# preprocessing

***Release 0.1.12***

**Oct 25, 2017**

# Contents


**1 Summary** **3**

**2 Installation** **5**

**3 Example** **7**

**4 Organisation** **9**

**5 Contributing** **11**

**6 License** **13**

**7 *preprocessing* Contents** **15**
- 7.1 Package Documentation . . . . . 15
- 7.2 License . . . . . 18

**Python Module Index** **19**


logo.png

# CHAPTER 1

## Summary

Text pre-processing package to aid in NLP package development for Python3. With this package you can order text cleaning functions in the order you prefer rather than relying on the order of an arbitrary NLP package.

# CHAPTER 2

# Installation

pip:

```
pip install preprocessing
```

PyPI - You can also download the source distribution from:

https://pypi.python.org/pypi/preprocessing/

You can then perform:

```
pip install <path_to_tar_file>
```

on the tar file, or

```
python setup.py install
```

on/inside, respectively, the extracted package to install *preprocessing*.

# CHAPTER 3

# Example

Once you have the package installed, implementing it with Python3 takes the following form:

```
import preprocessing.text as ptext
from preprocessing.text import keyword_tokenize, remove_unbound_punct, remove_urls

text_string = "important string at: http://example.com"

clean_string = ptext.preprocess_text(text_string, [
    remove_urls,
    remove_unbound_punct,
    keyword_tokenize
])
```

```
>>> print(clean_string)
"important string"
```

Should the functions be performed in a different order (i.e. keyword_tokenize -> remove_urls -> remove_non_bound_punct) :

```
>>> print(clean_string)
"important string http example.com"
```

CHAPTER 4

# Organisation

This package is comprised of a single module with no intended subpackages currently. The *preprocessing* package is dependent on NLTK for tokenizers and stopwords. However, ignoring this, the package only has built-in dependencies from Python 3.

CHAPTER 5

# Contributing

If you feel like contributing:

- Check for open issues or open a new issue
- Fork the preprocessing repository to start making your changes
- Write a test which shows the bug was fixed or that the feature works as expected
- Send a pull request and remember to add yourself to CONTRIBUTORS.md

CHAPTER 6

# License

This project is licensed under the MIT license (see LICENSE)

CHAPTER 7

*preprocessing* Contents

# Package Documentation

## Text Preprocessing

Text pre-processing module:

preprocessing.text.**convert_html_entities**(*text_string*)

Converts HTML5 character references within text_string to their corresponding unicode characters and returns converted string as type str.

Keyword argument:

•text_string: string instance

Exceptions raised:

•InputError: occurs should a non-string argument be passed

preprocessing.text.**convert_ligatures**(*text_string*)

Coverts Latin character references within text_string to their corresponding unicode characters and returns converted string as type str.

Keyword argument:

•text_string: string instance

Exceptions raised:

•InputError: occurs should a string or NoneType not be passed as an argument

preprocessing.text.**correct_spelling**(*text_string*)

Splits string and converts words not found within a pre-built dictionary to their most likely actual word based on a relative probability dictionary. Returns edited string as type str.

Keyword argument:

•text_string: string instance

Exceptions raised:

- InputError: occurs should a string or NoneType not be passed as an argument

preprocessing.text.**create_sentence_list**(*text_string*)

Splits text_string into a list of sentences based on NLTK's english.pickle tokenizer, and returns said list as type list of str.

Keyword argument:

- text_string: string instance

Exceptions raised:

- InputError: occurs should a non-string argument be passed

preprocessing.text.**keyword_tokenize**(*text_string*)

Extracts keywords from text_string using NLTK's list of English stopwords, ignoring words of a length smaller than 3, and returns the new string as type str.

Keyword argument:

- text_string: string instance

Exceptions raised:

- InputError: occurs should a non-string argument be passed

preprocessing.text.**lemmatize**(*text_string*)

Returns base from of text_string using NLTK's WordNetLemmatizer as type str.

Keyword argument:

- text_string: string instance

Exceptions raised:

- InputError: occurs should a non-string argument be passed

preprocessing.text.**lowercase**(*text_string*)

Converts text_string into lowercase and returns the converted string as type str.

Keyword argument:

- text_string: string instance

Exceptions raised:

- InputError: occurs should a non-string argument be passed

preprocessing.text.**preprocess_text**(*text_string*, *function_list*)

Given each function within function_list, applies the order of functions put forward onto text_string, returning the processed string as type str.

Keyword argument:

- function_list: list of functions available in preprocessing.text
- text_string: string instance

Exceptions raised:

- FunctionError: occurs should an invalid function be passed within the list of functions
- InputError: occurs should text_string be non-string, or function_list be non-list

preprocessing.text.**remove_esc_chars**(*text_string*)
Removes any escape character within text_string and returns the new string as type str.

Keyword argument:

•text_string: string instance

Exceptions raised:

•InputError: occurs should a non-string argument be passed

preprocessing.text.**remove_number_words**(*text_string*)
Removes any integer represented as a word within text_string and returns the new string as type str.

Keyword argument:

•text_string: string instance

Exceptions raised:

•InputError: occurs should a non-string argument be passed

preprocessing.text.**remove_numbers**(*text_string*)
Removes any digit value discovered within text_string and returns the new string as type str.

Keyword argument:

•text_string: string instance

Exceptions raised:

•InputError: occurs should a non-string argument be passed

preprocessing.text.**remove_time_words**(*text_string*)
Removes any word associated to time (day, week, month, etc.) within text_string and returns the new string as type str.

Keyword argument:

•text_string: string instance

Exceptions raised:

•InputError: occurs should a non-string argument be passed

preprocessing.text.**remove_unbound_punct**(*text_string*)
Removes all punctuation unattached from a non-whitespace or attached to another punctuation character unexpectedly (e.g. ".;';") within text_string and returns the new string as type str.

Keyword argument:

•text_string: string instance

Exceptions raised:

•InputError: occurs should a non-string argument be passed

preprocessing.text.**remove_urls**(*text_string*)
Removes all URLs within text_string and returns the new string as type str.

Keyword argument:

•text_string: string instance

Exceptions raised:

•InputError: occurs should a non-string argument be passed

preprocessing.text.**remove_whitespace**(*text_string*)

Removes all whitespace found within text_string and returns new string as type str.

Keyword argument:

- text_string: string instance

Exceptions raised:

- InputError: occurs should a string or NoneType not be passed as an argument

## License

# Python Module Index

## p

preprocessing.text, 15

# Index


## C

convert_html_entities() (in module preprocessing.text), 15
convert_ligatures() (in module preprocessing.text), 15
correct_spelling() (in module preprocessing.text), 15
create_sentence_list() (in module preprocessing.text), 16

## K

keyword_tokenize() (in module preprocessing.text), 16

## L

lemmatize() (in module preprocessing.text), 16
lowercase() (in module preprocessing.text), 16

## P

preprocess_text() (in module preprocessing.text), 16
preprocessing.text (module), 15

## R

remove_esc_chars() (in module preprocessing.text), 16
remove_number_words() (in module preprocessing.text), 17
remove_numbers() (in module preprocessing.text), 17
remove_time_words() (in module preprocessing.text), 17
remove_unbound_punct() (in module preprocessing.text), 17
remove_urls() (in module preprocessing.text), 17
remove_whitespace() (in module preprocessing.text), 17