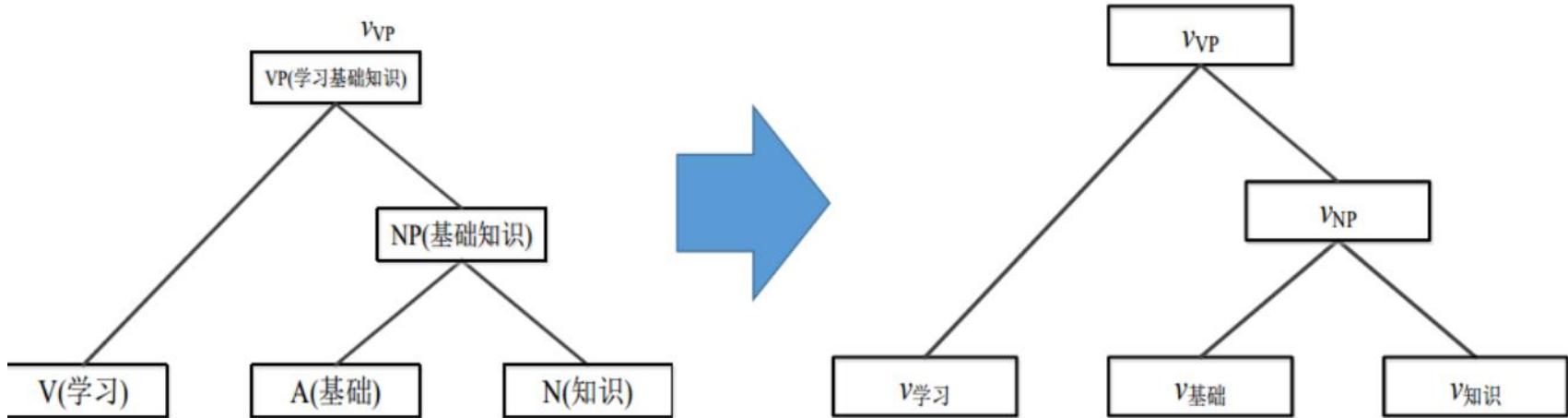
- 直接基于词表示构建

- 基于循环神经网络

- 基于递归神经网络

- 基于注意力网络

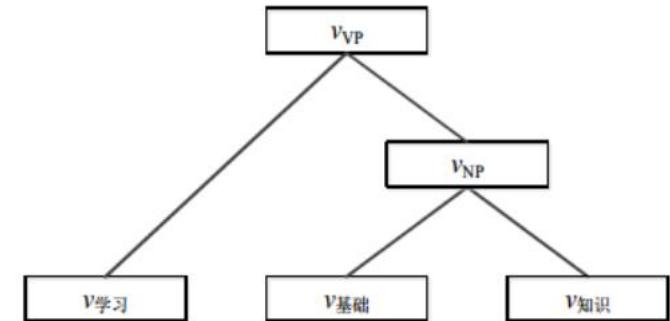
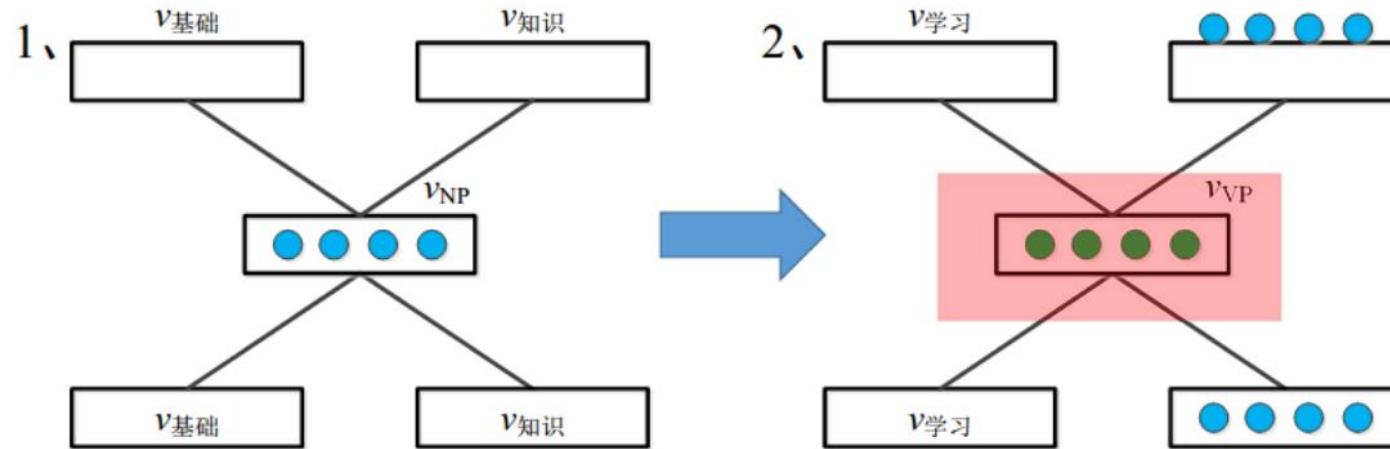
- 前述句子有句法结构，构建句子表示时能否利用
- 如果有句法树：



- 问题：如何训练获得 v_{VP} ？
- 一种可能：基于递归神经网络(Recursive Neural Networks, RecNN)

[RecursiveNN for sentence representation cs224n-2019-notes09]

■递归自编码器(Rec AutoEncoder)



■问题：

- 依赖句法结构：需要现有句子的句法树，基于句法树来构建RecNN
- 非终止符的词汇化表示：抽象单元表示是否存在，可能需要词汇化表示？

- 基于数值的句子语义表示与分析方法
 - 句子语义表示和语义分析的融合
 - 直接基于词表示构建
 - 基于循环神经网络
 - 基于递归神经网络
 - 基于注意力网络

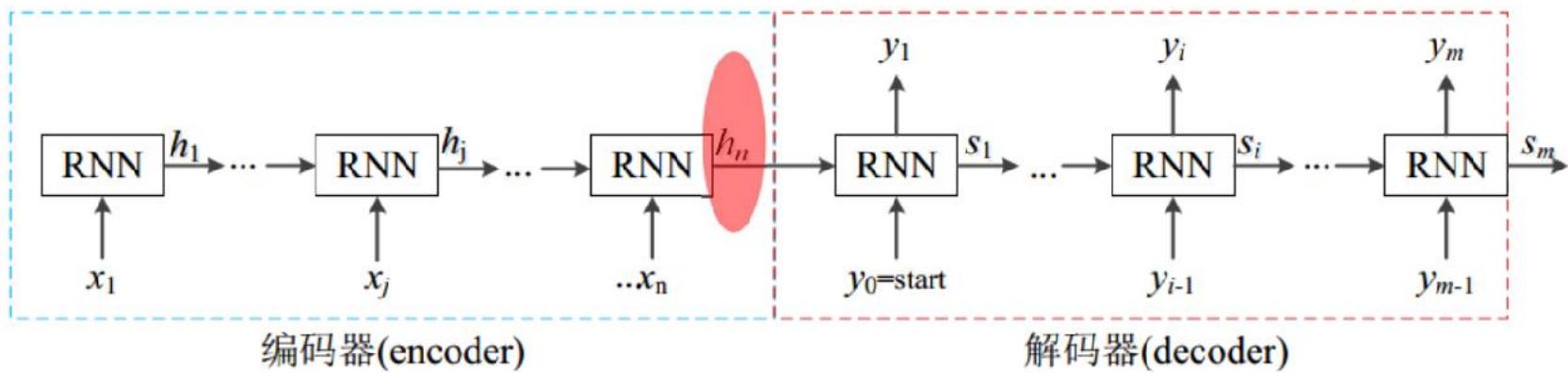
■前述面向变换任务的Seq2Seq编码-解码模型应用广泛

■用于机器翻译

■ Ilya Sutskever, Oriol Vinyals, Quoc V. Le. Sequence to Sequence Learning with Neural Networks. NIPS2014.

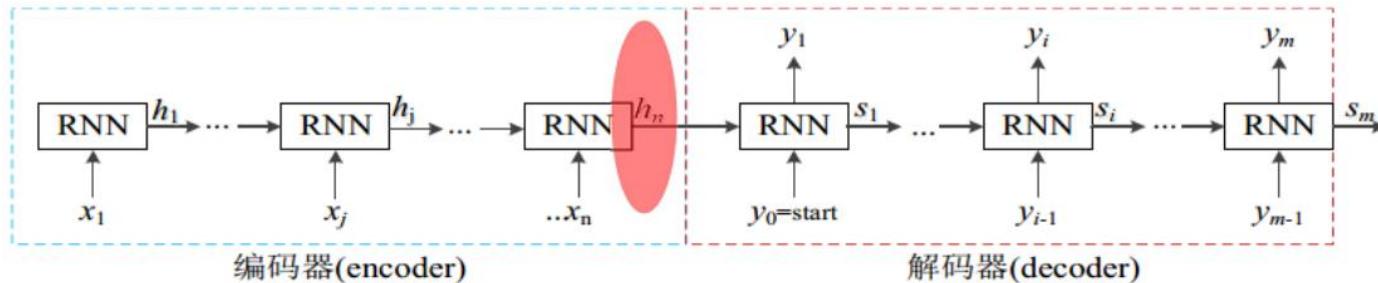
■用于应答任务(Chatbot)

■ Shang, L., Lu, Z., & Li, H.. Neural Responding Machine for Short-Text Conversation. ACL-IJCNLP2015.



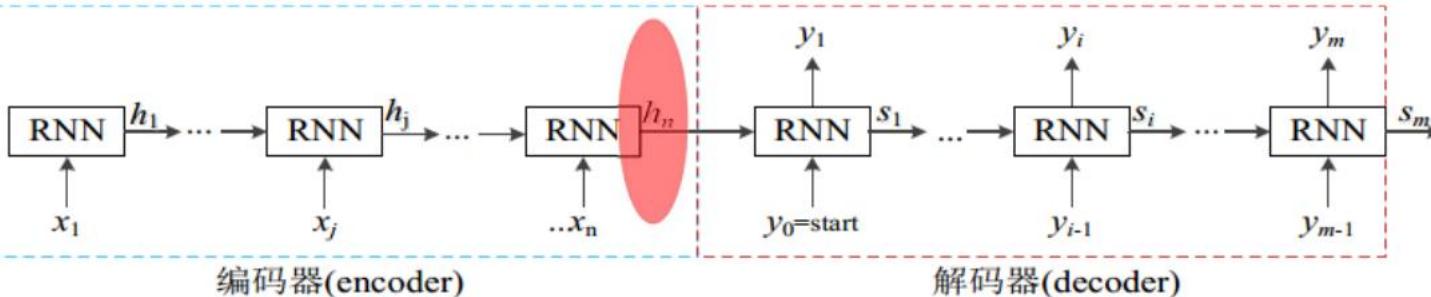
■ 编码器部分得到句子的编码，用于后面解码，但是存在一个重要的问题

■问题：以翻译为例



- $y_i = G(s_i, y_{i-1}, s_{i-1}, y_{i-2}, \dots, s_1, y_0, h_n)$
- 也就是说：解码每个单词时采用的输入句子编码是一样的 →
- 也就是说：生成每个译词时采用的原句信息是一样的 →
- 也就是说：生成不同译词时原句中不同词的作用是保持不变的？是否合理？
- 此外，随着句子长度增加， h_n 更难编码好句子信息，Seq2Seq模型性能下降

■ 上述seq2seq模型的问题：以翻译为例



■ 不太合理：生成某个译词时，其对应的原词作用会更大，也就是说：
生成不同译词时，利用的原句信息应该有所不同（差异化的句子编码）

■ 例如：I have a book → 我有一本书

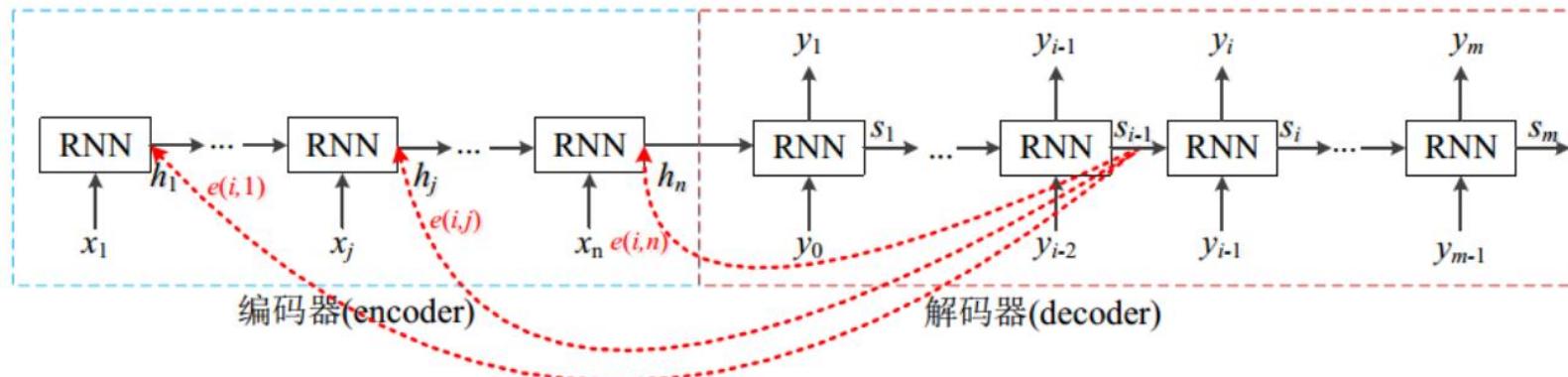
■ 生成译词“我”时，“I”应该更有价值，得到更多关注，而其他词作用小

■ 生成译词“有”时，“have”应该更有价值，得到更多关注，而其他词作用小

■ 如何在生成不同译词时体现原句各词的不同作用？注意力机制的引入：

[Bahdanau2015ICLR] Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. ICLR2015.

