NBCE：使用朴素贝叶斯扩展LLM的Context处理长度

23 May
By 苏剑林 | 2023-05-23 | 19863位读者

> 在LLM时代还玩朴素贝叶斯（Naive Bayes）？

这可能是许多读者在看到标题后的首个想法。确实如此，当古老的朴素贝叶斯与前沿的LLM相遇时，产生了令人惊讶的效果——我们可以直接扩展现有LLM模型的Context处理长度，无需对模型进行微调，也不依赖于模型架构，具有线性效率，而且效果看起来还不错——这就是本文所提出的NBCE（**N**aive **B**ayes-based **C**ontext **E**xtension）方法。

摸石过河

假设T为要生成的token序列，$S_1, S_2, \cdots, S_n$是给定的若干个相对独立的Context集合（比如n个不同的段落，至少不是一个句子被分割为两个片段那种），假设它们的总长度已经超过了训练长度，而单个S_k加T还在训练长度内。我们需要根据$S_1, S_2, \cdots, S_n$生成T，即估计$p(T|S_1, S_2, \cdots, S_n)$。

简单来说，朴素贝叶斯就是“贝叶斯公式+独立假设”。根据贝叶斯公式：

$$p(T|S_1, S_2, \cdots, S_n) \propto p(S_1, S_2, \cdots, S_n|T)p(T) \tag{1}$$

这里的$\propto$，是省去了与T无关的常数因子。根据（条件）独立假设：

$$p(S_1, S_2, \cdots, S_n|T) = \prod_{k=1}^n p(S_k|T) \tag{2}$$

所以有

$$p(T|S_1, S_2, \cdots, S_n) \propto p(T)\prod_{k=1}^n p(S_k|T) \tag{3}$$

再次根据贝叶斯公式$p(S_k|T) \propto \frac{p(T|S_k)}{p(T)}$，得到

$$p(T|S_1, S_2, \cdots, S_n) \propto \frac{1}{p^{n-1}(T)}\prod_{k=1}^n p(T|S_k) \tag{4}$$

或者

$$\log p(T|S_1, S_2, \cdots, S_n) = \sum_{k=1}^n \log p(T|S_k) - (n-1)\log p(T) + \text{常数} \tag{5}$$

这里的$p(T|S_k)$和$p(T)$都可以直接用现有的LLM进行计算，而且只要是语言模型都行，跟架构无关，也不需要用长文本微调。其中，$p(T|S_k)$是单个Context所预测的概率，$p(T)$则无Context（或者Context为空）的概率，并且多个Context可以放在同一个batch中并行计算，计算量随着Context数的增加是线性增长的。

抽丝剥茧

当然，朴素贝叶斯依赖于独立假设，这会限制它的实际效果。为了“青出于蓝而胜于蓝”，我们不妨将式(5)进一步“抽丝剥茧”、“去芜存菁”，以达到更好的效果。

首先我们记$\log p(T|S) = [\log p(T|S_1), \cdots, \log p(T|S_n)]$，以及

$$\overline{\log p(T|S)} = \frac{1}{n}\sum_{k=1}^{n} \log p(T|S_k) \tag{6}$$

并设$\beta = n - 1$，那么式(5)可以重写为

$$\log p(T|S_1, S_2, \cdots, S_n) = (\beta + 1)\overline{\log p(T|S)} - \beta \log p(T) + 常数 \tag{7}$$

重写为上述形式后，自然而言地引出了两个问题：

> 1、如果将β作为超参数来调，是否可能取得更好的效果？
>
> 2、$\overline{\log p(T|S)}$就是$\log p(T|S)$的Average Pooling，那么换成其他Pooling方法（简记为$\mathcal{P}$）是否有更好的效果？即
>
> $$\log p(T|S_1, S_2, \cdots, S_n) = (\beta + 1)\mathcal{P}[\log p(T|S)] - \beta \log p(T) + 常数 \tag{8}$$

于是笔者在7B模型上围绕这两个问题进行调试，得到的初步结论是：在阅读理解场景中Max Pooling配合$\beta = 0.25$，用Greedy Search总体表现比较好，然而Random Sample出来的结果基本不可读。

最终方案

为什么会出现Greedy Search好而Random Sample差的情况呢？我们知道，Random Sample是“按照分布采样”，它的效果差说明Max Pooling的结果不是一个合理的分布；而Greedy Search只关心最大概率者，而不关心分布的合理性，它的效果好告诉我们概率最大的token正确性较高。

