

Attention本质理解

笔记本： 自然语言处理
创建时间： 2019/7/1 15:33
作者： beyourselfwb@163.com

更新时间： 2019/7/19 11:31

Seq2Seq NMT 示例

参考几篇经典的论文：

- [1] BAHDANAU D, CHO K, BENGIO Y. Neural Machine Translation by Jointly Learning to Align and Translate[J]. arXiv:1409.0473 [cs, stat], 2014. <http://arxiv.org/abs/1409.0473>
- [2] LUONG M-T, PHAM H, MANNING C D. Effective Approaches to Attention-based Neural Machine Translation[J]. arXiv:1508.04025 [cs], 2015. <http://arxiv.org/abs/1508.04025>
- [3] VASWANI A, SHAZEER N, PARMAR N, 等. Attention Is All You Need[J]. arXiv:1706.03762 [cs], 2017. <http://arxiv.org/abs/1706.03762>

代码来源：

https://pytorch.org/tutorials/intermediate/seq2seq_translation_tutorial.html

Attention出现是为了解决什么问题：

传统的Encoder Decoder 框架，在解码时仅仅依赖Encoder生成的固定长度的向量表示，当输入序列比较长时，性能很差。

注意力模型，也被称作对齐模型，有硬对齐（非0即1），软对齐（不同权重，综合考虑），也叫Hard Attention、Soft Attention。

自注意力模型，Self-attention，即 $K=V=Q$ ，使用自注意力机制的原因，论文中提到主要从三个方面考虑（每一层的复杂度，是否可以并行，长距离依赖学习）（出自论文 Attention is All You Need）。

区别于之前的Attention是Decoder某个阶段的状态跟Encoder的所有隐藏状态的相关性，Self Attention指的是在Encoder/Decoder 内部的相关性计算，可以理解为Target=Source这种特殊情况下的注意力计算机制。

对于Target不等于Source的Attention，物理含义可以理解作为一种对齐，比如英语love和中文爱的一种对齐关系；
 那么 Target=Source，即self-attention 可以捕获同一个句子中单词之间的一些句法特征（比如图11展示的有一定距离的短语结构）或者语义特征（比如图12展示的its的指代对象Law）。比RNN、LSTM更好地捕捉句子中长距离的相互依赖的特征，因为只需一步计算就能直接联系起来，而不是一步步累积起来。