■ 在 seq2seq 模型中引入注意力机制(Attention Mechanism)

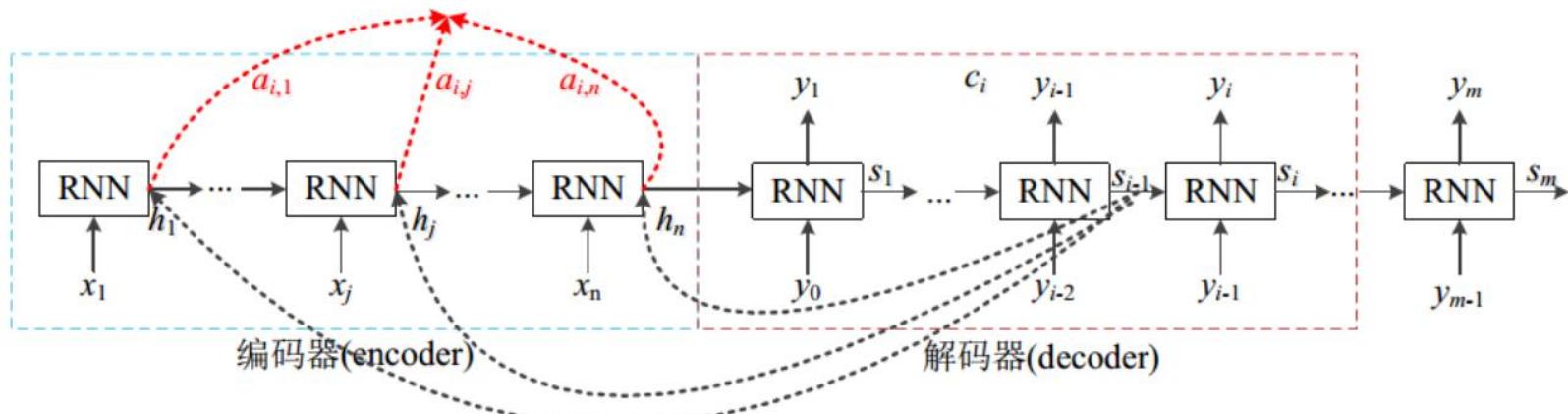


■ 注意力机制：在解码的 i 时刻，

■ 1) 计算： $e(i, j) = s_{i-1} \cdot h_j$

- 每个词的注意力得分
 - 不同词在解码当前词时的不同作用
- 为何如此计算?
 - 可以探索不同的选择：注意力打分函数

■ 在seq2seq模型中引入注意力机制(Attention Mechanism)



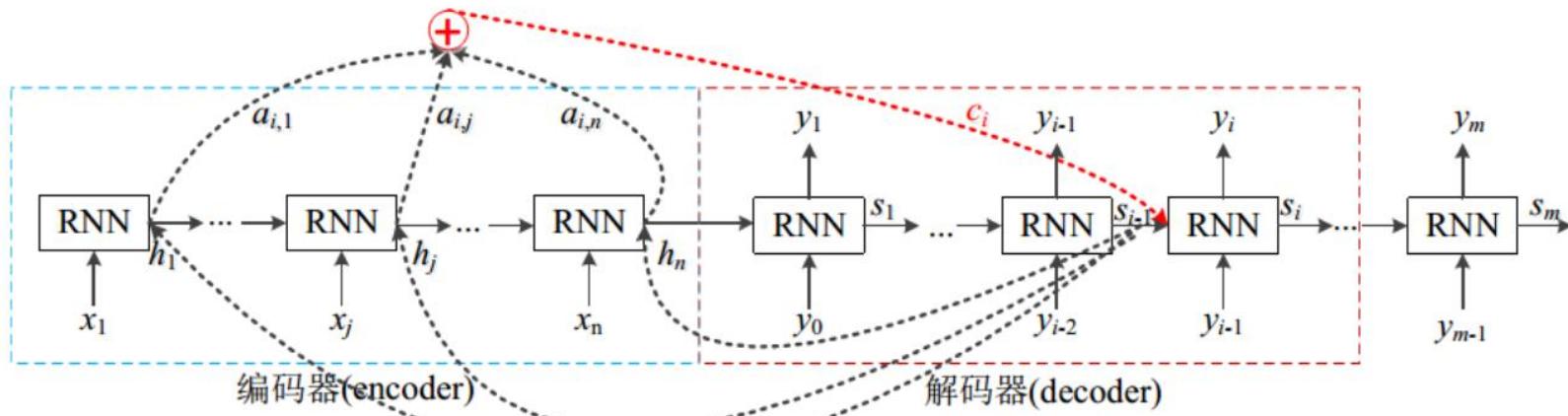
■ 注意力机制：在解码的*i*时刻，

■ 1) 计算： $e(i, j) = s_{i-1} \cdot h_j$

■ 2) softmax归一化： $a_{i,j} = \frac{\exp(e(i,j))}{\sum_{k=1}^n \exp(e(i,k))}$

- 解码 y_i 时对词 x_j 的关注程度
- 词 x_j 对解码 y_i 的影响程度
- 注意力是一个分布
- 相斥(注意资源有限)

■ 在 seq2seq 模型中引入注意力机制(Attention Mechanism)



■ 注意力机制：在解码的 i 时刻，

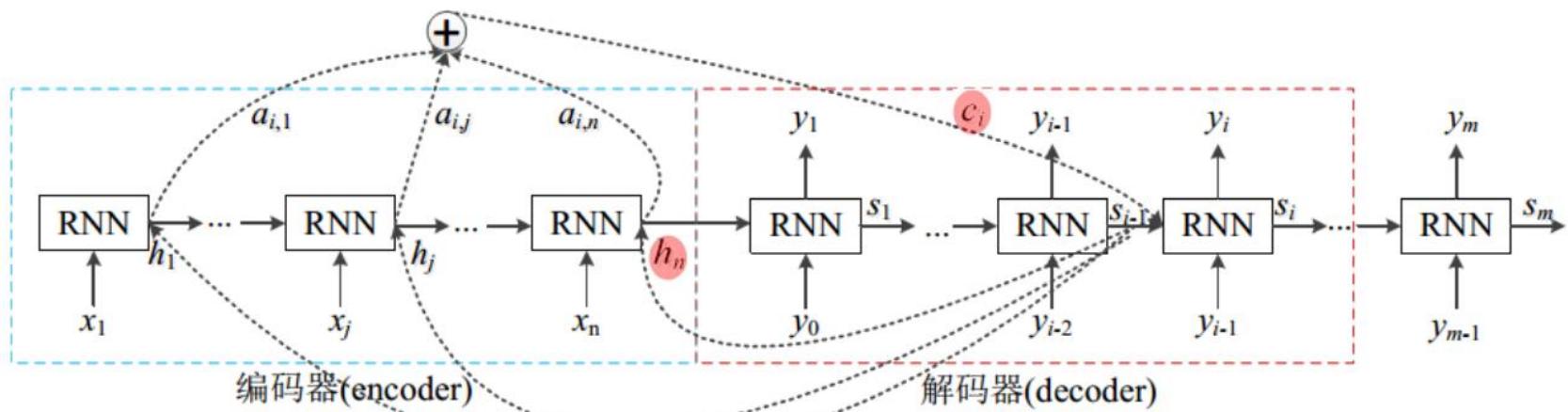
■ 1) 计算： $e(i, j) = s_{i-1} \cdot h_j$

■ 2) softmax 归一化： $a_{i,j} = \frac{\exp(e(i,j))}{\sum_{k=1}^n \exp(e(i,k))}$

■ 3) 加权求和： $c_i = \sum_{j=1}^n a_{i,j} h_j$

■ 综合各个词的作用

■ 在 seq2seq 模型中引入注意力机制(Attention Mechanism)



■ 注意力机制：在解码的 i 时刻，

■ 1) 计算： $e(i, j) = s_{i-1} \cdot h_j$

■ 2) softmax 归一化： $a_{i,j} = \frac{\exp(e(i,j))}{\sum_{k=1}^n \exp(e(i,k))}$

■ 3) 加权求和： $c_i = \sum_{j=1}^n a_{i,j} h_j$

解码： $y_i = g(y_{i-1}, s_i, h_n)$



解码： $y_i = f(y_{i-1}, s_i, c_i)$

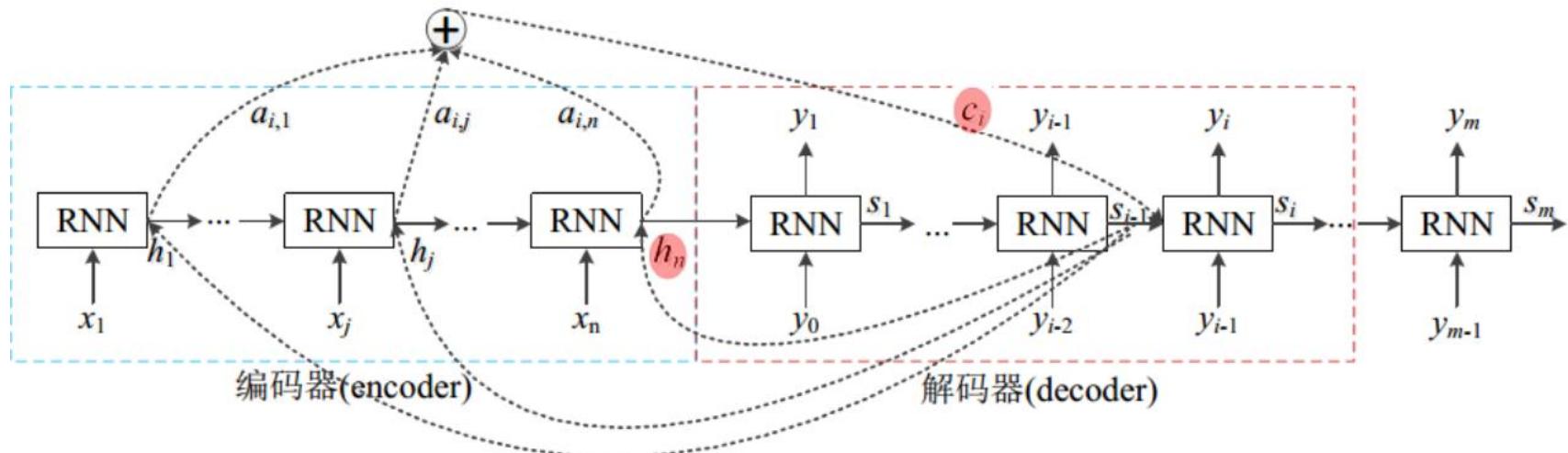
■ 在 seq2seq 模型中引入注意力机制(Attention Mechanism)

■ $e(i, j) = s_{i-1} \cdot h_j$: 与 s_{i-1} 越相似的 h_j 获得越大的注意力，是 c_i 中越主要的部分，对解码 y_i 的作用也越大

■ 例: $j: h_1 = (0.1, 0.2, 0.6), h_2 = (0.8, 0.2, 0.4), h_3 = (0.4, 0.5, 0.2)$

■ $i = 2: s_{2-1} = s_1 = (0.9, 0.3, 0.5)$

■ $e(2,1) = s_1 \cdot h_1 = 0.45, e(2,2) = s_1 \cdot h_2 = 0.98, e(2,3) = s_1 \cdot h_3 = 0.61$



■ 在 seq2seq 模型中引入注意力机制(Attention Mechanism)

