GeneralNewsExtractor

发布 *0.0.6*

2020 年 03 月 11 日

Contents:

1 如何使用 3

2 注意事项 5

3 运行截图 9
- 3.1 网易新闻 . . . 9
- 3.2 今日头条 . . . 10
- 3.3 新浪新闻 . . . 10
- 3.4 凤凰网 . . . 11

4 API 13

5 配置文件 15

6 已知问题 17

7 Changelog 19
- 7.1 2020.03.11 . . . 19
- 7.2 2020.02.21 . . . 19
- 7.3 2020.02.13 . . . 19
- 7.4 2020.01.04 . . . 20
- 7.5 2019.12.31 . . . 20
- 7.6 2019.12.29 . . . 20
- 7.7 2019.11.24 . . . 20

8 交流沟通 23

9 目录 25

GeneralNewsExtractor（GNE）是一个通用新闻网站正文抽取模块，输入一篇新闻网页的 HTML，输出正文内容、标题、作者、发布时间、正文中的图片地址和正文所在的标签源代码。GNE 在提取今日头条、网易新闻、游民星空、观察者网、凤凰网、腾讯新闻、ReadHub、新浪新闻等数百个中文新闻网站上效果非常出色，几乎能够达到 100% 的准确率。

使用方式也非常简单：

```
from gne import GeneralNewsExtractor

extractor = GeneralNewsExtractor()
html = '网站源代码'
result = extractor.extract(html)
print(result)
```

本项目取名为 抽取器，而不是 爬虫，是为了规避不必要的风险，因此，本项目的输入是 HTML 源代码，输出是一个字典。请自行使用恰当的方法获取目标网站的 HTML。

GNE 现在不会，将来也不会提供主动请求网站 HTML 的功能。

CHAPTER 1

如何使用

如果你想体验 GNE 的功能，请按照如下步骤进行：

0. 在线体验

如果你想先体验 GNE 的提取效果，那么你可以访问 http://122.51.39.219。一般情况下，你只需要把网页粘贴到最上面的多行文本框中，然后点 提取按钮即可。通过附加更多的参数，可以让提取更精确。具体参数的写法与作用，请参阅 API

1. 安装 GNE

```
# 以下两种方案任选一种即可

# 使用 pip 安装
pip install --upgrade gne

# 使用 pipenv 安装
pipenv install gne
```

2. 使用 GNE

```
>>> from gne import GeneralNewsExtractor
>>> html = '''经过渲染的网页 HTML 代码'''
>>> extractor = GeneralNewsExtractor()
>>> result = extractor.extract(html, noise_node_list=['//div[@class="comment-list"]'])
```

(下页继续)

