Bimpm

发表年份：2017

题目：Bilateral Multi-Perspective Matching for Natural Language Sentences

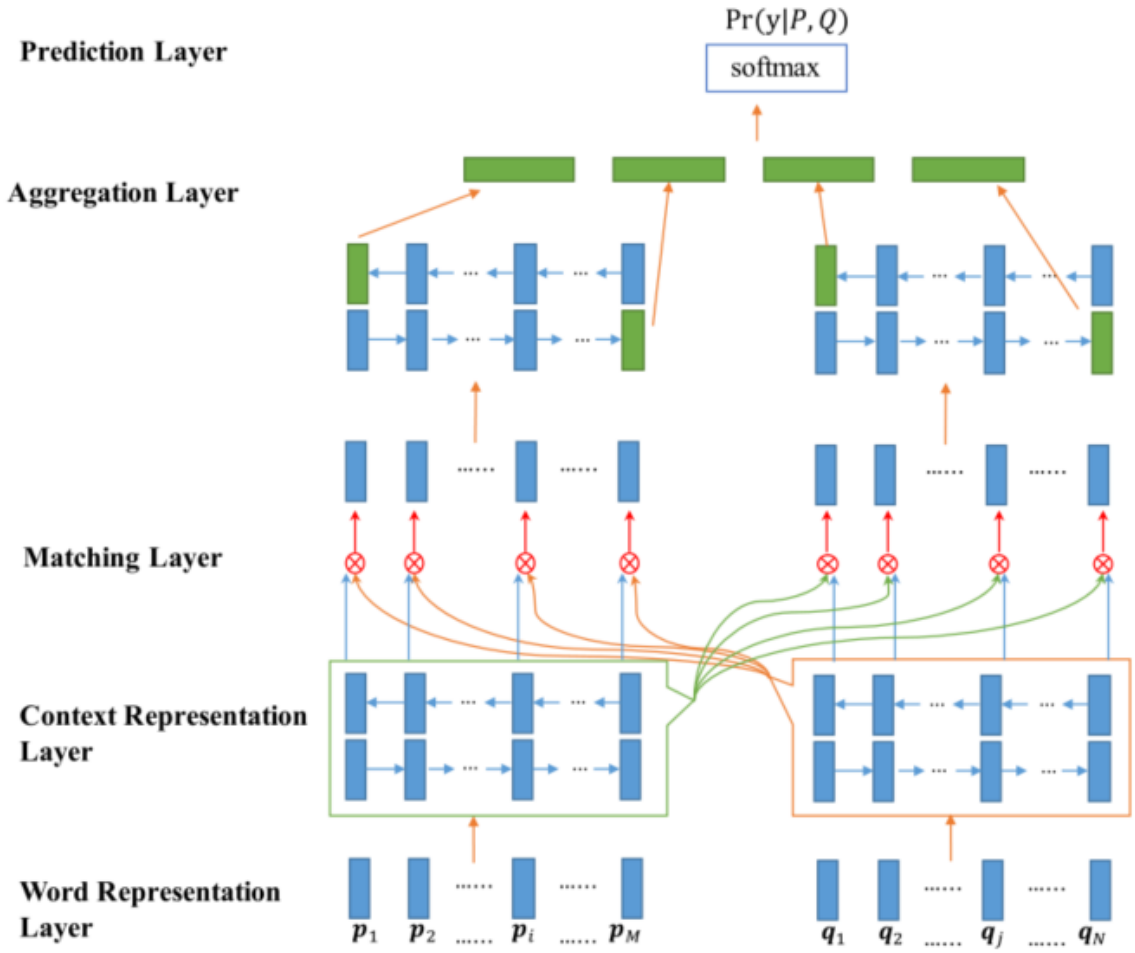
类型：compare-aggregate

1. 概述

Bilateral Multi-perspective Matching(BiMPM)是一个基于“matching-aggregation”框架的语义匹配模型。对于输入的两个句子P和Q，在采用预训练语言模型embedding后，模型在表示层采用双向的LSTM分别对P,Q进行representation。然后分别在两个方向进行matching，即P-->Q和Q-->P，同时结合四种匹配方式，这就是multi-perspective matching。将matching结果输入双向LSTM进行aggregation，然后经过全连接层softmax的到结果。此论文的核心就在matching的地方，在representation和aggregation其实并没有什么新意，后面将着重介绍四种matching的方式，以及multi-perspective的思想。

