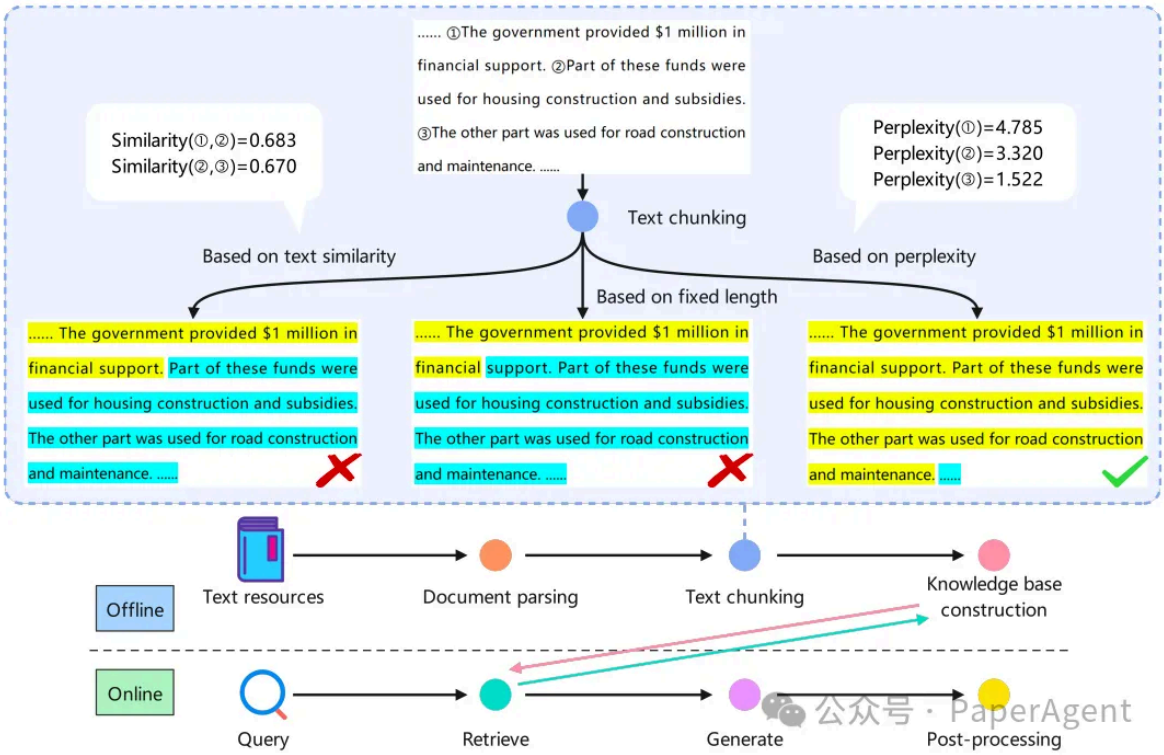
又快又准的RAG Meta-Chunking，1.3倍提升，耗时减半

深度学习与NLP 2024年10月21日 00:01 北京

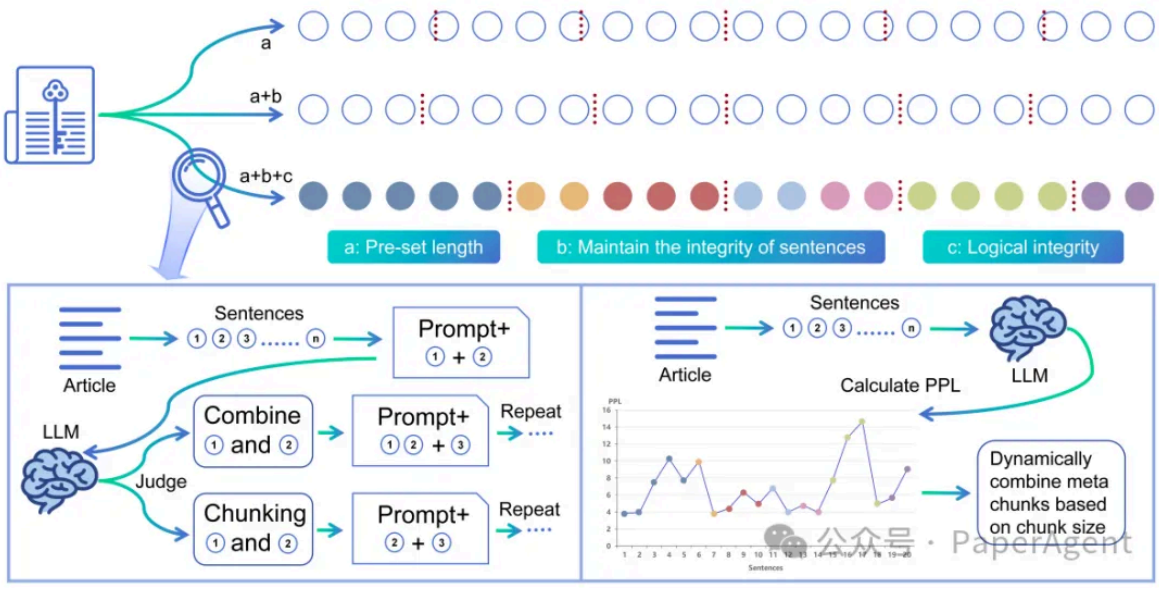
RAG效果在很大程度上依赖于检索到的文档的相关性和准确性。传统的基于规则或语义相似性的文本分块方法在捕捉句子间微妙的逻辑关系上存在不足。

