

Lecture10 Question Answering and the Default Final Project

SQuAD

一个问题的答案总是来自该段落的一系列单词序列——提取问题回答

SQuAD version1.1

- 收集了3个黄金答案
- 系统在两个指标上计算得分
 - 精确匹配：1/0的准确度，您是否匹配三个答案中的一个
 - F1：将系统和每个答案都视为词袋，并评估
- 这两个指标都忽略标点符号和冠词 (a , an the)

SQuAD version2

SQuAD1.0的一个缺陷是，所有问题都有答案的段落；而在version2中对于开发集和测试集，一半的问题有答案，一半的问题在文章中没有答案

对于没有答案的问题，无答案得分为1，其他答案得分为0

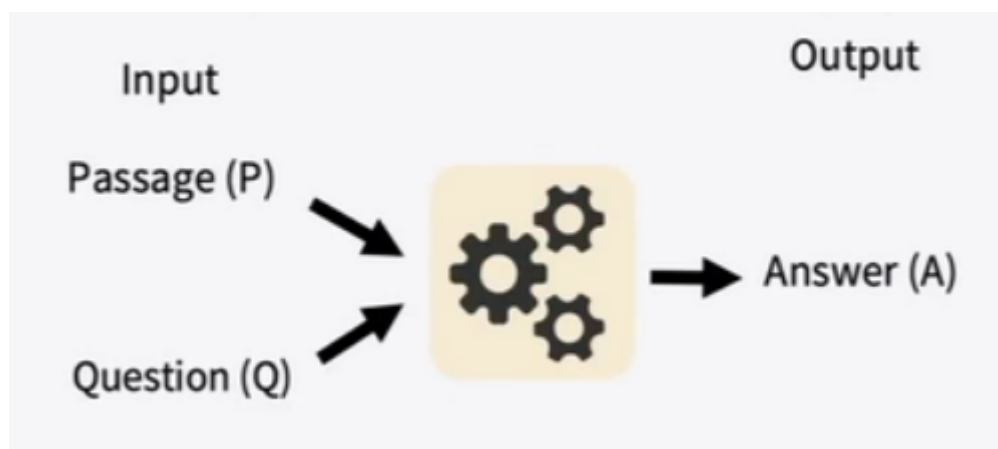
使用阈值来判断是否回答了一个问题

数据集结构良好，干净

limitation:

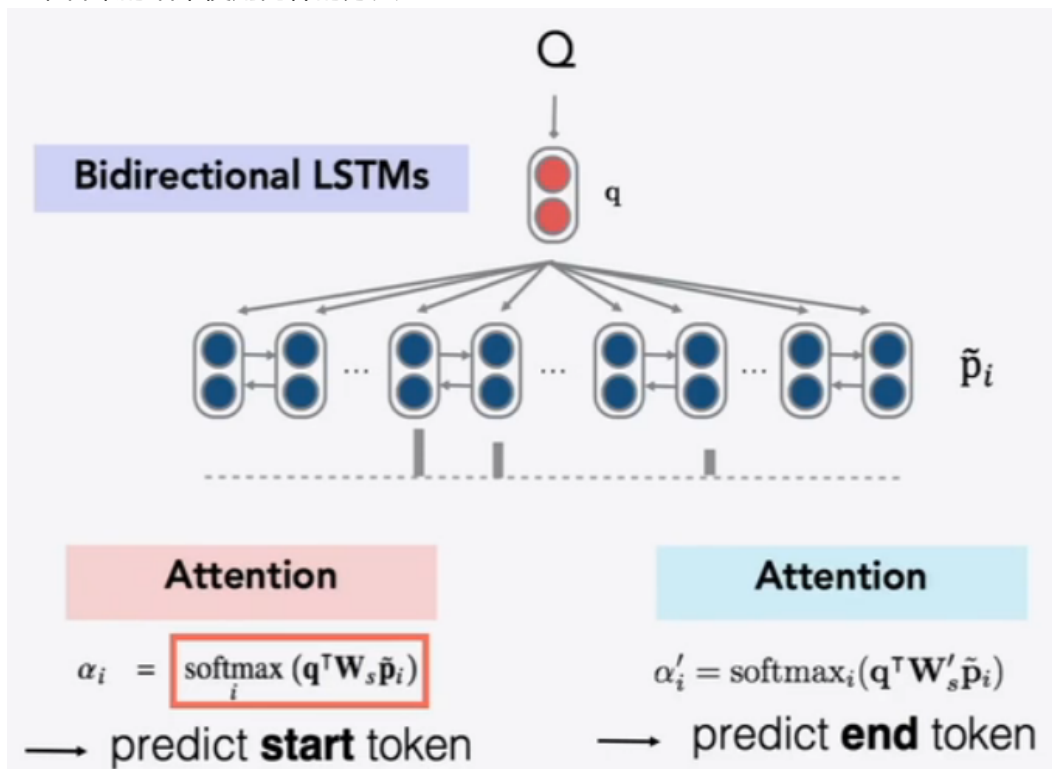
能提问的问题有限制：不能有yes/no问题，不能有计数问题，不能有困难的隐含问题