2. 模型

2.1 模型结构



2.2 模型详解

- **Word Representation Layer**（这个感觉无所谓，现在都用bert表示就完了）

  这一层采用预训练的word embedding, 将每个词转化为一个向量，对于OOV采用随机初始化的方式。同时还有char embedding, char embedding对于每一个词将其中的每个字母输入LSTM，采用LSTM的最后一个输出作为char embedding, word embedding 和 char embedding进行拼接作为最后的word representation。

- **Context Representation Layer**

  这一层采用双向LSTM来做内容表示，两个双向LSTM分别对P和Q做处理。这里的**两个LSTM是同一个网络，共享参数的**。之前看过一篇文章解释为什么两个网络需要共享参数，因为需要把两个句子放到同一个向量空间，才有比较的意义。这里也不确定这解释对不对，后面我会再研究下。这层没什么特殊的，需要每个时间步的输出以及最后一步的输出。

$$\begin{aligned} \overrightarrow{\boldsymbol{h}}_i^p &= \overrightarrow{\mathrm{LSTM}}(\overrightarrow{\boldsymbol{h}}_{i-1}^p, \boldsymbol{p}_i) \qquad i = 1, ..., M \\ \overleftarrow{\boldsymbol{h}}_i^p &= \overleftarrow{\mathrm{LSTM}}(\overleftarrow{\boldsymbol{h}}_{i+1}^p, \boldsymbol{p}_i) \qquad i = M, ..., 1 \end{aligned} \tag{1}$$

Meanwhile, we apply the same BiLSTM to encode Q:

$$\begin{aligned} \overrightarrow{\boldsymbol{h}}_j^q &= \overrightarrow{\mathrm{LSTM}}(\overrightarrow{\boldsymbol{h}}_{j-1}^q, \boldsymbol{q}_j) \qquad j = 1, ..., N \\ \overleftarrow{\boldsymbol{h}}_j^q &= \overleftarrow{\mathrm{LSTM}}(\overleftarrow{\boldsymbol{h}}_{j+1}^q, \boldsymbol{q}_j) \qquad j = N, ..., 1 \end{aligned} \tag{2}$$

- **MatchingLayer（重点）**

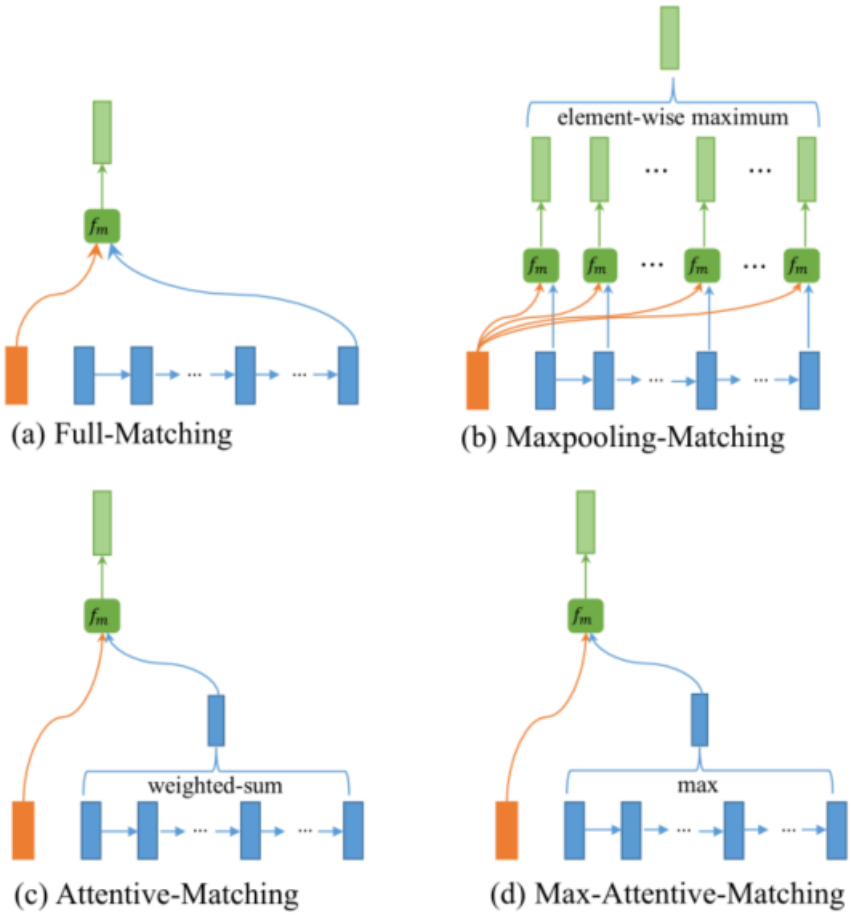
  matching layer是这篇论文的精华中的精华了。这里采用了双向多角度匹配（Bilateral multi-perspective matching），下面来具体解释下这个匹配方式。首先作者定义了一个新的cosine function ：*multi-perspective cosine matching function*

