传统分块已死？Agentic Chunking拯救语义断裂，实测RAG准确率飙升40%，LLM开发者必看！

原创 longyunfeigu AI 博物院 2025年02月20日 08:18 江苏

最近公司处理LLM项目的同事咨询了我一个问题：明明文档中多次提到同一个专有名词，RAG却总是漏掉关键信息。排查后发现，问题出在传统的分块方法上——那些相隔几页却密切相关的句子，被无情地拆散了。我给了一些通用的建议，比如使用混合检索代替单一的语义检索，基于chunk生成QA对等等。接着他又提出了一个问题，有没有通过分块技术能减少这类问题的发生？我说你也可以试试最近新提出的一种分块策略：**Agentic Chunking**.

为什么分块如此重要?

在RAG模型中，文本分块是第一步，也是最关键的一步。传统的分块方法，比如**递归字符分割（Recursive character splitting）**，虽然简单易用，但它有一个明显的缺点：它依赖于固定的token长度进行分割，这可能导致一个主题被分割到不同的文本块中，从而破坏了上下文的连贯性。

另一种常见的分块方法是**语义分割（semantic splitting）**，它通过检测句子之间的语义变化来进行分割。这种方法虽然比递归字符分割更智能，但它也有局限性。比如，当文档中的话题来回切换时，语义分割可能会将相关内容分割到不同的块中，导致信息不连贯。

比如遇到下面这种场景时，它们就会集体失灵：

> *"小明介绍了Transformer架构...（中间插入5段其他内容）...最后他强调，Transformer的核心是自注意力机制。"*

传统方法要么把这两句话拆到不同区块，要么被中间内容干扰导致语义断裂。而人工分块时，我们自然会将它们归为"模型原理"组——**这种跨越文本距离的关联性，正是Agentic Chunking要解决的**。

Agentic Chunking的工作原理

Agentic Chunking的核心思想是让大语言模型（LLM）主动评估每一句话，并将其分配到最合适的文本块中。与传统的分块方法不同，Agentic Chunking不依赖于固定的token长度或语义变化，而是通过LLM的智能判断，将文档中相隔较远但主题相关的句子归入同一组。

举个例子，假设我们有以下文本：

```
On July 20, 1969, astronaut Neil Armstrong walked on the moon. He was leading the NASA’s Apol
```

在Agentic Chunking中，LLM会将这些句子进行**propositioning**处理，即将每个句子独立化，确保每个句子都有自己的主语。处理后的文本如下：