the Stanford Attentive Reader model



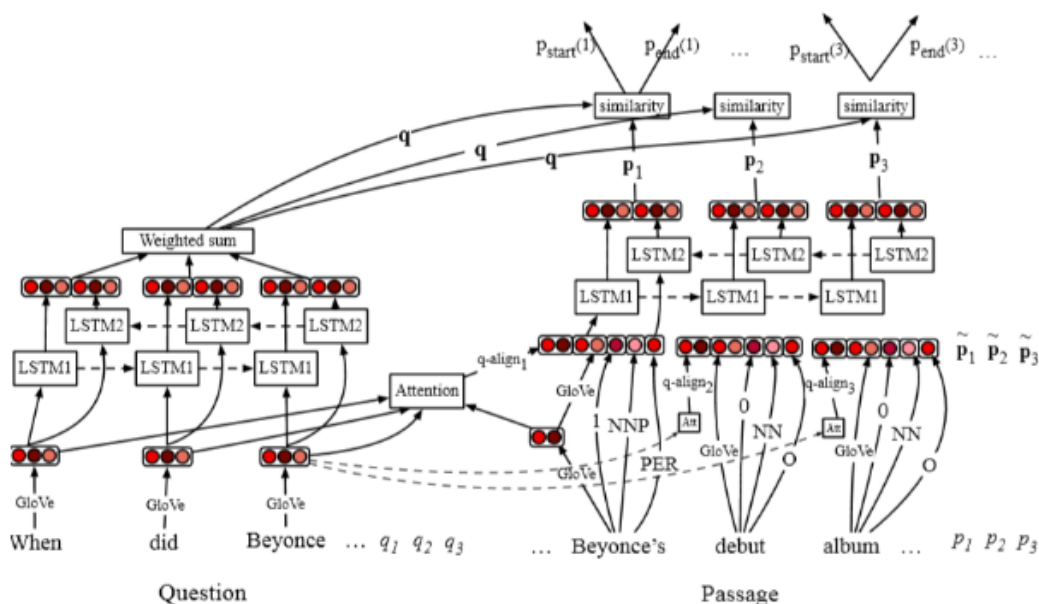
- 建立一个表示问题的向量：对于问题中的每个单词，找到一个word embedding特别的使用了GloVe-GloVe300维字嵌入，然后运行双向LSTM(每个维度为d)，得到两个LSTM的结果连接成2d维度的向量

- 查找每个单词的词嵌入并输入到双向LSTM中。对于每个单词的双向LSTM表示与问题表示，计算出一个注意力的得分，获得不同位置的注意力，从而获得答案的开始位置，答案的结束使用同样的方法。



双向线性注意在跨度开始时获得一个大的分数

Stanford Attentive Reader++



Training objective: $\mathcal{L} = - \sum \log P^{(\text{start})}(a_{\text{start}}) - \sum \log P^{(\text{end})}(a_{\text{end}})$