$$m = f_m(v1, v2; W)$$

$$m_k = cosine(Wk \circ v1, Wk \circ v2)$$

$$W \in R^{l \times d}$$

  上面这个公式其实很简单，首先先解释一下这个W, W其实就是**multi-perspective**，W的维度是$l \times d$的，其中l就是你有多少个perspective，d就是向量v_1和v_2的维度。公式f就是consine function，在计算向量v_1和v_2的cosine相似度的时候，分别乘上向量W_k为v_1,v_2附上权重，一共有l个perspective，也就意味着需要做l次cosine相似度的计算，最后生成一个$m = [m_1, \ldots, m_k, \ldots, m_l]$。一句话就是这个多角度的cosine相似度不再是只有一个结果，而是一个维度为l的向量，向量中的每个元素都是特定perspective下的结果。



- **Full-Matching**

  这个匹配策略是对于P中BiLSTM的每个时间步与Q中BiLSTM的最后一个时间步计算相似度（既有前向也有后向），然后Q的每个时间步与P的最后一个时间步计算相似度：

$$m_i^{\rightarrow full} = f_m(h_i^{\rightarrow p}, h_n^{\rightarrow q}; W^1)$$

$$m_i^{\leftarrow full} = f_m(h_i^{\leftarrow p}, h_1^{\leftarrow q}; W^2)$$

- **Maxpooling-Matching**

  这个匹配策略对于P中BiLSTM的每个时间步与Q中BiLSTM的每个时间步分别计算相似度，然后只返回最大的一个相似度：

$$m_i^{\rightarrow max} = \max_{j \in (1...N)} f_m(h_i^{\rightarrow p}, h_j^{\rightarrow q}; W^3)$$

$$m_i^{\leftarrow max} = \max_{j \in (1...N)} f_m(h_i^{\leftarrow p}, h_j^{\leftarrow q}; W^4)$$

- **Attentive-Matching**

  这个匹配策略先计算P和Q中BiLSTM中每个时间步的cosine(传统的)相似度，

$$\alpha_{i,j}^{\rightarrow} = cosine(h_i^{p\rightarrow}, h_j^{q\rightarrow})$$

$$\alpha_{i,j}^{\leftarrow} = cosine(h_i^{p\leftarrow}, h_j^{q\leftarrow})$$

生成一个相关性矩阵，然后用这个相关矩阵计算Q的加权求和（如果是P-->Q），最后用P的每个时间步分别于Q的加权求和计算相似度：

$$h_i^{\rightarrow mean} = \frac{\sum_{j=1}^{N} \alpha_{i,j}^{\rightarrow} \cdot h_j^{\rightarrow q}}{\sum_{j=1}^{N} \alpha_{i,j}^{\rightarrow}}$$

$$h_i^{\leftarrow mean} = \frac{\sum_{j=1}^{N} \alpha_{i,j}^{\leftarrow} \cdot h_j^{\leftarrow q}}{\sum_{j=1}^{N} \alpha_{i,j}^{\leftarrow}}$$

