

## 背景知识

encoder回顾/区分

RNN 变长序列

## Conditional Generation by RNN&Attention

### Transformer-> Bert

Attention的计算原理

在未应用attention时的encoder-decoder框架

在应用了attention后的encoder-decoder框架

## Deep learning for Question Answering System

### Doamin Adaptation

# 背景知识

---

## encoder回顾/区分

---

### [Difference between auto-encoder and encoder-decoder](#)

auto-encoder是一种特殊的encoder-decoder, 它的输入与输出内容长得一样(或者说decoder尽可能还原encoder压缩出来的 $\mathbb{C}$ , 可以认为auto-encoder是一种有损压缩算法),

$Auto - encoder \in Encoder - Decoder$

auto-encoder用途: 把中间的 hidden layer抽出来玩 [cite from](#)

encoder-decoder是一系列模型的统称, 是一种设计模式, 包含encoder与decoder这两个部分, encoder负责把属于映射至latent-space([在latent -space中, 相似的数据在latent空间中会得到相近的向量](#))中拿到向量 $\mathbb{C}$ , decoder负责解析 $\mathbb{C}$ ,转换为其他东西

seq2seq的严格定义: seq2seq 是一个 **Encoder-Decoder 结构**的网络, 它的输入是一个序列, 输出也是一个序列, Encoder 中将一个可变长度的信号序列变为固定长度的向量表达, Decoder 将这个固定长度的向量变成可变长度的目标的信号序列. [cite from](#)

$seq2seq \in encoder - decoder$

seq2seq与  $auto - encoder$ 的区别: seq2seq的输入与输出内容不一样, seq2seq不符合auto-encoder的定义

### Seq2Seq和 Encoder-Decoder的关系 [cite from](#)

Seq2Seq (强调目的) 不特指具体方法, 满足「输入序列、输出序列」的目的, 都可以统称为 Seq2Seq 模型。

而 Seq2Seq 使用的具体方法基本都属于Encoder-Decoder 模型（强调方法）的范畴。

总结一下的话：

- Seq2Seq 属于 Encoder-Decoder 的大范畴
- Seq2Seq 更强调目的，Encoder-Decoder 更强调方法

## RNN 变长序列

---

1. 为什么需要处理变长序？

因为神经网络需要输入内容具有较为一致的大小与格式

2. 变长序列的处理方法称为 <sup>1</sup>

## Conditional Generation by RNN&Attention

---

[李宏毅P51](#)

