简单高效的多段落阅读理解
Simple and Effective Multi-Paragraph Reading Comprehension

Guo Tianyi

2018年6月22日

大纲

简介

- 传统的信息检索方法可以轻松定位包含答案的文档；
- 神经模型的成功应用表明它有成为解决问题的方法的重要部件的潜力 (Wang et al., 2017b; Tan et al., 2017)；
- 然而将整篇文档作为输入进行训练与测试会导致极端的计算复杂度。

简介

多段阅读理解的两种基础解决方案：

- **流水线方法** 从文档中选择一个段落，传递到段落模型进行答案提取 (Joshi et al., 2017; Wang et al., 2017a)
- **基于置信度的方法** 在多段上使用模型，返回具有最高置信度的答案 (Chen et al., 2017)

置信度方法对不太复杂的段落选择过程具有一定的鲁棒性，但是需要一个能够生成精确的段落置信度的段落模型。

简介

本文的**流水线方法**着力于解决文档规模数据的训练带来的问题：

- 提出了一个基于TF-IDF的启发式方法，用于选择在哪些段落上进行训练与测试；
- 使用求和目标函数来对答案出现的所有位置进行边缘化；
- 并且采用了一些较新的解决方案，例如自注意力机制 (Cheng et al., 2016)，双向注意力机制 (Seo et al., 2016)等。

简介

先前的流水线方法使用称为“先验”的段落进行问答，这种方式有几个不足之处：(1)不能为不包含答案的段落生成较低的置信度；(2)目标函数不包含置信度，无法在段落之间进行比较。

本文的**置信度方法**对上述方法进行扩展以更好地适应多段落问答：

- 从上下文文档中采样段落进行训练，其中包括不含答案的段落；
- 使用共享标准化目标函数对所有的独立处理的段落进行合并处理；

本文的模型在TriviaQA数据集上 (Joshi et al., 2017)进行了验证。模型在测试集上达到了71.3的F1值，相比之前的工作有15%的提升。
本文还对流水线方法进行了模型简化测试（消融测试），同时还将本文的模型与TriviaQA中使用的检索机制进行组合，构建了一个端到端的问答系统原型。
本文模型代码在GitHub上公开：
`https://www.github.com/allenai/document-qa`

大纲

流水线方法

段落选择策略

选择与问题的TF-IDF余弦距离最小的段落作为候选段落。

噪音处理策略

通过最优化目标函数$-\log(\frac{\sum_{k\in A} e^{s_k}}{\sum_{i=1}^{n} e^{s_i}})$来处理由于远程监督策略导致的噪音。