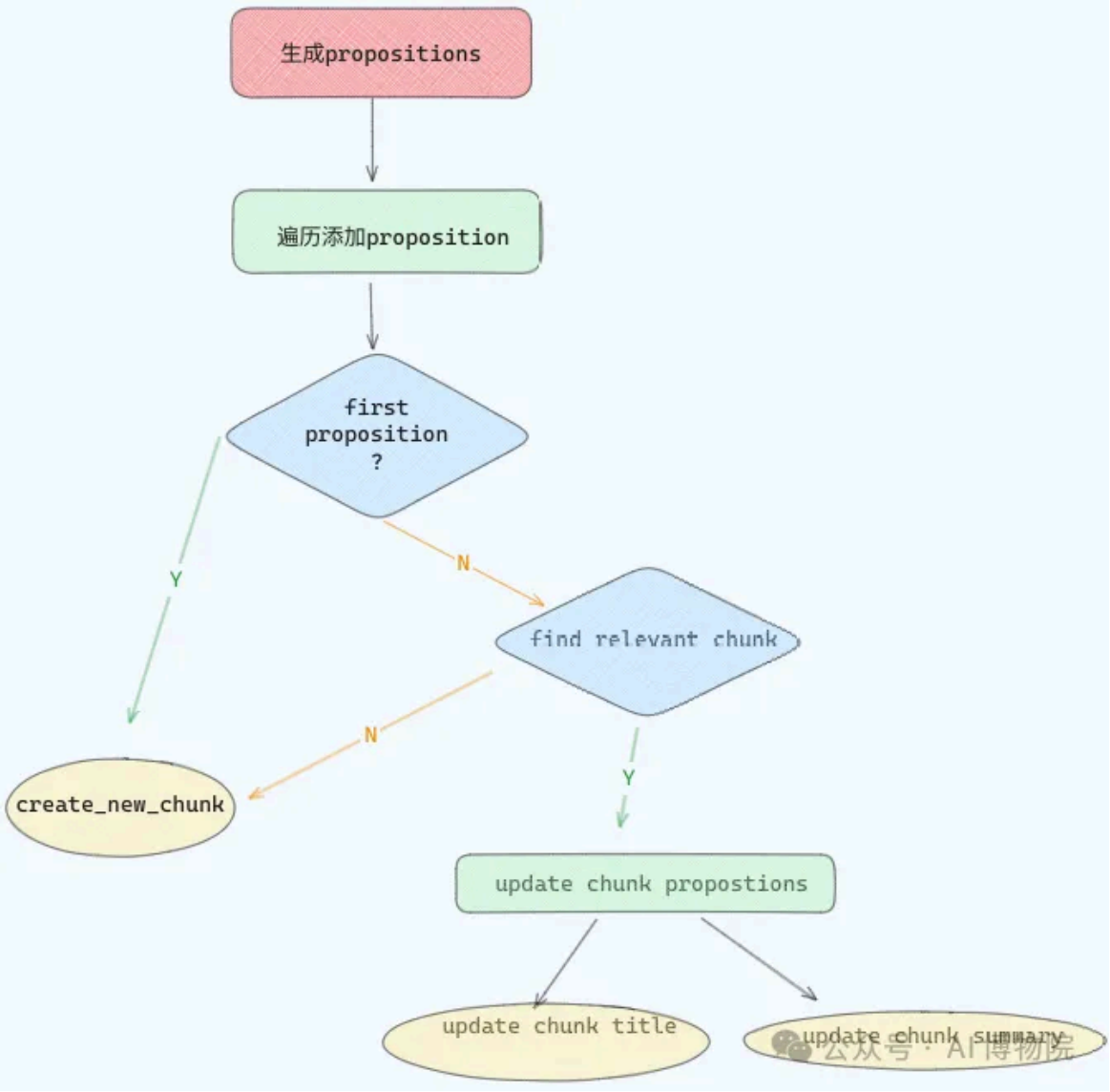
```
On July 20, 1969, astronaut Neil Armstrong walked on the moon.
Neil Armstrong was leading the NASA’s Apollo 11 mission.
Neil Armstrong famously said, “That’s one small step for man, one giant leap for mankind” as
```

这样，LLM就可以单独检查每一个句子，并将其分配到最合适的文本块中。

> propositioning 可以看做是对文档进行“句子级整容”，确保每个句子独立完整

如何实现Agentic Chunking?

实现Agentic Chunking的关键在于**propositioning**和**文本块的动态创建与更新**。我们可以使用Langchain和Pydantic等工具来实现这一过程。流程图如下：



1. Propositioning文本

首先，我们需要将文本中的每个句子进行propositioning处理。我们可以使用Langchain提供的提示词模板，让LLM自动完成这项工作。以下是一个简单的代码示例：

```
from langchain.chains import create_extraction_chain_pydantic
from langchain_core.pydantic_v1 import BaseModel
from typing import Optional
from langchain.chat_models import ChatOpenAI
import uuid
import os
from typing import List

from langchain import hub
from langchain_core.prompts import ChatPromptTemplate
from langchain_openai import ChatOpenAI
```

```
from pydantic import BaseModel

obj = hub.pull("wfh/proposal-indexing")
llm = ChatOpenAI(model="gpt-4o")

class Sentences(BaseModel):
    sentences: List[str]

extraction_llm = llm.with_structured_output(Sentences)
extraction_chain = obj | extraction_llm

sentences = extraction_chain.invoke(
    """
    On July 20, 1969, astronaut Neil Armstrong walked on the moon.
    He was leading the NASA's Apollo 11 mission.
    Armstrong famously said, "That's one small step for man, one giant leap for mankind" as h
    """
)
```

2. 创建和更新文本块

接下来，我们需要创建一个函数来动态生成和更新文本块。每个文本块包含主题相似的propositions，并且随着新propositions的加入，文本块的标题和摘要也会不断更新。

```
def create_new_chunk(chunk_id, proposition):
    summary_llm = llm.with_structured_output(ChunkMeta)
    summary_prompt_template = ChatPromptTemplate.from_messages([
        ("system", "Generate a new summary and a title based on the propositions."),
        ("user", "propositions:{propositions}"),
    ])
    summary_chain = summary_prompt_template | summary_llm
    chunk_meta = summary_chain.invoke({"propositions": [proposition]})
    chunks[chunk_id] = {
        "summary": chunk_meta.summary,
        "title": chunk_meta.title,
        "propositions": [proposition],
    }
```