■ 注意力 $a_{i,j}$ 可视化：

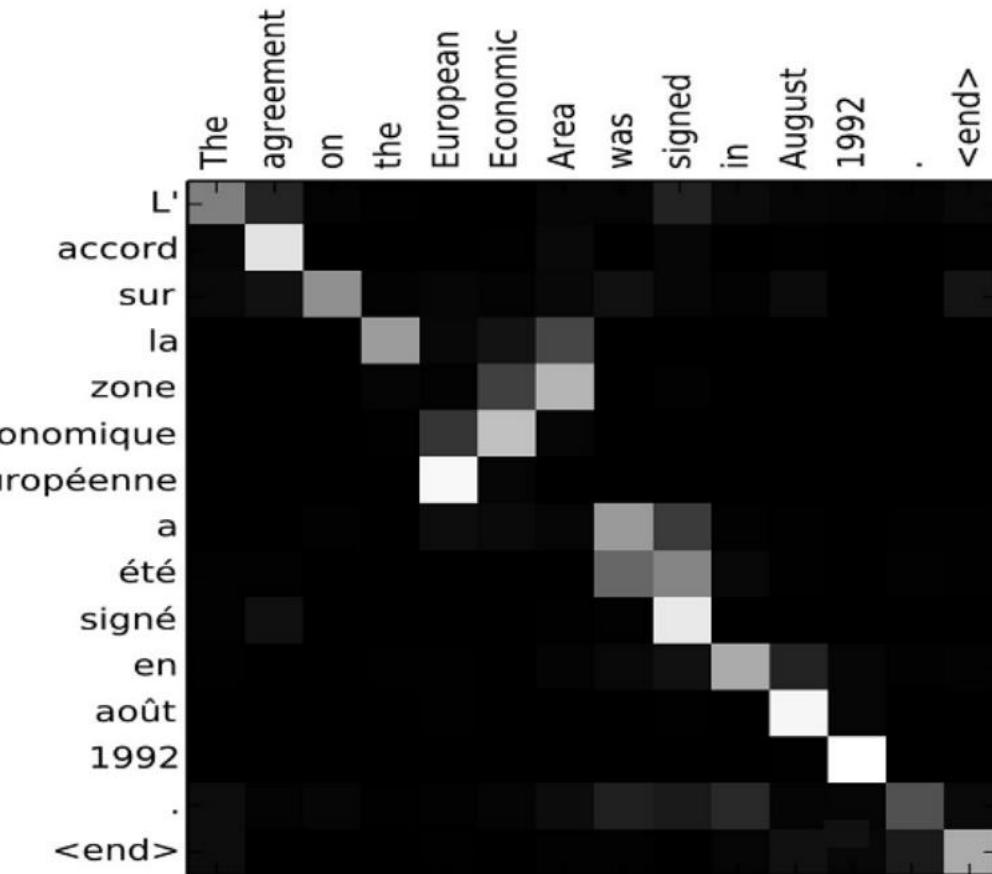
- 解码目标语言词时对源语言上词的注意力值，
- 颜色越浅注意力值越大

■ 例如

- 解码 L' 时 The 上的注意力值最大，其余较小；

■ 实现了：

- 解码不同词时句子中不同词的作用是不同的
- 边解码边对齐互译部分



注意力机制的发展

- 注意力网络
- 基于注意力网络的Seq2Seq模型-Transformer
- 基于Transformer的预训练语言模型-BERT/GPT

■ 认知神经科学中的注意(Attention)

■ 在认知过程只加工特定信息而忽视另一些信息的能力，是生物能够完成认知任务的原因(结果)。

■ 鸡尾酒会效应(自主注意)

■ 关键特点

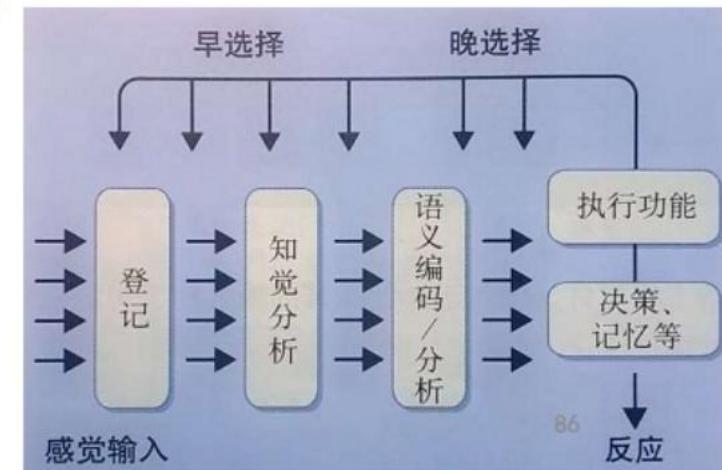
■ 选择性(基于情景进行选择性加工)

■ 为何选择：处理容量有限

■ 选择的时间

■ 早选择

■ 晚选择



■两类注意

■自主注意(有意注意)

- 自上而下:额叶发起

- 目标驱动:为着特定期望的目标选择接受输入

- 例如:视觉搜索

■自动注意(反射性注意)

- 自下而上:感知端发起

- 刺激驱动:为特定刺激(突出的或意外的)特异化的接受能力

- 例如:注意捕获(人脸、雷等异常声音)

■二者各有作用，在日常混合工作

注意力机制(Attention Mechanism)

- 自主注意(有目标地选择注意对象):
 - 软注意(Soft attention)
 - 前述seq2seq模型中引入的注意力就是一种软注意力
 - 硬注意(Hard attention)
 - ...
- 自动注意(注意对象之间的比较) :
 - 自注意(Self attention)
 - ...

软注意力 (Soft attention)

■由 q 引导的对 X 的软注意力两要素：

■ q 为查询向量(注意目标), $q \in \mathbb{R}^{d_1 \times 1}$

■ $X = [x_1, \dots, x_N]$ 为 N 个输入信息(注意对象), $x_i \in \mathbb{R}^{d_1 \times 1}$

■计算软注意力的两步骤：

■注意力分布计算:

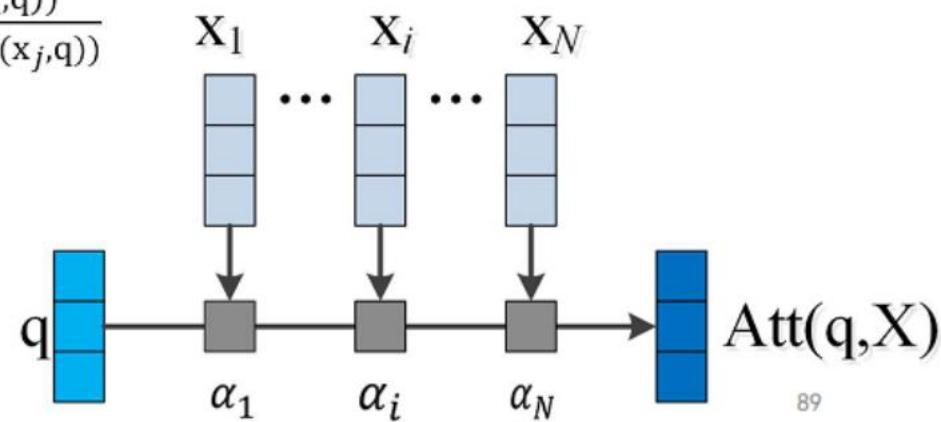
$$\alpha_i = softmax(s(x_i, q)) = \frac{\exp(s(x_i, q))}{\sum_{j=1}^N \exp(s(x_j, q))}$$

$$\sum_{i=1}^N \alpha_i = 1$$

■ $s(x_i, q)$ 为注意力打分函数(下页)

■注意力输出为

$$Att(q, X) = \sum_{i=1}^N \alpha_i x_i$$



■ 几种常见的打分函数

名称	$s(x_i, q)$ 函数
线性	$v^T \tanh(Wx_i + Uq)$
点积	$x_i^T q$
缩放点积	$x_i^T q / \sqrt{d}$ (d为向量维数)
双线性	$x_i^T Wq$

■ 点积

- 计算相对简单
- 但是方差大(d)
 - 设 q 和 x_i 是均值为0, 方差为1的独立高斯随机变量, 则 $\sum_{i=1}^d qx_i$ 均值为0, 方差为 d ;
 - 这样softmax函数比较容易进入饱和区域。
- 为抵消维度的影响
- → 缩放点积 (后文均假设用此)

■ 采用缩放点积时, 上页: $\text{Att}(q, X) = X \text{softmax}(X^T q / \sqrt{d_1})$

■ 注: 当 q 、 x_i 用行向量时, $\text{Att}(q, X) = \text{softmax}(q X^T / \sqrt{d_1}) X$

■硬注意力 (Hard attention)

■第一步：同样方式获得 α_i

■第二步：基于 α_i 只关注某一个 x_i ，两种方式：

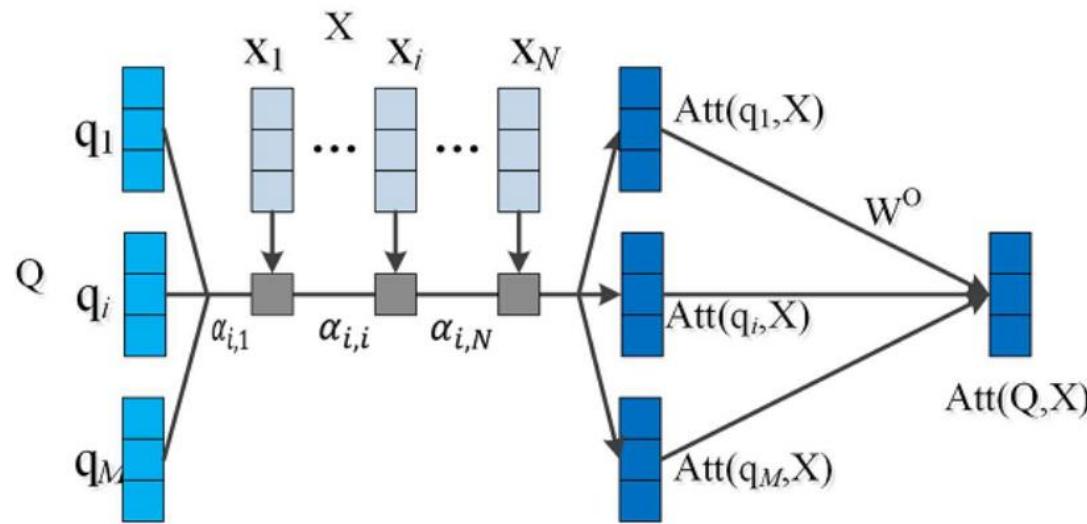
■1) 基于注意力分布 $(\alpha_1, \dots, \alpha_N)$ 采样获得特定 x_i

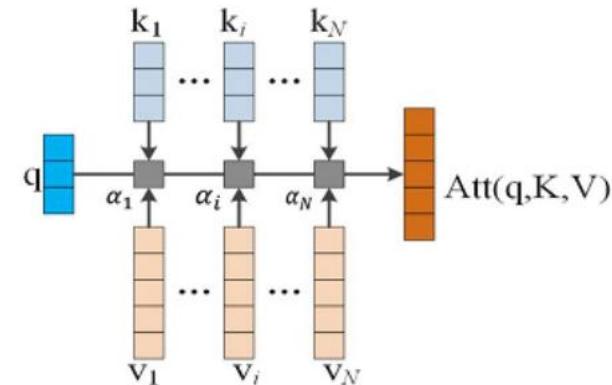
■2) 取最大，即：

■ $\text{Att}(q, X) = x_i$, 其中 $i = \operatorname{argmax}_{i=1,\dots,N} \alpha_i$

■多头注意力(Multi-Head Attention, MHA)

- 用多个查询向量 $Q = [q_1, \dots, q_M]$ ($q_i \in \mathbb{R}^{d_1 \times 1}$)分别引导注意力后拼接:
- $\text{Att}(Q, X) = \text{Att}(q_1, X) \oplus \dots \oplus \text{Att}(q_M, X) \in \mathbb{R}^{M \times d_1 \times 1}$
- 之后再通过变换映射到 $R^{d_1 \times 1}$:
- $\text{Att}(Q, X) = W^O \text{Att}(Q, X)$ 其中 $W^O \in \mathbb{R}^{d_1 \times M \times d_1}$





■键值对注意力(注意力的一般形式)

■两要素变为三要素

■查询 $q \in \mathbb{R}^{d_1 \times 1}$ ；

■输入信息为键-值对： $x_i = (k_i, v_i), k_i \in \mathbb{R}^{d_1 \times 1}, v_i \in \mathbb{R}^{d_2 \times 1}$ ，即
 $X=(K, V)=[(k_1, v_1), \dots, (k_N, v_N)]$ 为N个输入信息；

■注意力计算两步：

■用**键**来计算注意力分布： $\alpha_i = \frac{\exp(k_i^T q / \sqrt{d_1})}{\sum_j \exp(k_j^T q / \sqrt{d_1})}$

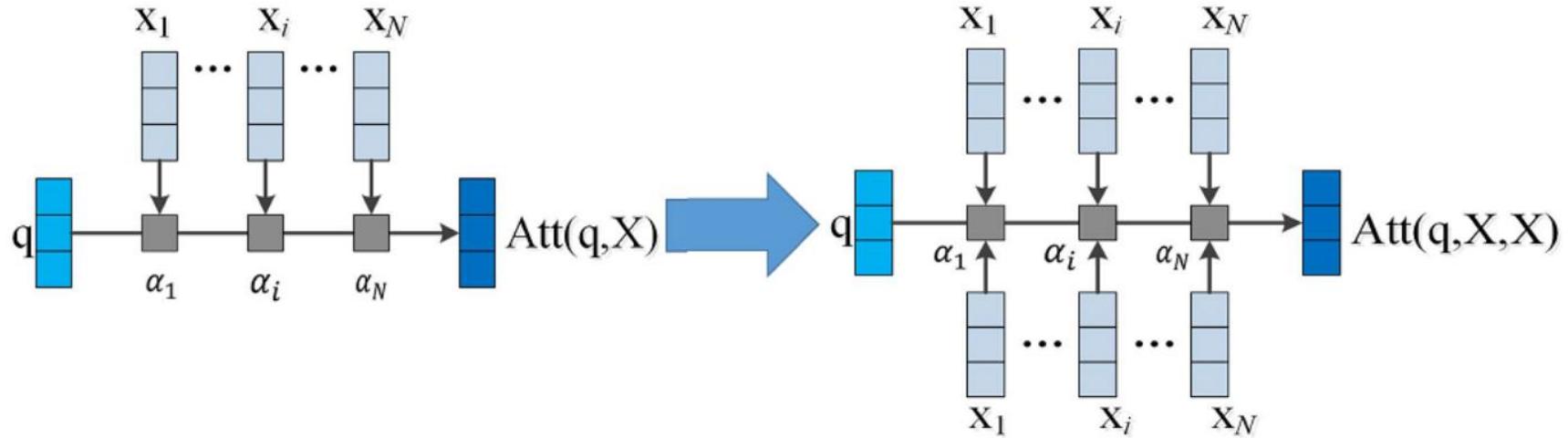
■在**值**上求和： $\text{Att}(q, K, V) = \sum_{i=1}^N \alpha_i v_i = V \text{softmax}\left(K^T q / \sqrt{d_1}\right)$

