
GeoNLPlify Documentation

Rémy Decoupes

Aug 10, 2023

CONTENTS:

1	Installation	3
1.1	Pip install	3
1.2	Virtual Environment	3
1.3	Conda environment	4
2	Devolppers	5
2.1	Building and installing the python package	5
2.2	Uninstall the python package	5
2.3	Push package to pypi	5
3	API	7
4	Acknowledgements	9
	Python Module Index	11
	Index	13



A NLP library for data augmentation focusing on spatial information contained in text.

Usage:

```
1 >>> import geonlplify
2 >>> my_text = "My name is Clara and I live in Berkeley."
3 >>> geonlplify.geonlplify(my_text)
4 'My name is Clara and I live in Bristol'
```


INSTALLATION

You can install GeoNLPlify in three ways:

- using pip
- in a *virtual environment (venv)*
- in a *conda environment* or

1.1 Pip install

```
pip install GeoNLPlify
python3 -c "from geonlplify import download_simplemaps_data; download_simplemaps_data()"
↪ # download simplemaps data
python -m spacy download en_core_web_trf # download spacy model
```

1.2 Virtual Environment

1. Git clone this repository

```
git clone https://github.com/remydecoupes/GeoNLPlify.git
```

2. Create a virtual env

```
python -m venv geonlplify_venv source geonlplify_venv/bin/activate pip install --
↪ upgrade pip
```

3. Install dependencies

```
cd GeoNLPlify pip install -r virtual_env_requirements.txt
```

4. Download world-cities from simple maps <<https://simplemaps.com/data/world-cities>>

```
wget -qO- https://simplemaps.com/static/data/world-cities/basic/simplemaps_worldcities_
↪ basicv1.75.zip | bsdtar -xvf- -C ./geonlplify/simplemaps/
```

1.3 Conda environment

1. Git clone this repository

```
git clone https://github.com/remydecoupes/GeoNLPlify.git
```

2. Create a conda with all the required dependencies

```
cd GeoNLPlify
conda env create -n geonlplify_conda --file conda_environment.yml python==3.10.6
conda activate geonlplify_conda
```

3. Install spacy models

```
python -m spacy download en_core_web_trf
```

4. Download world-cities from simple maps

```
wget -qO- https://simplemaps.com/static/data/world-cities/basic/simplemaps_worldcities_
↪basicv1.75.zip | bsdtar -xvf- -C ./geonlplify/simplemaps/
```

DEVELOPPERS

If you want to contribute to the project, here you can find some guidelines

2.1 Building and installing the python package

```
python3 -m build # build the wheel
pip install ./dist/GeoNLPlify-0.0.1-py3-none-any.whl # install the package
python3 -c "from geonlplify import download_simplemaps_data; download_simplemaps_data()"
↪ # download simplemaps data
python -m spacy download en_core_web_trf # download spacy model
```

2.2 Uninstall the python package

```
pip uninstall geonlplify
```

```
rm -r [your_site_package]/geonlplify/simplempas
```

2.3 Push package to pypi

```
python3 -m build
python3 -m twine upload dist/*
```


`geonlplify.find_sne(text)`

Return list of SNE

Parameters

text – input text

Returns

list of SNE

`geonlplify.geocode(sne_name)`

Input a place name and retrieves OSM properties. Exemple for `sne_name="Montpellier"`:

```
{'osm_id': 65442261, 'osm_type': 'N', 'country': 'France', 'osm_key': 'place', 'city': 'Montpel-  
lier', 'countrycode': 'FR', 'osm_value': 'city', 'postcode': '34062', 'name': 'Montpellier', 'county':  
'Hérault', 'state': 'Occitanie', 'type': 'district'}
```

Parameters

sne_name –

Returns

a dictionary with OSM properties

`geonlplify.geonlplify(text, method='spatial_synonym')`

GeoNLPlify aims to make variations of an input sentence working on spatial information contained in words

Examples:

```
>> geonlplify.geonlplify("5 cases of avian influenza found in Montpellier") 5 cases of avian influenza found  
in Bangalore
```

Parameters

- **text** – Input
- **method** – Between those 3 methods [generalization, specialization, spatial_synonym]

Returns

the variation of the input text

`geonlplify.importing_error_check()`

Check if there is any importing issue (like missing data or spacy model) :return:

`geonlplify.load_simplemaps()`

Load simplemaps from <https://simplemaps.com/data/world-cities> CC-BY 4.0 attribution from <https://simplemaps.com/data/world-cities> :return:

`geonlplify.replace_variants(text, list_of_variant, method, conserve_n_gram=True)`

Parameters

- **text** –
- **list_of_variant** – example: - error:
[{'name': 'South Sudan', 'label': 'GPE', 'start_char': 56, 'end_char': 67, 'generalization': nan, 'generalization_failed': 'sea'}]

– ok:
[{'name': 'New Jersey', 'label': 'GPE', 'start_char': 39, 'end_char': 49, 'generalization': 'United States'}]

Returns

ACKNOWLEDGEMENTS

This library use those terrific tools/libraries/data

- [Spacy](#)
- [The Komoot geocoder Photon](#)
- [OpenStreetMap](#)
- [Simplemaps data](#)

PYTHON MODULE INDEX

g

geonlplify, [7](#)

F

`find_sne()` (*in module geonlplify*), 7

G

`geocode()` (*in module geonlplify*), 7

`geonlplify`
 module, 7

`geonlplify()` (*in module geonlplify*), 7

I

`importing_error_check()` (*in module geonlplify*), 7

L

`load_simplemaps()` (*in module geonlplify*), 7

M

module
 geonlplify, 7

R

`replace_variants()` (*in module geonlplify*), 7