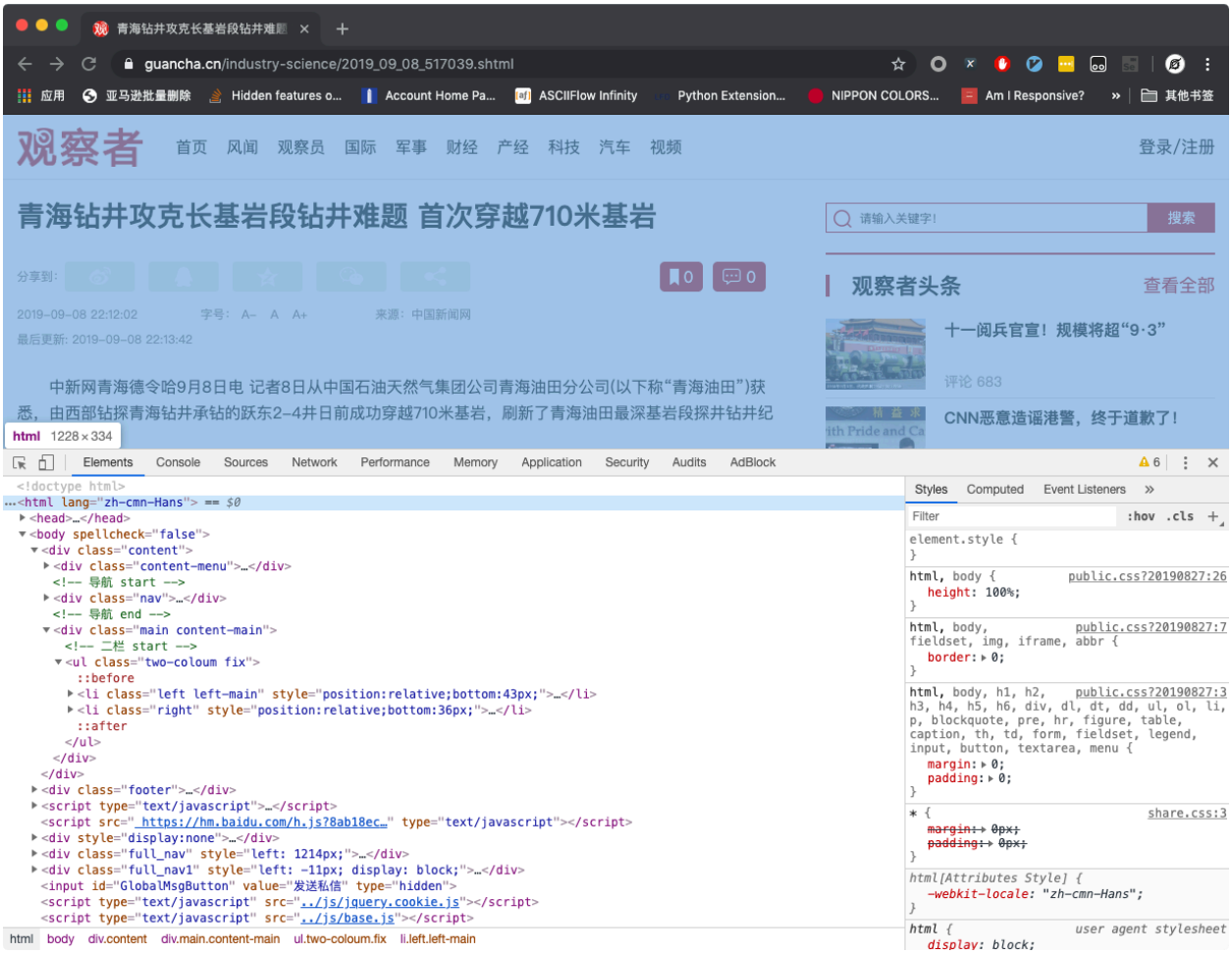
(续上页)

```
>>> print(result)
{"title": "xxxx", "publish_time": "2019-09-10 11:12:13", "author": "yyy", "content":
↪"zzzz", "images": ["/xxx.jpg", "/yyy.png"]}
```