模型架构

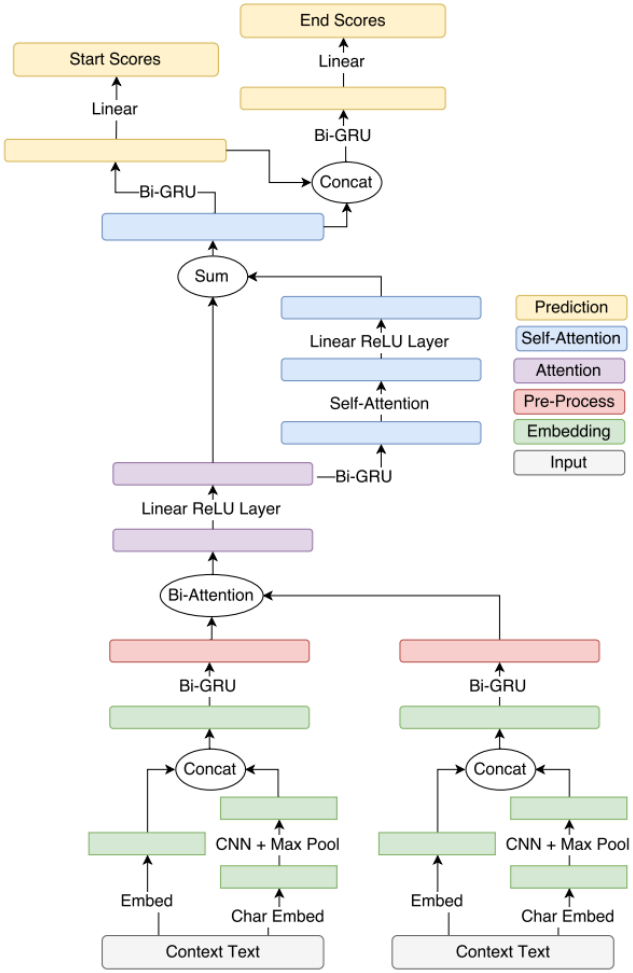


图: 流水线模型的整体架构

模型包含嵌入层、预处理层、注意力层、自注意力层与预测层。

- 嵌入层使用词嵌入与字符嵌入连接的方式生成嵌入；
- 预处理层用共享的双向GRU(Cho et al., 2014)对嵌入进行处理；
- 注意力层使用BiDAF模型建立一个查询相关的上下文表示；
- 自注意力层对上一层的输出进行一次自注意力，与输出求和；
- 预测层再次进行双向GRU处理，最后预测答案的范围

大纲

本文对前述模型进行修改以适应于多段文本阅读理解任务。处理文本的时候，从文本段落中进行采样来进行处理。将开始位置与结束位置的打分进行求和作为一个答案区域的得分。

潜在的问题

对各段独立处理得到的置信度在归一化时的系数不统一会导致偏差。

造成这种问题的原因推测是各段独立进行了softmax归一化，导致段落之间的置信度的相对大小关系发生了变化。
文章提供了四种解决该问题的方案：

- 共享归一化方法
- 合并方法
- “没有答案”选择方法
- Sigmoid方法

共享归一化

共享归一化的方法仍然是独立处理所有段落，然而对归一化方式进行了修改：

$$\frac{e^{s_{ap}}}{\sum_{j\in P}\sum_{i=1}^{n_j} e^{s_{ij}}}$$

其中P是所有段落，s_{ij}是打分。
这种方法相当于简单地将所有段落连接在一起进行打分与归一化处理，强制保持段落间的置信度的相对大小关系。

合并

该方法在训练时就将段落合并在一起，在段落之间添加一个预先训练好的分隔符。整体作为一段来进行训练与打分。

“没有答案”选择

该方法允许模型对每段增加一个“没有答案”选项。
对开始与结束标签的目标函数加以修改：

$$
\begin{aligned}
& -\log\left(\frac{e^{s_a}}{\sum_{i=1}^{n} e^{s_i}}\right) - \log\left(\frac{e^{g_b}}{\sum_{j=1}^{n} e^{g_j}}\right) \\
= & -\log\left(\frac{e^{s_a g_b}}{\sum_{i=1}^{n}\sum_{j=1}^{n} e^{s_i g_j}}\right) \\
\rightarrow & -\log\left(\frac{(1-\delta)e^{z} + \delta e^{s_a g_b}}{e^{z} + \sum_{i=1}^{n}\sum_{j=1}^{n} e^{s_i g_j}}\right)
\end{aligned}
$$

其中答案存在，δ是1，否则为0。
计算z的方法是在模型最后增加一层额外层来计算。

Sigmoid

最后一种方法是考虑使用sigmoid函数来作为目标函数。通过对开始/结束得分进行sigmoid操作，并不涉及段落内的归一化，于是这样得到的置信度仍然是可比较的。

大纲

数据集

本文分别在TriviaQA unfiltered (Joshi et al., 2017)，TriviaQA web和SQuAD (Rajpurkar et al., 2016)上对自己的模型进行了评估。

TriviaQA unfiltered数据集是一个大规模的远程监督的开放域问答数据集。TriviaQA web是在TriviaQA unfiltered的基础上增加了文本是否存在答案这一字段用来进行阅读理解训练的阅读理解数据集。

