
pyLDAPvis Documentation

Release 2.1.2

Ben Mabey

Feb 06, 2018

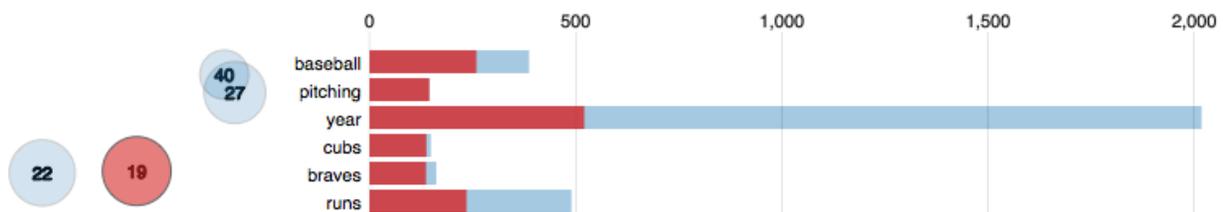
Contents

1	pyLDavis	3
1.1	Installation	3
1.2	Usage	4
1.3	Video demos	4
1.4	More documentation	4
2	Contributing	5
2.1	Types of Contributions	5
2.2	Get Started!	6
2.3	Pull Request Guidelines	7
3	Credits	9
3.1	Development Lead	9
3.2	Contributors	9
4	History	11
5	2.1.2 (2018-02-06)	13
6	2.1.1 (2017-02-13)	15
7	2.1.0 (2016-06-30)	17
8	2.0.0 (2016-06-30)	19
9	1.5.1 (2016-04-15)	21
10	1.5.0 (2016-02-20)	23
11	1.4.1 (2016-01-31)	25
12	1.4.0 (2016-01-31)	27
13	1.3.4 (2015-11-16)	29
14	1.3.3 (2015-11-13)	31
15	1.3.2 (2015-11-09)	33

16	1.3.1 (2015-11-02)	35
17	1.3.0 (2015-08-20)	37
18	1.2.0 (2015-06-13)	39
19	1.1.0 (2015-06-02)	41
20	1.0.0 (2015-05-29)	43
21	Indices and tables	45

Contents:

Python library for interactive topic model visualization. This is a port of the fabulous R package by Carson Sievert and Kenny Shirley.



pyLDavis is designed to help users interpret the topics in a topic model that has been fit to a corpus of text data. The package extracts information from a fitted LDA topic model to inform an interactive web-based visualization.

The visualization is intended to be used within an IPython notebook but can also be saved to a stand-alone HTML file for easy sharing.

build failing

1.1 Installation

- Stable version using pip:

```
pip install pyldavis
```

- Development version on GitHub

Clone the repository and run `python setup.py`

1.2 Usage

The best way to learn how to use **pyLDAvis** is to see it in action. Check out this [notebook for an overview](#). Refer to the [documentation](#) for details.

For a concise explanation of the visualization see this [vignette](#) from the LDAvis R package.

1.3 Video demos

Ben Mabey walked through the visualization in this short talk using a Hacker News corpus:

- [Visualizing Topic Models](#)
- [Notebook and visualization used in the demo](#)
- [Slide deck](#)

Carson Sievert <<https://cpsievert.me/>> created a video demoing the R package. The visualization is the same and so it applies equally to pyLDAvis:

- [Visualizing & Exploring the Twenty Newsgroup Data](#)

1.4 More documentation

To read about the methodology behind pyLDAvis, see the [original paper](#), which was presented at the [2014 ACL Workshop on Interactive Language Learning, Visualization, and Interfaces](#) in Baltimore on June 27, 2014.

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

2.1 Types of Contributions

2.1.1 Report Bugs

Report bugs at <https://github.com/bmabey/pyLDAPvis/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.