■ $\text{Att}(q, K, V) \in \mathbb{R}^{d_2}$ **注意力值的维度和值的维度一致**

■当 $K=V$ 时，回到前面的软注意力

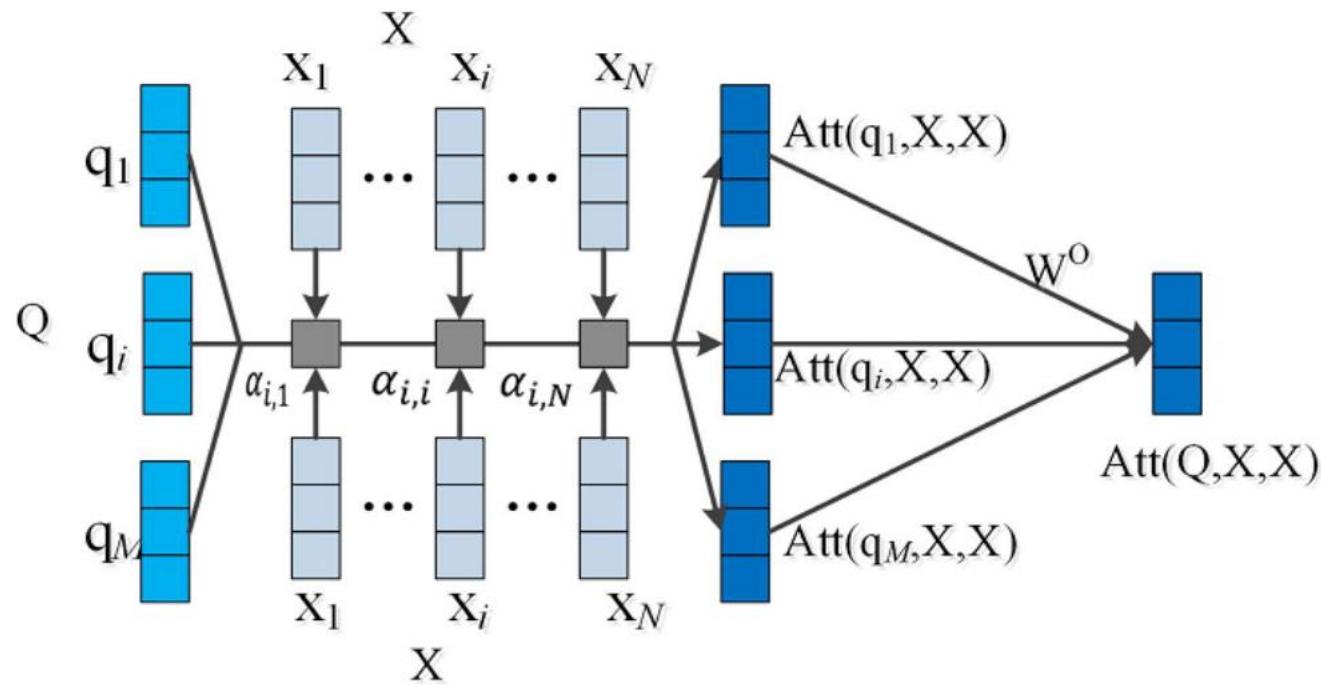
在键值注意力方式下看软注意力

$$K=V=X$$



■多头注意力(Multi-Head Attention, MHA)

■用QKV方式描述



■前述注意力均有一个查询q指引：

■关注输入X(K,V)，是一种自顶向下的方式。

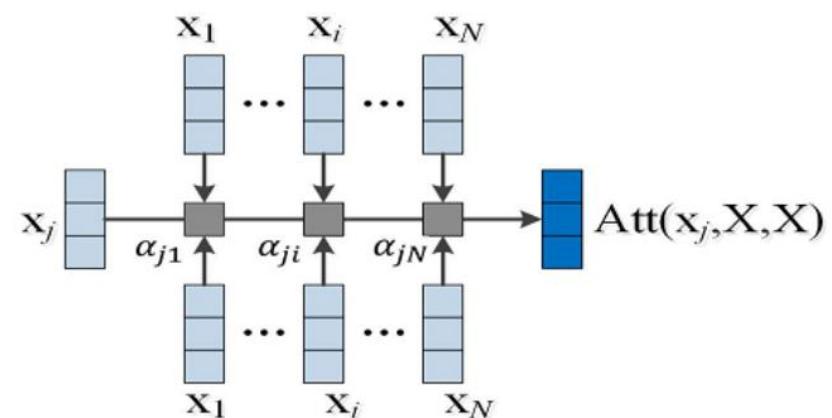
■自底向上，对输入自身进行注意力选择，如何？

■→自注意力(Self Attention, SA)

■输入： $X = [x_1, \dots, x_N] \in \mathbb{R}^{d_1 \times N}$

■套用前述qKV注意力的构建方式

■首先需要构建q，因为要对输入自身进行注意力选择，自然的想法是以输入自身为q



■ 计算 x_j 引导的自注意力

■ 用键来计算注意力分布

$$\alpha_{ji} = \frac{\exp(x_j, x_i)}{\sum_l \exp(x_j, x_l)},$$

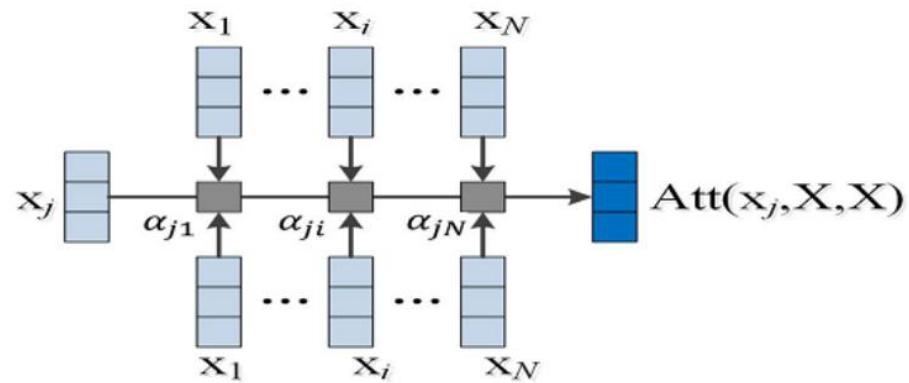
■ 这里为简化直接用了点积打分函数，一般要用缩放。

■ 在值上求和

$$\text{Att}(x_j, X, X) = \sum_{i=1}^N \alpha_{ji} x_i =$$

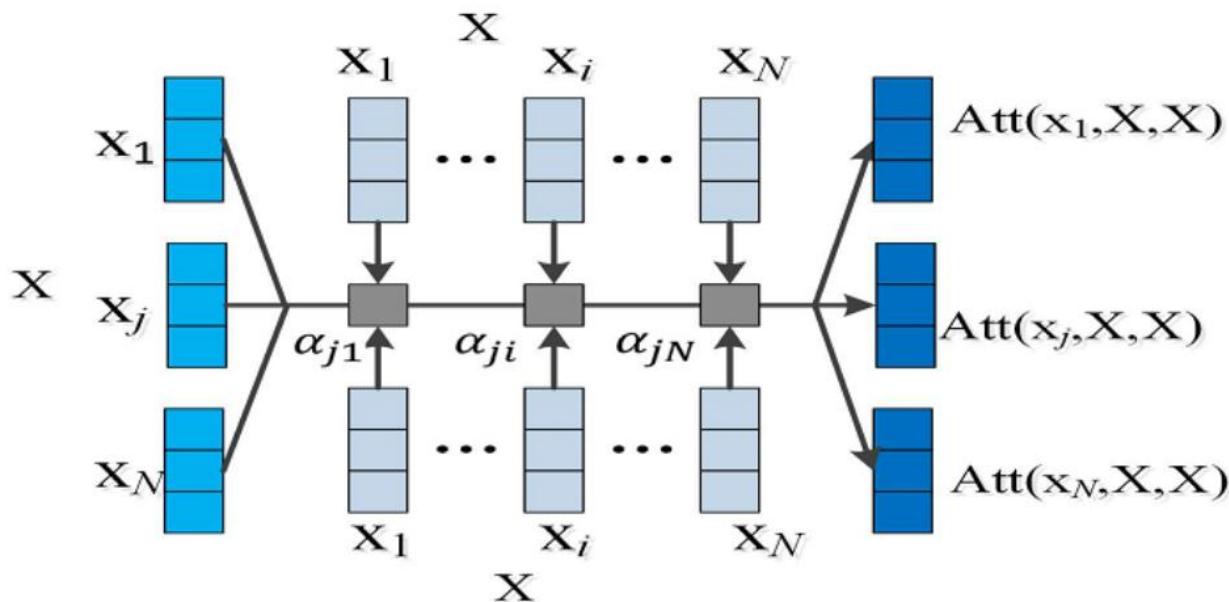
$$\sum_{i=1}^N \frac{\exp(x_j, x_i)}{\sum_l \exp(x_j, x_l)} x_i = X \text{ softmax}(X^T x_j)$$

■ 即：将 x_j 更新为 x_j 引导的自注意力 $\text{Att}(x_j, X, X)$

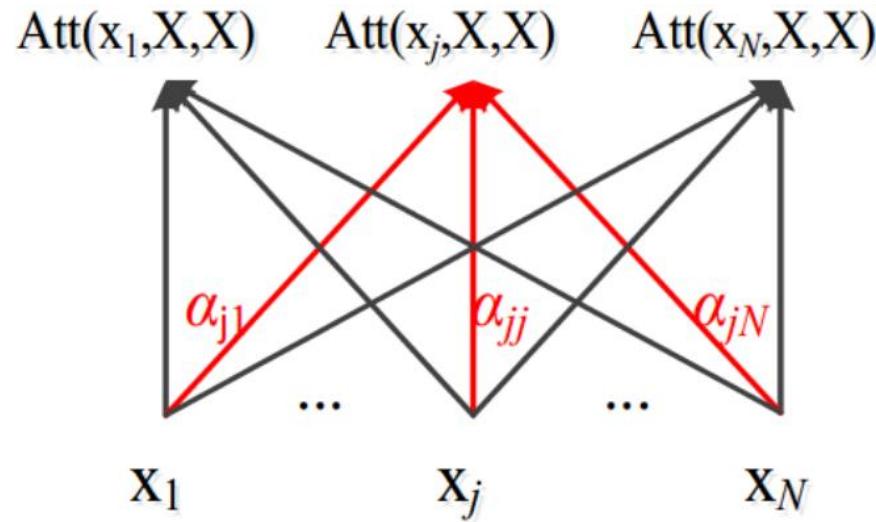


如果对每个 x_j 进行自注意力： (注意这和多头注意力不一样)

$$\text{Att}(x_j, X, X) \quad j=1, \dots, N \rightarrow \text{Att}(X, X, X) = X \text{ softmax}(X^T X)$$



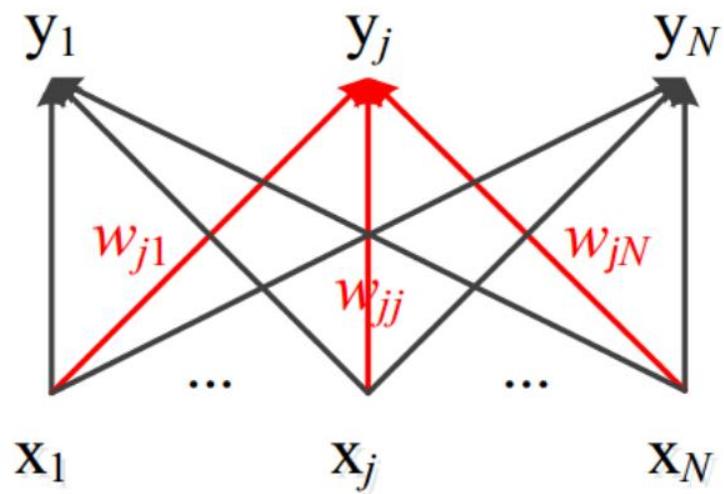
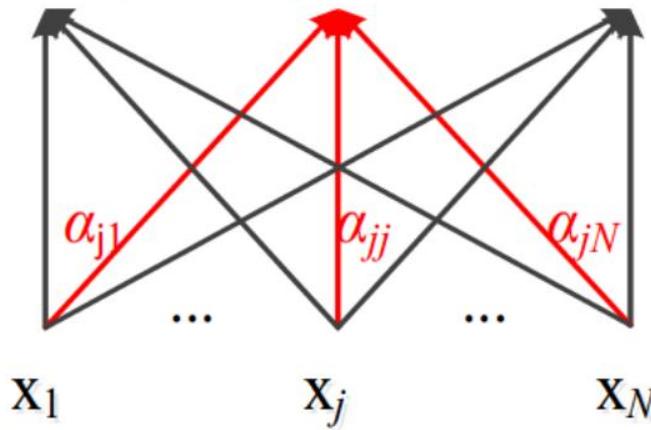
Att(X,X,X)
其中
Q=X
K=X
V=X