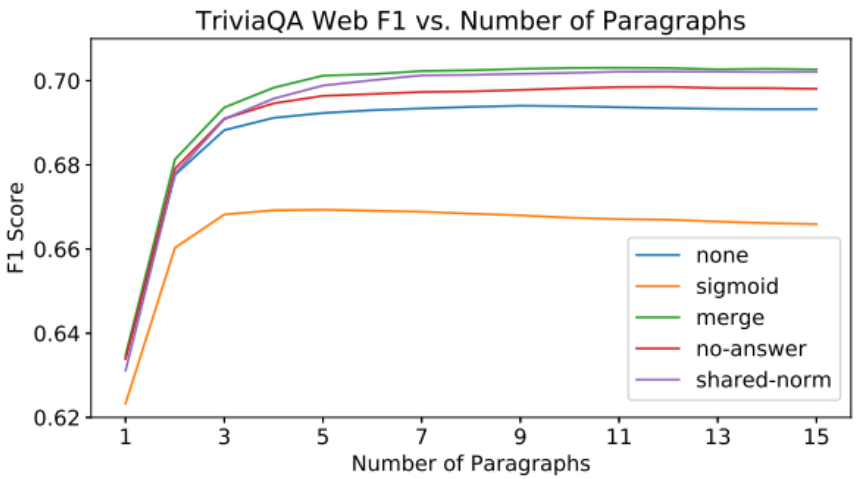
SQuAD数据集是一个维基百科文章与众包问题回答的数据集。

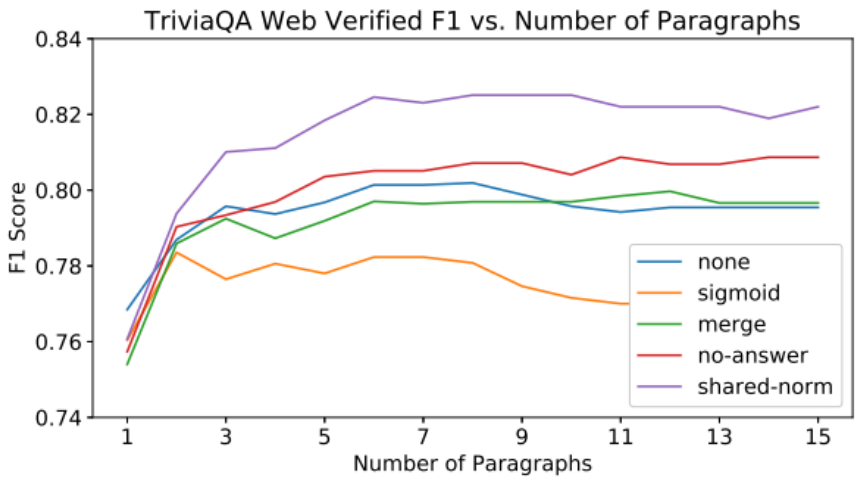
处理方法

- **预处理**：不进行下采样；对文本归一化进行修正处理；合并小段落直到一个指定范围，同时插入分隔符保留格式信息。
- **采样**：SQuAD与TriviaQA web数据集依据各段TF-IDF进行采样；TriviaQA unfiltered使用线性函数对段落进行打分，然后在打分并排序后的段落中进行采样。

