

---

# **labMTsimple Documentation**

***Release 2.8.4***

**Andy Reagan**

**Jun 08, 2018**



---

## Contents

---

<b>1</b>	<b>Getting Started</b>	<b>3</b>
1.1	Usage . . . . .	3
1.2	Installation . . . . .	3
1.3	Running tests . . . . .	4
1.4	Developing with labMT-simple locally . . . . .	4
1.5	Building these docs . . . . .	4
<b>2</b>	<b>Detailed Examples</b>	<b>5</b>
2.1	Preparing texts . . . . .	5
2.2	Loading dictionaries . . . . .	5
<b>3</b>	<b>Making Wordshifts</b>	<b>7</b>
3.1	Installing phantom-crowbar . . . . .	7
3.2	Installing inkscape . . . . .	8
3.3	Installing rsvg . . . . .	8
3.4	Installing labMTsimple . . . . .	8
3.5	Making your first shift . . . . .	8
3.6	Full Automation . . . . .	9
<b>4</b>	<b>Advanced Usage</b>	<b>11</b>
4.1	About Tries . . . . .	11
4.2	Advanced Parsing . . . . .	11
<b>5</b>	<b>labMTsimple package</b>	<b>13</b>
5.1	labMTsimple.speedy module . . . . .	13
5.2	labMTsimple.storyLab module . . . . .	24



Contents:



# CHAPTER 1

---

## Getting Started

---

TL;DR a simple labMT usage script

### 1.1 Usage

This script uses the language assessment by Mechanical Turk (labMT) word list to score the happiness of a corpus. The labMT word list was created by combining the 5000 words most frequently appearing in four sources: Twitter, the New York Times, Google Books, and music lyrics, and then scoring the words for sentiment on Amazon's Mechanical Turk. The list is described in detail in the publication Dodds' et al. 2011, PLOS ONE, "Temporal Patterns of Happiness and Information in a Global-Scale Social Network: Hedonometrics and Twitter."

Given two corpora, the script "storylab.py" creates a word-shift graph illustrating the words most responsible for the difference in happiness between the two corpora. The corpora should be large (e.g. at least 10,000 words) in order for the difference to be meaningful, as this is a bag-of-words approach. As an example, a random collection of English tweets from both Saturday January 18 2014 and Tuesday January 21 2014 are included in the "example" directory. They can be compared by moving to the test directory, using the command

```
python example.py example-shift.html
```

and opening the file `example-shift.html` in a web browser. For an explanation of the resulting plot, please visit <http://www.hedonometer.org/shifts.html>

### 1.2 Installation

Cloning the github directly is recommended, and then installing locally:

```
git clone https://github.com/andyreagan/labMT-simple.git
cd labMT-simple
python setup.py install
```

This repository can also be installed using pip

```
pip install labMTsimple
```

in which case you can download the tests from github and run them, if desired.

## 1.3 Running tests

Tests are based on nose2, `pip install nose2`, and can be run inside the by executing

```
nose2
```

in the root directory of this repository.

This will compare the two days in test/data and print test.html which shifts them, allowing for a changable lens.

## 1.4 Developing with labMT-simple locally

I find it really useful to reload the library when testing it interactively:

```
try:
    reload
except NameError:
    # Python 3
    from importlib import reload
```

## 1.5 Building these docs

Go into the docs directory (activate local `virtualenv` first), and do the following:

```
\rm -rf _build/*
make html
make latexpdf
git add -f *
git commit -am "new docs, probably should just add a pre-commit hook"
```

Note that these docs will build locally in python 2 because the dependencies exist. With python 3 available, these dependencies will be mocked (and this is set for the online readthedocs site).

(`sphinx-apidoc -o . ../labMTsimple` was run once.)



### 2.1 Preparing texts

This is simple really: just load the text to be scored into python. I'm using a subset of a couple days of public tweets to text, and I've already put the tweet text into `.txt` files that I load into strings:

```
f = codecs.open("data/18.01.14.txt", "r", "utf8")
saturday = f.read()
f.close()

f = codecs.open("data/21.01.14.txt", "r", "utf8")
tuesday = f.read()
f.close()
```

### 2.2 Loading dictionaries

Again this is really simple, just use the `emotionFileReader` function:

```
lang = 'english'
labMT, labMTvector, labMTwordList = emotionFileReader(stopval=0.0, lang=lang,
↳returnVector=True)
```

Then we can score the text and get the word vector at the same time:

```
saturdayValence, saturdayFvec = emotion(saturday, labMT, shift=True,
↳happsList=labMTvector)
tuesdayValence, tuesdayFvec = emotion(tuesday, labMT, shift=True, happsList=labMTvector)
```

But we don't want to use these happiness scores yet, because they included all words (including neutral words). So, set all of the neutral words to 0, and generate the scores:

```
tuesdayStoppedVec = stopper(tuesdayFvec, labMTvector, labMTwordList, stopVal=1.0)
saturdayStoppedVec = stopper(saturdayFvec, labMTvector, labMTwordList, stopVal=1.0)

saturdayValence = emotionV(saturdayStoppedVec, labMTvector)
tuesdayValence = emotionV(tuesdayStoppedVec, labMTvector)
```

## CHAPTER 3

---

### Making Wordshifts

---

I just merged updates to the d3 wordshift plotting into labMTsimple, and combined with phantom crowbar (see previous post), it's easier than ever to use the labMT data set to compare texts.

To make an html page with the shift, you'll just need to have labMT-simple installed. To automate the process into generating svg files, you'll need the phantom crowbar, which depends on phantomjs. To go all the way to pdf, you'll also need inkscape for making vectorized pdfs, or rsvg for making better formatted, but rasterized, versions.

Let's get set up to make shifts automatically. Since they're aren't many dependencies all the way down, start by getting phantomjs installed, then the phantom-crowbar.

### 3.1 Installing phantom-crowbar

For the phantomjs, I prefer to use homebrew:

```
brew update
brew upgrade
brew install phantomjs
```

Then to get the crowbar, clone the git repository.

```
cd ~
git clone https://github.com/andyreagan/phantom-crowbar
```

To use it system-wide, I use the bash alias:

```
alias phantom-crowbar="/usr/local/bin/phantomjs ~/phantom-crowbar/phantom-crowbar.js"
```

Without too much detail, I recommend adding this to your ~/.bash\_profile so that it's loaded every time you start a terminal session.

## 3.2 Installing inkscape

You only need inkscape if you want to go from svg to pdf (and there are other ways too), but this one is easy with, again, homebrew.

```
brew install inkscape
```

## 3.3 Installing rsvg

You only need inkscape if you want to go from svg to pdf (and there are other ways too), but this one is easy with, again, homebrew.

```
brew install librsvg
```

## 3.4 Installing labMTsimple

There are two ways to get it: using pip or cloning the git repo. If you're not sure, use pip. I think pip makes it easier to keep it up to date, etc.

```
pip install labMTsimple
```

## 3.5 Making your first shift

If you cloned the git repository, install the thing and then you can check out the example in `examples/example.py`. If you went with pip, see that file on [github](#).