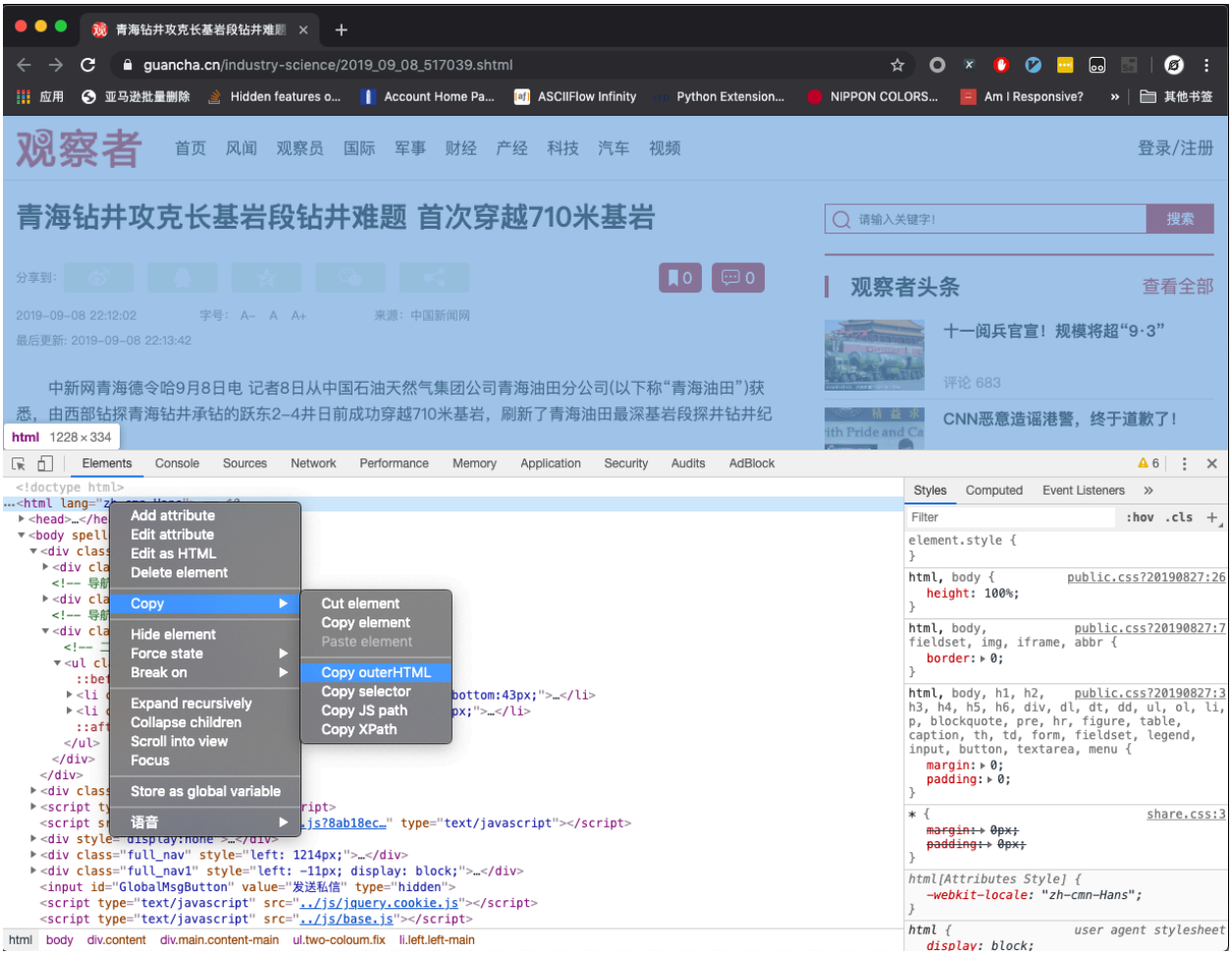
CHAPTER 2

注意事项

- 本项目的输入 HTML 为经过 JavaScript 渲染以后的 HTML，而不是普通的网页源代码。所以无论是后端渲染、Ajax 异步加载都适用于本项目。
- 如果你要手动测试新的目标网站或者目标新闻，那么你可以在 Chrome 浏览器中打开对应页面，然后开启 *开发者工具*，如下图所示：



在 Elements 标签页定位到 <html> 标签，并右键，选择 Copy - Copy OuterHTML ，如下图所示



- 当然，你可以使用 Puppeteer/Pyppeteer、Selenium 或者其他任何方式获取目标页面的 JavaScript 渲染后的源代码。
- 获取到源代码以后，通过如下代码提取信息：

```
from gne import GeneralNewsExtractor

extractor = GeneralNewsExtractor()
html = '你的目标网页正文'
result = extractor.extract(html)
print(result)
```

- 如果标题自动提取失败了，你可以指定 XPath：

```
from gne import GeneralNewsExtractor

extractor = GeneralNewsExtractor()
html = '你的目标网页正文'
```

(下页继续)

(续上页)

```
result = extractor.extract(html, title_xpath='//h5/text()')
print(result)
```

对大多数新闻页面而言，以上的写法就能够解决问题了。

但某些新闻网页下面会有评论，评论里面可能存在长篇大论，它们会看起来比真正的新闻正文更像是正文，因此 `extractor.extract()` 方法还有一个默认参数 `noise_node_list` ，用于在网页预处理时提前把评论区域整个移除。`noise_mode_list` 的值是一个列表，列表里面的每一个元素都是 XPath，对应了你需要提前移除的，可能会导致干扰的目标标签。