参考 张俊林博客：

<https://blog.csdn.net/malefactor/article/details/78767781>

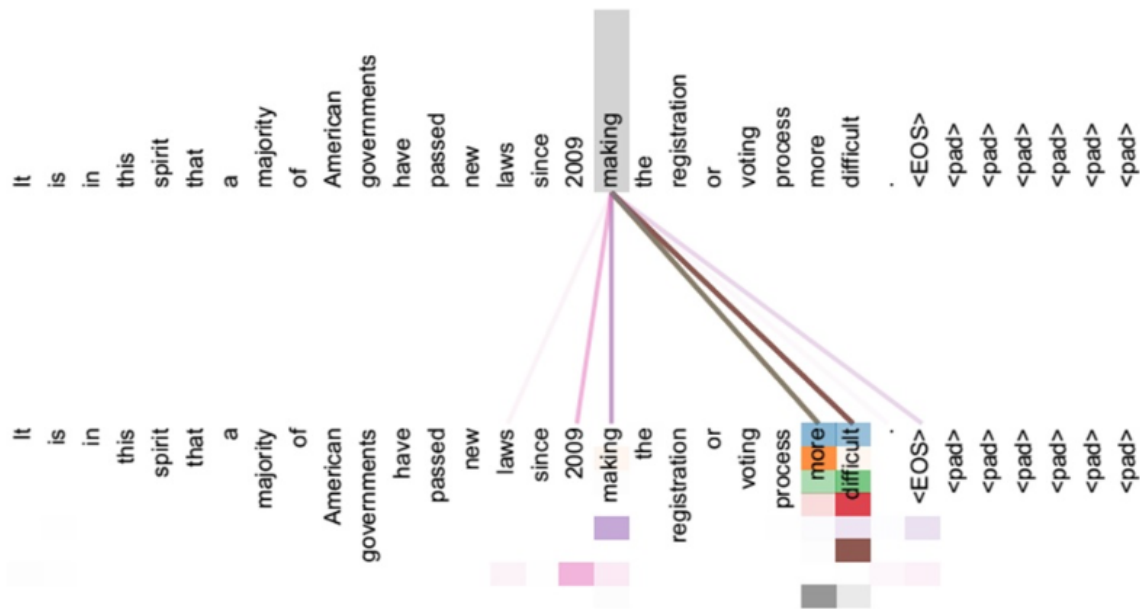


图11

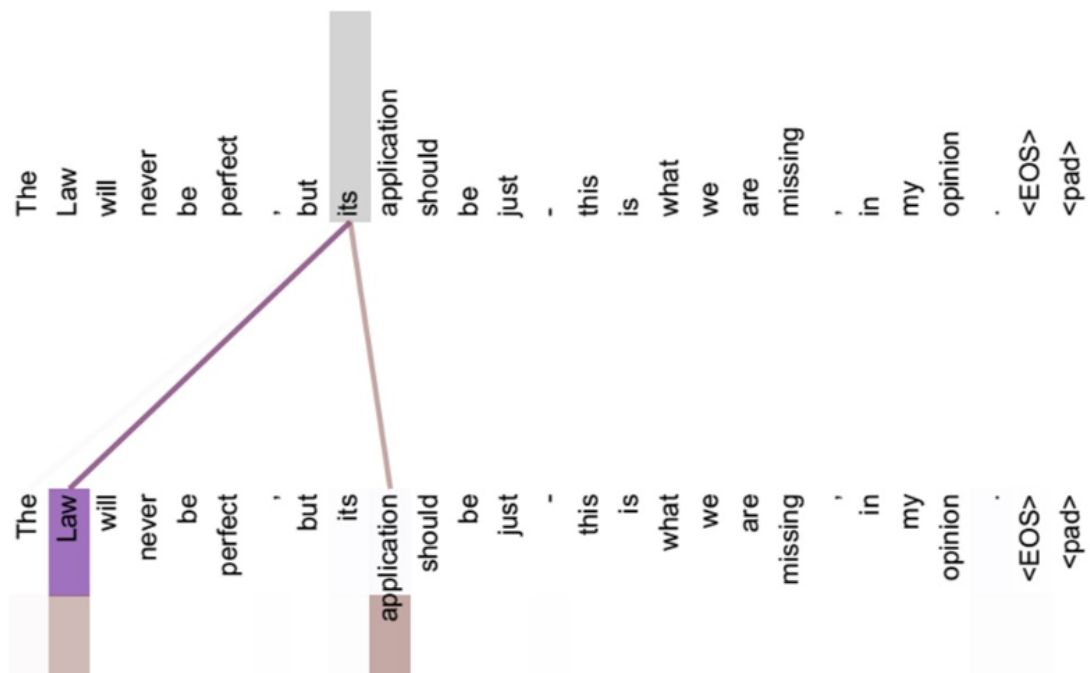
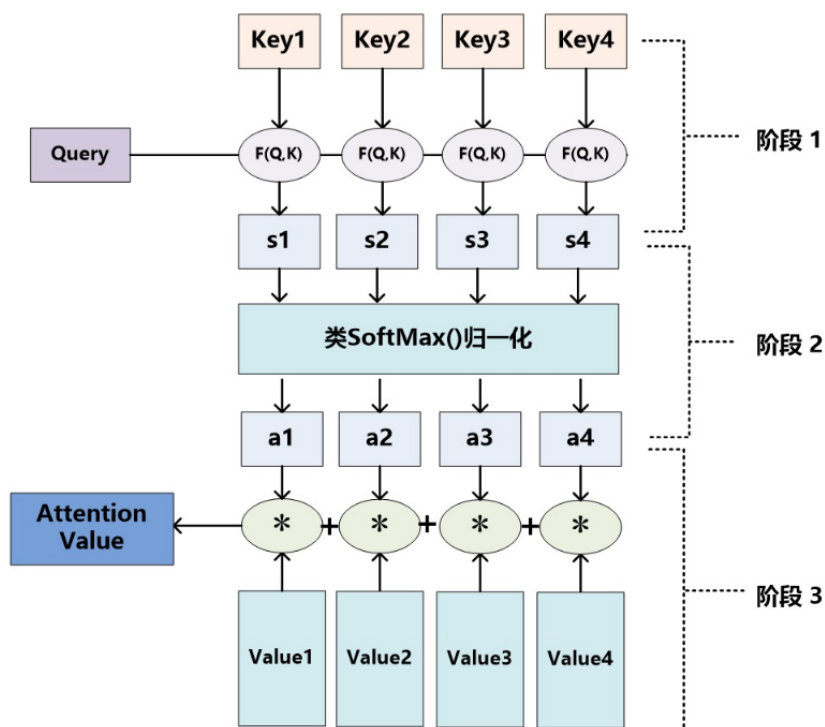