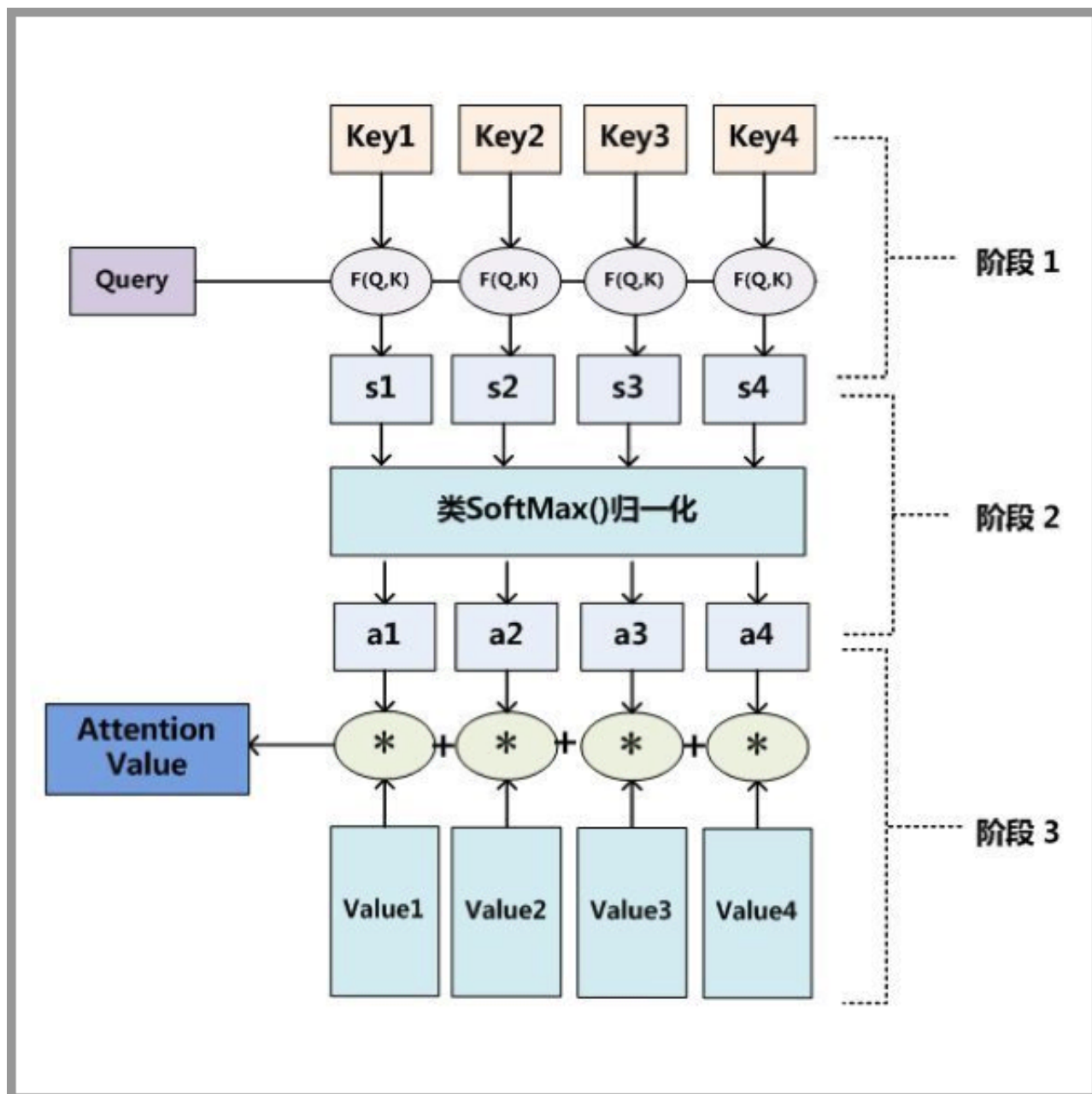
- character/words
  - 英文中character组成word, word之间以空格相互分隔,组成句子
  - 中文: e.g. "葡萄" 是一个word, "葡"与"萄"是characters

## Transformer-> Bert

---

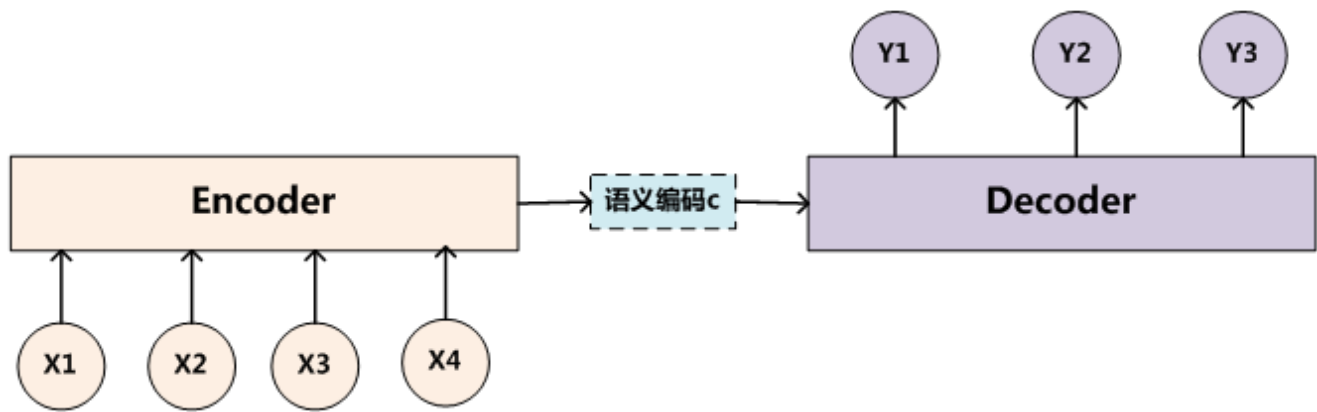
transfoermet: seq2seq model with self-attention

