⌂ Modules Retrieval Text Splitters

MarkdownHeaderTextSplitter

MarkdownHeaderTextSplitter

Motivation

Many chat or Q+A applications involve chunking input documents prior to embedding and vector storage.

These notes from Pinecone provide some useful tips:

```
When a full paragraph or document is
embedded, the embedding process considers
both the overall context and the
relationships between the sentences and
phrases within the text. This can result in a
more comprehensive vector representation that
captures the broader meaning and themes of
the text.
```

As mentioned, chunking often aims to keep text with common context together. With this in mind, we might want to specifically honor the structure of the document itself. For example, a markdown file is organized by headers. Creating chunks within specific header groups is an intuitive idea. To address this challenge, we can use `MarkdownHeaderTextSplitter`. This will split a markdown file by a specified set of headers.

For example, if we want to split this markdown:

```
md = '# Foo\n\n ## Bar\n\nHi this is Jim
\nHi this is Joe\n\n ## Baz\n\n Hi this is
Molly'
```

We can specify the headers to split on:

```
[("#", "Header 1"),("##", "Header 2")]
```

And content is grouped or split by common headers:

```
{'content': 'Hi this is Jim  \nHi this is
Joe', 'metadata': {'Header 1': 'Foo', 'Header
2': 'Bar'}}
{'content': 'Hi this is Molly', 'metadata':
{'Header 1': 'Foo', 'Header 2': 'Baz'}}
```

Let’s have a look at some examples below.

```
from langchain.text_splitter import
MarkdownHeaderTextSplitter
```

```
markdown_document = "# Foo\n\n    ## Bar\n\nHi
this is Joe\n\n ### Boo \n\n Hi this is Lance \
this is Molly"

headers_to_split_on = [
    ("#", "Header 1"),
    ("##", "Header 2"),
    ("###", "Header 3"),
]

markdown_splitter =
MarkdownHeaderTextSplitter(headers_to_split_on=
md_header_splits = markdown_splitter.split_text
md_header_splits
```

```
[Document(page_content='Hi this is Jim  \nHi
this is Joe', metadata={'Header 1': 'Foo',
'Header 2': 'Bar'}),
 Document(page_content='Hi this is Lance',
metadata={'Header 1': 'Foo', 'Header 2':
'Bar', 'Header 3': 'Boo'}),
 Document(page_content='Hi this is Molly',
metadata={'Header 1': 'Foo', 'Header 2':
'Baz'})]
```

```
type(md_header_splits[0])
```

```
langchain.schema.document.Document
```

By default, `MarkdownHeaderTextSplitter` strips headers being split on from the output chunk's content. This can be disabled by setting `strip_headers = False`.

```
markdown_splitter = MarkdownHeaderTextSplitter(
    headers_to_split_on=headers_to_split_on,
strip_headers=False
)
md_header_splits =
markdown_splitter.split_text(markdown_document)
md_header_splits
```

```
[Document(page_content='# Foo  \n## Bar  \nHi
this is Jim  \nHi this is Joe', metadata=
{'Header 1': 'Foo', 'Header 2': 'Bar'}),
 Document(page_content='### Boo  \nHi this is
Lance', metadata={'Header 1': 'Foo', 'Header
2': 'Bar', 'Header 3': 'Boo'}),
 Document(page_content='## Baz  \nHi this is
Molly', metadata={'Header 1': 'Foo', 'Header
2': 'Baz'})]
```

Within each markdown group we can then apply any text splitter we want.

```
markdown_document = "# Intro \n\n    ## History \n\n Markdown[9] is a lightweight markup language for creating formatted text using a plain-text editor. John Gruber created Markdown in 2004 as a markup language that is appealing to human readers in its source code form.[9] \n\n Markdown is widely used in blogging, instant messaging, online forums, collaborative software, documentation pages, and readme files. \n\n ## Rise and divergence \n\n As Markdown popularity grew rapidly, many Markdown implementations appeared, driven mostly by the need for \n\n additional features such as tables, footnotes, definition lists,[note 1] and Markdown inside HTML blocks. \n\n #### Standardization \n\n From 2012, a group of people, including Jeff Atwood and John MacFarlane, launched what Atwood characterised as a standardisation effort. \n\n ## Implementations \n\n Implementations of Markdown are available for over a dozen programming languages."

headers_to_split_on = [
    ("#", "Header 1"),
    ("##", "Header 2"),
]
```

```
# MD splits
markdown_splitter = MarkdownHeaderTextSplitter(
    headers_to_split_on=headers_to_split_on,
strip_headers=False
)
md_header_splits =
markdown_splitter.split_text(markdown_document)

# Char-level splits
from langchain.text_splitter import
RecursiveCharacterTextSplitter

chunk_size = 250
chunk_overlap = 30
text_splitter = RecursiveCharacterTextSplitter(
    chunk_size=chunk_size,
chunk_overlap=chunk_overlap
)

# Split
splits =
text_splitter.split_documents(md_header_splits)
splits
```

```
[Document(page_content='# Intro  \n## History
\nMarkdown[9] is a lightweight markup
language for creating formatted text using a
plain-text editor. John Gruber created
Markdown in 2004 as a markup language that is
appealing to human readers in its source code
form.[9]', metadata={'Header 1': 'Intro',
'Header 2': 'History'}),
 Document(page_content='Markdown is widely
used in blogging, instant messaging, online
forums, collaborative software, documentation
pages, and readme files.', metadata={'Header
1': 'Intro', 'Header 2': 'History'}),
 Document(page_content='## Rise and
divergence  \nAs Markdown popularity grew
rapidly, many Markdown implementations
appeared, driven mostly by the need for
\nadditional features such as tables,
footnotes, definition lists,[note 1] and
Markdown inside HTML blocks.', metadata=
{'Header 1': 'Intro', 'Header 2': 'Rise and
divergence'}),
 Document(page_content='#### Standardization
\nFrom 2012, a group of people, including
Jeff Atwood and John MacFarlane, launched
what Atwood characterised as a
standardisation effort.', metadata={'Header
1': 'Intro', 'Header 2': 'Rise and
divergence'}),
 Document(page_content='## Implementations
\nImplementations of Markdown are available
for over a dozen programming languages.',
metadata={'Header 1': 'Intro', 'Header 2':
'Implementations'})]
```