
ai*news*_{writer}*Documentation*

Release 0.1.1

R. Brent Stephenson

Nov 12, 2019

Contents:

1	ai_news_writer	1
1.1	Features	1
1.2	Credits	2
2	Installation	3
2.1	Stable release	3
2.2	From sources	3
3	Usage	5
4	Contributing	7
4.1	Types of Contributions	7
4.2	Get Started!	8
4.3	Pull Request Guidelines	9
4.4	Tips	9
4.5	Deploying	9
5	Credits	11
5.1	Development Lead	11
5.2	Contributors	11
6	History	13
6.1	0.1.2 (2019-11-01)	13
6.2	0.1.1 (2019-10-19)	13
6.3	0.1.0 (2019-10-18)	14
7	Indices and tables	15

The primary purpose of this project is to create a news aggregator that gathers headlines and summaries announcements, news, and trending information about artificial intelligence and other cool technology stuff. After it has collected the articles and announcements, I would like it to generate a summary giving the highlights of that week's news. Ideally, after it has collected and summarized the information, I would like to generate a script based on the summaries and hopefully do a weekly online show discussing the happenings and other current events.

Additionally, this is an exercise in for me to learn Cookiecutter, Git, ReadTheDocs, and Travis.CI. I have primarily worked with TFS, SVN, and Jenkins for CI. This will be a *big* learning experience for me. I'm ready to bang my head against my desk and learn everything I can.

I am also taking this opportunity to learn *pytest* and *pytest-bdd* to help me take the time a plan out what I want to do.

And finally, I have decided that I will also build a *django* front-end for this project. This will allow me to maintain the RSS feed collection, gather articles, and generate my output (scripts). I'm really jumping in this whole hog and trying to see what's going to stick. In any case, this will be fun and a great learning experience.

If you would like to contribute this project, please feel free and have fun!

- Free software: MIT license
- Documentation: <https://ai-news-writer.readthedocs.io>.

1.1 Features

- RSS Aggregator - Collecting news and other information on a configured topic
- Article Summarizing - Analyzes the content of each of the articles and summarizes the articles
- Content Generation - Generate basic script based on the gathered articles and processed summaries

1.2 Credits

This package was created with [Cookiecutter](#) and the [audreyr/cookiecutter-pypackage](#) project template.

2.1 Stable release

To install `ai_news_writer`, run this command in your terminal:

```
$ pip install ai_news_writer
```

This is the preferred method to install `ai_news_writer`, as it will always install the most recent stable release.

If you don't have `pip` installed, this [Python installation guide](#) can guide you through the process.

2.2 From sources

The sources for `ai_news_writer` can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/RBStephensno/ai_news_writer
```

Or download the [tarball](#):

```
$ curl -OJL https://github.com/RBStephensno/ai_news_writer/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```


CHAPTER 3

Usage

To use `ai_news_writer` in a project:

```
import ai_news_writer
```


Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

4.1 Types of Contributions

4.1.1 Report Bugs

Report bugs at https://github.com/RBStephensno/ai_news_writer/issues.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

4.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” and “help wanted” is open to whoever wants to implement it.

4.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “enhancement” and “help wanted” is open to whoever wants to implement it.

4.1.4 Write Documentation

ai_news_writer could always use more documentation, whether as part of the official ai_news_writer docs, in docstrings, or even on the web in blog posts, articles, and such.

4.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/RBStephensno/ai_news_writer/issues.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

4.2 Get Started!

Ready to contribute? Here's how to set up ai_news_writer for local development.

1. Fork the ai_news_writer repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/ai_news_writer.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv ai_news_writer
$ cd ai_news_writer/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 ai_news_writer tests
$ python setup.py test or pytest
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

4.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.7, 3.5, 3.6 and 3.7, and for PyPy. Check https://travis-ci.org/RBStephensno/ai_news_writer/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Tips

To run a subset of tests:

```
$ pytest tests.test_ai_news_writer
```

4.5 Deploying

A reminder for the maintainers on how to deploy. Make sure all your changes are committed (including an entry in HISTORY.rst). Then run:

```
$ bump2version patch # possible: major / minor / patch
$ git push
$ git push --tags
```

Travis will then deploy to PyPI if tests pass.

5.1 Development Lead

- R. Brent Stephenson <brent.stephenson@brenttheprogrammer.com>

5.2 Contributors

None yet. Why not be the first?

6.1 0.1.2 (2019-11-01)

- Added new RSS News Feed class
- Added unit tests for RSS News Feed
- Added RSS News Feed List feature
- Added RSS News Feed step definitions

6.1.1 Reconsidering

After adding *pytest-bdd* I became overwhelmed with trying to write the actual test cases for some reason. I have decided to write regular unit tests for the time being and take a look at BDD at a later date. I don't want all progress to halt while I try to figure out BDD. I am keeping everything in place up to this point and will make the build pass, other than that BDD is on hold.

- Started using *pytest-bdd* for feature and test development
- Updated readme file to include some more details and ideas of what I want to accomplish with this project.

6.2 0.1.1 (2019-10-19)

- Corrected issues with the readme file
- Several minor updates for build configurations
- Bumped the version number to test out bump2version
- Removed PyPI link from readme because I don't plan on publishing this to PyPI

6.3 0.1.0 (2019-10-18)

- First release on PyPI.

CHAPTER 7

Indices and tables

- `genindex`
- `modindex`
- `search`