Who Not to Follow Documentation

Release 0.1.0

Anthony Shaw

May 16, 2016

Contents:

CHAPTER 1

Who Not to Follow

Twitter has a recommendation algorithm. Based on who you follow and your description it recommends more people that you might find interesting.

This is really helpful, allows you to build up a lot of followers (or followees) in your world, but there is one fatal flaw in the algorithm.

It assumes you want to follow people who are all more or less the same. I realised that the 1,000 people I am following on twitter are guys, in their 20's and 30's, with kids, interesting in cloud, technology, programming and Python. So basically copies of myself. What am I really going to learn from them? Sure I'll polish my technical skills and find out about the lastest cool new utility and project going around. But it won't expand my world view on anything and I'll become a more narrow minded individual. An anti-recommendation algorithm for twitter

- Free software: MIT license
- Documentation: https://wntf.readthedocs.org.

1.1 How it works

I initially thought I could just search for the opposite (antonym) of my profile and search for that. It's not that simple! What is the opposite of 'enthusiast' and does that even make sense?

The algorithm takes the profile description of your followers on Twitter, uses a natural language processing tool (NLTK) to understand characteristics of your followers.

The first thing to look at is the nouns in the followers description and the most common nouns. For me, this is :

{'NN': [('https', 80), ('cloud', 56), ('@', 39), ('technology', 36), ('http', 31), ('software', 30), ('developer', 28), ('business', 28), ('world', 26), ('father', 24), ('news', 23), ('fan', 21), ('account', 20), ('source', 20), ('husband', 20), ('enthusiast', 18), ('team', 18), ('web', 18), ('geek', 17), ('code', 17)], }

So what can we tell about those nouns?

We then filter out certain nouns that commonly occur, such as 'tweet', 'views', 'opinions', since a lot of people have a statement about their views not representing their employer etc. etc.

Once you filter that list I can see that my followers' characteristics in a few traits:

- Their industry 'business', 'technology'
- Their role 'developer'
- Their gender 'husband', 'father'
- Their interests 'web', 'code', 'software'

- The way they describe themselves 'geek', 'enthusiast'

Looking at the Proper nouns I can also get some other interesting information:

> **'NNP': [('@', 313),** ('Cloud', 92), ('I', 74), ('Data', 63), ('IT', 44), ('Dimension', 39), ('Software', 36), ('Microsoft', 35), ('Director', 35), ('Python', 32), ('Manager', 31), ('Husband', 26), ('Developer', 25), ('CTO', 25), ('Architect', 25), ('CEO', 24), ('Engineer', 24), ('/', 24), ('Technology', 23), ('Dad', 23)],

Again filtering out some of the fluff, like @ and /

- Company data Microsoft, Dimension (Data)
- Role 'CTO', 'CEO', Engineer, Architect

Now for each of these words we build up a synset. A Synset is a set of synonyms that share a common meaning. So for 'technology', the synset includes the nouns 'technology' and 'engineering'.

Then we look at the following characteristics of the word 'technology':

- The hypernyms, in this case 'application' and 'profession' (we are interested in this)
- The hyponyms (subsets), 'aeronautical engineering', 'automotive technology' 'chemical engineering' etc.

1.2 The diversity wheel

The diversity wheel has many characterstics, such as :

- gender
- background
- interests
- religion

If we put the nouns into buckets in the diversity wheel based on their hypernyms then we find a better view of your following.

Also, we don't want to assume that all of your followers are the same, so we'll weight each instance based on it's frequency.

1.3 TODO

This is half finished (my flight leaves soon):

- build up a map of wheels and the words in those wheels (e.g professions, regligion, interests)
- map those in a chord where the size of the difference between the angles in the chord represents the difference, e.g. carpenter is very different to IT, but banker is similar to accountant
- rank all your words into the chords, show the chord diagrams and then plot points furthest away and use those words to make recommendations.

The wheels will represent the diversity wheel (http://web.jhu.edu/sebin/t/m/DiversityWheel_Small.jpg)

1.4 Credits

This package was created with Cookiecutter and (a fork of) the **'audreyr/cookiecutter-pypackage'_** project template.

CHAPTER 2

Installation

At the command line:

```
$ easy_install wntf
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv wntf
$ pip install wntf
```

CHAPTER 3

Usage

To use Who Not to Follow in a project:

```
import wntf
```

CHAPTER 4

Contributing

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/tonybaloney/wntf/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" is open to whoever wants to implement it.

4.1.3 Implement Features

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

4.1.4 Write Documentation

Who Not to Follow could always use more documentation, whether as part of the official Who Not to Follow 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/tonybaloney/wntf/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 *wntf* for local development.

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

```
$ git clone git@github.com:your_name_here/wntf.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 wntf
$ cd wntf/
$ 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 wntf 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.

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 2.6, 2.7, 3.3, 3.4 and 3.5, and for PyPy. Check https://travis-ci.org/tonybaloney/wntf/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Tips

To run a subset of tests:

```
$ python -m unittest tests.test_wntf
```

CHAPTER 5

Credits

5.1 Development Lead

- Anthony Shaw <anthonyshaw@apache.org>

5.2 Contributors

None yet. Why not be the first?

CHAPTER 6

History

6.1 0.1.0 (2016-04-04)

- First release on PyPI.

CHAPTER 7

Indices and tables

- genindex
- modindex
- search