大纲

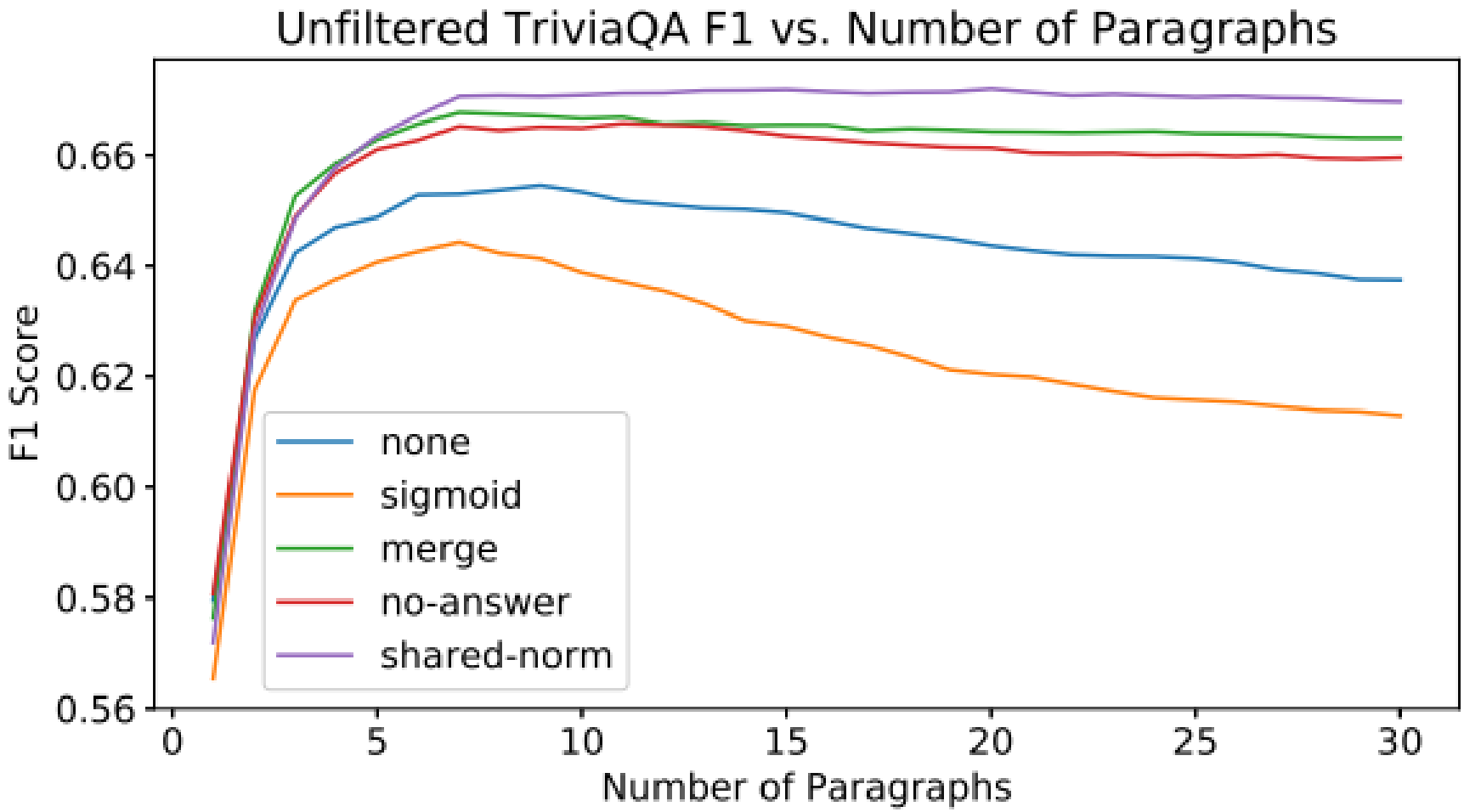
TriviaQA Web数据集





Model	All		Verified	
	EM	F1	EM	F1
baseline(Joshi et al., 2017)	40.74	47.06	49.54	55.80
MEMEN* (Pan et al., 2017)	43.16	46.90	49.28	55.83
Mnemonic Reader (Hu et al., 2017)	46.94	52.85	54.45	59.46
Reading Twice for NLU (Weissenborn et al., 2017a)	50.56	56.73	63.20	67.97
S-Norm (ours)	66.37	71.32	79.97	83.70

TriviaQA Unfiltered数据集



SQuAD数据集

Model	Dev		Test	
	EM	F1	EM	F1
none	71.60	80.78	72.14	81.05
sigmoid	70.28	79.05	-	-
merge	71.20	80.26	-	-
no-answer	71.51	80.71	-	-
shared-norm	71.16	80.23	-	-

讨论

1. 只在包含答案的段落上进行训练的模型在多段落配置上表现很差，在SQuAD数据集上表现尤其差；
2. 共享归一化的方法整体表现是最好的
3. “没有答案”与合并方法表现中等，但是并没有解决先前提到的softmax问题
4. sgimoid方法表现最差，推测因为它对噪音非常敏感

大纲

相关工作

- 阅读理解数据集：
  第一个大规模完形填空阅读理解数据集(Hermann et al., 2015; Hill et al., 2015)；
  SQuAD(Rajpurkar et al., 2016)；
  WikiReading(Hewlett et al., 2016)；
  MS Marco(Nguyen et al., 2016)；
  TriviaQA(Joshi et al., 2017)；
  Quasar-T(Dhingra et al., 2017)
- 神经阅读理解模型：
  变分Dropout(Gal and Ghahramani, 2016)
  双向注意力(Seo et al., 2016)
  自注意力(Cheng et al., 2016; Wang et al., 2017b; Pan et al., 2017)等

相关工作

- 开放域问答：
  TREC问答(Voorhees et al., 1999)
  使用知识库的问答(Berant et al., 2013)
  YodaQA(Baudiš, 2015)

大纲

结论

针对段落级别的多段问答模型，结合本文的共享归一化策略可以有非常好的表现。将其与本文的段落选择策略和求和目标函数可以大幅提升模型在TriviaQA上的表现。