

2015MV-LSTM

题目：A Deep Architecture for Semantic Matching with Multiple Positional Sentence Representations

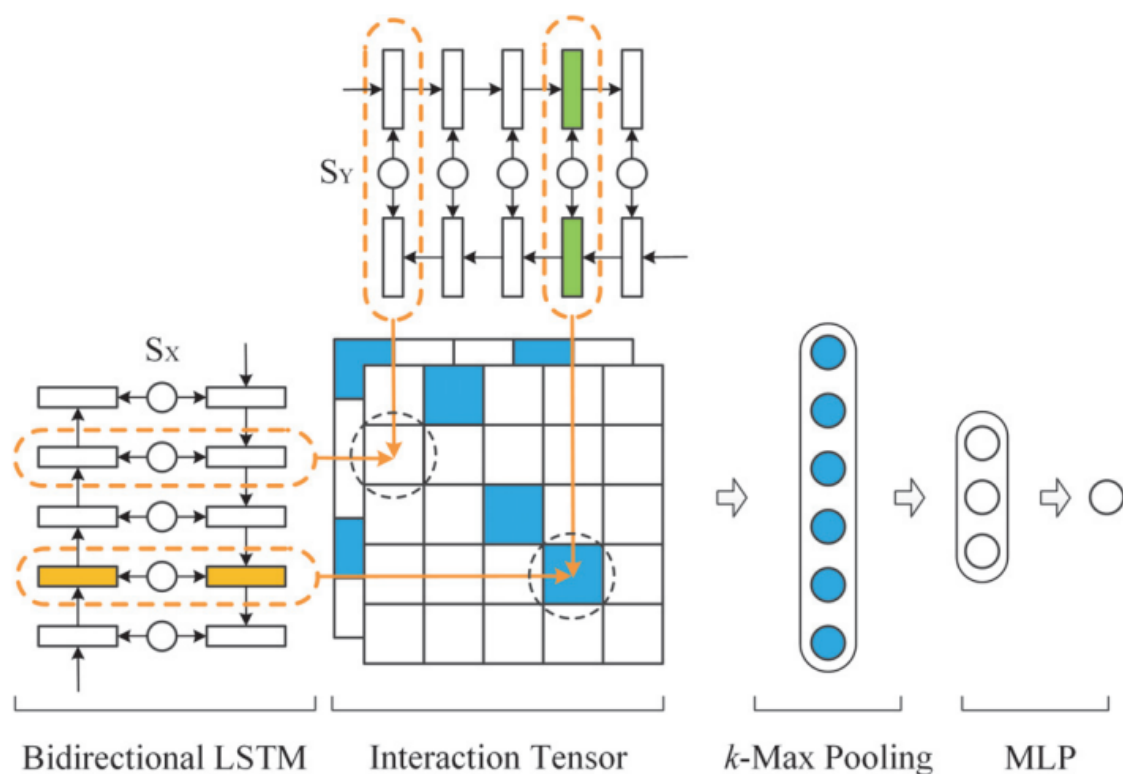
年份：2015

思想：

这篇论文采用**双向LSTM**处理两个句子，然后对LSTM隐藏层的输出进行两两计算匹配度，作者认为这是一个**Multi-View(MV)**的过程，能够考察每个单词在不同语境下的含义。同时用双向LSTM处理句子，相当于用变长的窗口逐步的解读句子，实现多颗粒度考察句子的效果。用每个位置的两个隐藏层输出代表这个位置的位置向量信息，从而关注到整个句子的不同位置，从而实现一个句子的多语义表达。在此基础上进行交互，分类。

这篇论文有3个**创新点**：1、使用了当时比较新型的双向LSTM模型，这种模型能考虑到句子的时序关系，能捕捉到长距离的单词依赖，而且采用双向的模型，能够减轻这种有顺序的模型权重偏向句末的问题。2、用Interaction tensor计算句子之间的两两匹配度，从多个角度解读句子。3、匹配计算公式采用了带参数的计算公式。

模型结构



分别对两个句子用**双向LSTM**提取语义表达特征。然后对这两个语义特征向量进行交互操作。然后对每一个交互矩阵上做K-MAX Pooling。最后用Pooling的结果输入MLP做分类。

具体流程

• Step1: Positional Sentence Representation

将句子输入Bi-LSTM，每个词结点都对应两个隐藏层输出。用这个两个隐藏层输出代表每个词所在位置的位置向量表示。如模型图的橙色虚线部分表示。

• Step2: Interactions Between Two Sentences

如模型图所示，两个sentence中的每一对位置向量表示两两交互，文章提供了三种交互方式：

1. 余弦相似度Cosine

$$s(u, v) = \frac{u^T v}{||u|| \cdot ||v||}, \quad \text{双线性函数Bilinear**}$$

2. 双线性函数Bilinear

$$s(u, v) = u^T M v + b,$$

M矩阵初始化以后根据训练调整参数

3. 张量层TensorLayer

$$s(u, v) = f(u^T M^{[1:c]} v + W_{uv} \begin{bmatrix} u \\ v \end{bmatrix} + b),$$

where $M^i, i \in [1, \dots, c]$ is one slice of the tensor parameters, W_{uv} and b are parameters of the linear part. f is a non-linear function, and we use **rectifier** $f(z) = \max(0, z)$ (Glorot,

结果是TensorLayer效果较好，结合了前两种方式，paper里采用的也是TensorLayer

• Step3: Interaction Aggregation

在得到了交互矩阵的基础上，即模型图中的Interaction Tensor，接着再进行下面两部操作：

1.k-Max Pooling:

相当于是选出交互矩阵中最大的K个元素，构成向量q

原文描述：Specifi-cally for the interaction matrix, we scan the whole matrix and the top k values are directly returned to form a vector q according to the descending order.

2. 多层感知机MLP：

将k-Max Pooling的结果q传入MLP，计算出最终的Matching Score，记为s。s的计算过程如下：

$$r = f(W_r q + b_r), \quad s = W_s r + b_s,$$

首先将k-Max Pooling的结果q传入一个全连接层得到向量r，再对r做一次线性变换即可，得到s即最终的matching score

