astroph_new

Release 0.2.4

Shinyoung Kim

Mar 23, 2023

CONTENTS

Release
0.2.4

date
Mar 23, 2023

`astropy_new` is a python module to search for new astrophysics submissions in arXiv based on the user interests and to create a summarized document that can be easily opened in a browser.

CHAPTER

ONE

FEATURES

- Managing user interests (subjects, author names, and keywords)
- Searching for new submissions in astro-ph based on the user interests
- Creating a summarized HTML document of the interested submissions
- Daily automatic searching and summary on a set time.

CHAPTER

TWO

INSTALLATION

2.1 Using pip

Assuming you have Python already, install `astroph_new` with `pip` simply run:

```
pip install -U astroph-new
```

2.2 Checking installation

If your installation is OK, you can import this module without any error:

```
python -c 'import astroph_new'
```

CHAPTER

THREE

DOCUMENTATION

3.1 Getting started

3.1.1 Configuring interests

3.1.1.1 Creating folder for astro-ph

Create a folder where the astro-ph daily summaries will be stored in an easy-to-access and frequently viewed path (e.g., `~/Desktop/`).

```
% cd ~/Desktop
Desktop % mkdir astro-ph
Desktop % cd astro-ph
```

Note: Initializing and managing interests in the REPL mode of the Python shell is convenient.

```
astro-ph % python
```

3.1.1.2 Initializing interests

`astroph_new` searches interested new submissions in astro-ph based on the user interests, a list of interested subjects, authors, and keywords. To initialize your interests, you can make an empty interests file by using `init_interest()`:

```
>>> import astroph_new as apn
>>> apn.init_interest()
'.interests' was initialized.
```

The default filename for saving your interests is `.interests`, but you can use different filenames with `file='filename'` for various purposes, e.g., separating interests by area of interest:

```
>>> apn.init_interest(file='keywords_for_polarization')
'keywords_for_polarization' was initialized.
```

The `init_interest()` has three options:

file
: [string] file name to save the interests

default
[bool] update the default filename to `file`, default is `False`

overwrite
[bool] overwrite to the existing file, default is `False`

You can change the default filename with an option of `default=True` or with the `set_params()` function.

```
>>> apn.init_interest(file='.mykeywords', default=True)
The default file name was changed to '.mykeywords'.
'.mykeywords' was initialized.
```

```
>>> apn.set_params('file', '.interests')
>>> apn.init_interest(overwrite=True)
'.interests' was initialized.
```

3.1.1.3 Managing interests

The `astroph_new` module contains functions helpful in managing user interests stored in the file name (or path) given by `file`.

Note: Functions that manage user interests, such as `read_interest()`, `add_interest()`, and `remove_interest()`, have an input parameter `file` specifying the name of the user interests file. If it is not given, the default file name is used, which was set in `init_interest()` or `set_params()` and can be checked with `get_params()`.

```
>>> apn.get_params('file')
'.interests'
```

You can add the interested categories of astro-ph, interested author names, and search keywords for the titles or abstracts into your interests list using `add_interest()`.

```
>>> subject = ['SR', 'GA']
>>> author = ['Di Francesco, James', 'Neal J. Evans II', 'Caselli, Paola']
>>> keyword = ['molecular cloud', 'filament', 'dense core', 'prestellar']
>>> apn.add_interest(subject=subject, author=author, keyword=keyword)
'SR' is added to the 'subject' list.
'GA' is added to the 'subject' list.
'Di Francesco, James' is added to the 'author' list.
'Neal J. Evans II' is added to the 'author' list.
'Caselli, Paola' is added to the 'author' list.
'molecular cloud' is added to the 'keyword' list.
'filament' is added to the 'keyword' list.
'dense core' is added to the 'keyword' list.
'prestellar' is added to the 'keyword' list.
'interests.txt' was updated.
```

The `add_interest()` has the following inputs,

subject
[string or list of strings] interested categories in astro-ph

- `'GA'` : Astrophysics of Galaxies

- `'CO'` : Cosmology and Nongalactic Astrophysics
- `'EP'` : Earth and Planetary Astrophysics
- `'HE'` : High Energy Astrophysical Phenomena
- `'IM'` : Instrumentataion and Methods for Astrophysics
- `'SR'` : Solar and Stellar Astrophysics

author
[string or list of strings] interested authors (`Family, Given` or `Given Family`)

keyword
[string or list of strings] interested keywords for the title and abstract (case insensitive)

To check the interests list:

```
>>> interest = apn.read_interest()
>>> print(interest['author'])
['Caselli, Paola', 'Di Francesco, James', 'Neal J. Evans II']
```