概率越大说明不确定性越低，所以为了改善Random Sample的效果，我们将Pooling方式改为直接输出不确定性最低的那个分布：

$$\begin{aligned}
&\mathcal{P}[\log p(T|S)] = \log p(T|S_k)\\
&k = \arg\min\left\{H_1, H_2, \cdots, H_n\right\}\\
&H_i = -\sum_T p(T|S_i)\log p(T|S_i)
\end{aligned} \tag{9}$$

代入到式(8)，就是最终的NBCE（**N**aive **B**ayes-based **C**ontext **E**xtension）。

值得指出的是，虽然我们的出发点是朴素贝叶斯，但一般化后的式(8)已经超出了常规的朴素贝叶斯的范畴，同时保留了朴素贝叶斯的可解释性。不难看出，式(8)的形式很是直观：

> 1、不同Context的预测结果通过方法$\mathcal{P}$聚合（或者说投票）在一起（权重为$\beta + 1$），并减去无Context的预测结果（权重为β）；
>
> 2、之所以要减去无Context预测结果，是为了让模型更加倾向于结合Context而不是纯粹根据自身知识储备来回答（注：3天后出现在Arxiv的论文《Trusting Your Evidence: Hallucinate Less with Context-aware Decoding》也提

出了相同的技巧用来减少幻觉）；

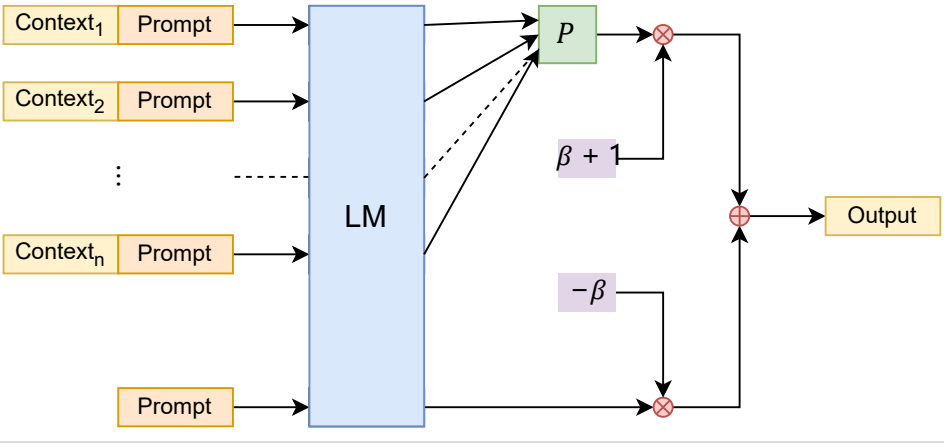
3、不同场景可以选择不同的β，比如需要结合Context做阅读理解的，可以考虑较大的β，如果偏向于自由创作，则选择较小的β，笔者认为$\beta \geq -1$都是合理的。

参考实现

下面给出NBCE的参考实现：

> Github: **https://github.com/bojone/NBCE**

从演示代码可以看出，NBCE的实现很简单，只需要修改一下解码函数中的logits构建方式，跟解码算法的选择并不冲突。



Naive Bayes-based Context Extension（NBCE）示意图

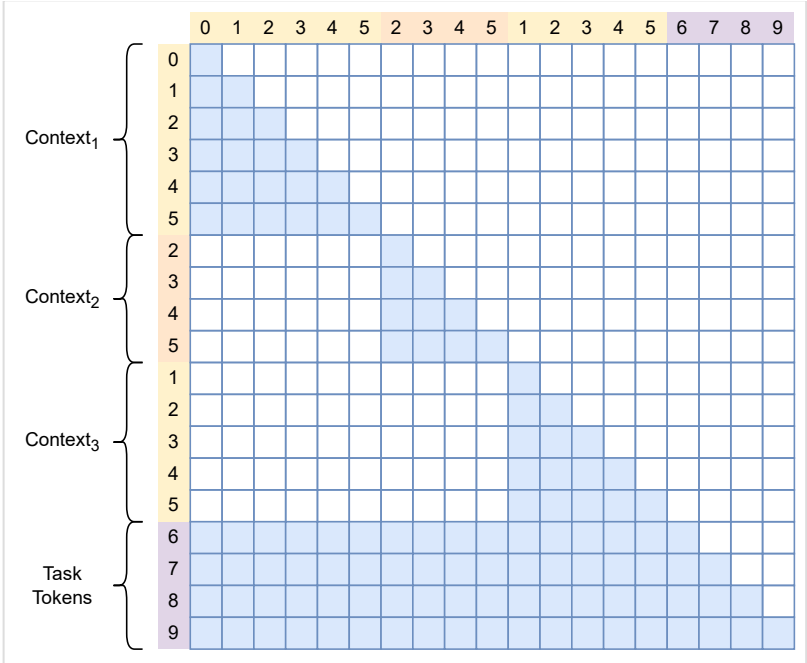
所给的Demo包含12段不同的Context，总长度为9000多字，连同8个问题一次性输入到模型中（模型训练长度为2048，参数量为7B，可以在OpenBuddy下载），模型能够逐一根据所给Context正确回答这8个问题。值得指出的是，所有的Context、问题和答案加起来，超过了1万字！另外，有朋友简单尝试了简历匹配和作文打分应用，效果也尚可，非常建议大家亲自调试一下。