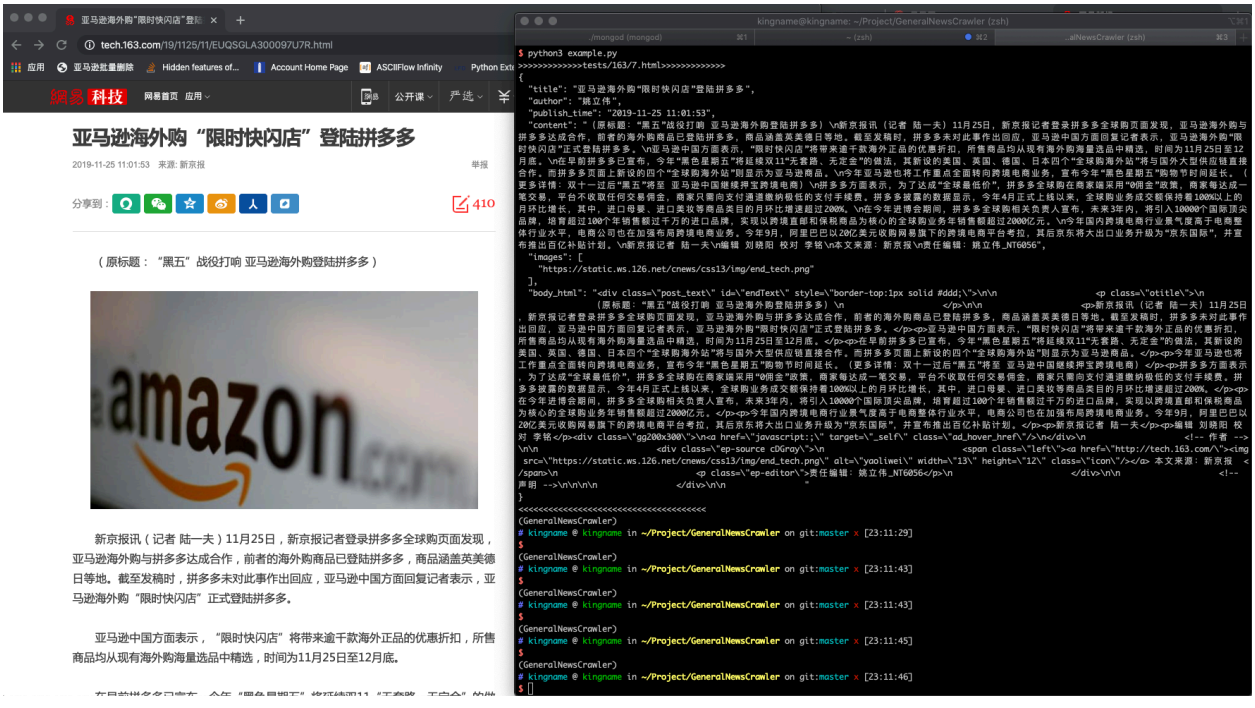
例如，观察者网下面的评论区域对应的 Xpath 为 `//div[@class="comment-list"]` 。所以在提取观察者网时，为了防止评论干扰，就可以加上这个参数：

```
result = extractor.extract(html, noise_node_list=['//div[@class="comment-list"]'])
```

