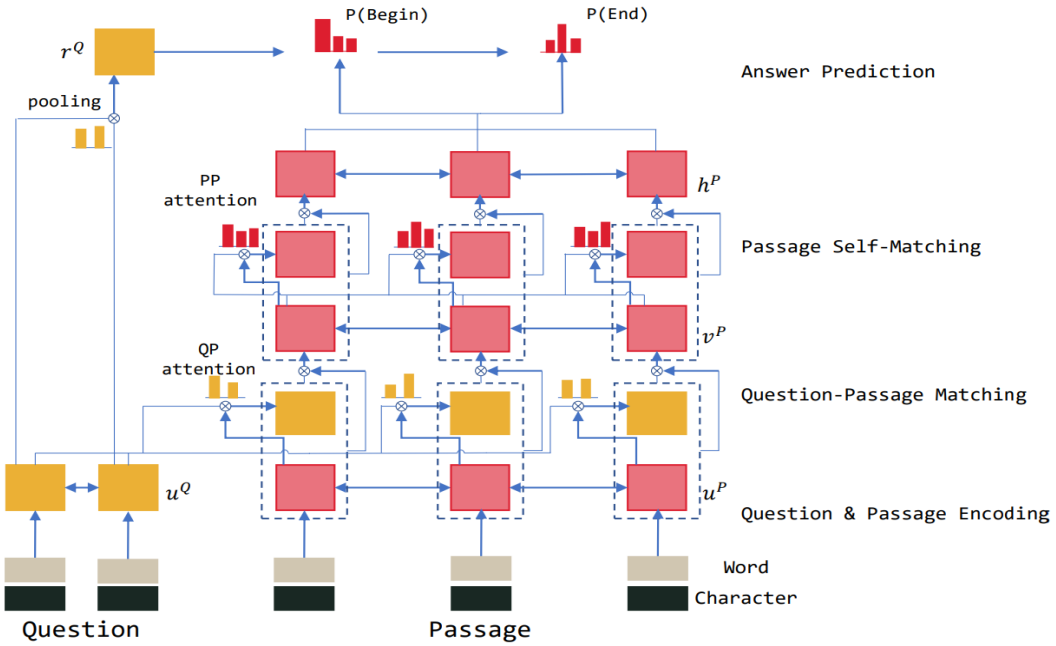
Week3

本章介绍了 Bert 之前比较经典的几个模型：R-net、FusionNet 和 QANet。同时介绍了两个前导内容 Multi-Hop 机制和 Pointer network。本笔记聚焦在 R-net 模型。

R-net 模型在注意力计算中加入门控机制。整体框架图如下：



整体流程分为编码层、两个交互层以及最后输出层：

1.编码层(Question & Passage Matching)

word 级别，通过 glove 模型作为嵌入层输出。Char(字符)级别通过 RNN 编码，把其最后位置的结果向量，作为 Char（字符）向量输出。拼接 word 级别向量和 char 级别向量，然后和上一个状态的 u_{t-1} 作为双向 RNN(用的是双向 GRU)的输入，生成 ut。此过程分别用于生成 passage 和 question 的 u_t。

$$u_t^Q = \mathrm{BiRNN}_Q(u_{t-1}^Q, [e_t^Q, c_t^Q])$$
$$u_t^P = \mathrm{BiRNN}_P(u_{t-1}^P, [e_t^P, c_t^P])$$

2.交互层(Question-Passage Matching)

此过程先是把 question 结果融入到 passage 中，采用 attention 的方式融入。同时采用门控机制，控制 u^P 和 c_t 的重要性。具体流程如下：

a. 通过把 u_t^P、v_{t-1}^P 和 u_j^Q 做类似相似度计算输出，得到 t 时刻 question 的第 j 个 u^Q 向量对应的 passage 的 S_j^t 向量；

b. 对 S_j^t 做 softmax 处理，得到 quesiton 中第 j 个 u^Q 对应 passage 的概率，重复这个过程，最终得到 t 时刻输入的 question 的 u^Q 所有向量对应 passage 的概率；

c. 和 t 时刻输入的 question 的 u^Q 向量做加权和处理，最终得到 t 时刻，question 对 passage 的注意力得分 c_t。

d. 把 t 时刻的 u^P 和 c_t 放入 sigmoid 函数中，得到门控概率，进而再得到通过门控概率的 u^P 和 c_t 的拼接向量。此门控机制能控制 u^P 和 c_t 的重要性；

e. 把 d 步得到的拼接向量和 v_{t-1} 放入 RNN(LSTM)中，输出 v_t^P；

最终得到 Question 和 Passage 匹配向量 v_t^P。整个过程公式见下图。

$$v_t^P = \mathrm{RNN}(v_{t-1}^P, c_t)$$

$$s_j^t = \mathrm{v}^{\mathrm{T}} \tanh(W_u^Q u_j^Q + W_u^P u_t^P + W_v^P v_{t-1}^P)$$
$$a_i^t = \exp(s_i^t) / \Sigma_{j=1}^m \exp(s_j^t)$$
$$c_t = \Sigma_{i=1}^m a_i^t u_i^Q$$