Go ahead and run that script!

```
python example-002.py
```

You can open the html file to see the shift in any browser, with your choice of local webserver. Python's SimpleHTTPServer works fine, and I've found that the node based http-server is a bit more stable.

To take out the svg, go ahead and use the `phantom-crowbar.js` file copied to the `example/static` directory. Running it looks like this, for me:

```
/usr/local/bin/phantomjs js/shift-crowbar.js example-002.html shiftsvg wordshift.svg
```

Using inkscape or librsvg on my computer look like this:

```
/Applications/Inkscape.app/Contents/Resources/bin/inkscape -f $(pwd)/wordshift.svg -A  
→ $(pwd)/wordshift-inkscape.pdf  
  
rsvg-convert --format=eps wordshift.svg > wordshift-rsvg.eps  
epstopdf wordshift-rsvg.eps
```

And again, feel free to tweet suggestions at [@andyreagan](#), and submit pull requests to the [source code](#)!

## 3.6 Full Automation

I've wrapped up all of this into what is potentially the most backwards way to generate figure imaginable. The `shiftPDF()` function operates the same way as the `shiftHTML()`, but uses the headless web server to render the d3 graphic, then executes a piece of injected JS to save a local SVG, and uses command line image manipulation libraries to massage it into a PDF.

On my macbook, this works, but your mileage will most certainly vary.



### 4.1 About Tries

For dictionary lookup of scores from phrases, the fastest benchmarks I've found and that were reasonable stable were from the libraries `datrie` and `marisatrie` which both have python bindings.

They're used in the `speedy` module in an attempt to both speed things up, and match against word stems.

### 4.2 Advanced Parsing

Some dictionaries use word stems to cover the multiple uses of a single word, with a single score. We can very quickly match these word stems using a prefix match on a trie. This is much better than using many compiled RE matches, which in my testing took a very long time.





## 5.1 labMTsimple.speedy module

```

class labMTsimple.speedy.AFINN (datastructure='dict', stopVal=0.0, bananas=False, loadFrom-
                                File=False, saveFile=False, lang='english')
    Bases: labMTsimple.speedy.sentiDict
    center = 0.0
    citation = "@inproceedings{nielsen2011new,\n\tAuthor = {Nielsen, Finn {\AA}rup},\n\tB
    citation_key = 'nielsen2011new'
    construction_note = 'Manual'
    license = 'ODbL v1.0'
    loadDict (bananas, lang)
    note = 'Words manually rated -5 to 5 with impact scores by Finn Nielsen'
    score_range_type = 'integer'
    stems = False
    title = 'AFINN'

class labMTsimple.speedy.ANEW (datastructure='dict', stopVal=0.0, bananas=False, loadFrom-
                                File=False, saveFile=False, lang='english')
    Bases: labMTsimple.speedy.sentiDict
    ANEW class.
    center = 5.0
    citation = '@techreport{bradley1999affective,\n\tAddress = {Gainesville, FL},\n\tAutho
    citation_key = 'bradley1999affective'
    construction_note = 'Survey: FSU Psych 101'

```

```
license = 'Free for research'
loadDict (bananas, lang)
    Load the corpus into a dictionary, straight from the origin corpus file.
note = 'Affective Norms of English Words'
score_range_type = 'continuous'
stems = False
title = 'ANEW'
class labMTsimple.speedy.EmoLex (datastructure='dict', stopVal=0.0, bananas=False, loadFrom-
                                File=False, saveFile=False, lang='english')
    Bases: labMTsimple.speedy.sentiDict
center = 0.0
citation = '@article{mohammad2013crowdsourcing,\n title={Crowdsourcing a word--emotion
citation_key = 'mohammad2013crowdsourcing'
construction_note = 'Survey: MT'
license = 'Free for research'
loadDict (bananas, lang)
note = 'NRC Word-Emotion Association Lexicon: emotions and sentiment evoked by common
score_range_type = 'integer'
stems = False
title = 'EmoLex'
class labMTsimple.speedy.EmoSenticNet (datastructure='dict', stopVal=0.0, ba-
                                      nanas=False, loadFromFile=False, saveFile=False,
                                      lang='english')
    Bases: labMTsimple.speedy.sentiDict
center = 0.0
citation = '@article{poria2013enhanced,\n\tAuthor = {Poria, Soujanya and Gelbukh, Alex
citation_key = 'poria2013enhanced'
construction_note = 'Bootstrapped extension'
license = 'Non-commercial'
loadDict (bananas, lang)
note = 'extends SenticNet words with WNA labels'
score_range_type = 'integer'
stems = False
title = 'EmoSenticNet'
url = 'http://www.gelbukh.com/emosenticnet/'
class labMTsimple.speedy.Emoticons (datastructure='dict', stopVal=0.0, bananas=False, load-
                                   FromFile=False, saveFile=False, lang='english')
    Bases: labMTsimple.speedy.sentiDict
center = 0.0
```

```

citation = "@inproceedings{gonccalves2013comparing,\n title={Comparing and combining s
citation_key = 'gonccalves2013comparing'
construction_note = 'Manual'
license = 'Open source code'
loadDict (bananas, lang)
note = 'Commonly used emoticons with their positive, negative, or neutral emotion'
score_range_type = 'integer'
stems = False
title = 'Emoticons'

class labMTsimple.speedy.GI (datastructure='dict', stopVal=0.0, bananas=False, loadFrom-
                           File=False, saveFile=False, lang='english')
Bases: labMTsimple.speedy.sentiDict
center = 0.0
citation = '@article{stone1966general,\n\tAuthor = {Stone, Philip J and Dunphy, Dexter
citation_key = 'stone1966general'
construction_note = 'Harvard-IV-4'
license = 'Unspecified'
loadDict (bananas, lang)
note = 'General Inquirer: database of words and manually created semantic and cogniti
score_range_type = 'integer'
stems = False
title = 'GI'

class labMTsimple.speedy.HashtagSent (datastructure='dict', stopVal=0.0, bananas=False,
                                     loadFromFile=False, saveFile=False, lang='english')
Bases: labMTsimple.speedy.sentiDict
center = 0.0
citation = '@inproceedings{zhu2014nrc,\n title={Nrc-canada-2014: Recent improvements
citation_key = 'zhu2014nrc'
construction_note = 'PMI with hashtags'
license = 'Free for research'
loadDict (bananas, lang)
note = 'NRC Hashtag Sentiment Lexicon: created from tweets using Pairwise Mutual Info
score_range_type = 'continuous'
stems = False
title = 'HashtagSent'

class labMTsimple.speedy.LIWC (datastructure='dict', stopVal=0.0, bananas=False, loadFrom-
                             File=False, saveFile=False, lang='english')
Bases: labMTsimple.speedy.sentiDict

```

LIWC class.

