

---

# nefnir Documentation

*Release 1.0.1*

**Sverrir Á. Berg**

**Sep 14, 2019**



---

## Contents:

---

<b>1</b>	<b>nefnir</b>	<b>1</b>
1.1	Features . . . . .	1
1.2	Example . . . . .	1
1.3	Credits . . . . .	1
<b>2</b>	<b>Installation</b>	<b>3</b>
2.1	Stable release . . . . .	3
2.2	From sources . . . . .	3
<b>3</b>	<b>Usage</b>	<b>5</b>
<b>4</b>	<b>nefnir</b>	<b>7</b>
4.1	nefnir package . . . . .	7
<b>5</b>	<b>Contributing</b>	<b>11</b>
5.1	Types of Contributions . . . . .	11
5.2	Get Started! . . . . .	12
5.3	Pull Request Guidelines . . . . .	13
5.4	Tips . . . . .	13
5.5	Deploying . . . . .	13
<b>6</b>	<b>History</b>	<b>15</b>
6.1	1.0.0 (2019-09-14) . . . . .	15
<b>7</b>	<b>Indices and tables</b>	<b>17</b>
	<b>Python Module Index</b>	<b>19</b>
	<b>Index</b>	<b>21</b>



This is a package wrapping <https://github.com/jonfd/nefnir>

All credits to the author of that excellent project.

A lemmatizer for Icelandic text See e.g. `_IceNLP_bridge` for help creating tokenized and tagged text.

- Free software: Apache Software License 2.0
- Documentation: <https://nefnir.readthedocs.io>.

## 1.1 Features

- See [Nefnir](#) for full description.
- Simple lemmatizer for Icelandic.

## 1.2 Example

```
import nefnir
nefnir.lemmatize('Hvað', 'fshen')
```

## 1.3 Credits

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



### 2.1 Stable release

To install nefnir, run this command in your terminal:

```
$ pip install nefnir
```

This is the preferred method to install nefnir, 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 nefnir can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/sverrirab/nefnir-package
```

Or download the [tarball](#):

```
$ curl -OJL https://github.com/sverrirab/nefnir-package/tarball/master
```

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

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





## CHAPTER 3

---

### Usage

---

To use nefnir in a project:

```
import nefnir

nefnir.lemmatize('Hvað', 'fshen')
```



## 4.1 nefnir package

### 4.1.1 Submodules

### 4.1.2 nefnir.nefnir module

**class** nefnir.nefnir.Nefnir

Bases: object

A rule-based lemmatizer

**lemmatize** (*form, tag*)

Lemmatize a word form given its part-of-speech tag.

**Parameters**

- **form** – A word form.
- **tag** – The word form’s part-of-speech tag.

**Returns** The word form’s lemma.

**recase** (*form, tag, lemma*)

Determine how to properly case a lemma given the word form and part of speech tag it was derived from.

Nefnir transforms words into lowercase prior to lemmatization. Some words, such as proper nouns, abbreviations and foreign words therefore need to be re-capitalized or changed back into uppercase.

**Parameters**

- **form** – A word form, cased as it was written.
- **tag** – The word form’s part-of-speech tag.
- **lemma** – The word form’s lemma, in lowercase.

**Returns** A properly cased lemma.

`nefnir.nefnir.get_suffixes(s)`

Return an iterator yielding a string's suffixes, from the largest to the smallest.

**Parameters** *s* – A text string.

**Returns** An iterator for the string's suffixes.

`nefnir.nefnir.main()`

### 4.1.3 nefnir.wrapper module

`nefnir.wrapper.init()` → None

Read configuration files.

`nefnir.wrapper.lemmatize(form: str, tag: str) → str`

Lemmatize a word form given its part-of-speech tag.

**Parameters**

- **form** – A word form.
- **tag** – The word form's part-of-speech tag.

**Returns** The word form's lemma.

`nefnir.wrapper.lemmatize_line(line: str, separator: str = '\t') → Tuple[Optional[str], Optional[str], Optional[str]]`

Lemmatize a word form given its part-of-speech tag.

**Parameters**

- **line** – A line with form and tag separated by separator.
- **separator** – The token separator.

**Returns** Tuple with form, tag, lemma (any can be None if data invalid).

`nefnir.wrapper.recase(form: str, tag: str, lemma: str) → str`

Determine how to properly case a lemma given the word form and part of speech tag it was derived from.

Nefnir transforms words into lowercase prior to lemmatization. Some words, such as proper nouns, abbreviations and foreign words therefore need to be re-capitalized or changed back into uppercase.

**Parameters**

- **form** – A word form, cased as it was written.
- **tag** – The word form's part-of-speech tag.
- **lemma** – The word form's lemma, in lowercase.

**Returns** A properly cased lemma.

### 4.1.4 Module contents

Top-level package for nefnir (nefnir-package).

`nefnir.init()` → None

Read configuration files.

`nefnir.lemmatize(form: str, tag: str) → str`

Lemmatize a word form given its part-of-speech tag.

**Parameters**

- **form** – A word form.
- **tag** – The word form’s part-of-speech tag.

**Returns** The word form’s lemma.

`nefnir.lemmatize_line(line: str, separator: str = '\t') → Tuple[Optional[str], Optional[str], Optional[str]]`

Lemmatize a word form given its part-of-speech tag.

**Parameters**

- **line** – A line with form and tag separated by separator.
- **separator** – The token separator.

**Returns** Tuple with form, tag, lemma (any can be None if data invalid).

`nefnir.recase(form: str, tag: str, lemma: str) → str`

Determine how to properly case a lemma given the word form and part of speech tag it was derived from.

Nefnir transforms words into lowercase prior to lemmatization. Some words, such as proper nouns, abbreviations and foreign words therefore need to be re-capitalized or changed back into uppercase.

**Parameters**

- **form** – A word form, cased as it was written.
- **tag** – The word form’s part-of-speech tag.
- **lemma** – The word form’s lemma, in lowercase.

**Returns** A properly cased lemma.



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

## 5.1 Types of Contributions

### 5.1.1 Report Bugs

Report bugs at <https://github.com/sverrirab/nefnir-package/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.

### 5.1.2 Fix Bugs

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

### 5.1.3 Implement Features

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

### 5.1.4 Write Documentation

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

### 5.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/sverrirab/nefnir-package/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 :)

## 5.2 Get Started!

Ready to contribute? Here's how to set up *nefnir-package* for local development.

1. Fork the *nefnir-package* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/nefnir-package.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 nefnir-package
$ cd nefnir-package/
$ 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 nefnir 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.



## 5.3 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/sverrirab/nefnir/pull\\_requests](https://travis-ci.org/sverrirab/nefnir/pull_requests) and make sure that the tests pass for all supported Python versions.

## 5.4 Tips

To run a subset of tests:

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

## 5.5 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.



#### 6.1 1.0.0 (2019-09-14)

- First release on PyPI.



## CHAPTER 7

---

### Indices and tables

---

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



### n

- `nefnir`, 8
- `nefnir.nefnir`, 7
- `nefnir.wrapper`, 8





## G

`get_suffixes()` (in module *nefnir.nefnir*), 7

## I

`init()` (in module *nefnir*), 8

`init()` (in module *nefnir.wrapper*), 8

## L

`lemmatize()` (in module *nefnir*), 8

`lemmatize()` (in module *nefnir.wrapper*), 8

`lemmatize()` (*nefnir.nefnir.Nefnir* method), 7

`lemmatize_line()` (in module *nefnir*), 9

`lemmatize_line()` (in module *nefnir.wrapper*), 8

## M

`main()` (in module *nefnir.nefnir*), 8

## N

*Nefnir* (class in *nefnir.nefnir*), 7

*nefnir* (module), 8

*nefnir.nefnir* (module), 7

*nefnir.wrapper* (module), 8

## R

`recase()` (in module *nefnir*), 9

`recase()` (in module *nefnir.wrapper*), 8

`recase()` (*nefnir.nefnir.Nefnir* method), 7