To add additional interests:

```
>>> apn.add_interest(subject='IM')
'IM' is added to the 'subject' list.
'interests.txt' was updated.
```

```
>>> apn.add_interest(author='Philip C. Myers')
'Philip C. Myers' is added to the 'author' list.
'interests.txt' was updated.
```

Obvious duplicates cannot be added:

```
>>> apn.add_interest(author=['Paola Caselli', 'Myers, P. C.'])
'Paola Caselli' is already exist in the 'author' list.
'Myers, P. C.' is already exist in the 'author' list.
Nothing changed!
```

Keyword searches for titles and abstracts ignore cases, but are sensitive to spaces and hyphens:

```
>>> apn.add_interest(keyword=['starless', 'protostellar'])
'starless' is added to the 'keyword' list.
'protostellar' is added to the 'keyword' list.
'interests.txt' was updated.
```

```
>>> apn.add_interest(keyword=['proto-stellar', 'pre-stellar'])
'proto-stellar' is added to the 'keyword' list.
'pre-stellar' is added to the 'keyword' list.
'interests.txt' was updated.
```

```
>>> print(apn.read_interest()['keyword'])
['dense core', 'filament', 'molecular cloud', 'pre-stellar', 'prestellar',
 'proto-stellar', 'protostellar', 'starless']
```

To remove no longer used keywords of interest:

```
>>> apn.remove_interest(subject='IM', keyword='proto-stellar')
'IM' is removed from the 'subject' list.
'proto-stellar' is removed from the 'keyword' list.
'interests.txt' was updated.
```

Note: You can access and update the saved user interests using text editors, such as `vim` or `emacs`.

3.1.2 Searching based on interests

Important: In this `astroph_new` module, downloading new submissions from astro-ph, searching based on user interests, and making a summary report of the search results are performed in three steps by `get_new()`, `search_new()`, and `make_report()`. However, since each step automatically calls the previous step function, **you can skip this section and only need to run** `make_report()`.

The `get_new()` function opens new page of astro-ph in the virtual web browser using the `selenium` module and downloads the page source. Abstracts in astro-ph are compiled with mathjax, which takes some running time. So, `get_new()` will repeat the download with an interval of 5 seconds and check no changes to the page source. The `get_new()` returns the processed download result as a Python dictionary with key names `class`, `link`, `title`, `author`, and `abstract`.

```
>>> newsub = apn.get_new()
>>> newsub.keys()
dict_keys(['class', 'link', 'title', 'author', 'subject', 'abstract'])
```

You can set an input argument `refresh` for `get_new()` to avoid getting duplicate data.

`refresh`
: [int] If new submission data received from astro-ph was stored within the refresh time (minutes), it returns the saved data without receiving the data again. If you want to receive the data again, set `refresh=0`. Default is `30` (minutes).

The `search_new()` function searches interested new submissions based on the user interests, which is saved in the running folder with the given name by `file`. The `search_new()` returns a python tuple that contains the output of `get_new()` and an index list of the interested submissions.

```
>>> newsub, idx = apn.search_new()
keyword 'molecular cloud' is found in the title of [10] (https://arxiv.org/abs/2303.
↪07410)
keyword 'starless' is found in the title of [19] (https://arxiv.org/abs/2303.07501)
keyword 'molecular cloud' is found in the abstract of [10] (https://arxiv.org/abs/2303.
↪07410)
keyword 'starless' is found in the abstract of [19] (https://arxiv.org/abs/2303.07501)
keyword 'molecular cloud' is found in the abstract of [36] (https://arxiv.org/abs/2303.
↪07628)
keyword 'starless' is found in the abstract of [36] (https://arxiv.org/abs/2303.07628)
keyword 'molecular cloud' is found in the abstract of [41] (https://arxiv.org/abs/2303.
↪07752)
keyword 'filament' is found in the abstract of [60] (https://arxiv.org/abs/2303.08088)
```

```
>>> print(idx)
[10, 19, 36, 41, 60]
```

3.1.3 Making summary report

The `make_report()` function creates an `HTML` document, which is the summary report of the interested new submissions and can be quickly and conveniently opened in any browser, such as `Google Chrome` or `Safari`.

```
>>> apn.make_report()
keyword 'molecular cloud' is found in the title of [10] (https://arxiv.org/abs/2303.
↪07410)
keyword 'starless' is found in the title of [19] (https://arxiv.org/abs/2303.07501)
keyword 'molecular cloud' is found in the abstract of [10] (https://arxiv.org/abs/2303.
↪07410)
keyword 'starless' is found in the abstract of [19] (https://arxiv.org/abs/2303.07501)
keyword 'molecular cloud' is found in the abstract of [36] (https://arxiv.org/abs/2303.
↪07628)
keyword 'starless' is found in the abstract of [36] (https://arxiv.org/abs/2303.07628)
keyword 'molecular cloud' is found in the abstract of [41] (https://arxiv.org/abs/2303.
↪07752)
keyword 'filament' is found in the abstract of [60] (https://arxiv.org/abs/2303.08088)
'astro-ph_20230315.html' was saved.
```