**affect** = 125

**center** = 0.0

**citation** = '@article{pennebaker2001linguistic},\n\tAuthor = {Pennebaker, James W and F

**citation\_key** = 'pennebaker2001linguistic'

**construction\_note** = 'Manual'

**license** = 'Paid, commercial'

**loadDict** (*bananas, lang*)

Load the corpus into a dictionary, straight from the origin corpus file.

**negative** = 127

**note** = 'Linguistic Inquiry and Word Count, three version'

**positive** = 126

**score\_range\_type** = 'integer'

**stems** = True

**title** = 'LIWC'

**word\_types** = {}

**year** = '07'

```
class labMTsimple.speedy.LIWC01 (datastructure='dict', stopVal=0.0, bananas=False, loadFrom-  
File=False, saveFile=False, lang='english')  
    Bases: labMTsimple.speedy.LIWC
```

**affect** = 12

**negative** = 16

**positive** = 13

**title** = 'LIWC01'

**year** = '01'

```
class labMTsimple.speedy.LIWC07 (datastructure='dict', stopVal=0.0, bananas=False, loadFrom-  
File=False, saveFile=False, lang='english')  
    Bases: labMTsimple.speedy.LIWC
```

This is the default, define it anyway

**affect** = 125

**negative** = 127

**positive** = 126

**title** = 'LIWC07'

**year** = '07'

```
class labMTsimple.speedy.LIWC15 (datastructure='dict', stopVal=0.0, bananas=False, loadFrom-  
File=False, saveFile=False, lang='english')  
    Bases: labMTsimple.speedy.LIWC
```

**affect** = 30

**negative** = 32

```

    positive = 31
    title = 'LIWC15'
    year = '15'
class labMTsimple.speedy.LabMT (datastructure='dict', stopVal=0.0, bananas=False, loadFrom-
                                File=False, saveFile=False, lang='english')
    Bases: labMTsimple.speedy.sentiDict
    LabMT class.
    Now takes the full name of the language.
    center = 5.0
    citation = '@article{dodds2015human,\n\tAuthor = {Dodds, P. S. and Clark, E. M. and De
    citation_key = 'dodds2015human'
    construction_note = 'Survey: MT, 50 ratings'
    license = 'CC'
    loadDict (bananas, lang)
    note = 'language assessment by Mechanical Turk'
    score_range_type = 'continuous'
    stems = False
    title = 'labMT'
class labMTsimple.speedy.MPQA (datastructure='dict', stopVal=0.0, bananas=False, loadFrom-
                                File=False, saveFile=False, lang='english')
    Bases: labMTsimple.speedy.sentiDict
    MPQA class.
    center = 0.0
    citation = '@article{wilson2005recognizing,\n\tAuthor = {Theresa Wilson and Janyce Wier
    citation_key = 'wilson2005recognizing'
    construction_note = 'Manual + ML'
    license = 'GNU GPL'
    loadDict (bananas, lang)
        Load the corpus into a dictionary, straight from the origin corpus file.
    note = 'The Multi-Perspective Question Answering (MPQA) Subjectivity Dictionary'
    score_range_type = 'integer'
    stems = True
    title = 'MPQA'
class labMTsimple.speedy.MaxDiff (datastructure='dict', stopVal=0.0, bananas=False, load-
                                FromFile=False, saveFile=False, lang='english')
    Bases: labMTsimple.speedy.sentiDict
    center = 0.0
    citation = '@article{kiritchenko2014sentiment,\n title={Sentiment analysis of short in
    citation_key = 'kiritchenko2014sentiment'

```

```
    construction_note = 'Survey:  MT, MaxDiff'
    license = 'Free for research'
    loadDict (bananas, lang)
    note = 'NRC MaxDiff Twitter Sentiment Lexicon:  crowdsourced real-valued scores using
    score_range_type = 'continuous'
    stems = False
    title = 'MaxDiff'
class labMTsimple.speedy.OL (datastructure='dict',  stopVal=0.0,  bananas=False,  loadFrom-
                           File=False, saveFile=False, lang='english')
    Bases: labMTsimple.speedy.sentiDict
    center = 0.0
    citation = '@article{liu2010sentiment,\n\tAuthor = {Liu, Bing},\n\tJournal = {Handbook
    citation_key = 'liu2010sentiment'
    construction_note = 'Dictionary propagation'
    license = 'Free'
    loadDict (bananas, lang)
        Load the corpus into a dictionary, straight from the origin corpus file.
    note = 'Opinion Lexicon, developed by Bing Liu)'
    score_range_type = 'integer'
    stems = False
    title = 'OL'
class labMTsimple.speedy.PANASX (datastructure='dict',  stopVal=0.0,  bananas=False, loadFrom-
                               File=False, saveFile=False, lang='english')
    Bases: labMTsimple.speedy.sentiDict
    center = 0.0
    citation = '@jurthesis{watson1999panas,\n\tAuthor = {Watson, David and Clark, Lee Anna
    citation_key = 'watson1999panas'
    construction_note = 'Manual'
    license = 'Copyrighted paper'
    loadDict (bananas, lang)
    note = 'The Positive and Negative Affect Schedule --- Expanded'
    score_range_type = 'integer'
    stems = False
    title = 'PANAS-X'
class labMTsimple.speedy.Pattern (datastructure='dict',  stopVal=0.0,  bananas=False, load-
                                FromFile=False, saveFile=False, lang='english')
    Bases: labMTsimple.speedy.sentiDict
    center = 0.0
    citation = '@article{de2012pattern,\n\tAuthor = {De Smedt, Tom and Daelemans, Walter},
```

```

citation_key = 'de2012pattern'
construction_note = 'Unspecified'
license = 'BSD'
loadDict (bananas, lang)
note = 'A web mining module for the Python programming language, version 2.6'
score_range_type = 'continuous'
stems = False
title = 'Pattern'

class labMTsimple.speedy.SOCAL (datastructure='dict', stopVal=0.0, bananas=False, loadFrom-
                               File=False, saveFile=False, lang='english')
    Bases: labMTsimple.speedy.sentiDict
    center = 0.0
    citation = '@article{taboada2011lexicon,\n Author = {Taboada, Maite and Brooke, Julian
    citation_key = 'taboada2011lexicon'
    construction_note = 'Manual'
    license = 'GNU GPL'
    loadDict (bananas, lang)
    note = 'Manually constructed general-purpose sentiment dictionary'
    score_range_type = 'continuous'
    stems = True
    title = 'SOCAL'

class labMTsimple.speedy.Sent140Lex (datastructure='dict', stopVal=0.0, bananas=False, load-
                                   FromFile=False, saveFile=False, lang='english')
    Bases: labMTsimple.speedy.sentiDict
    center = 0.0
    citation = '@inproceedings{MohammadKZ2013,\n\tAddress = {Atlanta, Georgia, USA},\n\tAu
    citation_key = 'MohammadKZ2013'
    construction_note = 'PMI with emoticons'
    license = 'Free for research'
    loadDict (bananas, lang)
    note = "NRC Sentiment140 Lexicon:  created from the ``sentiment140'' corpus of tweets,
    score_range_type = 'continuous'
    stems = False
    title = 'Sent140Lex'

class labMTsimple.speedy.SentiStrength (datastructure='dict', stopVal=0.0, ba-
                                       nanas=False, loadFromFile=False, saveFile=False,
                                       lang='english')
    Bases: labMTsimple.speedy.sentiDict
    center = 0.0