### Attention的计算原理



key-value, query与key做F(Q,K) 匹配度计算, 拿到 $s_i$ , 最后得到 $Attention_{value} = \sum_i s_i \cdot value_i$

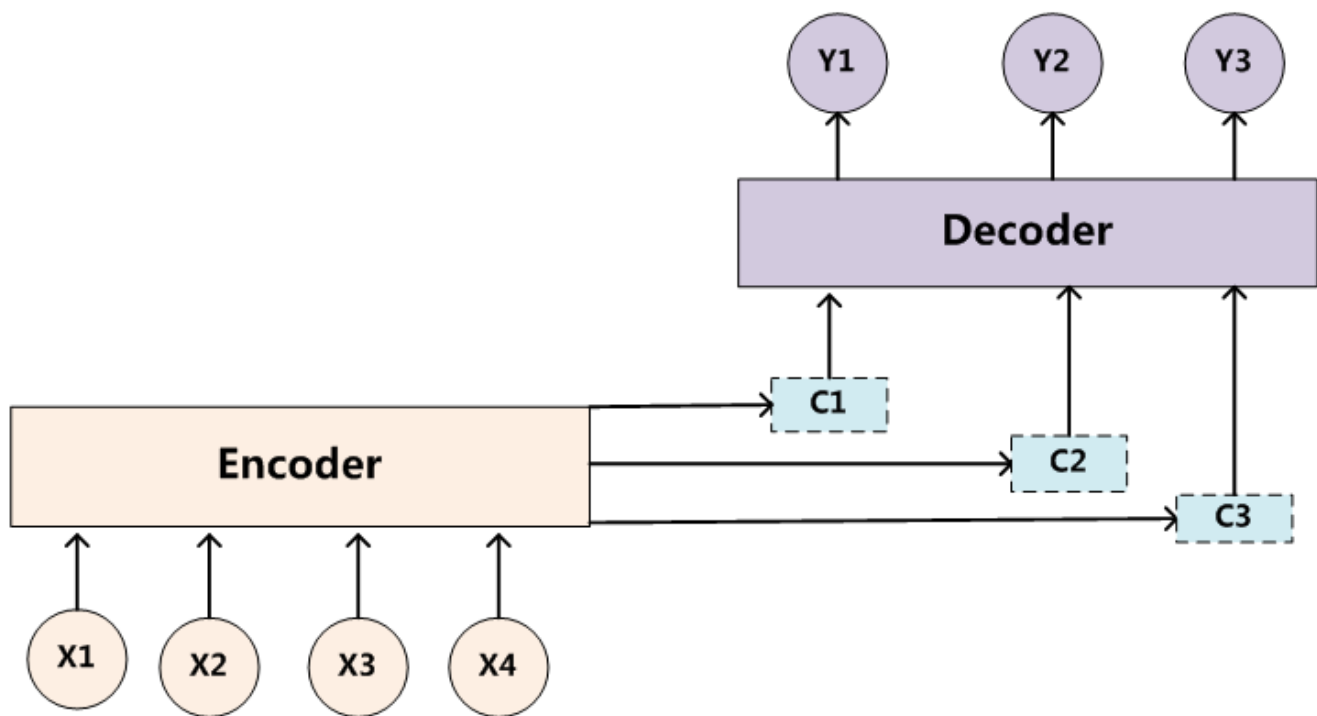
在未应用attention时的encoder-decoder框架



这是对于decoder  $f$  来说,  $Y_i$  的产生顺序:

$$\begin{aligned} y_1 &= f(C) \\ y_2 &= f(C, y_1) \\ y_3 &= f(C, y_1, y_2) \end{aligned}$$