Match-LSTM $v_t^P = \mathrm{RNN}(v_{t-1}^P, [u_t^P, c_t])$

替换

$$g_t = \mathrm{sigmoid}(W_g[u_t^P, c_t])$$
$$[u_t^P, c_t]^* = g_t \odot [u_t^P, c_t]$$

3. 交互层(Passage Self-Matching)

在上一个交互层，所有 question 的内容已经融入到 passage 中了，在这个交互层，passage 做自我匹配，来得到 passage 中自己的重要性部分。具体流程如下：

a. 对不同位置的 v^P 做类似相似度计算，得到 t 时刻，passage 中第 j 个 v^P 向量对应的 v_t 的 S_j^t 向量；

b. 对 S_j^t 做 softmax 处理，得到 passage 中第 j 个 v^P 向量对应 v_t 的概率，重复这个过程，最终得到 t 时刻输入的 passage 的 v^P 所有向量对应 v_t 的概率；

c. 和 t 时刻所有输入的 passage 的 v^P 向量做加权和处理，最终得到 t 时刻，passage 对 v_t 的注意力得分 c_t；

d. 把 v_t^P 和 c_t 做拼接后的向量，和 h_{t-1}^P 一起输入到双向 RNN 中，得到结果 h_t^P；

最终得到 Passage 的自我匹配向量 h_t^P。整个过程公式见下图。

$$h_t^P = \mathrm{BiRNN}(h_{t-1}^P, [v_t^P, c_t])$$

$$s_j^t = \mathrm{v}^{\mathrm{T}} \tanh(W_v^P v_j^P + W_v^{\tilde{P}} v_t^P)$$
$$a_i^t = \exp(s_i^t) / \Sigma_{j=1}^n \exp(s_j^t)$$
$$c_t = \Sigma_{i=1}^n a_i^t v_i^P$$

4. 输出层

此层级融入 question，再用上一层级得到的 h_t^P 做注意力交互，最终获得开始和结束位置的最大概率。具体过程如下：

a. 把编码层得到的 u^Q 向量和一个参数向量 V_r^Q 做类似相似度计算，得到 t 时刻 question 的第 j 个 u^Q 向量对应 V_r^Q 的 s_j 向量，进而和之前的 attention 计算类似，最终得到 question 每个位置的注意力得分 r^Q；

b. 对 h^P 做自我相似度的计算，得到 t 时刻第 j 个 h^P 对应的 h_{t-1}^a 的相似向量 s_j，再对 s_j 做 softmax 处理，得到 t 时刻 s_j 的概率 a_i^t，重复此过程，进而得到 t 时刻所有 h^P 对 h_{t-1}^a 的概率 a_i^t，选择概率最大的位置 p^t 作为 t 时刻的输出；

c. 在这里需注意的是，b 过程只计算两个时刻，即 t=1 和 2，表示开始位置和结束位置。在 t=1 时，h_{t-1}^a 为 h_0^a 等于 a 步骤 r^Q，带入 b 步骤中，得到 p^1。同时，用 b 步骤中得到的所有 h^P 对 h_0^a 的概率 a_i^1，和 t=1 时刻所有的 h^P 做加权和，得到 t=1 时刻，h_0^a 对 h^P 注意力得分 c_1，再把 h_0^a 和 c_1 输入到 RNN 中，得到 h_1^a，进而重复之前 b 的计算，得到 t=2 时刻的 p^2；

此过程得到了 p^1 和 p^2 两个位置。整个过程公式见下图。

$$s_j^t = \mathrm{v}^{\mathrm{T}} \tanh(W_h^P h_j^P + W_{\mathrm{h}}^a h_{t-1}^a)$$
$$a_i^t = \exp(s_i^t) / \Sigma_{j=1}^n \exp(s_j^t)$$
$$p^t = \operatorname{argmax}(a_1^t, \ldots, a_n^t)$$

$$c_t = \Sigma_{i=1}^n a_i^t h_i^P$$
$$h_t^a = \mathrm{RNN}(h_{t-1}^a, c_t)$$

$$r^Q = att(u^Q, V_r^Q) \Rightarrow$$

$$s_j = \mathrm{v}^{\mathrm{T}} \tanh(W_u^Q u_j^Q + W_{\mathrm{v}}^Q V_r^Q)$$
$$a_i = \exp(s_i) / \Sigma_{j=1}^m \exp(s_j)$$
$$r^Q = \Sigma_{i=1}^m a_i u_i^Q$$

以上为 R-net 模型的全过程。最效果来看，此模型结果优于 BiDAF。