■一个两层网络结构：注意力机制、注意力网络

■与全连接前馈网络对比：

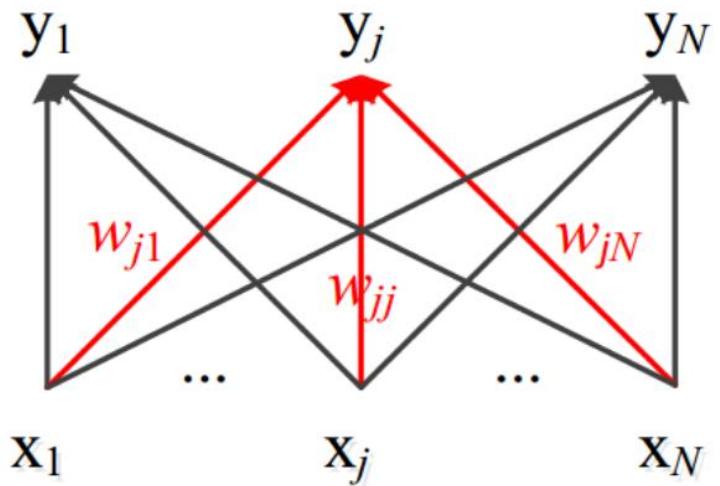
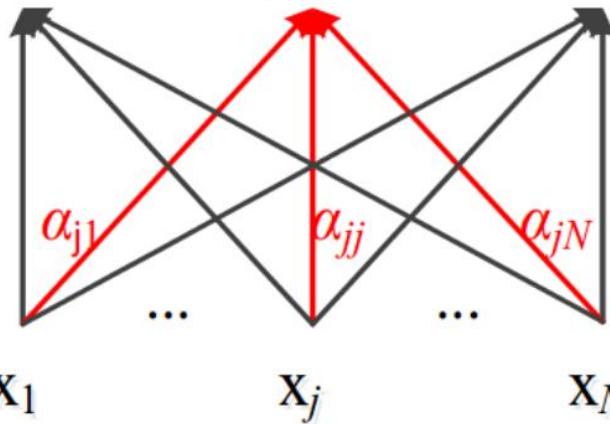
$\text{Att}(x_1, X, X)$ $\text{Att}(x_j, X, X)$ $\text{Att}(x_N, X, X)$



■一个两层网络结构：注意力机制、注意力网络

■与全连接前馈网络对比：

$$\text{Att}(x_1, X, X) \quad \text{Att}(x_j, X, X) \quad \text{Att}(x_N, X, X)$$



$$\blacksquare \text{Att}(x_j, X, X) = \sum_{i=1}^N \frac{\exp(x_j \cdot x_i)}{\sum_{l=1}^N \exp(x_j \cdot x_l)} x_i \quad \text{表示学习}$$

$$\blacksquare y_j = \sum_{i=1}^N w_{ji} x_i \quad \text{连接学习}$$

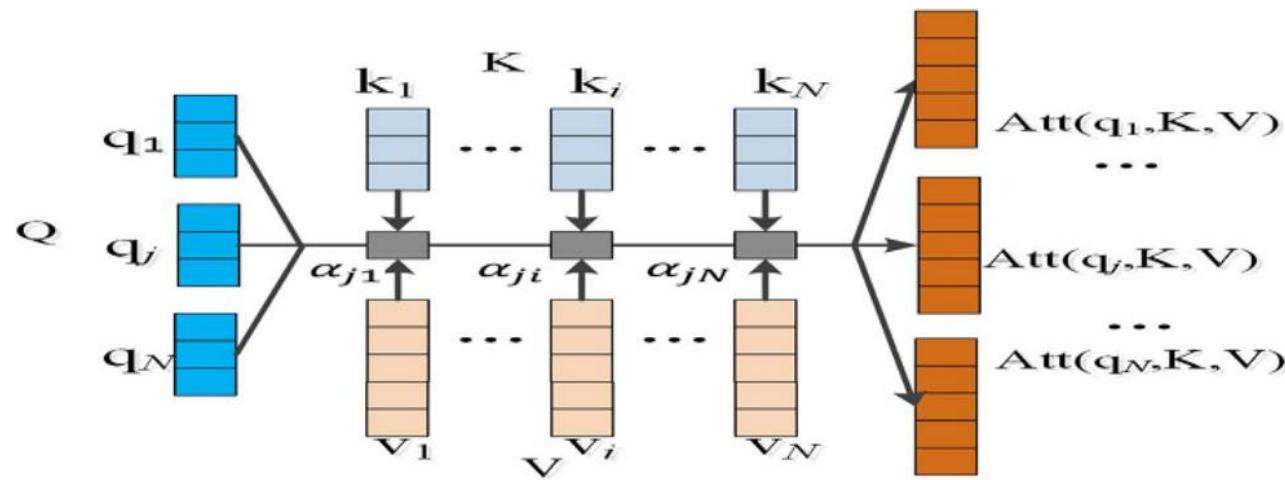
■ $\text{Att}(X, X, X)$ ($Q=K=V=X$) 更一般化：先对 X 进行变换构建三要素：

■ 查询向量序列： $Q = W_Q X \in \mathbb{R}^{d_k \times N}$, 其中 $W_Q \in \mathbb{R}^{d_k \times d_1}$

■ 键向量序列： $K = W_K X \in \mathbb{R}^{d_k \times N}$, 其中 $W_K \in \mathbb{R}^{d_k \times d_1}$

■ 值向量序列： $V = W_V X \in \mathbb{R}^{d_v \times N}$, 其中 $W_V \in \mathbb{R}^{d_v \times d_1}$

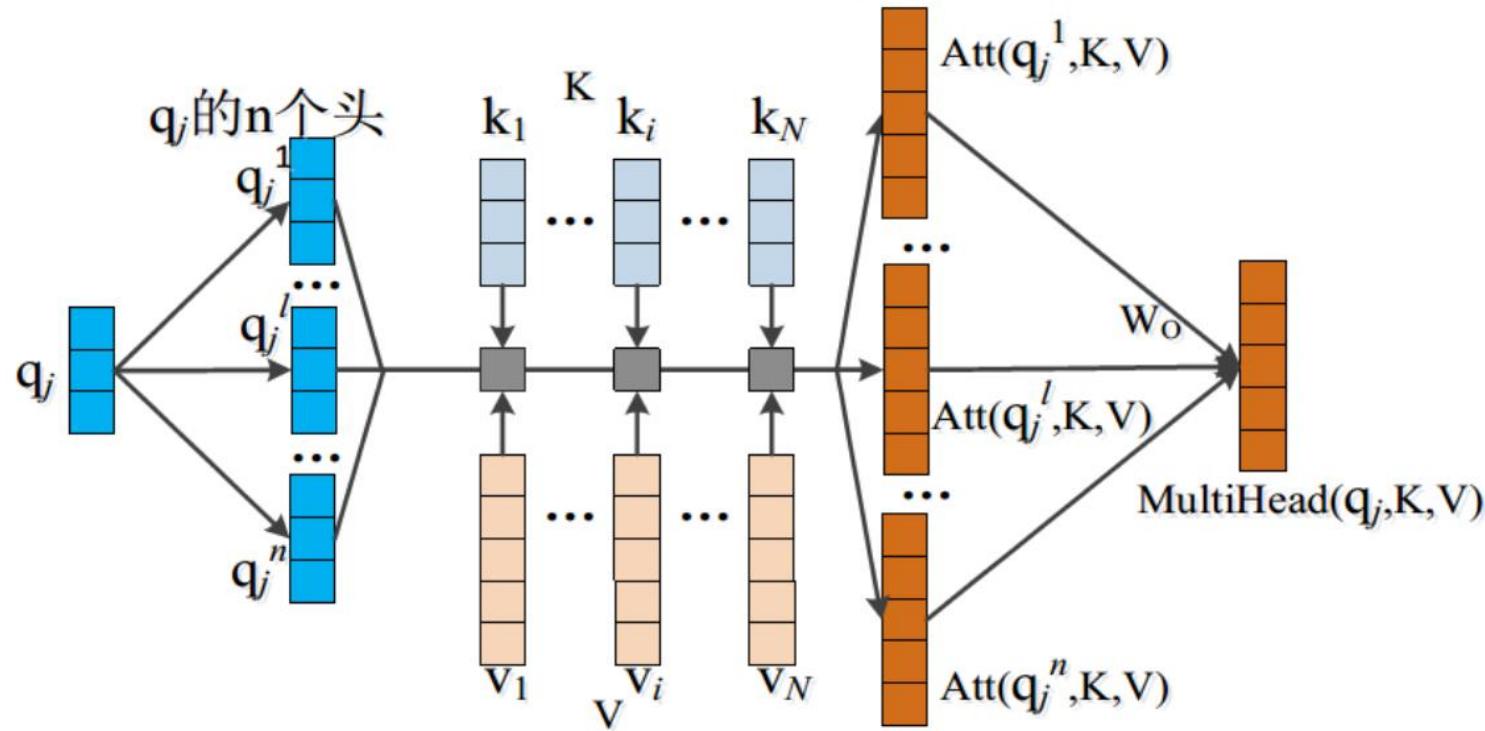
■ 同前方法可计算 $\text{Att}(Q, K, V) = V \text{ softmax}(Q^T K) = W_V X \text{ softmax}(X^T W_Q^T W_K X)$



■多头自注意力(Multi-Head Self Attention, MHSA)

- 前述用一组($W_Q \setminus W_K \setminus W_V$)对 X 进行变换可以得到一个 $X = [x_1, \dots, x_N]$ 的自注意力
- 采用多组($W_Q^i \setminus W_K^i \setminus W_V^i$)($i=1, \dots, n$, n 为组数量)可以得到 X 的多组自注意力，其允许获得 X 在不同变换子空间中的信息，在后面一些应用中的经验表明其有效。
- 操作方法：每组矩阵分别可以得到任意 x_i 的一个 d_v 维注意力，之后拼接得到 $n * d_v$ 维的注意力，为了保持最后仍为 d_v 维，后接一个变换，即：
- $\text{MultiHead}(Q, K, V) = W_0 \text{Concat}(\text{head}_1; \dots; \text{head}_n)$, 其中
 - $\text{head}_i = \text{Att}(W_Q^i X, W_K^i X, W_V^i X)$, Concat 是对应列拼接
 - $W_0 \in \mathbb{R}^{d_v \times n d_v}$

q_j 的多头注意力



列拼接

■更多的注意力

■指针网络[Vinyals2015NIPS]

■利用注意力分布获得位置信息，常用于指出一个片段的开头和结束位置，来做片段的选择、拷贝等

■层次注意力[YangZC2016NAACL]

■借以实现注意力从上到下的传递，常用于具有层次结构的对象，例如文本-句子-词

■...

[Vinyals2015NIPS]Oriol Vinyals, Meire Fortunato, Navdeep Jaitly. Pointer Networks. NIPS2015.

[YangZC2016NAACL]Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, Eduard Hovy. Hierarchical Attention Networks for Document Classification. NAACL2016.

注意力网络的发展

- 注意力机制(网络)
- 基于注意力网络的Seq2Seq模型-Transformer
- 基于Transformer的预训练语言模型-BERT/GPT

- Seq2Seq + 注意力机制 取得显著效果

- 序贯建模(序列标注、语言模型)取得了当时的SOTA
 - 转导问题(Machine Translation, Chatbot) 取得了当时的SOTA

- 但是：

- 1、基于RNN的Seq2Seq中RNN的按时序处理的本质排除了并行计算的可能，计算效率是一个问题(大规模数据的利用能力)。
 - 2、RNN建模远距离词之间的关系能力弱

- 注意到：

- 1、注意力机制具有天然的并行性
 - 2、注意力机制与距离无关

- 因此：可否用注意力机制替换Seq2seq

- 发展出：基于注意力网络的Seq2seq模型:Transformer

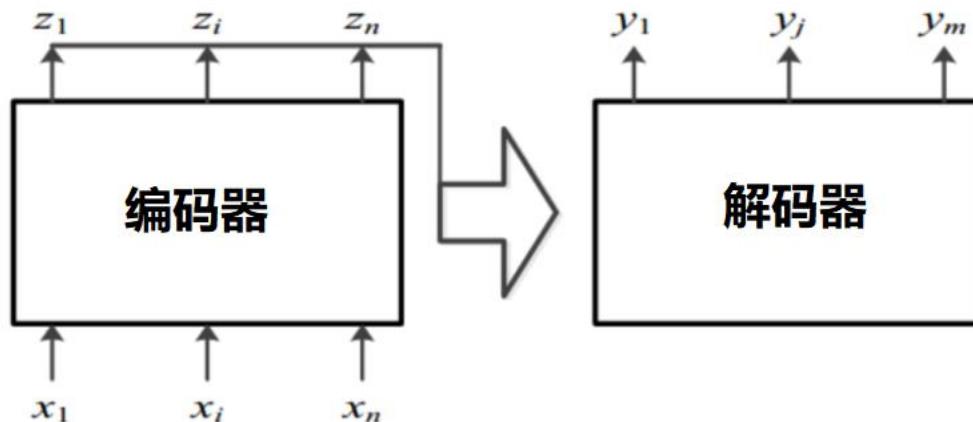
■ 基于注意力网络的Seq2seq模型:Transformer(转导器)