```

```

citation = '@article{thelwall2010sentiment,\n\tAuthor = {Thelwall, Mike and Buckley, K
citation_key = 'thelwall2010sentiment'
construction_note = 'LIWC+GI'
license = 'Free for research'
loadDict (bananas, lang)
note = 'an API and Java program for general purpose sentiment detection (here we use o
score_range_type = 'integer'
stems = True
title = 'SentiStrength'
url = 'http://sentistrength.wlv.ac.uk/'
class labMTsimple.speedy.SentiWordNet (datastructure='dict', stopVal=0.0, ba-
                                     nanas=False, loadFromFile=False, saveFile=False,
                                     lang='english')
Bases: labMTsimple.speedy.sentiDict
center = 0.0
citation = '@inproceedings{baccianella2010sentiwordnet,\n title={SentiWordNet 3.0: An
citation_key = 'baccianella2010sentiwordnet'
construction_note = 'Synset synonyms'
license = 'CC BY-SA 3.0'
loadDict (bananas, lang)
note = 'WordNet synsets each assigned three sentiment scores: positivity, negativity,
score_range_type = 'continuous'
stems = False
title = 'SentiWordNet'
class labMTsimple.speedy.SenticNet (datastructure='dict', stopVal=0.0, bananas=False, load-
                                   FromFile=False, saveFile=False, lang='english')
Bases: labMTsimple.speedy.sentiDict
center = 0.0
citation = '@inproceedings{cambria2014senticnet,\n title={SenticNet 3: a common and c
citation_key = 'cambria2014senticnet'
construction_note = 'Label propogation'
license = 'Citation requested'
loadDict (bananas, lang)
note = 'Sentiment dataset labelled with semantics and 5 dimensions of emotions by Camb
score_range_type = 'continuous'
stems = False
title = 'SenticNet'
```



```
class labMTsimple.speedy.USent (datastructure='dict', stopVal=0.0, bananas=False, loadFrom-
                               File=False, saveFile=False, lang='english')
    Bases: labMTsimple.speedy.sentiDict

    center = 0.0
    citation = '@inproceedings{pappas2013distinguishing,\n title={Distinguishing the popul
    citation_key = 'pappas2013distinguishing'
    construction_note = 'Manual'
    license = 'CC'
    loadDict (bananas, lang)
    note = 'set of emoticons and bad words that extend MPQA'
    score_range_type = 'integer'
    stems = False
    title = 'USent'

class labMTsimple.speedy.Umigon (datastructure='dict', stopVal=0.0, bananas=False, loadFrom-
                               File=False, saveFile=False, lang='english')
    Bases: labMTsimple.speedy.sentiDict

    center = 0.0
    citation = '@inproceedings{levallois2013umigon,\n\tAuthor = {Levallois, Clement},\n\tB
    citation_key = 'levallois2013umigon'
    construction_note = 'Manual'
    license = 'Public Domain'
    loadDict (bananas, lang)
    note = 'Manually built specifically to analyze Tweets from the sentiment140 corpus'
    score_range_type = 'integer'
    stems = False
    title = 'Umigon'

class labMTsimple.speedy.VADER (datastructure='dict', stopVal=0.0, bananas=False, loadFrom-
                               File=False, saveFile=False, lang='english')
    Bases: labMTsimple.speedy.sentiDict

    center = 0.0
    citation = '@inproceedings{hutto2014vader,\n title={Vader:  A parsimonious rule-based
    citation_key = 'hutto2014vader'
    construction_note = 'MT survey, 10 ratings'
    license = 'Freely available'
    loadDict (bananas, lang)
    note = 'method developed specifically for Twitter and social media analysis'
    score_range_type = 'continuous'
    stems = False
```

```

title = 'VADER'

class labMTsimple.speedy.WDAL (datastructure='dict', stopVal=0.0, bananas=False, loadFrom-
                               File=False, saveFile=False, lang='english')
    Bases: labMTsimple.speedy.sentiDict

    center = 1.5

    citation = "@article{whissell1986dictionary,\n\tAuthor = {Whissell, Cynthia and Fourni
    citation_key = 'whissell1986dictionary'

    construction_note = 'Survey: Columbia students'

    license = 'Unspecified'

    loadDict (bananas, lang)

    note = "Whissel's Dictionary of Affective Language: words rated in terms of their Ple
    score_range_type = 'continuous'

    stems = False

    title = 'WDAL'

class labMTsimple.speedy.WK (datastructure='dict', stopVal=0.0, bananas=False, loadFrom-
                              File=False, saveFile=False, lang='english')
    Bases: labMTsimple.speedy.sentiDict

    center = 5.0

    citation = '@article{warriner2013norms,\n\tAbstract = {Information about the affective
    citation_key = 'warriner2013norms'

    construction_note = 'Survey: MT, 14--18 ratings'

    license = 'CC'

    loadDict (bananas, lang)

    note = 'Warriner and Kuperman rated words from SUBTLEX by Mechanical Turk'

    score_range_type = 'continuous'

    stems = False

    title = 'WK'

class labMTsimple.speedy.sentiDict (datastructure='dict', stopVal=0.0, bananas=False, load-
                                      FromFile=False, saveFile=False, lang='english')
    Bases: object

    An abstract class to score them all.

    computeStatistics (stopVal)

    data = {}

    fmt = 'Hf'

    makeListsFromDict ()
        Make lists from a dict, used internally.

    makeMarisaTrie (save_flag=False)
        Turn a dictionary into a marisa_trie.

```

**matcherDictBool** (*word*)

MatcherTrieDict(word) just checks if a word is in the dict.

**matcherTrieBool** (*word*)

MatcherTrieBool(word) just checks if a word is in the list. Returns 0 or 1.

Works for both trie types. Only one needed to make the plots. Only use this for coverage, so don't even worry about using with a dict.

**matcherTrieDict** (*word, wordVec, count*)

Not sure what this one does.

**matcherTrieMarisa** (*word, wordVec, count*)

Not sure what this one does.

**my\_marisa** = (<marisa\_trie.RecordTrie object>, <marisa\_trie.RecordTrie object>)

Declare this globally.

**openWithPath** (*filename, mode*)

Helper function for searching for files.

**scoreTrieDict** (*wordDict, idx=1, center=0.0, stopVal=0.0*)

Score a wordDict using the dict backend.

INPUTS:

-wordDict is a favorite hash table of word and count.

**scoreTrieMarisa** (*wordDict, idx=1, center=0.0, stopVal=0.0*)

Score a wordDict using the marisa\_trie backend.