RAG流水线的概览，以及基于规则、相似性和PPL分割的示例。相同的背景色表示位于同一个块中。

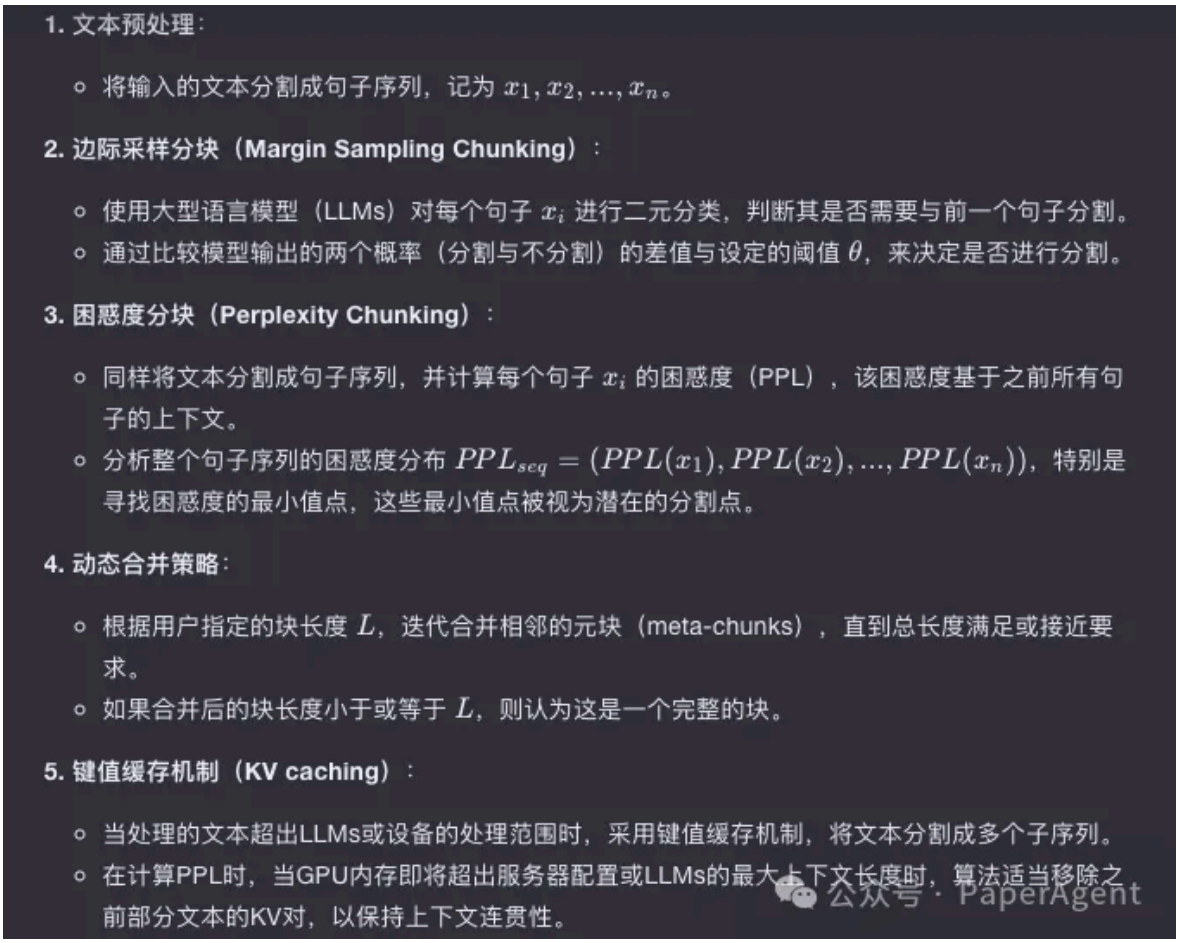


为了解决现有方法的局限性，提出了一种名为**Meta-Chunking**的概念，它在句子和段落之间定义了一种粒度：**由段落内具有深层语言逻辑联系的句子集合组成**，旨在增强文本分割过程中的逻辑连贯性。Meta-Chunking包括基于LLMs的两种策略：**边际采样分块**（Margin Sampling Chunking）和**困惑度分块**（Perplexity Chunking）。