段落表示：

单词表示：不只使用GloVe还加入了语言特征以便运行命名实体识别器和词性标记器，同时还加入了词频；

完全匹配：对于段落中的每个单词，是否出现在问题中，有三种方式来完成：完全匹配、

忽略大小写匹配、词干匹配

使用嵌入相似性来计算问题与答案之间的一种相似

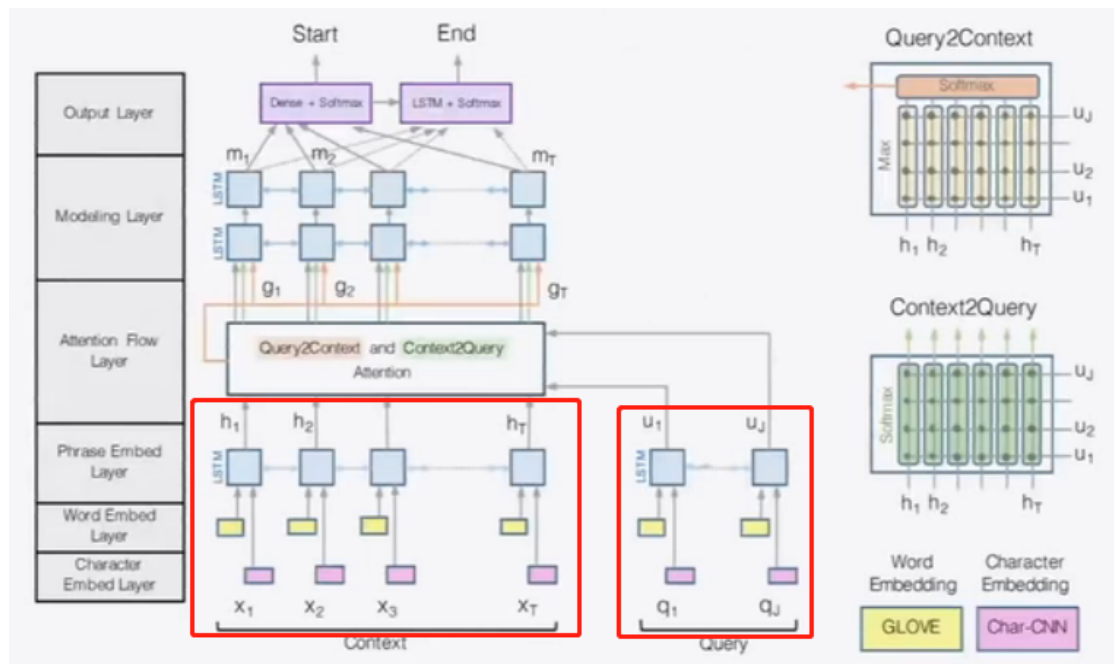
$$f_{\text{align}}(p_i) = \sum_j a_{i,j} \mathbf{E}(q_j) \quad q_{i,j} = \frac{\exp(\alpha(\mathbf{E}(p_i)) \cdot \alpha(\mathbf{E}(q_j)))}{\sum_{j'} \exp(\alpha(\mathbf{E}(p_i)) \cdot \alpha(\mathbf{E}(q'_j)))}$$

为什么神经网络模型性能更好？

单词相似度的语义匹配更好 或 与语义相关但不使用相同词语的改写

BiDAF

在SAT中，利用注意力来讲问题表示映射到段落文字中，但是可以通过在单次级别的两个方向进行映射来做得更好。在注意力流动的两个方向上，找到可以映射到问题单词的段落词和可以映射到段落词的问题单词，再运行另一轮的序列模型，就可以在两者之间做更好地匹配。



idea :

- 对于每个段落词和每个问题词，计算出相似性得分的方式是 建立一个大的连接向量，所以有段落词和问题词的LSTM表示 c_i, q_i ，并对它们做Hadamard product，将这个巨大的向量与学到的权重矩阵点积，得到在问题中的每个位置和上下文之间的相似度得分 S_{ij} ，并使用它来定义两个方向的注意力

$$S_{ij} = w_{\text{sim}}^T [c_i; q_i; c_i \circ q_j] \in \mathbb{R}$$

$$\alpha^i = \text{softmax}(S_{i,:}) \in \mathbb{R}^M \quad \forall i \in \{1, \dots, N\}$$

$$a_i = \sum_{j=1}^M \alpha_j^i q_j \in \mathbb{R}^{2h} \quad \forall i \in \{1, \dots, N\}$$

- 在反方向

$$m_i = \max_j S_{ij} \in \mathbb{R} \quad \forall i \in \{1, \dots, N\}$$

$$\beta = \text{softmax}(m) \in \mathbb{R}^N$$

$$c' = \sum_{i=1}^N \beta_i c_i \in \mathbb{R}^{2h}$$

这些做完之后将得到BiDAF layer的输出为

$$b_i = [c_i; a_i; c_i \circ a_i; c_i \circ c'] \in \mathbb{R}^{8h} \quad \forall i \in \{1, \dots, N\}$$

- 然后有一个“建模”层：2层的BiLSTM
 start：将BiDAF和建模层的输出连接到一个稠密FF层，然后softmax;
 end：将建模层M的输出通过另一个BiLSTM o give M，然后与BiDAF层连接，再次通过稠密FF层和一个softmax

Recent, more advanced crchitectures

FusionNet :

Attention functions

MLP (Additive) form:

$$S_{ij} = s^T \tanh(W_1 c_i + W_2 q_j)$$

Space: $O(mnk)$, W is $k \times d$

Bilinear (Product) form:

$$S_{ij} = c_i^T W q_j$$

$$S_{ij} = c_i^T U^T V q_j$$

Space: $O((m+n)k)$

$$S_{ij} = c_i^T W^T D W q_j$$

1. Smaller space
2. Non-linearity

$$S_{ij} = \text{Relu}(c_i^T W^T) D \text{Relu}(W q_j)$$

ELMo and BERT preview

产生上下文单词表示的算法，可以在特定的上下文中对每个单词进行表示