■ 结构: 编码器-解码器结构

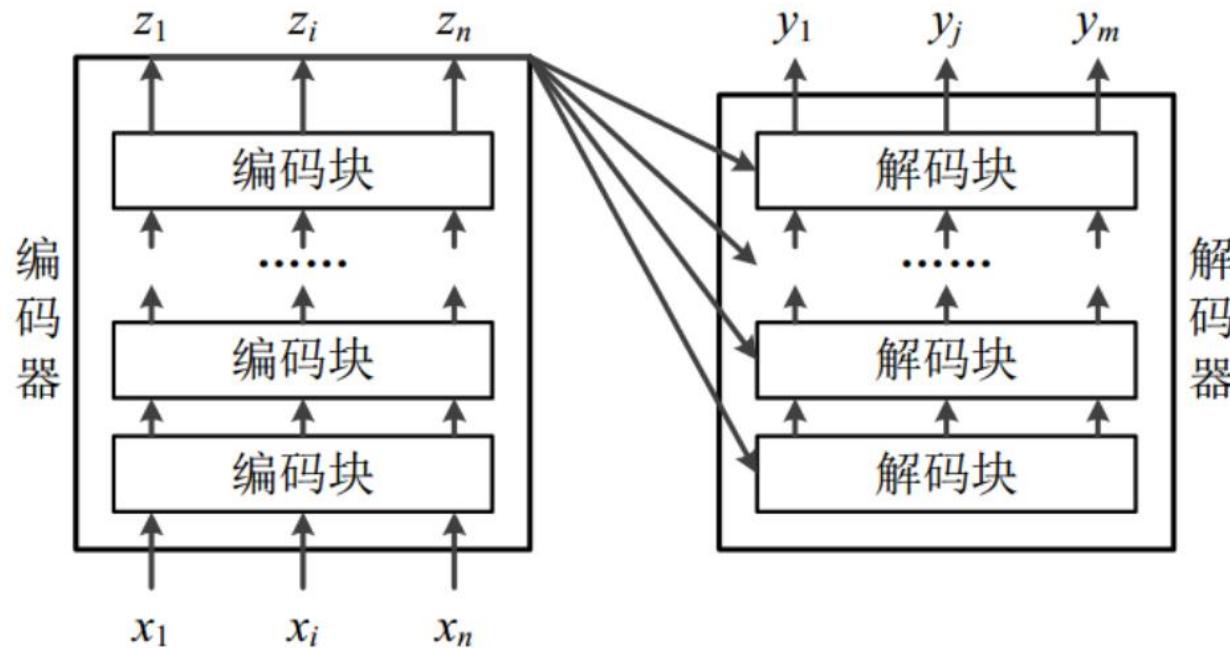
■ Transformer编码器: 将输入 $x = (x_1, \dots, x_n)$ 映射到隐层 $z = (z_1, \dots, z_n)$;

■ x_i 为某个word 或 subword 的 embedding(+ 位置编码)

■ Transformer解码器: 基于 z 解码生成输出序列 (y_1, \dots, y_m)

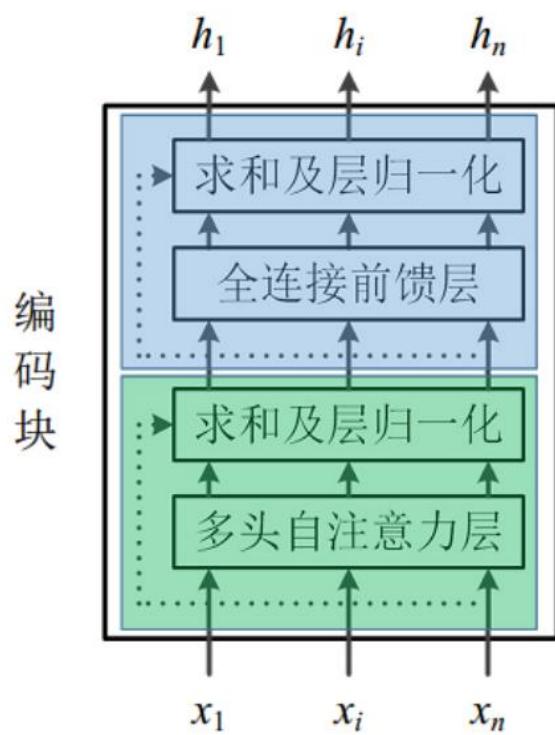


■ 编码器和解码器分别由N层Transformer编码块和Transformer解码块组成 (原文N=6)

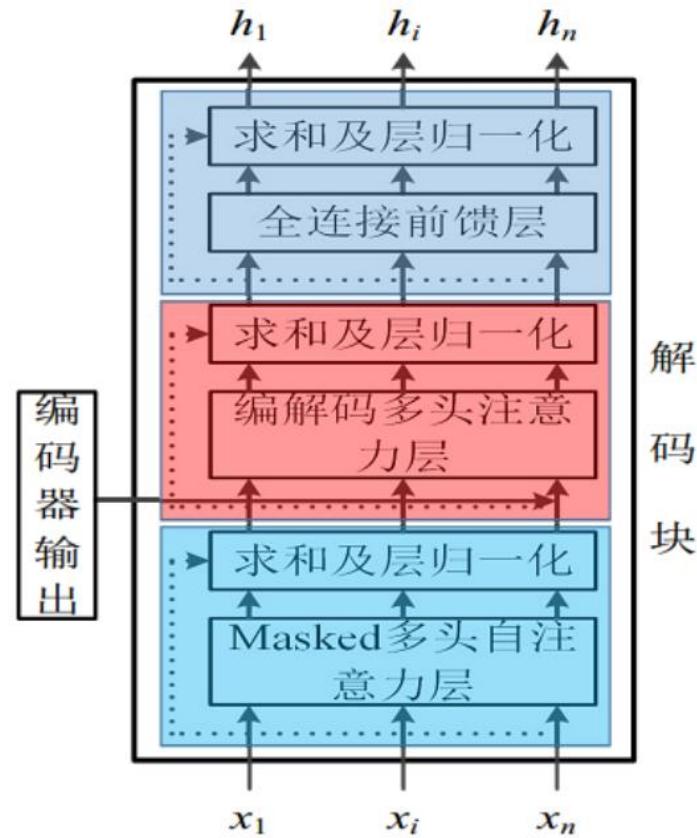


■ 下面分别介绍Transformer编码块和Transformer解码块

■Transformer编码

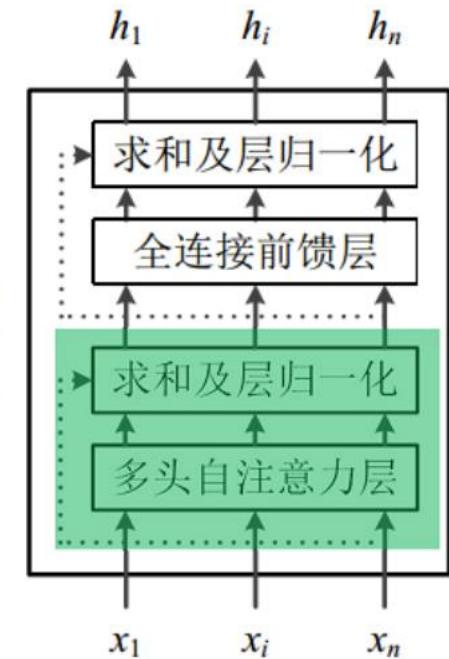
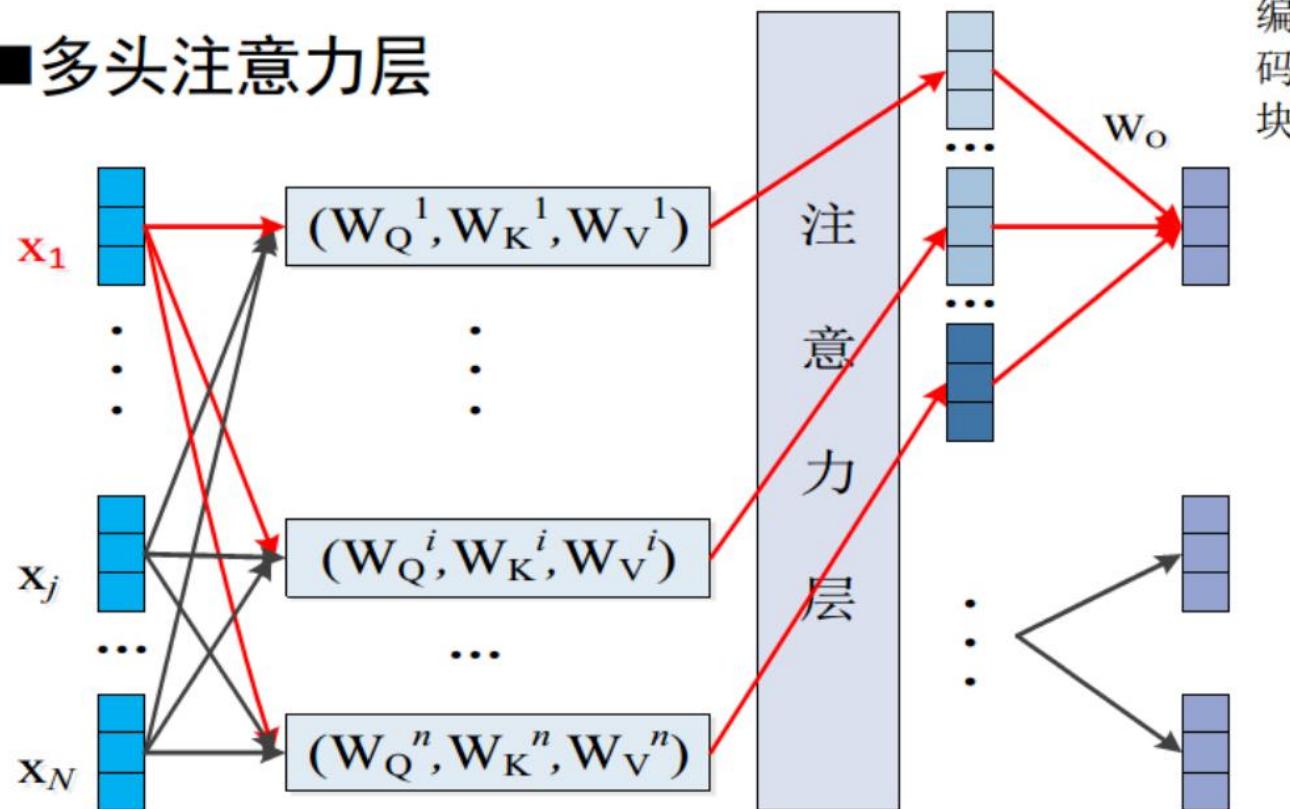


■Transformer解编码



Transformer编码块

■ 多头注意力层



Transformer编码块

■求和及归一化层

■求和: $y = x + a$

■ x : short-cut直连

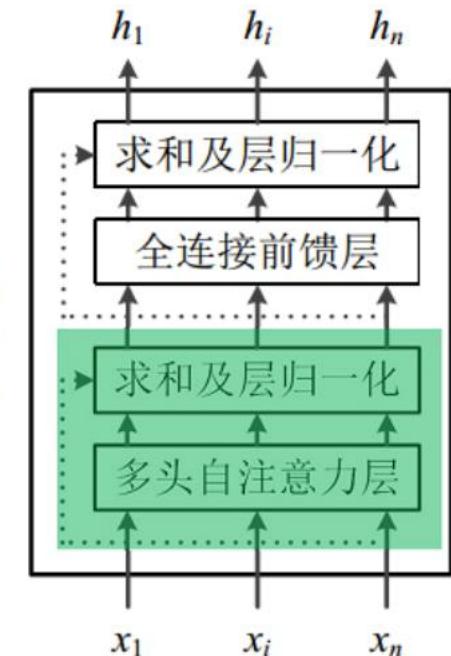
■ a : 多头注意力层输出

■层归一化: LayerNorm(y)

■ $h = g \odot N(y) + b, \quad N(x) = \frac{y - \mu}{\sigma},$

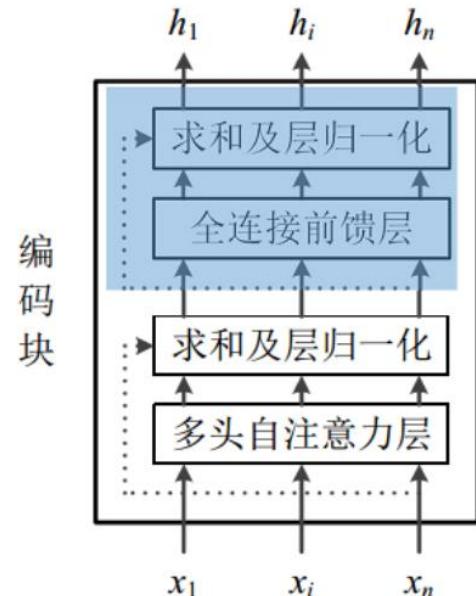
■ $\mu = \frac{1}{H} \sum_{i=1}^H y_i, \quad \sigma = \sqrt{\frac{1}{H} \sum_{i=1}^H (y_i - \mu)^2}, \quad H \text{为层神经元数}, \quad y = (y_1, \dots, y_i, \dots, y_H)$

■为何有效? [XuJJ2019NIPS]