INPUTS:

-wordDict is a favorite hash table of word and count.

**stopData** ()

**stopper** (*tmpVec, stopVal=1.0, ignore=[]*)

Take a frequency vector, and 0 out the stop words.

Will always remove the nig\* words.

Return the 0'ed vector.

**wordVecifyTrieDict** (*wordDict*)

Make a word vec from word dict using dict backend.

INPUTS:

-wordDict is our favorite hash table of word and count.

**wordVecifyTrieMarisa** (*wordDict*)

Make a word vec from word dict using marisa\_trie backend.

INPUTS:

-wordDict is our favorite hash table of word and count.

**labMTsimple.speedy.u** (*x*)

Python 2/3 agnostic unicode function

## 5.2 labMTsimple.storyLab module

`labMTsimple.storyLab.copy_static_files()`

Deprecated method to copy files from this module's static directory into the directory where shifts are being made.

`labMTsimple.storyLab.emotion(tmpStr, someDict, scoreIndex=1, shift=False, happsList=[])`

Take a string and the happiness dictionary, and rate the string.

If `shift=True`, will return a vector (also then needs the `happsList`).

`labMTsimple.storyLab.emotionFileReader(stopval=1.0, lang='english', min=1.0, max=9.0, returnVector=False)`

Load the dictionary of sentiment words.

Stopval is our lens, `$Delta_h$`, read the labMT dataset into a dict with this lens (must be tab-delimited).

With `returnVector = True`, returns `tmpDict,tmpList,wordList`. Otherwise, just the dictionary.

`labMTsimple.storyLab.emotionV(frequencyVec, scoreVec)`

Given the frequency vector and the score vector, compute the happs.

Doesn't use numpy, but equivalent to `np.dot(freq,happs)/np.sum(freq)`.

`labMTsimple.storyLab.link_static_files()`

Same as `copy_static_files`, but makes symbolic links.

`labMTsimple.storyLab.shift(refFreq, compFreq, lens, words, sort=True)`

Compute a shift, and return the results.

If `sort=True`, will return the three sorted lists, and `sumTypes`. Else, just the two shift lists, and `sumTypes` (words don't need to be sorted).

`labMTsimple.storyLab.shiftHtml(scoreList, wordList, refFreq, compFreq, outFile, corpus='LabMT', advanced=False, customTitle=False, title="", ref_name='reference', comp_name='comparison', ref_name_happs="", comp_name_happs="", isare="")`

Make an interactive shift for exploring and sharing.

The most insane-o piece of code here (lots of file copying, writing vectors into html files, etc).

Accepts a score list, a word list, two frequency files and the name of an HTML file to generate

**\*\* will make the HTML file, and a directory called static that hosts a bunch of .js, .css that is useful.**

`labMTsimple.storyLab.shiftHtmlJupyter(scoreList, wordList, refFreq, compFreq, outFile, corpus='LabMT', advanced=False, customTitle=False, title="", ref_name='reference', comp_name='comparison', ref_name_happs="", comp_name_happs="", isare="", saveFull=True, selfshift=False, bgcolor='white')`

Shifter that generates HTML in two pieces, designed to work inside of a Jupyter notebook.

Saves the filename as given (with .html extension), and sneaks in a filename-wrapper.html, and the wrapper file has the html headers, everything to be a standalone file. The filenameed html is just the guts of the html file, because the complete markup isn't need inside the notebook.

`labMTsimple.storyLab.shiftHtmlPreshifted(scoreList, wordList, refFreq, compFreq, outFile, corpus='LabMT', advanced=False, customTitle=False, title="", ref_name='reference', comp_name='comparison', ref_name_happs="", comp_name_happs="", isare="")`

Make an interactive shift for exploring and sharing.

The most insane-o piece of code here (lots of file copying, writing vectors into html files, etc).

Accepts a score list, a word list, two frequency files and the name of an HTML file to generate

\*\* will make the HTML file, and a directory called static that hosts a bunch of .js, .css that is useful.

`labMTsimple.storyLab.stopper(tmpVec, score_list, word_list, stopVal=1.0, ignore=[], center=5.0)`

Take a frequency vector, and 0 out the stop words.

Will always remove the nig\* words.

Return the 0'ed vector.

`labMTsimple.storyLab.stopper_mat(tmpVec, score_list, word_list, stopVal=1.0, ignore=[], center=5.0)`

Take a frequency vector, and 0 out the stop words.

A sparse-aware matrix stopper. F-vecs are rows: [i,:]

Will always remove the nig\* words.

Return the 0'ed matrix, sparse.



## A

affect (labMTsimple.speedy.LIWC attribute), 16  
 affect (labMTsimple.speedy.LIWC01 attribute), 16  
 affect (labMTsimple.speedy.LIWC07 attribute), 16  
 affect (labMTsimple.speedy.LIWC15 attribute), 16  
 AFINN (class in labMTsimple.speedy), 13  
 ANEW (class in labMTsimple.speedy), 13

## C

center (labMTsimple.speedy.AFINN attribute), 13  
 center (labMTsimple.speedy.ANEW attribute), 13  
 center (labMTsimple.speedy.EmoLex attribute), 14  
 center (labMTsimple.speedy.EmoSenticNet attribute), 14  
 center (labMTsimple.speedy.Emoticons attribute), 14  
 center (labMTsimple.speedy.GI attribute), 15  
 center (labMTsimple.speedy.HashtagSent attribute), 15  
 center (labMTsimple.speedy.LabMT attribute), 17  
 center (labMTsimple.speedy.LIWC attribute), 16  
 center (labMTsimple.speedy.MaxDiff attribute), 17  
 center (labMTsimple.speedy.MPQA attribute), 17  
 center (labMTsimple.speedy.OL attribute), 18  
 center (labMTsimple.speedy.PANASX attribute), 18  
 center (labMTsimple.speedy.Pattern attribute), 18  
 center (labMTsimple.speedy.Sent140Lex attribute), 19  
 center (labMTsimple.speedy.SenticNet attribute), 20  
 center (labMTsimple.speedy.SentiStrength attribute), 19  
 center (labMTsimple.speedy.SentiWordNet attribute), 20  
 center (labMTsimple.speedy.SOCAL attribute), 19  
 center (labMTsimple.speedy.Umigon attribute), 21  
 center (labMTsimple.speedy.USent attribute), 21  
 center (labMTsimple.speedy.VADER attribute), 21  
 center (labMTsimple.speedy.WDAL attribute), 22  
 center (labMTsimple.speedy.WK attribute), 22  
 citation (labMTsimple.speedy.AFINN attribute), 13  
 citation (labMTsimple.speedy.ANEW attribute), 13  
 citation (labMTsimple.speedy.EmoLex attribute), 14  
 citation (labMTsimple.speedy.EmoSenticNet attribute), 14  
 citation (labMTsimple.speedy.Emoticons attribute), 14