CHAPTER 3

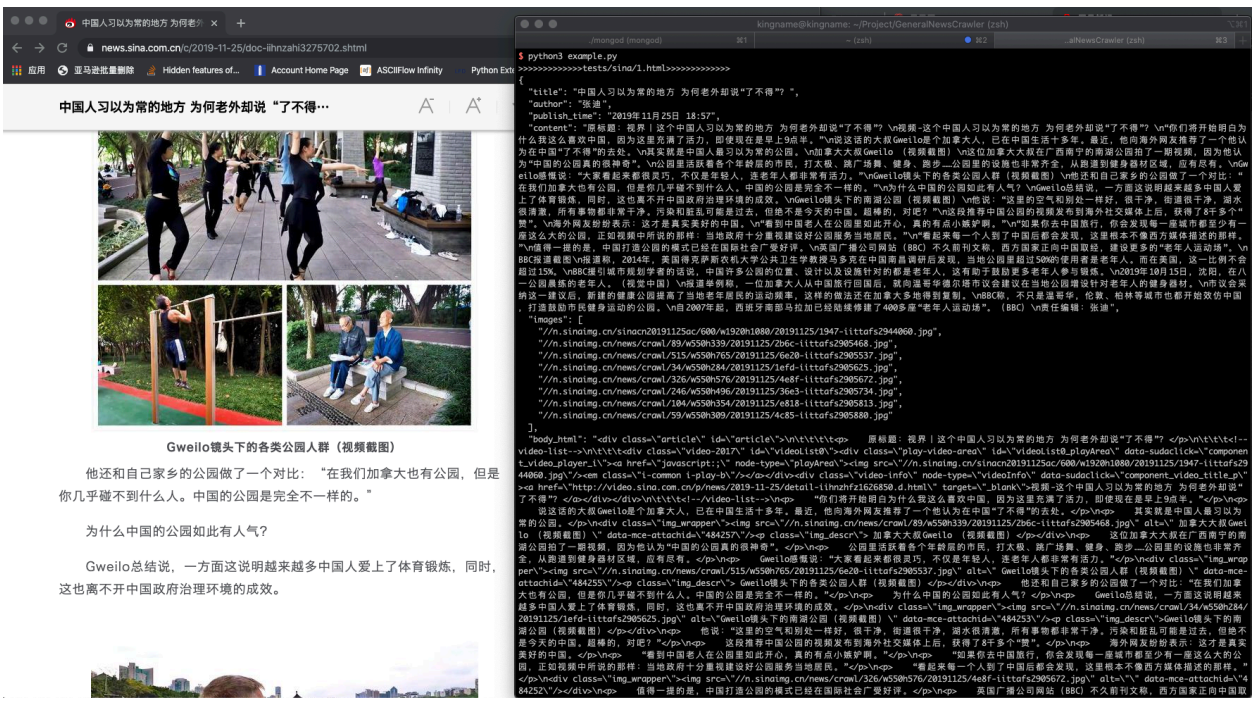
运行截图

3.1 网易新闻

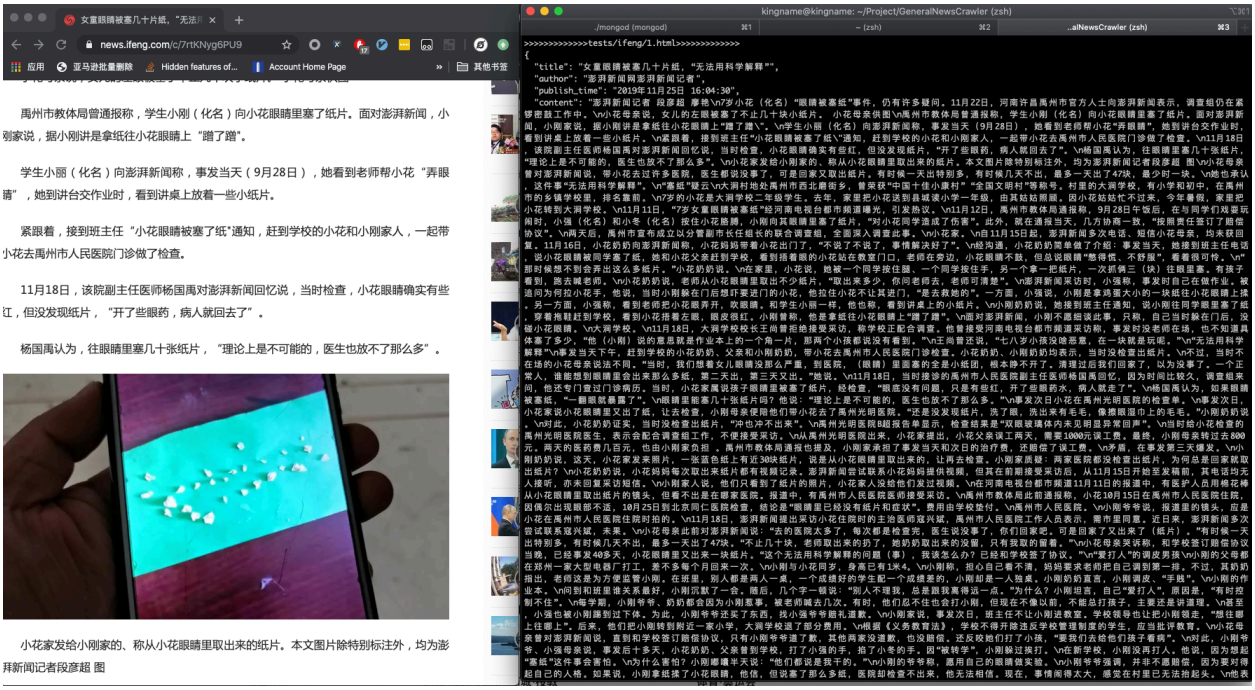


3.2 今日头条

3.3 新浪新闻



3.4 凤凰网



禹州市教体局曾通报称，学生小刚（化名）向小花眼睛里塞了纸片。面对澎湃新闻，小刚家说，据小刚讲是拿纸往小花眼睛上“蹭了蹭”。

学生小丽（化名）向澎湃新闻称，事发当天（9月28日），她看到老师帮小花“弄眼睛”，她到讲台交作业时，看到讲桌上放着一些小纸片。

紧跟着，接到班主任“小花眼睛被塞了纸”通知，赶到学校的小花和小刚家人，一起带小花去禹州市人民医院门诊做了检查。

11月18日，该院副主任医师杨国禹对澎湃新闻回忆说，当时检查，小花眼睛确实有些红，但没发现纸片，“开了些眼药，病人就回去了”。

杨国禹认为，往眼睛里塞几十张纸片，“理论上是不可能的，医生也放不了那么多”。

小花家发给小刚家的、称从小花眼睛里取出来的纸片。本文图片除特别标注外，均为澎拜新闻记者段彦超 图

CHAPTER 4

API

GNE 的函数原型为：

```
class GeneralNewsExtractor:
    def extract(self, html, title_xpath='', host='', author_xpath='', publish_time_xpath=
↪'', noise_node_list=None, with_body_html=False)
```

各个参数的意义如下：

- **html(str)**：目标网站的源代码
- **title_xpath(str)**：新闻标题的 XPath，用于定向提取标题
- **host(str)**：图片所在的域名，例如 `https://www.kingname.info`，那么，当 GNE 从新闻网站提取到图片的相对连接 ‘‘/images/123.png‘‘时，会把 `host` 拼接上去，变成 ‘‘https://www.kingname.info/images/123.png‘‘