3. 将proposition推送到合适的文本块

最后，我们需要一个AI Agent来判断新的proposition应该被添加到哪个文本块中。如果没有合适的文本块，Agent会创建一个新的文本块。

```
def find_chunk_and_push_proposition(proposition):
    class ChunkID(BaseModel):
```

```
    chunk_id: int = Field(description="The chunk id.")
allocation_llm = llm.with_structured_output(ChunkID)
allocation_prompt = ChatPromptTemplate.from_messages([
    ("system", "Find the chunk that best matches the proposition. If no chunk matches, re
    ("user", "proposition:{proposition} chunks_summaries:{chunks_summaries}"),
])
allocation_chain = allocation_prompt | allocation_llm
chunks_summaries = {chunk_id: chunk["summary"] for chunk_id, chunk in chunks.items()}
best_chunk_id = allocation_chain.invoke({"proposition": proposition, "chunks_summaries":
if best_chunk_id not in chunks:
    create_new_chunk(best_chunk_id, proposition)
else:
    add_proposition(best_chunk_id, proposition)
```

实测效果如何

我选择了新加坡圣淘沙著名景点 Wings of Time 的介绍文本作为测试对象，使用 GPT-4 模型进行处理。这段文本包含了景点介绍、票务信息、开放时间等多个方面的内容，是一个很好的测试样本。

```
Product Name: Wings of Time

Product Description: Wings of Time is one of Sentosa's most breathtaking attractions, combini

Product Category: Shows

Product Type: Attraction

Keywords: Wings of Time, Sentosa night show, Sentosa attractions, laser show Sentosa, water s

Meta Description: Experience Wings of Time at Sentosa! A breathtaking night show featuring wa


Product Tags: Family Fun,Popular experiences,Frequently Bought

Locations: Beach Station

[Tickets]

Name: Wings of Time (Std)
Terms: • All Wings of Time (WOT) Open-Dated tickets require prior redemption at Singapore Cab
Pax Type: Standard
Promotion A: Enjoy $1.90 off when you purchase online! Discount will automatically be applied
Price: 19




Opening Hours: Daily  Show 1: 7.40pm  Show 2: 8.40pm




Accessibilities: Wheelchair
```

```
[Information]

Title: Terms & Conditions
Description: For more information, click  (https://www.sentosa.com.sg/en/promotional-general-


Title: Getting Here
Description: By Sentosa Express: Alight at Beach Station  By Public Bus: Board Bus 123 and al


Title: Contact Us
Description: Beach Station  +65 6361 0088   (mailto:guestrelations@mflg.com.sg) guestrelatior
```

