[RAG] Meta-Chunking: 通过逻辑感知学习高效的文本分段

原创 简单的机器学习 简单的机器学习 2024年11月22日 09:55 浙江

简单的机器学习
深度学习、视觉、OCR、NLP、标注工具，一起加油吧！
35篇原创内容

公众号

本文是由人大提出的，旨在解决在检索增强生成（RAG）系统中，文本分段这一关键方面被忽视的问题。具体来说，传统文本分段方法（如基于规则或语义相似性）在捕捉句子间深层语言逻辑联系方面存在不足，导致在知识密集型任务（如开放域问答）中的性能受到影响。本文通过引入Meta-Chunking的概念及其两种实现策略（边际采样分段和困惑度分段），解决了以下几个关键问题：

逻辑连贯性问题：

- 问题：传统文本分段方法往往基于规则或语义相似性，难以捕捉句子间的深层逻辑联系（如因果、过渡、并行和渐进关系）。
- 解决方案：Meta-Chunking通过利用LLMs的强大理解和推理能力，设计了边际采样分段和困惑度分段策略，精确识别文本分段边界，确保分段后的文本块具有逻辑连贯性。

资源和时间效率问题：

- 问题：现有的文本分段方法（如LumberChunker）需要使用高性能的LLMs（如Gemini模型），导致资源和时间成本显著增加。
- 解决方案：边际采样分段有效减少了文本分段对模型大小的依赖，使推理能力相对较弱的小型语言模型也能胜任此任务。困惑度分段进一步提高了处理效率，实现了资源和时间的节省。

细粒度和粗粒度分段的平衡问题：

- 问题：仅通过调整阈值来控制块大小有时会导致块大小不均匀，难以满足用户的多样化分段需求。
- 解决方案：提出了一种结合Meta-Chunking与动态合并的策略，旨在灵活应对不同的分段要求，在细粒度和粗粒度文本分段之间取得有效平衡。

长文本处理问题：

- 问题：处理较长文本时，传统的分段方法可能导致上下文连贯性丧失或GPU内存溢出。
- 解决方案：在困惑度分段中引入键值（KV）缓存机制，在保持句子间逻辑连贯性的前提下计算困惑度，从而优化GPU内存和计算准确性。

跨语言适应性问题：

- 问题：小模型在跨语言适应性方面存在局限性，难以直接应用于多语言文本分段。
- 解决方案：通过实验验证，中等规模的模型（如1.5B参数级别）在处理不同长度的文本分段时能在性能和效率之间保持出色平衡。

通过上述解决方案，本文提出的Meta-Chunking方法显著提升了基于RAG的单跳和多跳问答性能，同时在效率和成本节约方面表现出优越性能，解决了传统文本分段方法在逻辑连贯性、资源和时间效率、细粒度和粗粒度分段平衡、长文本处理以及跨语言适应性等方面的不足。

Meta-Chunking

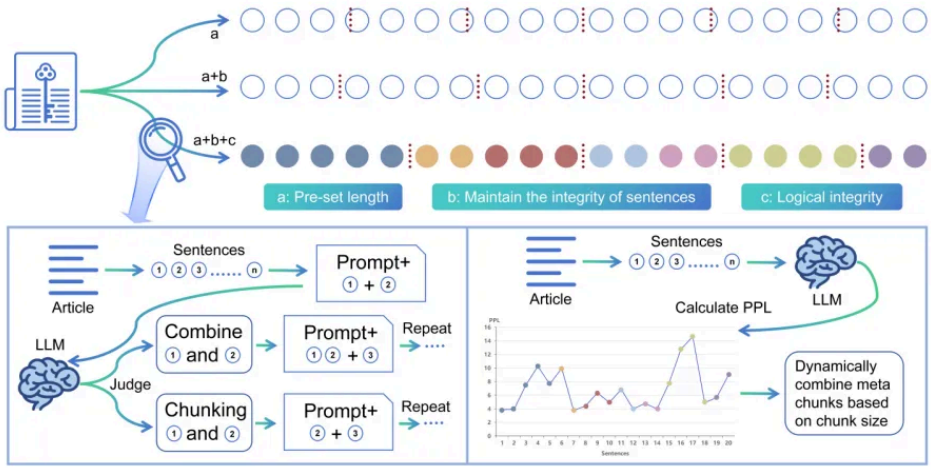

Figure 2: Overview of the entire process of Meta-Chunking. Each circle represents a complete sentence, and the sentence lengths are not consistent. The vertical lines indicate where to segment. The two sides at the bottom of the figure reveal Margin Sampling Chunking and Perplexity Chunking. Circles with the same background color represent a meta-chunk, which is dynamically combined to make the final chunk length meet user needs.