整个元块分割（Meta-Chunking）过程的概览。每个圆圈代表一个完整的句子，句子的长度并不一致。垂直线表示在哪里进行分割。图底部的两侧揭示了边缘采样分割（Margin Sampling Chunking）和困惑度分割（Perplexity Chunking）。具有相同背景色的圆圈代表一个元块，它们被动态组合以使最终的块长度满足用户需求。



Meta-Chunking的工作流程：



通过在11个数据集上的实验，证明了**Meta-Chunking**能够更有效地提升基于RAG的单跳和多跳问答任务的性能。例如，在2WikiMultihopQA数据集上，它比**相似性分块（Similarity Chunking）**的性能提高了**1.32倍**，同时**只消耗了45.8%的时间。**

在五个问答（QA）数据集中展示了主要的实验结果。前四个数据集来源于LongBench。sent. 表示是否适合将两个句子分开，而chunk表示后一个句子是否适合与前面的块合并。comb. 指的是首先使用PPL分割（PPL Chunking）且阈值为0来分割文本，然后进行动态组合的过程。

Dataset	2WikiMultihopQA		Qasper		MultiFieldQA-en		MultiFieldQA-zh		MultiHop-RAG			
Chunking Method	F1	Time	F1	Time	F1	Time	F1	Time	Hits@10	Hits@4	MAP@10	MRR@10
Baselines with rule-based or similarity-based chunking												
Original	11.89	0.21	9.45	0.13	29.89	0.16	22.45	0.06	0.6027	0.4523	0.1512	0.3507
Llama_index	11.74	8.12	10.15	5.81	28.30	6.25	21.85	5.53	0.7366	0.5437	0.1889	0.4068
Similarity Chunking	12.00	416.45	9.93	307.05	29.19	318.41	22.39	134.80	0.7232	0.5362	0.1841	0.3934
Margin Sampling Chunking based on different models												
Pythia-0.16B$_{sent.}$	13.14	478.91	9.15	229.68	31.19	273.10	-	-	0.6993	0.5069	0.1793	0.3773
Pythia-0.41B$_{sent.}$	11.86	926.29	9.76	498.46	29.30	545.15	-	-	0.7259	0.5596	0.1934	0.4235
Qwen2-0.5B$_{sent.}$	11.74	788.30	9.67	599.97	31.28	648.76	23.35	480.35	0.7162	0.5246	0.1830	0.3913
Qwen2-1.5B$_{sent.}$	11.18	1908.25	10.09	1401.30	32.19	1457.31	22.27	1081.64	**0.7805**	**0.6089**	**0.2106**	**0.4661**
Qwen2-7B$_{sent.}$	**13.22**	7108.37	10.58	5207.87	32.32	5316.62	23.24	4212.00	0.6993	0.5197	0.1794	0.3835
Qwen2-1.5B$_{chunk}$	11.30	2189.29	9.49	1487.27	32.81	1614.01	22.08	1881.15	0.7109	0.5517	0.1970	0.4252
Qwen2-7B$_{chunk}$	12.94	8781.82	**11.37**	5755.79	**33.56**	6287.31	**24.24**	5084.95	0.7175	0.5415	0.1903	0.4141
Perplexity Chunking based on different models												
Internlm2-1.8B$_{comb.}$	12.37	355.53	10.02	200.69	30.81	251.06	22.53	161.15	0.7237	0.5499	0.1897	0.4121
Qwen2-1.5B$_{comb.}$	13.32	190.93	9.82	122.44	31.30	136.96	22.57	107.94	**0.7366**	0.5570	0.1979	0.4300
Baichuan2-7B$_{comb.}$	12.98	858.99	**10.04**	569.72	**32.55**	632.80	**23.36**	569.72	0.7206	**0.5636**	**0.2048**	**0.4406**
Qwen2-7B$_{comb.}$	**13.41**	736.69	9.39	486.48	32.35	523.74	22.81	424.96	0.7215	0.5521	0.1967	0.4229

此外，还探讨了不同模型大小对文本分块任务的影响，发现中型模型（如**1.5B参数级别**）在性能和效率之间展现了更平衡的表现。

```
META-CHUNKING: LEARNING EFFICIENT TEXT SEGMENTATION VIA LOGICAL PERCEPTION
https://arxiv.org/pdf/2410.12788
https://github.com/IAAR-Shanghai/Meta-Chunking.
```

来源 | PaperAgent

RAG专栏 38

RAG专栏 · 目录

上一篇
o1推理扩展的风吹到了RAG，性能飙升58.9%！

下一篇
告别文档解析，VisRAG带飞RAG，性能飙升37%