系统首先将原文转化为 50 多个独立的陈述句（propositions）。有趣的是，在这个过程中，系统自动将每句话的主语统一为"Wings of Time"，这显示出了 AI 对文本主题的准确把握。

```
[
    "Wings of Time is one of Sentosa's most breathtaking attractions.",
    'Wings of Time combines water, laser, fire, and music to create a mesmerizing night show.
    'The night show of Wings of Time is about friendship and courage.',
    'Wings of Time is situated on the scenic Siloso Beach.',
    'Wings of Time is an award-winning spectacle staged nightly.',
    'Wings of Time promises an unforgettable experience for visitors of all ages.',
    'Wings of Time features spellbinding laser, fire, and water effects set to a majestic sou
    'Wings of Time includes a jaw-dropping fireworks display.',
    'Wings of Time is a fitting end to a day out at Sentosa.',
    'Wings of Time is possibly the only place in Singapore where such an awe-inspiring perfor
    'Wings of Time will offer an even better experience starting 1 February 2025.',
    'Wings of Time Fireworks Symphony is Singapore’s only daily fireworks show.',
    'Wings of Time Fireworks Symphony now features a fireworks display that is four times lor
    'Visitors should visit the provided link if they need to change their visit date to Wings
    'All changes to the visit date must be made at least 1 day prior to the visit date.',
    'Wings of Time is categorized as a show.',
    'Wings of Time is a type of attraction.',
    'Keywords for Wings of Time include: Wings of Time, Sentosa night show, Sentosa attractic
    'The meta description for Wings of Time is: Experience Wings of Time at Sentosa! A breath
    'Product tags for Wings of Time include: Family Fun, Popular experiences, Frequently Boug
    'Wings of Time is located at Beach Station.',
    'Wings of Time (Std) tickets require prior redemption at Singapore Cable Car Ticketing co
    'Wings of Time (Std) tickets are subjected to seats availability on a first come first se
    'Wings of Time is a rain or shine event.',
    'Tickets for Wings of Time are non-exchangeable or nonrefundable under any circumstances.
    'Once the timeslot for Wings of Time is confirmed, no further amendments are allowed.',
    'Visitors should proceed to Wings of Time admission gates to scan their issued QR code vi
    'Gates for Wings of Time will open 15 minutes prior to the start of the show.',
    'The show duration for Wings of Time is 20 minutes per show.',
    'Visitors should be punctual for their booked time slot for Wings of Time.',
    'Admission to Wings of Time will be on a first come first serve basis within the allocate
    'Standard seats for Wings of Time are applicable to guests aged 4 years and above.',
    'No outside food and drinks are allowed at Wings of Time.',
    'More information on Wings of Time can be found at the provided link.',
    'The pax type for Wings of Time is Standard.',
    'Promotion A for Wings of Time offers $1.90 off when purchased online.',
```

```
'The discount for Promotion A will automatically be applied upon checkout.',
'The price for Wings of Time is 19.',
'Wings of Time has opening hours daily with Show 1 at 7.40pm and Show 2 at 8.40pm.',
'Wings of Time is accessible by wheelchair.',
"The title for terms and conditions is 'Terms & Conditions'.",
'More information on terms and conditions can be found at the provided link.',
"The title for getting to Wings of Time is 'Getting Here'.",
'Visitors can get to Wings of Time by Sentosa Express by alighting at Beach Station.',
'Visitors can get to Wings of Time by Public Bus by boarding Bus 123 and alighting at Bea
'Visitors can get to Wings of Time by Intra-Island Bus by boarding Sentosa Bus A or B anc
'The nearest car park to Wings of Time is Beach Station Car Park.',
"The title for contacting Wings of Time is 'Contact Us'.",
'The contact location for Wings of Time is Beach Station.',
'The contact phone number for Wings of Time is +65 6361 0088.',
'The contact email for Wings of Time is guestrelations@mflg.com.sg.']
```

经过 AI 的智能分块（agentic chunking），整个文本被自然地划分为四个主要部分：

1. 主体信息块：包含了 Wings of Time 的核心介绍、特色、位置等综合信息
2. 日程政策块：专门处理预约变更相关的信息
3. 价格优惠块：聚焦于折扣和支付相关内容
4. 法律条款块：归纳了各项条款和规定

```
Chunk (a641f): Sentosa's Wings of Time Show & Visitor Information
Summary: This chunk contains comprehensive details about the Wings of Time attraction in Sent

Chunk (ae2b8): Scheduling Policies
Summary: This chunk contains information about policies regarding changes to scheduled dates

Chunk (dadbb): Retail & Discounts
Summary: This chunk contains information about the application of discounts during the checkc

Chunk (3347c): Legal Terms & Conditions
Summary: This chunk contains information about terms and conditions, including their titles a
```

经过这样的分块之后，各个块的主题明确，不重叠，且重要信息优先，辅助信息分类存放。把这样的信息放在一起，也有助于提升向量库的召回率，从而提升RAG的准确率。

总结

Agentic Chunking是一种非常强大的文本分块技术，它能够将文档中相隔较远但主题相关的句子归入同一组，从而提升RAG模型的效果，但是这种方法在成本和延迟上相对较高。同事尝试了Agentic chunking之后，据他说准确率提升了40%，但成本也增加了3倍。那么我们时候应该使用Agentic chunking呢?

根据我的项目经验，以下场景特别适合：

- 非结构化文本（如客服对话记录）
- 主题反复横跳的内容（技术沙龙实录）
- 需要跨段落关联的QA系统

而面对结构清晰的论文、说明书等，传统分块和语义分块仍是性价比之选。


AI 博物院
专注于工作、技术、生活各种分享交流
57篇原创内容

公众号

> 我们建了交流群，目前群人数较多，感兴趣的朋友可以点赞关注后台回复加群即可，对源码感兴趣的可以关注私信我

RAG 33 AI 32 人工智能 47 Agentic 1 chunking 1

RAG · 目录

上一篇 · 从GraphRAG到PIKE-RAG，微软发布复杂企业场景下的私域知识提取与推理新突破