1. RAG的文本加载

langchain提供了一些文本加载器，使用文档加载器可以从源加载数据转换为Document。

Document包含两个属性page_content（文本）和metadata=（元数据，即来源、标题等等）

1.1 加载TXT

```
# 加载分割TXT
from langchain_community.document_loaders import TextLoader

# 加载TXT文件
txt_loader = TextLoader("test.txt", encoding="UTF-8")
documents = txt_loader.load()
```

1.2 加载 CSV

```
# 加载 CSV
from langchain_community.document_loaders import CSVLoader

# 加载CSV文件
csv_loader = CSVLoader(file_path='test.csv', encoding="UTF-8")
csv_data = csv_loader.load()

print(csv_data)
```

1.3 加载 PDF

安装依赖

```
pip install pypdf==5.1.0
```

```
from langchain_community.document_loaders import PyPDFLoader

# 加载 PDF 文件
loader = PyPDFLoader("test.pdf")
pages = loader.load_and_split()  # 按页面分割 PDF
```

按页分割后会在metadata有page的标签显示是哪一页。

1.4 加载Markdown

安装依赖

```
pip install unstructured==0.16.6 markdown==3.7
```

```
from langchain_community.document_loaders import
UnstructuredMarkdownLoader

loader = UnstructuredMarkdownLoader(file_path="test.md",
mode="elements")
data = loader.load()
print(data)
```

通过设置mode="elements"保留Unstructured 为不同文本块创建不同的属性，比较重要的是'category'指明了分块的类型是标题或正文。

结果：

```
[Document(metadata={'source': 'test.md', 'category_depth': 0,
'languages': ['zho'], 'filename': 'test.md', 'filetype':
'text/markdown', 'last_modified': '2024-11-22T13:11:54',
'category': 'Title', 'element_id':
'eee747715792a53455282fbd70333794'}, page_content='第一章'),
Document(metadata={'source': 'test.md', 'category_depth': 0,
'languages': ['zho'], 'filename': 'test.md', 'filetype':
'text/markdown', 'last_modified': '2024-11-22T13:11:54',
'category': 'Title', 'element_id':
'bd51a2b393e23c424cf308329132ffb9'}, page_content='test'),
Document(metadata={'source': 'test.md', 'category_depth': 0,
'languages': ['zho'], 'filename': 'test.md', 'filetype':
'text/markdown', 'last_modified': '2024-11-22T13:11:54',
'category': 'Title', 'element_id':
'5d8f390f1d75c98dac060cb011c1075b'}, page_content='在西天取经的几百年
前，黑风山上有一只黑熊精占山为王，自称黑风大王。'), Document(metadata=
{'source': 'test.md', 'category_depth': 0, 'languages': ['zho'],
'filename': 'test.md', 'filetype': 'text/markdown',
'last_modified': '2024-11-22T13:11:54', 'category': 'Title',
'element_id': '502f783f3550c72d55b76b25ec406a00'},
page_content='有一天，黑熊精碰到了一个小和尚。他觉得这个小和尚蛮有趣，于是给
了他一些金银财宝，又教给他一些长生的法门。这个小和尚就是后来的金池长老，二人从此
结缘。'), Document(metadata={'source': 'test.md', 'category_depth':
0, 'languages': ['zho'], 'filename': 'test.md', 'filetype':
'text/markdown', 'last_modified': '2024-11-22T13:11:54',
'category': 'Title', 'element_id':
'af4162b883d8ad90da342bcc7031228f'}, page_content='在这之后，金池也给
黑熊精讲一些佛法，黑熊精也有点兴趣，二人也算是有共同话题的朋友。'),
Document(metadata={'source': 'test.md', 'category_depth': 0,
'languages': ['zho'], 'filename': 'test.md', 'filetype':
'text/markdown', 'last_modified': '2024-11-22T13:11:54',
'category': 'Title', 'element_id':
'd154f2b8cdf6c6819e721f6965e405d4'}, page_content='金池一路坐到观音禅
院的长老（毕竟学过长生术，活的最久），受人顶礼膜拜，欲望也随之膨胀。')]
```

从打印的内容来看，`UnstructuredMarkdownLoader` 成功加载并解析了 Markdown 文件 `test.md`，将其内容转换成了一组 `Document` 对象列表。每个 `Document` 对象代表了原文档中的一个元素或段落，并包含了元数据（metadata）和页面内容（page_content）两个主要部分。

元数据（Metadata）包含的信息：

- **source**: 原始文件的路径或标识符，这里是 `test.md`。
- **category_depth**: 表示当前文档元素在文档结构中的层级深度，这里所有元素均为0，意味着它们都是顶级元素。
- **languages**: 文档元素的语言信息，这里的文档是中文（`zho`）。
- **filename**: 原始文件名，与 `source` 类似，但仅包含文件名部分。
- **filetype**: 文件类型，这里是 `text/markdown`，表明这是一个 Markdown 格式的文本文件。
- **last_modified**: 文件最后修改的时间戳，格式为 ISO8601。
- **category**: 文档元素的类别，这里所有的元素都被分类为标题（`Title`），这可能是因为 Markdown 文件中使用了特定的标记（如 `#`）来定义标题。
- **element_id**: 每个文档元素的唯一标识符，用于区分不同的文档元素。

页面内容（Page Content）：

页面内容即为文档元素的实际内容，也就是 Markdown 文件中对应部分的文字内容。可以看到，这些内容包括了一些故事性的叙述，如关于黑熊精和金池长老的故事片段。

1.5 加载 json

安装依赖

```
pip install jq==1.8.0 python-magic-bin==0.4.14
```

```
"""加载 JSON"""
from langchain_community.document_loaders import JSONLoader
# jq_schema 是一个字符串，用于指定如何从 JSON 文件中提取数据。你提供的
jq_schema 是 .skills，这意味着你希望提取 skills 数组。
loader = JSONLoader(file_path='test.json', jq_schema='.skills')

data = loader.load()
print(data)
```

1.6 加载 Html

```
"""加载 HTML"""
from langchain_community.document_loaders import
UnstructuredHTMLLoader, BSHTMLLoader

# 使用UnstructuredHTMLLoader加载HTML文件
loader = UnstructuredHTMLLoader("test.html")
data = loader.load()
print(data)

# 使用BSHTMLLoader加载HTML，提取页面标题
loader = BSHTMLLoader("test.html", open_encoding="UTF-8")
data = loader.load()
print(data)
```