- **Max-Attentive-Matching**

  这个和上面的attentive-matching很像，只不过这里不再是加权求和了，而是直接用cosine最大的embedding作为attentive vector，然后P的每个时间步分别于最大相似度的embedding求多角度cosine相似度

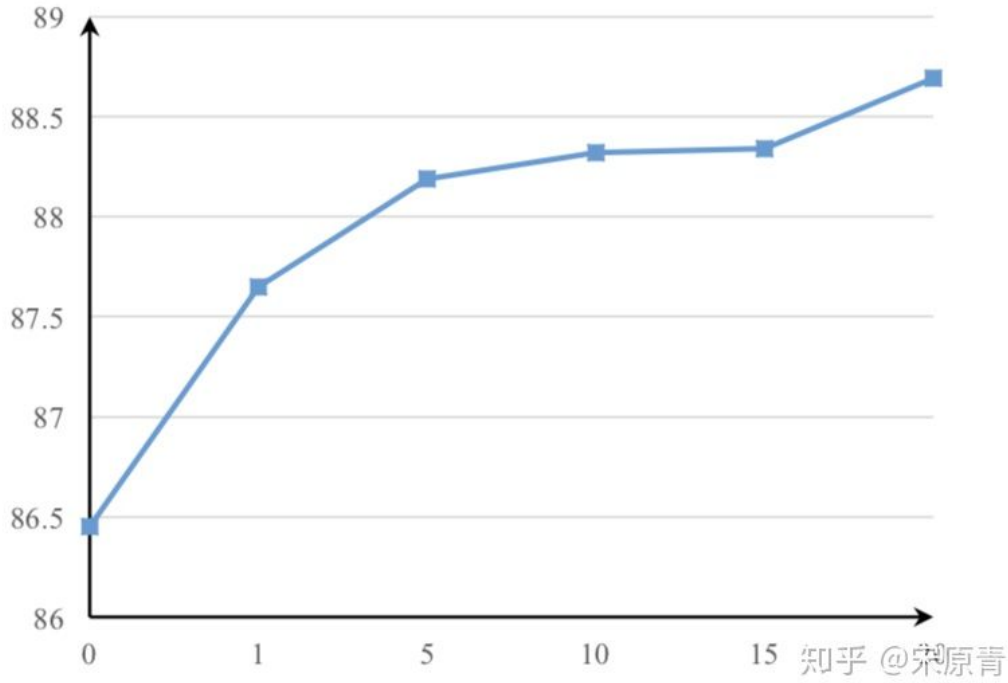
> 通过上面的双向多角度的matching，对于P的每个时间步，我们都能得到8个向量，然后我们把这8个项量拼接后作为P每个时间步的matching vector。

3. 论证方法

论文的逻辑认证还是非常的严谨的，这里主要讨论模型结构上的论证，主要从以下几方面展开论证：

- **证明perspective是有效果的**

  作者对于模型的perspective分别取{1, 5, 10, 15, 20}, 发现随着perspective上升模型的表现越来越好了：



- **证明双向以及四种匹配策略是否work**

  作者分别对单独使用两个方向观察模型表现，同时分别单独使用四种匹配模式观察模型表现：

Models	Accuracy
Only $P \rightarrow Q$	87.74
Only $P \leftarrow Q$	87.47
w/o Full-Matching	87.86
w/o Maxpooling-Matching	87.64
w/o Attentive-Matching	87.87
w/o MaxAttentive-Matching	87.98
Full Model	**88.69**

- **与其它传统模型比较**

  作者分别实现了孪生CNN以及孪生LSTM，同时又将四种mathing方法加到了两个传统模型中，发现对传统网络也有效果提升，进一步证明了新matching方法是有效果的。