citation (labMTsimple.speedy.GI attribute), 15  
 citation (labMTsimple.speedy.HashtagSent attribute), 15  
 citation (labMTsimple.speedy.LabMT attribute), 17  
 citation (labMTsimple.speedy.LIWC attribute), 16  
 citation (labMTsimple.speedy.MaxDiff attribute), 17  
 citation (labMTsimple.speedy.MPQA attribute), 17  
 citation (labMTsimple.speedy.OL attribute), 18  
 citation (labMTsimple.speedy.PANASX attribute), 18  
 citation (labMTsimple.speedy.Pattern attribute), 18  
 citation (labMTsimple.speedy.Sent140Lex attribute), 19  
 citation (labMTsimple.speedy.SenticNet attribute), 20  
 citation (labMTsimple.speedy.SentiStrength attribute), 19  
 citation (labMTsimple.speedy.SentiWordNet attribute), 20  
 citation (labMTsimple.speedy.SOCAL attribute), 19  
 citation (labMTsimple.speedy.Umigon attribute), 21  
 citation (labMTsimple.speedy.USent attribute), 21  
 citation (labMTsimple.speedy.VADER attribute), 21  
 citation (labMTsimple.speedy.WDAL attribute), 22  
 citation (labMTsimple.speedy.WK attribute), 22  
 citation\_key (labMTsimple.speedy.AFINN attribute), 13  
 citation\_key (labMTsimple.speedy.ANEW attribute), 13  
 citation\_key (labMTsimple.speedy.EmoLex attribute), 14  
 citation\_key (labMTsimple.speedy.EmoSenticNet attribute), 14  
 citation\_key (labMTsimple.speedy.Emoticons attribute), 15  
 citation\_key (labMTsimple.speedy.GI attribute), 15  
 citation\_key (labMTsimple.speedy.HashtagSent attribute), 15  
 citation\_key (labMTsimple.speedy.LabMT attribute), 17  
 citation\_key (labMTsimple.speedy.LIWC attribute), 16  
 citation\_key (labMTsimple.speedy.MaxDiff attribute), 17  
 citation\_key (labMTsimple.speedy.MPQA attribute), 17  
 citation\_key (labMTsimple.speedy.OL attribute), 18  
 citation\_key (labMTsimple.speedy.PANASX attribute), 18  
 citation\_key (labMTsimple.speedy.Pattern attribute), 18  
 citation\_key (labMTsimple.speedy.Sent140Lex attribute), 19

- `citation_key` (labMTsimple.speedy.SenticNet attribute), 20
- `citation_key` (labMTsimple.speedy.SentiStrength attribute), 20
- `citation_key` (labMTsimple.speedy.SentiWordNet attribute), 20
- `citation_key` (labMTsimple.speedy.SOCAL attribute), 19
- `citation_key` (labMTsimple.speedy.Umigon attribute), 21
- `citation_key` (labMTsimple.speedy.USent attribute), 21
- `citation_key` (labMTsimple.speedy.VADER attribute), 21
- `citation_key` (labMTsimple.speedy.WDAL attribute), 22
- `citation_key` (labMTsimple.speedy.WK attribute), 22
- `computeStatistics()` (labMTsimple.speedy.sentiDict method), 22
- `construction_note` (labMTsimple.speedy.AFINN attribute), 13
- `construction_note` (labMTsimple.speedy.ANEW attribute), 13
- `construction_note` (labMTsimple.speedy.EmoLex attribute), 14
- `construction_note` (labMTsimple.speedy.EmoSenticNet attribute), 14
- `construction_note` (labMTsimple.speedy.Emoticons attribute), 15
- `construction_note` (labMTsimple.speedy.GI attribute), 15
- `construction_note` (labMTsimple.speedy.HashtagSent attribute), 15
- `construction_note` (labMTsimple.speedy.LabMT attribute), 17
- `construction_note` (labMTsimple.speedy.LIWC attribute), 16
- `construction_note` (labMTsimple.speedy.MaxDiff attribute), 17
- `construction_note` (labMTsimple.speedy.MPQA attribute), 17
- `construction_note` (labMTsimple.speedy.OL attribute), 18
- `construction_note` (labMTsimple.speedy.PANASX attribute), 18
- `construction_note` (labMTsimple.speedy.Pattern attribute), 19
- `construction_note` (labMTsimple.speedy.Sent140Lex attribute), 19
- `construction_note` (labMTsimple.speedy.SenticNet attribute), 20
- `construction_note` (labMTsimple.speedy.SentiStrength attribute), 20
- `construction_note` (labMTsimple.speedy.SentiWordNet attribute), 20
- `construction_note` (labMTsimple.speedy.SOCAL attribute), 19
- `construction_note` (labMTsimple.speedy.Umigon attribute), 21
- `construction_note` (labMTsimple.speedy.USent attribute), 21
- `construction_note` (labMTsimple.speedy.VADER attribute), 21
- `construction_note` (labMTsimple.speedy.WDAL attribute), 22
- `construction_note` (labMTsimple.speedy.WK attribute), 22
- `copy_static_files()` (in module labMTsimple.storyLab), 24
- ## D
- `data` (labMTsimple.speedy.sentiDict attribute), 22