图12

一切Attention的本质（不同的只是 $f(Q,K)$ ）：

参考：<https://blog.csdn.net/malefactor/article/details/78767781>



再结合实际例子理解 Q K V

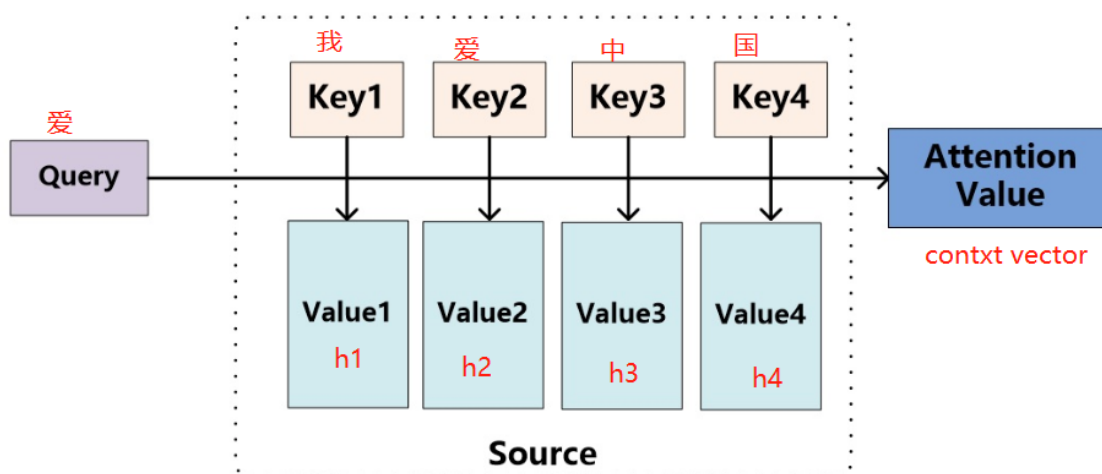


图9 Attention机制的本质思想

我们可以这样来看待Attention机制（参考图9）：将Source中的构成元素想象成是由一系列的 $\langle \text{Key}, \text{Value} \rangle$ 数据对构成，此时给定Target中的某个元素Query，通过计算Query和各个Key的相似性或者相关性，得到每

个Key对应Value的权重系数，然后对Value进行加权求和，即得到了最终的Attention数值。所以本质上Attention机制是对Source中元素的Value值进行加权求和，而Query和Key用来计算对应Value的权重系数。即可以将其本质思想改写为如下公式：

作者：张俊林博客

来源：CSDN

原文：<https://blog.csdn.net/malefactor/article/details/78767781>

版权声明：本文为博主原创文章，转载请附上博文链接！

$$\text{Attention}(\text{Query}, \text{Source}) = \sum_{i=1}^{L_x} \text{Similarity}(\text{Query}, \text{Key}_i) * \text{Value}_i$$