Meta-Chunking是一种创新文本分段技术，利用LLMs的能力灵活地将文档分割成逻辑连贯的独立块。方法是基于一个核心原则：允许块大小的可变性，以更有效地捕捉和保持内容的逻辑完整性。这种粒度的动态调整确保每个分段块包含一个完整且独立的表达，从而避免分段过程中逻辑链的中断。这不仅增强了文档检索的相关性，还提高了内容清晰度。

如上图所示，方法整合了传统文本分段策略的优势，如遵守预设块长度约束和确保句子结构完整性，同时在分段过程中增强了保证逻辑连贯性的能力。关键在于引入了一个介于句子级和段落级文本粒度之间的新概念：Meta-Chunking。一个元块由段落中顺序排列的句子集合组成，这些句子不仅共享语义相关性，更重要的是包含深层语言逻辑联系，包括但不限于因果、过渡、并行和渐进关系。这些关系超越了单纯的语义相似性。为了实现这一目标，论文中设计和实现了以下两种策略。

边际采样分段

给定一段文本，初始步骤将其分割成一系列句子，记为$(x_1, x_2, \ldots, x_n)$，最终目标是进一步将这些句子分割成若干块，形成新集合$(X_1, X_2, \ldots, X_k)$，每个块包含原始句子的连贯分组。该方法可以表述为：

$$\text{Margin}_M(x_i) = P_M(y = k_1 | \text{Prompt}(x_i, X')) - P_M(y = k_2 | \text{Prompt}(x_i, X'))$$

其中(k_1, k_2)表示二分类决策，$\text{Prompt}(x_i, X')$表示在$x_i \in \{x_l\}_{l=1}^{n}$和$X'$之间形成指令，关于它们是否应合并，其中$X'$包含单个句子或多个句子。通过模型$M$获得的概率$P_M$，我们可以推导出两个选项之间的概率差异$\text{Margin}_M(x_i)$。随后，通过将$\text{Margin}_M(x_i)$与阈值$\theta$进行比较，可以得出两个句子是否应分段的结论。对于$\theta$的设置，我们最初将其赋值为0，然后通过记录历史的$\text{Margin}_M(x_i)$并计算其平均值进行调整。

困惑度分段

同样，论文中将文本分割成句子，并使用模型计算每个句子x_i基于前面句子的困惑度：

$$\text{PPL}_M(x_i) = \frac{\sum_{k=1}^{K} \text{PPL}_M(t_k^i | t_{<k}^i, t_{<i})}{K}$$

其中K表示x_i中的总token数，t_k^i表示x_i中的第k个token，$t_{<i}$表示所有在x_i之前的token。为了定位文本分段的关键点，算法进一步分析$\text{PPL}_{seq} = (\text{PPL}_M(x_1), \text{PPL}_M(x_2), \ldots, \text{PPL}_M(x_n))$的分布特征，特别是识别最小值：

$$\text{Minima}_{index}(\text{PPL}_{seq}) = \left\{ i \;\middle|\; \min(\text{PPL}_M(x_{i-1}), \text{PPL}_M(x_{i+1})) - \text{PPL}_M(x_i) > \theta, or\ \text{PPL}_M(x_{i-1}) - \text{PPL}_M(x_i) > \theta\ and\ \text{PPL}_M(x_{i+1}) = \text{PPL}_M(x_i) \right\}$$

这些最小值被视为潜在的块边界。如果文本超出LLMs或设备的处理范围，论文策略性地引入键值（KV）缓存机制。具体来说，文本首先根据token分成若干部分，形成多个子序列。随着困惑度计算的进行，当GPU内存即将超过服务器配置或LLMs的最大上下文长度时，算法适当地移除先前部分文本的KV对，从而不会牺牲太多的上下文连贯性。

困惑度分段的理论分析

LLMs旨在学习一个分布￥Q￥，使其接近样本文本的经验分布P。为了量化这两个分布之间的接近程度，通常使用交叉熵作为度量。在离散场景下，Q相对于P的交叉熵正式定义如下：

$$H(P,Q) = \mathbb{E}_p[-logQ] = -\sum_x P(x) logQ(x) = H(P) + D_{KL}(P||Q)$$

其中$H(P)$表示经验熵，$D_{KL}(P||Q)$是Q和P之间的Kullback-Leibler（KL）散度。LLMs的困惑度在数学上定义为：

$$\text{PPL}(P,Q) = 2^{H(P,Q)}$$

需要注意的是，由于$H(p)$是不可优化的且有界，真正影响不同LLMs困惑度计算差异的是KL散度，它作为评估分布差异的度量。KL散度越大，两个分布之间的差异越大。此外，高困惑度表明LLMs对真实内容的认知幻觉，这些部分不应被分段。

另一方面，Shannon（1951）通过函数近似任何语言的熵：

$$G_K = -\sum_{T_k} P(T_k) log_2 P(t_k | T_{k-1}) = -\sum_{T_k} P(T_k) log_2 P(T_k) + \sum_{T_{k-1}} P(T_{k-1}) log_2 P(T_{k-1})$$

其中T_k表示文本序列中的k个连续token $(t_1, t_2, \ldots, t_k)$，熵可以表示为：

$$H(P) = \lim_{K \to \infty} G_K$$

然后，基于论文附录A.1中的证明，$G_{K+1} \leq G_K$对所有$K \geq 1$成立，可以推导出：

$$G_1 \geq G_2 \geq \cdots \geq \lim_{K \to \infty} G_K = H(P)$$

通过上面的公式可以观察到对于大规模文本处理任务，增加上下文长度往往会降低交叉熵或困惑度，这一现象反映了LLMs在捕获更广泛的上下文信息后进行更有效的逻辑推理和语义理解的能力。

实验

论文在十一个数据集上的广泛实验，验证了Meta-Chunking策略在提升基于RAG的单跳和多跳问答性能方面的有效性。具体数据请参看原论文。

> 论文地址：https://arxiv.org/pdf/2410.12788
>
> github： https://github.com/IAAR-Shanghai/Meta-Chunking