pyminer Documentation

Release 0.0.5.9000

Scott Chamberlain

Jun 07, 2017

Contents

Python client for the GBIF API.

Source on GitHub at sckott/pyminer

Other Crossref text mining (and related) clients:

- R: *rcrossref*, ropensci/rcrossref
- R: *crminer*, ropensci/crminer
- R: *fulltext*, ropensci/fulltext
- Ruby: *textminer*, sckott/textminer

CHAPTER 1

Installation

Stable from pypi

```
pip install pyminer
```

Development version

```
[sudo] pip install git+git://github.com/sckott/pyminer.git#egg=pyminer
```

CHAPTER 2

Search

```
from pyminer import miner
miner.search(filter = {'has_full_text': True}, limit = 5)
```

CHAPTER 3

Fetch

```
from pyminer import miner
url = "http://www.banglajol.info/index.php/AJMBR/article/viewFile/25509/17126"
out = miner.fetch(url)
out.url
out.path
out.type
out.parse()
```

CHAPTER 4

Extract

```
from pyminer import miner
url = 'http://www.nepjol.info/index.php/JSAN/article/viewFile/13527/10928'
x = miner.fetch(url)
miner.extract(x.path)
```

CHAPTER 5

Meta

- License: MIT, see LICENSE file
- Please note that this project is released with a Contributor Code of Conduct. By participating in this project you agree to abide by its terms.

Contents

miner module

`miner.`**`search`**(*ids=None*, *member=None*, *filter=None*, *limit=500*, ***kwargs*)

Search Crossref to get text mining links

Parameters

- **`ids`** – [Array] DOIs (digital object identifier) or other identifiers
- **`member`** – [String] member ids
- **`filter`** – [Hash] Filter options. See ...
- **`limit`** – [Fixnum] Number of results to return. Not relavant when searching with specific dois. Default: 20. Max: 1000
- **`kwargs`** – any additional arguments will be passed on to `requests.get`

Returns A dictionary, of results

Usage:

```
from pyminer import miner
miner.search(filter = {'has_full_text': True}, limit = 5)
miner.search(filter = {'full_text_type': 'text/plain', 'license_url': "http://
↪creativecommons.org/licenses/by-nc-nd/3.0"})
miner.search(filter = {'has_full_text': True, 'license_url': "http://
↪creativecommons.org/licenses/by/4.0"})
```

miner.**fetch**(*url*)

Get full text

Work easily for open access papers, but for closed. For non-OA papers, use Crossref's Text and Data Mining service, which requires authentication and pre-authorized IP address. Go to https://apps.crossref.org/clickthrough/researchers to sign up for the TDM service, to get your key. The only publishers taking part at this time are Elsevier and Wiley.

Parameters **url** – [String] A url for full text

Returns [Mined] An object of class Mined, with methods for extracting the url requested, the file path, and parsing the plain text, XML, or extracting text from the pdf.

XML returns object of class lxml.etree._Element, which you can parse using for example lxml

Usage:

```
from pyminer import miner

# pdf
url = "http://www.banglajol.info/index.php/AJMBR/article/viewFile/25509/17126"
out = miner.fetch(url)
out.url
out.path
out.type
out.parse()

# xml
url = "https://peerj.com/articles/cs-23.xml"
out = miner.fetch(url)
out.url
out.path
out.type
out.parse()
## or drop down to individual parsing methods
from pyminer import parsers as p
p.parse_xml(out.path)
p.parse_xml_string(out.path)

# search first, then pass links to fetch
res = miner.search()
miner.fetch(res['url'])
```

miner.**extract**(*path*)

Extract full text fro pdf's

Parameters **path** – [String] Path to a pdf file downloaded via {fetch}, or another way.

Returns [str] a string of text

Usage:

```
from pyminer import miner

# a pdf
url = "http://www.banglajol.info/index.php/AJMBR/article/viewFile/25509/17126"
out = miner.fetch(url)
out.parse()

# search first, then pass links to fetch
res = miner.search(filter = {'has_full_text': True, 'license_url': "http://
↪creativecommons.org/licenses/by/4.0"})
```

```
# url = res.links_pdf()[0]
url = 'http://www.nepjol.info/index.php/JSAN/article/viewFile/13527/10928'
x = miner.fetch(url)
miner.extract(x.path)
```

parsers module

parsers.**parse_xml**(*x*)
parse xml

Parameters **path** – [String] Path to an xml file

Returns object of class lxml.etree._Element

Usage:

```
from pyminer import fetch
from pyminer import parsers
url = "https://peerj.com/articles/cs-23.xml"
out = fetch(url)
parsers.parse_xml(out.path)
```

parsers.**parse_xml_string**(*x*, *encoding='UTF-8'*)
parse xml to a string

Parameters **path** – [String] Path to an xml file

Returns a string

Usage:

```
from pyminer import fetch
from pyminer import parsers
url = "https://peerj.com/articles/cs-23.xml"
out = fetch(url)
parsers.parse_xml_string(out.path)
```

parsers.**parse_plain**(*x*)
parse plain text

Parameters **path** – [String] Path to a plain text file

Returns a string

Usage:

```
from pyminer import fetch
from pyminer import parsers
url = "xx"
out = fetch(url)
parsers.parse_plain(out.path)
```

parsers.**parse_pdf**(*x*)
parse pdf

Parameters **path** – [String] Path to a pdf file

Returns a string

Usage:

```
from pyminer import fetch
from pyminer import parsers
url = "http://www.banglajol.info/index.php/AJMBR/article/viewFile/25509/17126"
out = fetch(url)
parsers.parse_pdf(out.path)
```

Changelog

0.1.0 (2016-02-02)

- in the works...

License

MIT

CHAPTER 6

Indices and tables

- genindex
- modindex
- search

Python Module Index

p

Index