- **noise_node_list(List[str])**：一个包含 XPath 的列表。这个列表中的 XPath 对应的标签，会在预处理时被直接删除掉，从而避免他们影响新闻正文的提取
- **with_body_html(bool)**：默认为 False，此时，返回的提取结果不含新闻正文所在标签的 HTML 源代码。当把它设置为 True 时，返回的结果会包含字段 `body_html`，内容是新闻正文所在标签的 HTML 源代码
- **author_xpath(str)**：文章作者的 XPath，用于定向提取文章作者
- **publish_time_xpath(str)**：文章发布时间的 XPath，用于定向提取文章发布时间

CHAPTER 5

配置文件

API 中的参数 `title_xpath`、`host`、`noise_node_list`、`with_body_html`、`author_xpath`、`publish_time_xpath` 除了直接写到 `extract` 方法中外，还可以通过一个配置文件来设置。

请在项目的根目录创建一个文件 `.gne`，配置文件可以用 YAML 格式，也可以使用 JSON 格式。

- YAML 格式配置文件

```
title:
    xpath: //title/text()
host: https://www.xxx.com
noise_node_list:
    - //div[@class=\"comment-list\"]
    - //*[@style=\"display:none\"]
with_body_html: true
author:
    xpath: //meta[@name="author"]/@content
publish_time:
    xpath: //em[@id="publish_time"]/text()
```