## E

`EmoLex` (class in labMTsimple.speedy), 14

`EmoSenticNet` (class in labMTsimple.speedy), 14

`Emoticons` (class in labMTsimple.speedy), 14

`emotion()` (in module labMTsimple.storyLab), 24

`emotionFileReader()` (in module labMTsimple.storyLab), 24

`emotionV()` (in module labMTsimple.storyLab), 24

## F

`fmt` (labMTsimple.speedy.sentiDict attribute), 22

## G

`GI` (class in labMTsimple.speedy), 15

## H

`HashtagSent` (class in labMTsimple.speedy), 15

## L

`LabMT` (class in labMTsimple.speedy), 17

`labMTsimple.speedy` (module), 13

`labMTsimple.storyLab` (module), 24

`license` (labMTsimple.speedy.AFINN attribute), 13

`license` (labMTsimple.speedy.ANEW attribute), 13

`license` (labMTsimple.speedy.EmoLex attribute), 14

`license` (labMTsimple.speedy.EmoSenticNet attribute), 14

`license` (labMTsimple.speedy.Emoticons attribute), 15

`license` (labMTsimple.speedy.GI attribute), 15

`license` (labMTsimple.speedy.HashtagSent attribute), 15

`license` (labMTsimple.speedy.LabMT attribute), 17

`license` (labMTsimple.speedy.LIWC attribute), 16

`license` (labMTsimple.speedy.MaxDiff attribute), 18

`license` (labMTsimple.speedy.MPQA attribute), 17

`license` (labMTsimple.speedy.OL attribute), 18

`license` (labMTsimple.speedy.PANASX attribute), 18

`license` (labMTsimple.speedy.Pattern attribute), 19

`license` (labMTsimple.speedy.Sent140Lex attribute), 19

`license` (labMTsimple.speedy.SenticNet attribute), 20

`license` (labMTsimple.speedy.SentiStrength attribute), 20

`license` (labMTsimple.speedy.SentiWordNet attribute), 20



[license \(labMTsimple.speedy.SOCAL attribute\), 19](#)  
[license \(labMTsimple.speedy.Umigon attribute\), 21](#)  
[license \(labMTsimple.speedy.USent attribute\), 21](#)  
[license \(labMTsimple.speedy.VADER attribute\), 21](#)  
[license \(labMTsimple.speedy.WDAL attribute\), 22](#)  
[license \(labMTsimple.speedy.WK attribute\), 22](#)  
[link\\_static\\_files\(\) \(in module labMTsimple.storyLab\), 24](#)  
[LIWC \(class in labMTsimple.speedy\), 15](#)  
[LIWC01 \(class in labMTsimple.speedy\), 16](#)  
[LIWC07 \(class in labMTsimple.speedy\), 16](#)  
[LIWC15 \(class in labMTsimple.speedy\), 16](#)  
[loadDict\(\) \(labMTsimple.speedy.AFINN method\), 13](#)  
[loadDict\(\) \(labMTsimple.speedy.ANEW method\), 14](#)  
[loadDict\(\) \(labMTsimple.speedy.EmoLex method\), 14](#)  
[loadDict\(\) \(labMTsimple.speedy.EmoSenticNet method\), 14](#)  
[loadDict\(\) \(labMTsimple.speedy.Emoticons method\), 15](#)  
[loadDict\(\) \(labMTsimple.speedy.GI method\), 15](#)  
[loadDict\(\) \(labMTsimple.speedy.HashtagSent method\), 15](#)  
[loadDict\(\) \(labMTsimple.speedy.LabMT method\), 17](#)  
[loadDict\(\) \(labMTsimple.speedy.LIWC method\), 16](#)  
[loadDict\(\) \(labMTsimple.speedy.MaxDiff method\), 18](#)  
[loadDict\(\) \(labMTsimple.speedy.MPQA method\), 17](#)  
[loadDict\(\) \(labMTsimple.speedy.OL method\), 18](#)  
[loadDict\(\) \(labMTsimple.speedy.PANASX method\), 18](#)  
[loadDict\(\) \(labMTsimple.speedy.Pattern method\), 19](#)  
[loadDict\(\) \(labMTsimple.speedy.Sent140Lex method\), 19](#)  
[loadDict\(\) \(labMTsimple.speedy.SenticNet method\), 20](#)  
[loadDict\(\) \(labMTsimple.speedy.SentiStrength method\), 20](#)  
[loadDict\(\) \(labMTsimple.speedy.SentiWordNet method\), 20](#)  
[loadDict\(\) \(labMTsimple.speedy.SOCAL method\), 19](#)  
[loadDict\(\) \(labMTsimple.speedy.Umigon method\), 21](#)  
[loadDict\(\) \(labMTsimple.speedy.USent method\), 21](#)  
[loadDict\(\) \(labMTsimple.speedy.VADER method\), 21](#)  
[loadDict\(\) \(labMTsimple.speedy.WDAL method\), 22](#)  
[loadDict\(\) \(labMTsimple.speedy.WK method\), 22](#)

## M

[makeListsFromDict\(\) \(labMTsimple.speedy.sentiDict method\), 22](#)  
[makeMarisaTrie\(\) \(labMTsimple.speedy.sentiDict method\), 22](#)  
[matcherDictBool\(\) \(labMTsimple.speedy.sentiDict method\), 22](#)  
[matcherTrieBool\(\) \(labMTsimple.speedy.sentiDict method\), 23](#)  
[matcherTrieDict\(\) \(labMTsimple.speedy.sentiDict method\), 23](#)  
[matcherTrieMarisa\(\) \(labMTsimple.speedy.sentiDict method\), 23](#)

[MaxDiff \(class in labMTsimple.speedy\), 17](#)  
[MPQA \(class in labMTsimple.speedy\), 17](#)  
[my\\_marisa \(labMTsimple.speedy.sentiDict attribute\), 23](#)

## N

[negative \(labMTsimple.speedy.LIWC attribute\), 16](#)  
[negative \(labMTsimple.speedy.LIWC01 attribute\), 16](#)  
[negative \(labMTsimple.speedy.LIWC07 attribute\), 16](#)  
[negative \(labMTsimple.speedy.LIWC15 attribute\), 16](#)  
[note \(labMTsimple.speedy.AFINN attribute\), 13](#)  
[note \(labMTsimple.speedy.ANEW attribute\), 14](#)  
[note \(labMTsimple.speedy.EmoLex attribute\), 14](#)  
[note \(labMTsimple.speedy.EmoSenticNet attribute\), 14](#)  
[note \(labMTsimple.speedy.Emoticons attribute\), 15](#)  
[note \(labMTsimple.speedy.GI attribute\), 15](#)  
[note \(labMTsimple.speedy.HashtagSent attribute\), 15](#)  
[note \(labMTsimple.speedy.LabMT attribute\), 17](#)  
[note \(labMTsimple.speedy.LIWC attribute\), 16](#)  
[note \(labMTsimple.speedy.MaxDiff attribute\), 18](#)  
[note \(labMTsimple.speedy.MPQA attribute\), 17](#)  
[note \(labMTsimple.speedy.OL attribute\), 18](#)  
[note \(labMTsimple.speedy.PANASX attribute\), 18](#)  
[note \(labMTsimple.speedy.Pattern attribute\), 19](#)  
[note \(labMTsimple.speedy.Sent140Lex attribute\), 19](#)  
[note \(labMTsimple.speedy.SenticNet attribute\), 20](#)  
[note \(labMTsimple.speedy.SentiStrength attribute\), 20](#)  
[note \(labMTsimple.speedy.SentiWordNet attribute\), 20](#)  
[note \(labMTsimple.speedy.SOCAL attribute\), 19](#)  
[note \(labMTsimple.speedy.Umigon attribute\), 21](#)  
[note \(labMTsimple.speedy.USent attribute\), 21](#)  
[note \(labMTsimple.speedy.VADER attribute\), 21](#)  
[note \(labMTsimple.speedy.WDAL attribute\), 22](#)  
[note \(labMTsimple.speedy.WK attribute\), 22](#)

## O

[OL \(class in labMTsimple.speedy\), 18](#)  
[openWithPath\(\) \(labMTsimple.speedy.sentiDict method\), 23](#)

## P

[PANASX \(class in labMTsimple.speedy\), 18](#)  
[Pattern \(class in labMTsimple.speedy\), 18](#)  
[positive \(labMTsimple.speedy.LIWC attribute\), 16](#)  
[positive \(labMTsimple.speedy.LIWC01 attribute\), 16](#)  
[positive \(labMTsimple.speedy.LIWC07 attribute\), 16](#)  
[positive \(labMTsimple.speedy.LIWC15 attribute\), 16](#)