Transformer编码块

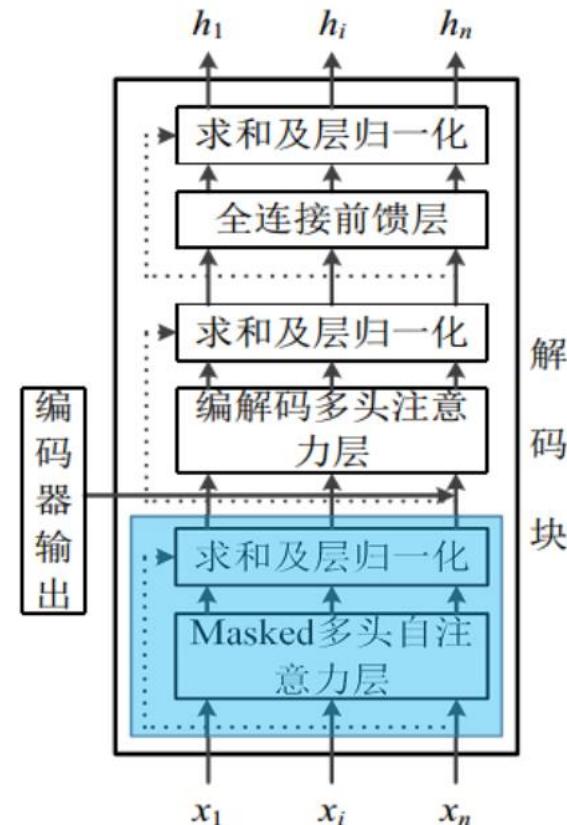
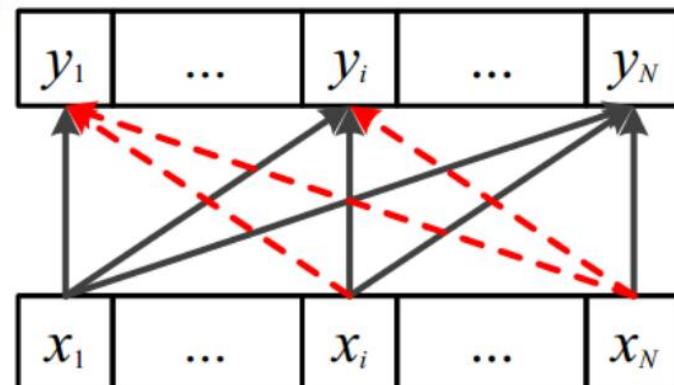
- 全连接前馈网络
 - 普通
- 求和及层归一化
 - 同前



Transformer解码块

■ Masked多头自注意力层：

■ 多头自注意力操作：和编码块中的类似，唯一差别是：在对 x_i 进行自注意力操作时， $x_j, j > i$ 部分被遮蔽掉(红虚线)



■ 原因：建模语言的时序性，解码 x_i 时是顺序进行，后面的词不能在解码前面的词时就已知了。

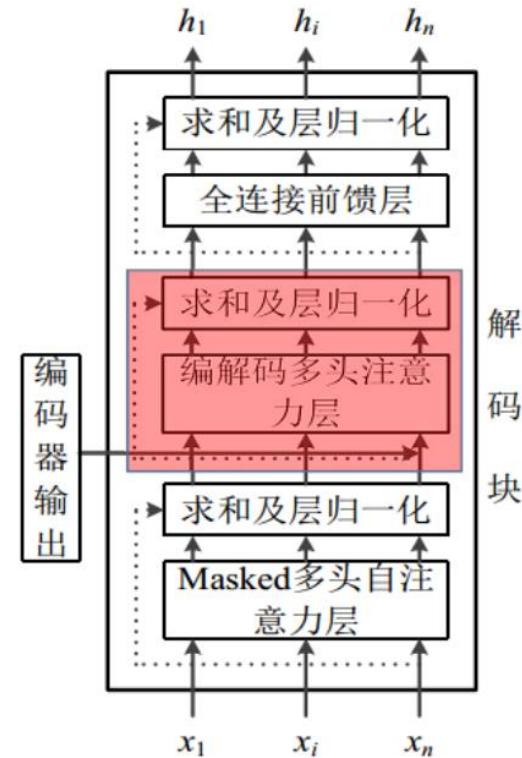
Transformer解码块

■ 编解码多头注意力层：

■ 多头注意力操作：其中的Q是来自下层的输出，而K、V是编码器的输出。

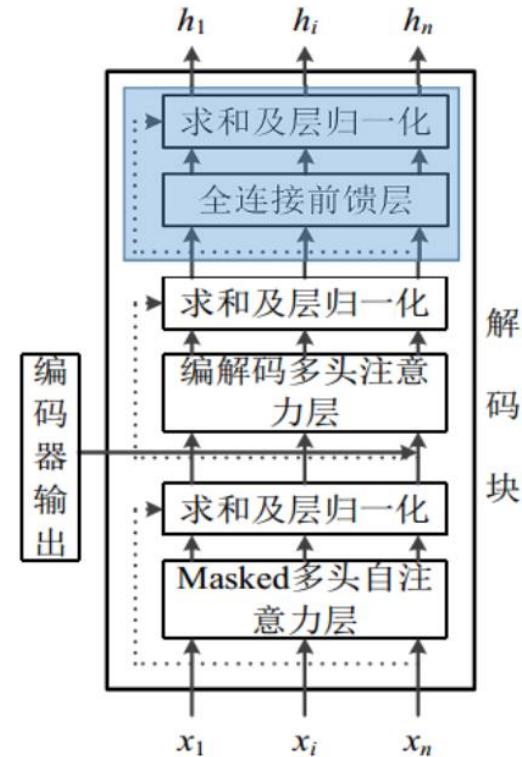
■ 求和及层归一化

■ 同前

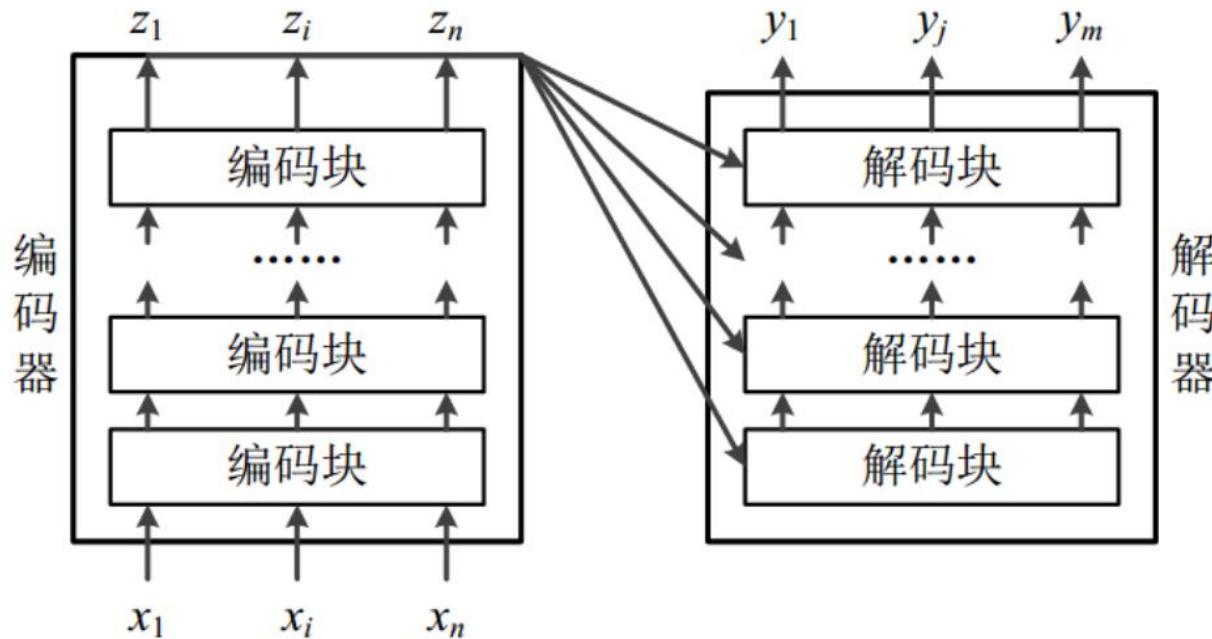


Transformer解码块

- 全连接前馈层
 - 和编码块中的一样
- 求和及层归一化层
 - 同前



■这样，就构成了完全没有RNN的主要由注意力网络(还有FNN)构成的Seq2Seq模型



■ 通过注意力机制

- 完全可并行化

- 跨越了词的远距离依赖

- 输入：句子 $s = w_1, \dots, w_i, \dots, w_N$ 的词向量 $x_1, \dots, x_i, \dots, x_N$ ，

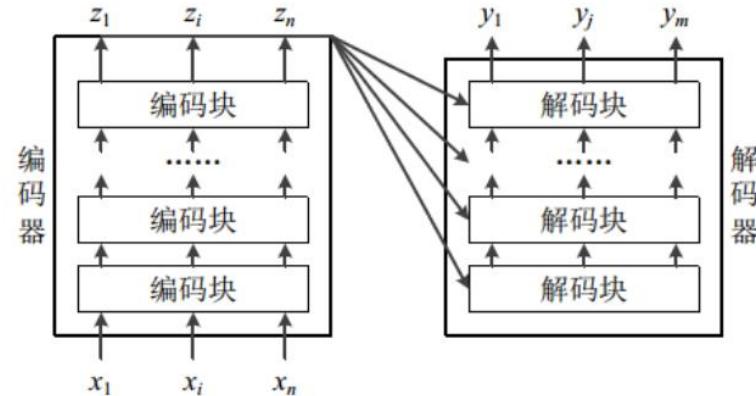
- 在计算其中每个词的自注意力时，其他词都可以产生作用。

■ 但是，问题也存在于此：

- 句子 $s = w_1, \dots, w_i, \dots, w_N$ 中的词序信息丢失了！

- 如何加入这种信息？

- 一般方案：词编码+位置编码(和词编码相同维度的向量)



■位置编码

■将位置 k 编码为一个 d 维向量 $PE_{(k)}$

■1、随机初始化位置编码向量

■2、sin-cos编码方案，各维定义：

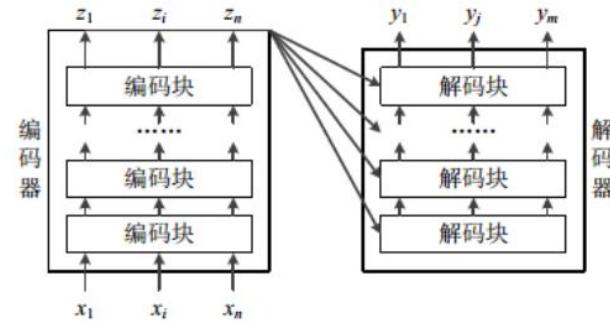
$$\text{■ } PE_{(k,2i)} = \sin\left(\frac{k}{1000d}\right) \quad PE_{(k,2i+1)} = \cos\left(\frac{k}{1000d}\right)$$

$$\text{■ } (\sin(k), \cos(k), \sin\left(\frac{k}{1000d}\right), \cos\left(\frac{k}{1000d}\right), \dots)$$

■二者性能上具有可比性

■随机初始化位置编码向量：可学习

■sin-cos编码：解析式，可理解？



■ 性能比较：翻译任务

Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [15]	23.75			
Deep-Att + PosUnk [32]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [31]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [8]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [26]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [32]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [31]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [8]	26.36	41.29	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1		$3.3 \cdot 10^{18}$
Transformer (big)	28.4	41.0		$2.3 \cdot 10^{19}$

模型规模：Transformer Base: $65 \cdot 10^6$; Transformer Big: $213 \cdot 10^6$

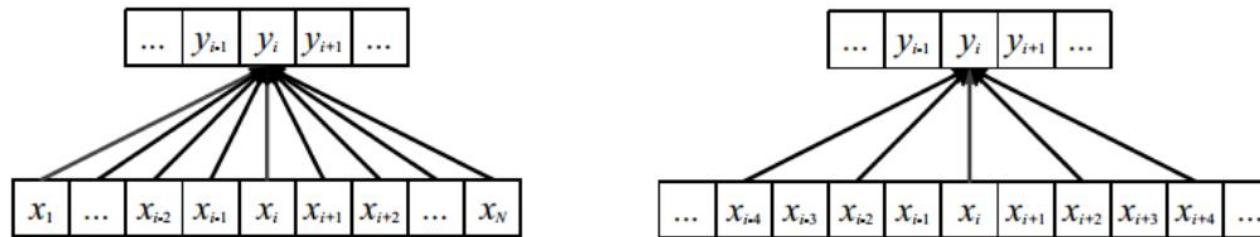
更多细节可参阅 [Vaswani2017NIPS] Ashish Vaswani, et al., Attention Is All You Need, NIPS2017.

■Transformer结构的问题与改进

■自注意力的计算复杂性高：输入符号长度N的平方

■稀疏化注意力、空洞注意力：

■间隔注意1、2、3、4.....