2.1.2 Fix Bugs

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

2.1.3 Implement Features

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

2.1.4 Write Documentation

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

2.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/bmabey/pyLDAPvis/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 :)

2.2 Get Started!

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

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

```
$ git clone git@github.com:your_name_here/pyLDAPvis.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 pyLDAPvis
$ cd pyLDAPvis/
$ 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 pyLDAPvis tests
$ python setup.py test
$ 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.

2.3 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 2.6, 2.7, 3.3, and 3.4, and for PyPy. Check https://travis-ci.org/bmabey/pyLDavis/pull_requests and make sure that the tests pass for all supported Python versions.

3.1 Development Lead

- Ben Mabey <ben@benmabey.com>

3.2 Contributors

- Paul English <paul@onfrst.com> - JS and CSS fixes and improvements.

CHAPTER 4

History

CHAPTER 5

2.1.2 (2018-02-06)

- Fix pandas deprecation warnings.

CHAPTER 6

2.1.1 (2017-02-13)

- Fix *gensim* module to work with a sparse corpus #82.

CHAPTER 7

2.1.0 (2016-06-30)

- Added missing dependency on *scipy*.
- Fixed term sorting that was incompatible with pandas 0.19.x.

CHAPTER 8

2.0.0 (2016-06-30)

- Removed dependency on *scikit-bio* by adding an internal PCoA implementation.
- Added helper functions for scikit-learn LDA model! See the new notebook for details.
- Extended gensim helper functions to work with HDP models.
- Added scikit-learn's Multi-dimensional scaling as another MDS option when scikit-learn is installed.

CHAPTER 9

1.5.1 (2016-04-15)

- Add `sort_topics` option to `prepare` function to allow disabling of topic re-ordering.

- Red Bar Width bug fix

In some cases, the widths of the red topic-term bars did not decrease (as they should have) from term #1 to term #R under the relevance ranking with $\lambda = 1$. In other words, when $\lambda = 1$, there were topics in which a narrow red bar was displayed above a wider red bar, which should never happen. The issue had to do with the way topic-term bar widths are computed, and is discussed in detail in #32.

In the end, we implemented a quick fix in which we compute term frequencies implicitly, rather than using those supplied in the `createJSON()` function. The upside is that the red bar widths are now explicitly controlled to produce the correct visualization. The downside is that the blue bar widths do not necessarily match the user-supplied term frequencies exactly – in fact, the new version of LDAvis ignores the user-supplied term frequencies entirely. In a few experiments, the differences are small, and decrease (as a proportion of the true term frequencies) as the true term frequencies increase.

CHAPTER 11

1.4.1 (2016-01-31)

- Included requirements.txt in MANIFEST to (hopefully) fix bad release.

CHAPTER 12

1.4.0 (2016-01-31)

- Updated to newest version of skibio for PCoA mds.
- requirements.txt cleanup
- New 'tsne' option for prepare, see docs and notebook for more info.

1.3.5 (2015-12-18)

- Add explicit version info for scikit-bio since the API has changed.

CHAPTER 13

1.3.4 (2015-11-16)

- Gensim Python typo fix in imports. :/

CHAPTER 14

1.3.3 (2015-11-13)

- Gensim Python 2.x fix for absolute imports.

CHAPTER 15

1.3.2 (2015-11-09)

- Gensim prepare 25% speed increase, thanks @mattilyra!
- Pandas deprecation warnings are now gone.
- Pandas v0.17 is now being used.

CHAPTER 16

1.3.1 (2015-11-02)

- Updates gensim and other logic to be python 3 compatible.

CHAPTER 17

1.3.0 (2015-08-20)

- Fixes gensim logic and makes it more robust.
- Faster graphlab processing.
- kargs for gensim and graphlab are passed down to underlying prepare function.
- Requires recent version of pandas to avoid problems with our use of the newer *DataFrame.to_dict* API.

CHAPTER 18

1.2.0 (2015-06-13)

- Updates gensim logic to be clearer and work with Python 3.x.

CHAPTER 19

1.1.0 (2015-06-02)

- Fixes bug with GraphLab function that was producing bogus visualizations.

CHAPTER 20

1.0.0 (2015-05-29)

- First release on PyPI. Faithful port of R version with IPython support and helper functions for GraphLab & gensim.

CHAPTER 21

Indices and tables

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