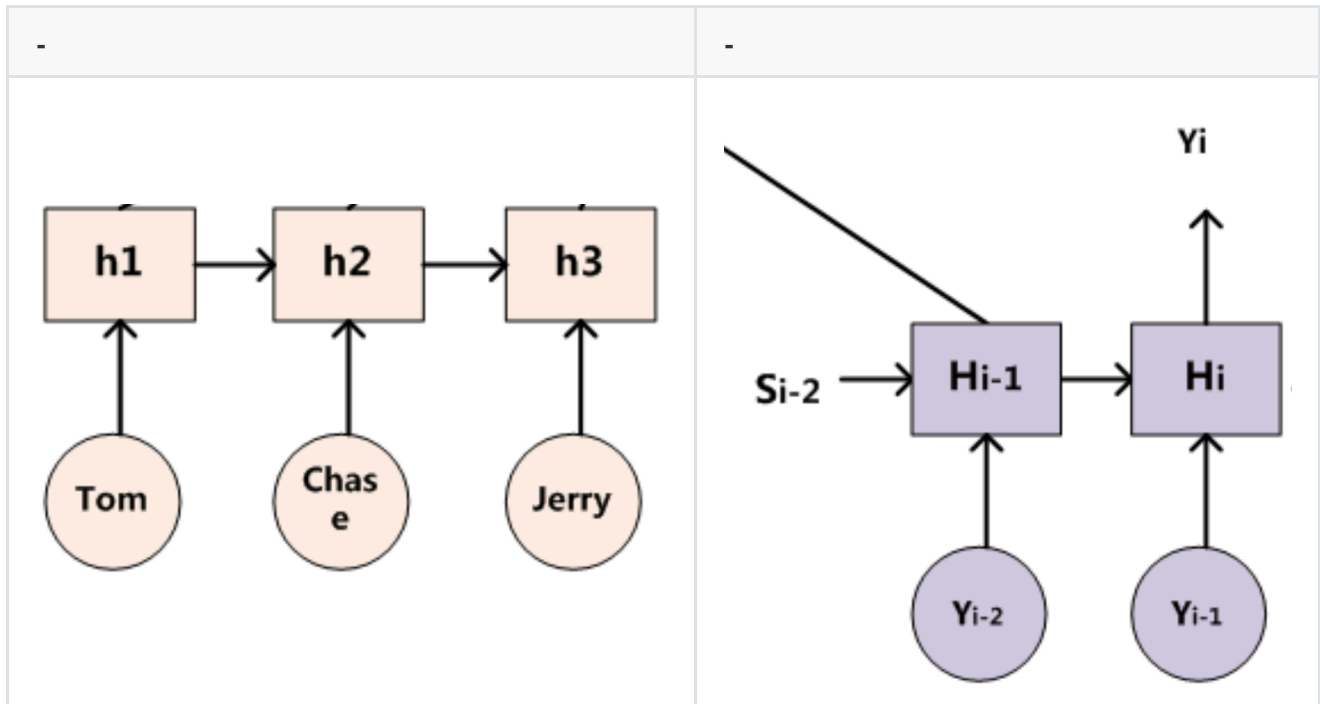
在应用了attention后的encoder-decoder框架



现在

$$\begin{aligned} y_1 &= f(C_1) \\ y_2 &= f(C_2, y_1) \\ y_3 &= f(C_3, y_1, y_2) \end{aligned}$$

假设decoder部分 $Y_i$ 部分具体展开如图



假设现在 $i=1$

所以我们手里有 $decoder$ 隐藏层  $H_0$ , 权重计算函数  $s(a, b)$ ,  $encoder$ 的每一个隐藏层  $h_i$

所以我们对  $H_0$  与每一个  $h_i$  代入  $s(a, b)$  计算匹配度, 并做  $softmax$  处理拿到概率分布  $A$

$$所以 C_0 = \sum_i h_i A_i = h^T \cdot A$$

$$于是 Y_0 = f(C_0, X_1)$$

Bert: unsupervised trained transformer

## Deep learning for Question Answering System

## Domain Adaptation

---

1. Padding As we know all the neural networks needs to have the inputs that should be in similar shape and size. When we pre-process the texts and use the texts as an inputs for our Model. Note that not all the sequences have the same length, as we can say naturally some of the sequences are long in lengths and some are short. Where we know that we need to have the inputs with the same size, now here padding comes into picture. The inputs should be in same size at that time padding is necessary. [cite from](#) [↩](#)