■条件计算：

■ MoE(Mixture of Experts) [Shazeer2017ICLR]: 不同样本通过Gated 网络来选择不同的少数 Experts进行响应，多个Experts共享结构，但是参数不同，这样极大增加了参数规模，从而可以在减少Transformer层数时保持参数规模。

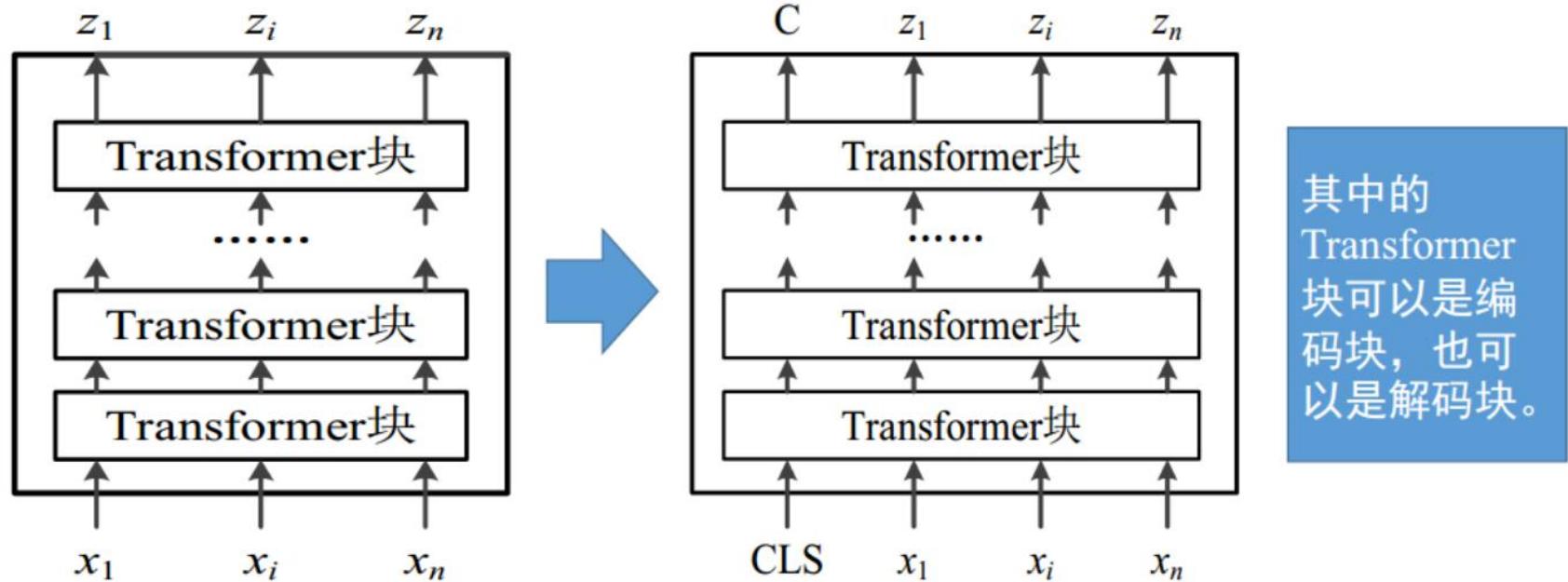
■解码时的串行问题

■.....

[Shazeer2017ICLR] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton and Jeff Dean. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. ICLR 2017.

■ 基于Transformer的词、句子编码

- Transformer的编码部分和解码部分分别可以都作为一种对词进行编码方法，进一步通过加入一个符号编码句子



注意力网络的发展

- 注意力机制(网络)
- 基于注意力网络的Seq2Seq模型-Transformer
- 基于Transformer的预训练语言模型-BERT/GPT

■Transformer结构的发展：

■基于Transformer的预训练语言模型

■GPT(Generative Pre-Training)

■**基于Transformer解码块**：单向生成的语言模型(生成)

■[Radford2018-GPT]Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever. Improving Language Understanding by Generative Pre-Training. 2018.

■BERT (Bidirectional Encoder Representations from Transformer)

■**基于Transformer编码块**：双向预测的语言模型(判别)

■[Devlin2018NAACL-BERT]Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL-HLT 2019, pages 4171–4186, Minneapolis, Minnesota, June 2 - June 7, 2019.

■BERT

■基础结构

- 基础的NN结构，独立于任务，输入-输出构成词和句子的再编码

■预训练任务

- 自监督学习任务对基础NN结构中的参数进行优化：即为预训练(Pre-Training)。
- 目的在于获得独立于特定任务的一般性知识。

■用于下游任务

- 结构和基础结构没有重要变化，只是为了特定任务能完成加上某个部件。
- 可以基于特定任务的标注数据进行再次的参数优化：即为微调(Fine-tuning)

■BERT结构：

■由多层Transformer解码块组成

■Transformer解码块的层数：L

■隐层单元数量：H

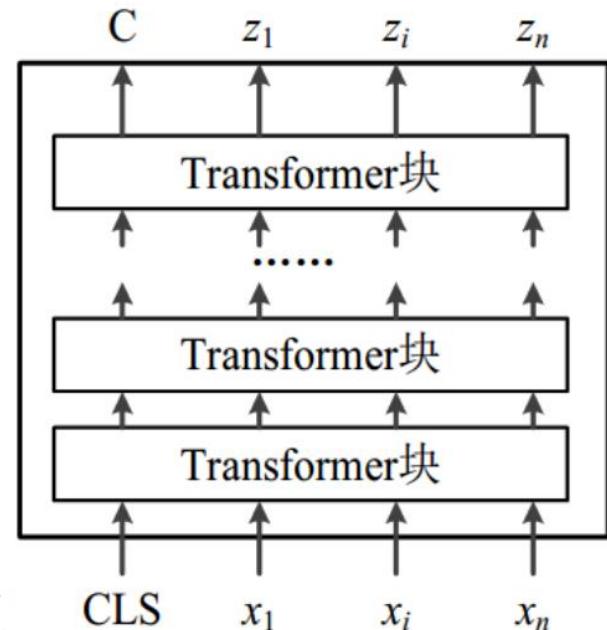
■self-attention头数：A

■BERTBASE

■ $L=12, H=768, A=12$, 总参数： 110M

■BERTLARGE

■ $L=24, H=1024, A=16$, 总参数： 340M



■ 预训练任务之MLM(Masked Language Model)

■ 遮蔽预测任务：

- 对于一个输入序列，遮蔽某个符号 x_i (以MASK代替)，输出端预测 $z_i=x_i$ ，交叉熵损失函数

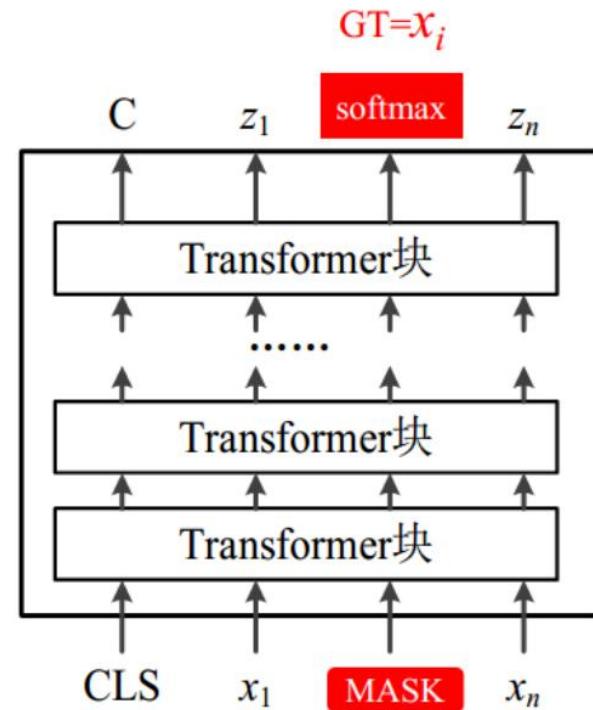
■ 样本构建：

- 随机选择M%(M=15)的输入符号

- 80%概率替换为MASK
- 10%概率替换为一个随机符号
- 10%概率不替换

- 对应的输出为原符号

- 无需额外的标注数据，自然的语句就可以用来直接构建为训练数据：自监督学习任务



■预训练任务之NSP(Next Sentence Prediction)

■下句预测任务：

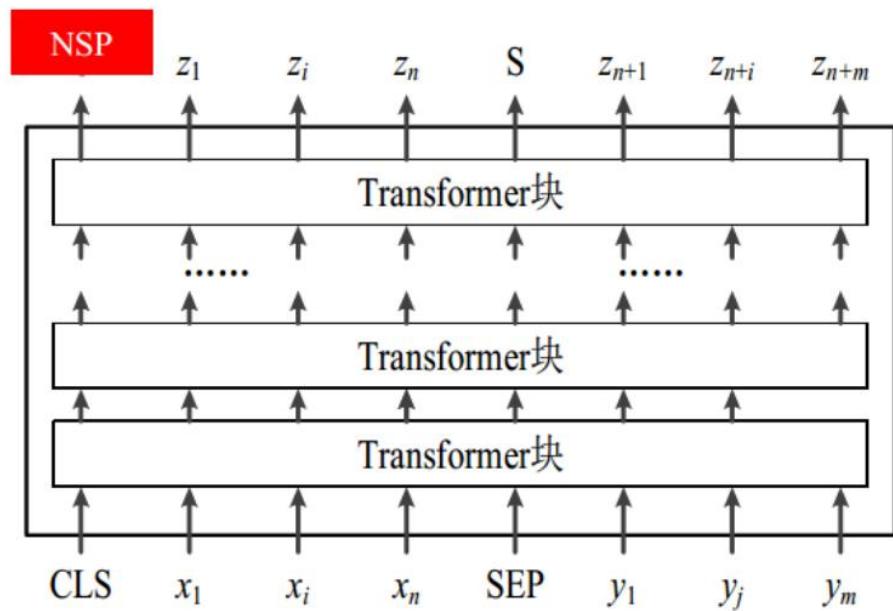
- 输入句A+句B，输出端C位预测B是否是A的下一句，二分类任务

■样本构建：

- 在语料中选择句A后：

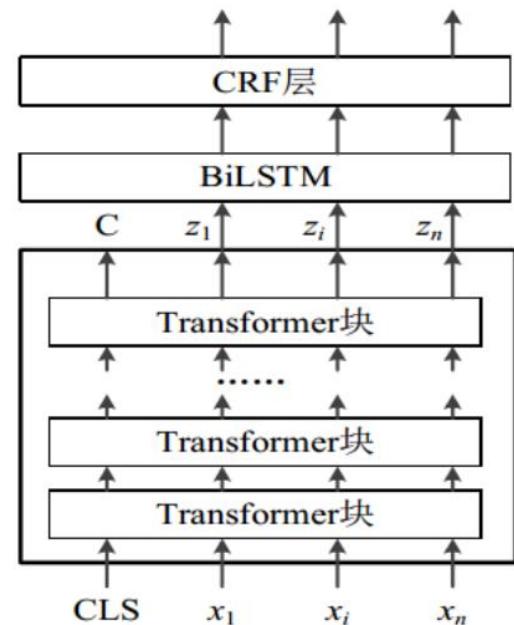
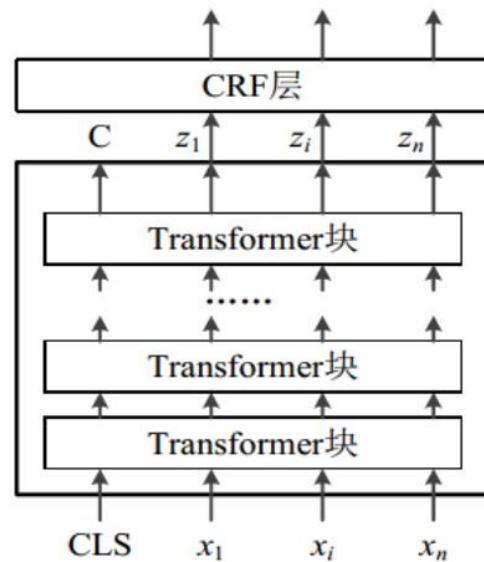
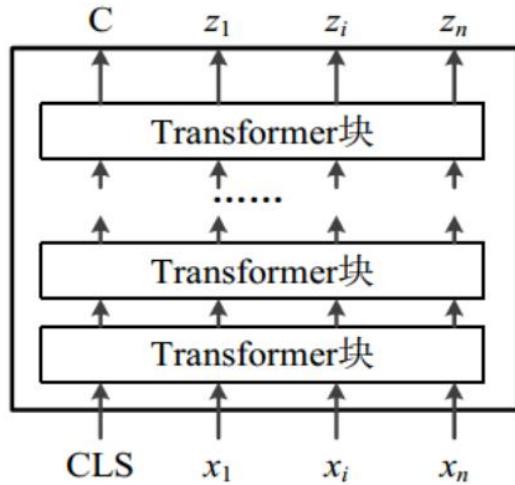
- 50%概率选择其后一句为句B（正例）
- 50%概率在其他位置选择一句为句B（反例）

- 无需额外的标注数据，自然的语句就可以用来直接构建为训练数据：自监督学习任务



■ 用于下游任务

- 句子分类：C处接一个softmax分类器
- 词性标注：
 - 每个 z_i 处接一个softmax分类器；
 - 进一步，建模标签转移约束，+CRF层：BERT-CRF；
 - 进一步，BERT不好微调，+BiLSTM：BERT-BiLSTM-CRF



■BERT模型性能评估

■ GLUE(The General Language Understanding Evaluation)多任务评测， 11个任务的数据集：

- 1、 CoLA(The Corpus of Linguistic Acceptability): 句子是否合语法的二分类
- 2、 SST-2(Stanford Sentiment Treebank): 句子情感二分类
- 3、 MNLI(Multi-Genre Natural Language Inference): 给定两个句子是否有蕴含、矛盾关系或无关
- 4、 STS-B(The Semantic Textual Similarity Benchmark)两个句子间的相似度判定(5级)
-

System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	Average
	392k	363k	108k	67k	8.5k	5.7k	3.5k	2.5k	-
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.9	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	88.1	91.3	45.4	80.0	82.3	56.0	75.2
BERT _{BASE}	84.6/83.4	71.2	90.1	93.5	52.1	85.8	88.9	66.4	79.6
BERT _{LARGE}	86.7/85.9	72.1	91.1	94.9	60.5	86.5	89.3	70.1	81.9

更多细节参阅[Devlin2018NAACL-BERT]Jacob Devlin, etc. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL-HLT 2019, pages 4171–4186, Minneapolis, Minnesota, June 2 - June 7, 2019.