- JSON 格式配置文件：

```
{
    "title": {
        "xpath": "//title/text()"
```

(下页继续)

(续上页)

```
    },
    "host": "https://www.xxx.com",
    "noise_node_list": ["//div[@class=\"comment-list\"]",
                        "//*[@style=\"display:none\"]"],
    "with_body_html": true,
    "author": {
        "xpath": "//meta[@name=\"author\"]/@content"
    },
    "publish_time": {
        "xpath": "//em[@id=\"publish_time\"]/text()"
    }
}
```

这两种写法是完全等价的。

配置文件与 `extract` 方法的参数一样，并不是所有字段都需要提供。你只需要填写你需要的字段即可。

如果一个参数，既在 `extract` 方法中，又在 `.gne` 配置文件中，但值不一样，那么 `extract` 方法中的这个参数的优先级更高。

CHAPTER 6

已知问题

1. 目前本项目只适用于新闻页的信息提取。如果目标网站不是新闻页，或者是今日头条中的相册型文章，那么抽取结果可能不符合预期。
2. 可能会有一些新闻页面出现抽取结果中的作者为空字符串的情况，这可能是由于文章本身没有作者，或者使用了已有正则表达式没有覆盖到的情况。

CHAPTER 7

Changelog

7.1 2020.03.11

1. 预处理可能会破坏 HTML 结构，导致用户自定义的 XPath 无法正确工作，因此需要把提取用户名、发布时间、标题的代码放在预处理之前。

7.2 2020.02.21

1. 感谢 @ 止水提供的 meta 对应的新闻时间属性，现在会从 HTML 的 meta 数据中检查是否有发布时间。

7.3 2020.02.13

1. 在 `GeneralNewsExtractor().extract()` 方法中传入参数 `author_xpath` 和 `publish_time_xpath` 强行指定抓取作者与发布时间的位置。
2. 在 `.gne` 配置文件中，通过如下两个配置分别指定作者与发布时间的 XPath：

```
author:
    xpath: //meta[@name="author"]/@content
publish_time:
    xpath: //em[@id="publish_time"]/text()
```

7.4 2020.01.04

1. 修复由于 `node.getparent().remove()` 会移除父标签中，位于自己后面的 text 的问题
2. 对于 class 中含有 `article` / `content` / `news_txt` / `post_text` 的标签，增加权重
3. 使用更科学的方法移除无效标签

7.5 2019.12.31

通用参数可以通过 YAML、JSON 批量设置了。只需要在项目的根目录下创建一个 `.gne` ，就可以实现函数默认参数的功能。

7.6 2019.12.29

1. 现在可以通过传入参数 `host` 来把提取的图片 url 拼接为绝对路径

例如：

```
extractor = GeneralNewsExtractor()
result = extractor.extract(html,
                           host='https://www.xxx.com')
```

返回数据中：

```
{
    ...
    "images": [
        "https://www.xxx.com/W020190918234243033577.jpg"
      ]
}
```

7.7 2019.11.24

1. 增加更多的 UselessAttr
2. 返回的结果包含 `images` 字段，里面的结果是一个列表，保存了正文中的所有图片 URL
3. 指定 `with_body_html` 参数，返回的数据中将会包含 `body_html` 字段，这是正文的 HTMl 源代码：

```
result = GeneralNewsExtractor().extract(html, with_body_html=True)
body_html = result['body_html']
print(f'正文的网页源代码为：{body_html}')
```

CHAPTER 8

交流沟通

如果您觉得 GNE 对您的日常开发或公司有帮助，请加作者微信 mxqiuchen（或扫描下方二维码）并注明”GNE”，作者会将你拉入群。

验证消息：GNE

如果你不用微信，那么可以加入 Telegram 交流群：https://t.me/joinchat/Bc5swww_XnVR7pEtDUl1vw

CHAPTER 9

目录

- genindex
- modindex
- search