Summary of MegaByte

沙之洲 2020012408

为了解决当前transformer模型对输入的限制，本篇工作提出的MetaByte将模型的输入序列分割成固定大小的patches，然后通过一个 global model，建立一个大的自回归transformer，把输入和输出从tokens变成patches。同时，引入了一个 local model，用于每个patch内部的字节的预测，其输入是从全局模块来的上下文patches表示结果，输出是预测下一个patch，这是一个小的自回归模型。

本篇工作的优点在于，提出了一种不同于 token 输入输出的范式，也即将 Byte 作为输入输出。用 Byte 的方式取代了传统的基于 BPE 的 tokenize 的方式，向更高效的模型无损压缩迈出了坚实的一步。

本篇工作的不足之处在于，patch size 在这篇工作中只有 8 Bytes，映射到 token 上的话只有 2 个左右的 token。这会导致两个问题，第一 patch size 带来的计算加速并不能抵消不用 token 带来的 Byte 长度的增长，也就是说 Byte + patch 并没有对模型的效率带来提高，第二，由于 local model 能看到的 sequence 太短，导致 local model 的能力将成为 scale up 的瓶颈。

一些可能的改进方案和课上讨论内容的总结。首先，local model 的能力限制实际上是来源于看到 sequence 长度太小，当不了解当前 query 的问题重点的时候，local model 不知道应该从提取什么样的信息。如果我们考虑人类的阅读过程，当我们读完一个自然段之后，即使不知道问题是什么，我们也能够比较好地提炼出这个段落的重点。所以如果我们能够将 local model 的 sequence 长度提升到几千的级别，再利用 global model 进行 hierarchy 的 attention，是否可以缓解 local model 的能力瓶颈呢？

其次，虽然 MegaByte 提出了基于 Byte 的无损压缩算法，但是并没有针对于此进行更好的压缩结构设计。类比于人类学习的过程，我们可以通过将字典作为辅助，帮助模型对 Byte 进行压缩。不过我因此产生了一个疑问，字典作为人类学习语言的重要且关键的工具，为什么字典迟迟没有进入 LLM 的训练当中呢？