The `make_report()` has the following options:

prefix
: [string] The prefix of the filename for saving the report HTML page. Default is `'astro-ph'`

datetag
: [bool] Add a date tag at the end of the saved file name to prevent overwriting and preserve older reports. Default is `True`.

timetag
: [bool] Add a time tag at the end of the saved file name. Default is `False`.

show
: [bool] After the report is created, it is automatically displayed in the browser. Default is `False`.

If you set the `show` option as `True`, the `make_report()` automatically displays the report page in the browser. For this feature, you should set the `'show'` parameter, which is a shell command for opening an HTML file in a browser, using `set_params()`. For example, if you use `Google Chrome` on the `mac`, it is `'open -a "Google Chrome"'` (default).

```
>>> apn.set_params('show', 'open -a "Google Chrome"')
>>> apn.get_params('show')
'open -a "Google Chrome"'
>>> apn.make_report(show=True)
```

3.1.4 Scheduling for astroph_new

The `run_apn()` function executes `make_report()` at user-designated time until the given end date. If the user-designated time has already passed at the time starting `run_apn()`, `make_report()` will be executed immediately, and from the next date, it will be executed at that time.

```
>>> apn.run_apn()
keyword 'molecular cloud' is found in the title of [10] (https://arxiv.org/abs/2303.
↪07410)
keyword 'starless' is found in the title of [19] (https://arxiv.org/abs/2303.07501)
keyword 'molecular cloud' is found in the abstract of [10] (https://arxiv.org/abs/2303.
↪07410)
keyword 'starless' is found in the abstract of [19] (https://arxiv.org/abs/2303.07501)
keyword 'molecular cloud' is found in the abstract of [36] (https://arxiv.org/abs/2303.
↪07628)
keyword 'starless' is found in the abstract of [36] (https://arxiv.org/abs/2303.07628)
keyword 'molecular cloud' is found in the abstract of [41] (https://arxiv.org/abs/2303.
↪07752)
keyword 'filament' is found in the abstract of [60] (https://arxiv.org/abs/2303.08088)
'astro-ph_20230315.html' was saved.
Next searching: 2023-03-16 11:00
Waiting ...
```

Warning: The `run_apn()` is not a daemon that runs in the background, so it is terminated when you close the running Python shell or terminal.

You can designate the time and end date for `run_apn()`.

at
: [string, `'HH:MM'`] a daily search and report generation time. Default is `'11:00'`.

end
: [string, `'yyyy-mm-dd'`] the end date of `run_apn()`. If not given, it is automatically set to 4 days later. For example, if you start `run_apn()` on Monday, the end date will be set Friday.

You can also set `end` as the end date of the year, like `2023-12-31`. However, these schedules can be terminated prematurely for many reasons, such as the Python kernel shutting down, closing a terminal window, or the system rebooting for an update.

```
>>> apn.run_apn(at='12:30', end='2023-12-31')
Next searching: 2023-03-16 12:30
Waiting ...
```

As mentioned in the description of the `end` argument, if you run `run_apn()` on Monday without specifying `end`, you can get a summary report at the given time every day until Friday. Additionally, if you're trying to have a daily habit of checking astro-ph, the `show=True` option is strongly recommended.

```
>>> apn.run_apn(at='11:00', show=True)
Next searching: 2023-03-22 11:00
Waiting ...
```

The `run_apn()` delivers input arguments to the `make_report()` and `search_new()`. So, you can set the `refresh`, `file`, `prefix`, `datetag`, `timetag`, and `show` options for all daily executions.

```
>>> apn.run_apn(file='keywords_ppdisk', prefix='ppdisk', show=True)
author 'Expert, P. Disk' is found in [#] (https://arxiv.org/abs/####.#####)
keyword 'protoplanetary disk' is found in [#] (https://arxiv.org/abs/####.#####)
keyword 'transitional disk' is found in [#] (https://arxiv.org/abs/####.#####)
'ppdisk_20230315.html' was saved.
Next searching: 2023-03-16 11:00
Waiting ...
```

Contents