## S

[score\\_range\\_type \(labMTsimple.speedy.AFINN attribute\), 13](#)  
[score\\_range\\_type \(labMTsimple.speedy.ANEW attribute\), 14](#)  
[score\\_range\\_type \(labMTsimple.speedy.EmoLex attribute\), 14](#)

score\_range\_type (labMTsimple.speedy.EmoSenticNet attribute), 14

score\_range\_type (labMTsimple.speedy.Emoticons attribute), 15

score\_range\_type (labMTsimple.speedy.GI attribute), 15

score\_range\_type (labMTsimple.speedy.HashtagSent attribute), 15

score\_range\_type (labMTsimple.speedy.LabMT attribute), 17

score\_range\_type (labMTsimple.speedy.LIWC attribute), 16

score\_range\_type (labMTsimple.speedy.MaxDiff attribute), 18

score\_range\_type (labMTsimple.speedy.MPQA attribute), 17

score\_range\_type (labMTsimple.speedy.OL attribute), 18

score\_range\_type (labMTsimple.speedy.PANASX attribute), 18

score\_range\_type (labMTsimple.speedy.Pattern attribute), 19

score\_range\_type (labMTsimple.speedy.Sent140Lex attribute), 19

score\_range\_type (labMTsimple.speedy.SenticNet attribute), 20

score\_range\_type (labMTsimple.speedy.SentiStrength attribute), 20

score\_range\_type (labMTsimple.speedy.SentiWordNet attribute), 20

score\_range\_type (labMTsimple.speedy.SOCAL attribute), 19

score\_range\_type (labMTsimple.speedy.Umigon attribute), 21

score\_range\_type (labMTsimple.speedy.USent attribute), 21

score\_range\_type (labMTsimple.speedy.VADER attribute), 21

score\_range\_type (labMTsimple.speedy.WDAL attribute), 22

score\_range\_type (labMTsimple.speedy.WK attribute), 22

scoreTrieDict() (labMTsimple.speedy.sentiDict method), 23

scoreTrieMarisa() (labMTsimple.speedy.sentiDict method), 23

Sent140Lex (class in labMTsimple.speedy), 19

SenticNet (class in labMTsimple.speedy), 20

sentiDict (class in labMTsimple.speedy), 22

SentiStrength (class in labMTsimple.speedy), 19

SentiWordNet (class in labMTsimple.speedy), 20

shift() (in module labMTsimple.storyLab), 24

shiftHtml() (in module labMTsimple.storyLab), 24

shiftHtmlJupyter() (in module labMTsimple.storyLab), 24

shiftHtmlPreshifted() (in module labMTsimple.storyLab), 24

SOCAL (class in labMTsimple.speedy), 19

stems (labMTsimple.speedy.AFINN attribute), 13

stems (labMTsimple.speedy.ANEW attribute), 14

stems (labMTsimple.speedy.EmoLex attribute), 14

stems (labMTsimple.speedy.EmoSenticNet attribute), 14

stems (labMTsimple.speedy.Emoticons attribute), 15

stems (labMTsimple.speedy.GI attribute), 15

stems (labMTsimple.speedy.HashtagSent attribute), 15

stems (labMTsimple.speedy.LabMT attribute), 17

stems (labMTsimple.speedy.LIWC attribute), 16

stems (labMTsimple.speedy.MaxDiff attribute), 18

stems (labMTsimple.speedy.MPQA attribute), 17

stems (labMTsimple.speedy.OL attribute), 18

stems (labMTsimple.speedy.PANASX attribute), 18

stems (labMTsimple.speedy.Pattern attribute), 19

stems (labMTsimple.speedy.Sent140Lex attribute), 19

stems (labMTsimple.speedy.SenticNet attribute), 20

stems (labMTsimple.speedy.SentiStrength attribute), 20

stems (labMTsimple.speedy.SentiWordNet attribute), 20

stems (labMTsimple.speedy.SOCAL attribute), 19

stems (labMTsimple.speedy.Umigon attribute), 21

stems (labMTsimple.speedy.USent attribute), 21

stems (labMTsimple.speedy.VADER attribute), 21

stems (labMTsimple.speedy.WDAL attribute), 22

stems (labMTsimple.speedy.WK attribute), 22

stopData() (labMTsimple.speedy.sentiDict method), 23

stopper() (in module labMTsimple.storyLab), 25

stopper() (labMTsimple.speedy.sentiDict method), 23

stopper\_mat() (in module labMTsimple.storyLab), 25

## T

title (labMTsimple.speedy.AFINN attribute), 13

title (labMTsimple.speedy.ANEW attribute), 14

title (labMTsimple.speedy.EmoLex attribute), 14

title (labMTsimple.speedy.EmoSenticNet attribute), 14

title (labMTsimple.speedy.Emoticons attribute), 15

title (labMTsimple.speedy.GI attribute), 15

title (labMTsimple.speedy.HashtagSent attribute), 15

title (labMTsimple.speedy.LabMT attribute), 17

title (labMTsimple.speedy.LIWC attribute), 16

title (labMTsimple.speedy.LIWC01 attribute), 16

title (labMTsimple.speedy.LIWC07 attribute), 16

title (labMTsimple.speedy.LIWC15 attribute), 17

title (labMTsimple.speedy.MaxDiff attribute), 18

title (labMTsimple.speedy.MPQA attribute), 17

title (labMTsimple.speedy.OL attribute), 18

title (labMTsimple.speedy.PANASX attribute), 18

title (labMTsimple.speedy.Pattern attribute), 19

title (labMTsimple.speedy.Sent140Lex attribute), 19

title (labMTsimple.speedy.SenticNet attribute), 20

title (labMTsimple.speedy.SentiStrength attribute), 20

title (labMTsimple.speedy.SentiWordNet attribute), 20

title (labMTsimple.speedy.SOCAL attribute), 19

title (labMTsimple.speedy.Umigon attribute), 21  
title (labMTsimple.speedy.USent attribute), 21  
title (labMTsimple.speedy.VADER attribute), 21  
title (labMTsimple.speedy.WDAL attribute), 22  
title (labMTsimple.speedy.WK attribute), 22

## U

u() (in module labMTsimple.speedy), 23  
Umigon (class in labMTsimple.speedy), 21  
url (labMTsimple.speedy.EmoSenticNet attribute), 14  
url (labMTsimple.speedy.SentiStrength attribute), 20  
USent (class in labMTsimple.speedy), 20

## V

VADER (class in labMTsimple.speedy), 21

## W

WDAL (class in labMTsimple.speedy), 22  
WK (class in labMTsimple.speedy), 22  
word\_types (labMTsimple.speedy.LIWC attribute), 16  
wordVecifyTrieDict() (labMTsimple.speedy.sentiDict  
method), 23  
wordVecifyTrieMarisa() (labMTsimple.speedy.sentiDict  
method), 23

## Y

year (labMTsimple.speedy.LIWC attribute), 16  
year (labMTsimple.speedy.LIWC01 attribute), 16  
year (labMTsimple.speedy.LIWC07 attribute), 16  
year (labMTsimple.speedy.LIWC15 attribute), 17