相关工作

扩展LLM的Context长度其实已有不少，但多数是通过结合检索或者摘要的方式来缩短样本的长Context，如Unlimiformer。由于不是直接处理长Context，因此通常无法做精细的阅读理解，而且这些方案往往需要在训练阶段就考虑进去，而不是事后即插即用到已有的LLM模型中。

在NBCE之前，能够不微调地扩展Context长度的方案是Parallel Context Window（下面简称PCW），出自论文《Parallel Context Windows for Large Language Models》和《Structured Prompting: Scaling In-Context Learning to 1,000 Examples》，两篇论文是同一时期不同作者的工作，但所提的方法只有细微的差别，因此这里都将它们叫做PCW。

PCW适用于Self Attention模型，主要修改包括Position Encoding和Attention Mask，如下图所示：



Parallel Context Window

首先确定Context的最大长度L（图中为6），然后每个Context的最后一个位置编码为$L-1$，倒数第二个位置编码为$L-2$，...，依此类推，这种编码方式我们称为“右对齐”（或者“左缩进”）；另一边，对于Task Tokens部分（Prompt+生成内容），我们的位置编码是$L, L+1, L+2, \cdots$。每个Context单独编码，所以对应的Attention Mask是分块对角矩阵，而因为是LM，所以是分块对角下三角阵；至于Task Tokens部分需要结合所有的Context，所以它需要Attention到所有Context（以及它自身）。这样一来，如果将每个Context单独拿出来，和Task Tokens拼在一起，其Attention模式就跟原本的LM一致了。

或许有读者看出，其实NBCE跟PCW有着很相似的特性，比如对于Context都是无序的、平权的。事实上，如果将NBCE应用到单层单头注意力模型中，那么结果大致上就是PCW。为了显示这一点，我们写出单层单头注意力的语言模型为

$$p(x_t|x_{<t}) = softmax\left(\sum_{i=1}^{t} a_{t,i} v_i W\right) \tag{10}$$

所以大致上有$\log p(x_t|x_{<t}) \sim \sum_{i=1}^{t} a_{t,i} v_i W$，接着代入到式$(7)$并取$\beta=0$，得到

$$\log p(T|S_1, S_2, \cdots, S_n) \sim \frac{1}{n}\sum_{k=1}^{n}\left(\sum_{i\in S_k} a_{T,i} v_i\right) W = \left(\sum_{i\in S_1\oplus\cdots\oplus S_n} \frac{a_{T,i}}{n} v_i\right) W \tag{11}$$

这里假设的是T是单个token，但其实已经不失一般性了，$\oplus$是拼接的意思。在上式中，$S_k \oplus T$是作为一个连续片段来推理的（NBCE的设定），所以它们的位置编码相邻，而$a_{T,i}/n$构成了T与所有S_i的一个整体Attention（求和同样是1），这些特性跟PCW其实是一致的，PCW只不过是以Attention Mask的方式更优雅地整合到每一层中。

因此，PCW大致上就是Average Pooling版的NBCE，我们实测也发现它跟Average Pooling版的NBCE有着相似的缺点——当Context数据增加时，输出的结果开始不够准确，具体表现为主题相关，但是作为问题的答案来说是错误的。

延伸思考

NBCE的一大缺点是无序性，即无法识别Context的输入顺序，这在续写故事等场景可能表现欠佳。为了缓解这一点，可以考虑在每一个Context前面加个能指示序信息的prefix，就好比小说中的“第一章”、“第二章”那样。

总的来说，目前笔者关于NBCE的测试都限于“阅读理解”场景，即“理解”长文本，能否用此方法来“生成”长文本，还是个未知数，期待大家的测试结果。

此外，还有一个有意思的问题是：

> 既然朴素贝叶斯都能在LLM领域能派上用场，那么其他传统概率模型（比如HMM）是否也能在LLM领域有它们的一席之地呢？

文章小结

本文提出了NBCE（Naive Bayes-based Context Extension），它基于朴素贝叶斯思想来扩展LLM的Context处理长度，有着即插即用、模型无关、无须微调、线性效率、实现简单等优点，并且看上去效果还不错，欢迎大家测试。